CS414 Assignment2 Notes

From professor, msg board:

RE: Data Gathering for Get Statistics

Actually there will be differences. you are only thinking of the internal implementation as a query. If I want a report that is based on a date range, I will need to select the type of data I want, give a start date and an end date. Think of the use case as something you can give someone else and have them implement it for you.

If you just tell a developer that you want to query, what do you think the developer will ask you before implementing your requirement? Every query has specific inputs and expects a certain kind of output. If nothing is specified, will you be happy if a developer just dumps the entire raw data on the screen?

To figure out which reports may be useful, you can use the Goal Question Metric idea. A garage administrator may have goals related to increasing revenue. To do that, the administrator may need to ask questions such as, when is the garage occupied at high levels because the answer to this may lead to different pricing strategies for different times of day. To answer that question, what kind of report will be useful? So, think of goals, some related questions, and then figure out what metrics (data) will help answer the questions.

One you have the types of data identified, you can figure out what kinds of queries are needed (i.e., what arguments need to be selected by the administrator, and what results are expected).

Otherwise, a generic query use case is useless in that anyone can implement it in any way s/he wants.

Re: Payments

Accept multiple forms of payment

If you do have credit card authorization, you can pretend to take the input (16-digit number and exp date) and the system just checks if it's a valid number (ie. it has 16 digits). No need and you can't use a real credit authorization system without getting into trouble.

All you need to do for cash payments is that the driver or cashier enters the amount in the system (like Larman's POS system) and the balance is reported. No need to show a cash register opening and closing.

RE: Policies

Policies are up to us (min amount due), Max amount due.

Size of the garage:

Size should never be hardcoded.

Impose a lost ticket fee.

I would treat the opening of the gate and subsequent closing (safely to let the car in) to be a step in the use case of "Enter garage".

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Do we require some sort of access control? So that only the Garage Admin can query into the system?

That would be natural to have (using simple login and password is okay) for any employee in the garage. This also brings up a use case for Add User, etc.

Normally one would not write use cases with out of scope elements. A separate document or section would be used to prioritize use cases/alternatives in a certain iteration, which would effectively rule out the out of scope elements for a particular iteration. I don't really care where you state what is in scope and what is out-of-scope, as long as you state them somewhere.

Yes, you may create a separate section for System Scope or Additional Constraints as the case may warrant.

2. If the customer is unable to pay, what would you like the system to do?

3. If the customer is unable to present a valid ticket, what would you like the system to do?

4. If the system crashes or is down, can parking customers leave without paying?

    a. If there are gates, will the gate to leave open and remain open, and the gate to come in remain closed, until the system is restored?

When we grade your alternative situations, we will be looking at your text to see if you thought about what things can go wrong. So, you need to mention the reasons for alternative situations.

Third: You will state what actions the system will take. For some cases, you will say that it is outside the scope of the system you will develop (power failure stopping gates from moving, etc). For others, you will show how the system will handle those cases. For your specific questions:

1. Gate: What's the question here? The assignment already mentions gates.

2. If the customer is unable to pay: There's not just one way to handle this. You are free to come up with ways, that involve the system in some way. For example, you can do an override to open the gate using the system, but the driver's information will be taken by a cashier or manager either using the system or outside the system.

3. You need to figure out some way. It doesn't matter what you do as long as you do something reasonable in the system. For example, charge a flat fee for fine. If you want to view video of cars entering the garage, that's outside the scope of the system. But some payment must be done.

4. If your system crashes, you will lose points (because that means some tests failed). However, for the sake of simulation, assume that gates don't crash or break apart.