SOM Lab Exercise II

September 17, 2019

Introduction

In this exercise you will work in pairs to practice *designing* software. Start by reading the case study below. There are several exercises that you will need to answer. Please write your report using the template included on the last page of this exercise.

Case study

In this exercise, you will start design a (fragment of a) ticket machines for the *Nederlandse Spoorwegen*. For the sake of simplicity, we will ignore many of the issues surrounding the real ticket machines – such as the OV chipcard or various different kinds of railcards (such as *trajectkaarten* or *jaarkaarten*). Instead, we limit ourself to ticket vending machines where customers may purchase paper tickets for journeys within the Netherlands. Not too long ago, it was only possible to buy tickets in person at a ticket desk. In this case study, we will consider the initial design for a first ticket vending machine.

A colleague has had several discussions with various stakeholders. You have been provided with the transcripts of these interviews. Read through these transcripts carefully and answer the questions starting on page 6.

1. Transcript: A ticket vendor

This is the transcript of an interview conducted with Rob Engelhart, an NS employee working at Utrecht CS.

Q: So Rob, tell me about yourself.
A: My name is Rob Engelhart. I'm married and have two children. I've been working at the NS for the last twelve years. At first, I was based in Leeuwarden, but I moved to Utrecht five years ago.
Q: And what do you do specifically?
A: I work behind the NS ticket desks, selling tickets and answering the customers questions.
Q: How do you do that?

A: Well, a customer comes up to my desk and tells me where they want to go. I enter this information into the terminal at my desk. This computes the price of the ticket. Once the customer has paid this amount, I print the ticket and give it to them. If they want, I can also provide advice about the next train, or which route to take exactly.

Q: What kind of tickets do you sell?

A: Oh, we sell tickets to anywhere in the Netherlands.

Q: You don't sell international tickets?

A: For international tickets, we have a separate service desk. I don't know anything about that.

Q: How is the price of a ticket computed?

A: I don't know really. I enter the destination into my terminal, and it tells me the answer.

Q: How do people pay for their ticket?

A: I don't understand the question...

Q: Well, do you accept cash?

A: Of course we accept cash!

Q: What about debit cards? Credit cards?

A: Well, we accept all Dutch debit cards. We can also take any credit card, but we do charge an additional 0.50 euro for credit card payments.

Q: So, what information do you enter to compute the price of a ticket?

A: I just enter the destination and the number of tickets that need to be sold.

Q: And that's all the information you need?

A: Yes.

Q: What about travelling first-class?

A: Oh, if a customer wants to travel first-class, I can enter that information too.

Q: Is there any other variation in the kind of tickets that you sell?

A: Sure. We sell all kinds of tickets: return tickets, single tickets, discounted tickets...

Q: Wait a minute. So, when is a ticket discounted?

A: If a customer has a railcard, they get either a 20% or a 40% discount on their ticket.

Q: And what does that depend on?

A: Well, on the rail card of course. Some customers have a more expensive railcard that gives them bigger discounts. But these discounts are only valid when travelling in the weekend or outside the morning rush hour.

Q: And this discount is *exactly* 20% or 40%?

A: I guess...

Q: And how is it rounded off?

A: I don't understand the question.

Q: Well, if a ticket costs 9,05 euro and I buy it with discount, it would cost 5,43 euro – but that seems a funny amount. I usually see prices advertised that are nice round numbers.

A: I really don't know how the price is calculated. I just enter the information.

Q: So in summary, customers can buy train tickets at your desk. These train tickets are only for trains travelling within the Netherlands...

A: (interrupts) Well, the train could be an international train, but then you would need an additional ticket.

Q: An additional ticket?

A: Yes. Take the Thalys for instance. It runs between Amsterdam and Paris. A traveller could get on the train in Amsterdam, and leave in Utrecht. He would just need a ticket from Utrecht to Amsterdam, but he'd need to pay an additional 2 euro supplementary charge.

Q: And why didn't you mention it earlier?

A: Well you never asked me about it!

Q: So, once again. Customers can buy tickets at your desk. They can pay for these tickets in a number of different ways. You sell tickets from Utrecht to anywhere in the Netherlands...

A: (interrupting) Not only from Utrecht! We can sell tickets starting from any train station

in the Netherlands, except for Utrecht Maliebaan. But that's because the station there is only for the museum and doesn't serve any real purpose.

Q: Right. So usually customers want tickets departing from Utrecht, but they can also buy a ticket departing from a different train station.

A: Exactly.

Q: And once a customer buys a ticket, you print their ticket, and they get on the next train.

A: Well, it doesn't need to be the next train. A ticket is valid for an entire day.

Q: So a ticket is not associated with any particular train, but is valid for the whole day on which it was purchased.

A: Well, not all tickets work that way. Sometimes a customer will ask for a ticket without a date on it. This costs the same, but they need to validate the ticket themself before getting on the train. There are separate machines for validating your ticket - there's one on every station.

Q: And what about return tickets?

A: Return tickets have to be used on the day that they are sold. They are valid for a one return trip from the station of departure to the station of your destination and back.

Q: And how is the price of a return ticket calculated?

A: I don't know exactly. They are a bit cheaper than buying two singles. My computer just tells me the price once I input the required information.

Q: Can the sale go wrong at any point?

A: Well sometimes a credit card is declined. I've also had a few people who didn't have enough cash to pay for their ticket. It was pretty embarrassing, I guess.

Q: Thanks for all your answers so far. I'd like to speak to someone responsible for the computer systems that you guys use.

A: I'm sure someone can help you with that.

2. Transcript: Software Architect

The second interview was with Paul van Dijk, chief software architect at the NS.

Q: Hi there. You must be Paul.

A: That is correct.

Q: Pleased to meet you.

A: Hmmph.

Q: So I hear you are the man responsible for the current software system.

A: It's a work of art. We distribute price information over an encrypted SSL connection to hundreds of different stations in the blink of an eye. We haven't had a single hour of unscheduled downtime in the past three years.

Q: That's very impressive. What is it your system does?

A: We've implemented it using D, an alternative to C++ that most people haven't heard of. It is so much better than C++. It's multi paradigm, you write much shorter code, and I couldn't live without type inference.

Q: That's really very interesting. But how does the system work?

A: The servers here at the NS are ancient. We're still running a OSF/1 on a Dec Alpha. Do you know we had to do to fork the DMD compiler and write our own backend to generate binaries?

Q: I had no idea...

A: It wasn't easy, I can tell you! We nearly lost our entire codebase when an intern screwed up our source control. It's a good thing that most of the important information is all stored in a separate database.

Q: Database?

A: Yeah, all the price calculations are really just SQL queries.

Q: How so?

A: There are two tables hosted in a MySQL Oracle Enterprise server. Those guys from Oracle set that part up, I have no idea how it works.

Q: But what is stored in the tables?

A: Oh yes. Prices are calculated in a two step process. Wait I have a map here somewhere... (Paul shows the map in Appendix A)

Q: And what is this map for?

A: Well one table on our server stores the distance between every two stations in the Netherlands.

Q: And how is that distance measured? These numbers don't look like kilometers.

A: That's right! Distance is expressed in *tariefeenheden*. This is not so much a unit of length, but rather captures how expensive it should be to travel along a certain route.

Q: But how do the *tariefeenheden* effect the overall price of a ticket?

A: That's where the second table comes in. Now you might think that this was just a simple table, but actually we've done something far more clever. When the Oracle guys moved the database server, there was an issue with orphaned SQL Server users on the new machine. With the new stored procedures in SQL Server 2000, we're now in a much better position to migrate away from the old Dec Alpha servers that...

Q: (interrupting) But what does this have to do with the ticket pricing?

A: Well, the *tariefeenheden* don't tell you anything about the price really. The second database table uses the number of *tariefeenheden* as a key to store how much different tickets cost for every possible *tariefeenheid*. Let me print out the first few entries in that table. (Print out the table in Appendix B)

Q: What do these numbers mean?

A: Do you see what I did there? Do you have any idea how hard it is to print Unicode euro symbols in D? I even managed to summarize the first eight identical rows into one single row. Pretty cool, huh!

Q: Amazing. I still don't know what these numbers mean.

A: This table converts *tariefeenheden* to euros.

Q: Right. So this table tells you how much a ticket will cost.

A: Exactly. The first column is the number of *tariefeenheden*, the next three columns are for second-class tickets, the final three columns are for first-class tickets.

Q: Why are there three columns for first-class and three columns for second-class?

A: Company policy. We are only allowed to charge amounts that are multiples of 0,10 euro. If we were to charge different amounts, the people selling tickets would need too much time counting out the change.

Q: But what do these three columns mean?

A: They are for the full price ticket, together with the 20% and 40% discount. These are the prices for all of 2013.

Q: So what happens next year?

A: Well there's been talk of abandoning the 20% discount rate, and introducing new railcards. Every year we are allowed to increase the prices once, but there is a government limit on how much we are allowed to raise the pricing.

Q: And what about return tickets?

A: A return ticket is just as expensive as two singles. There used to be some discount, but we got rid of that three years ago.

Q: Right. So the ticket machines at the stations are just querying these SQL tables and printing the appropriate ticket.

A: No! We do so much more than that! The communication has to be encrypted, plus all our information needs to be backed up ever hour for legal reasons. We also host the mail servers and NS Twitter account. On top of that, I'm the lead developer of our fork of the Dec Alpha backend of the DMD compiler. Without us, this company would fall apart.

Q: Thanks for your answers so far.

3. Transcript: Managing director

The day ends with an interview with Dik Hessels, CIO of the Nederlandse Spoorwegen.

Q: Hi Dik, thanks for seeing me today.
A: No problem. I do have another meeting coming up in fifteen minutes, so we don't have much time.
Q: So, what is it exactly you want these vending machines to do?
A: Well, we're looking to reduce the cost of running the NS. One of our biggest expenses is in staff. Some of the smaller trainstations hardly have any clients in the weekend, but we still need to man the ticket desks. To reduce our operational costs, we aim to start deploying a limited number of vending machines in Q2 of next year, but have complete coverage in less than two years.
Q: I see. But what should these ticket vending machines do?
A: They should be a complete replacement of the current personnel.
Q: So they should also provide information about train times?
A: Of course not! They're just ticket machines. We make quite a lot of money selling books containing all our traintables. We would lose that source of revenue if the ticket machines could also do that. The ticket machines offer our customers a way to pay for tickets, which get printed on the spot.
Q: So it should only sell tickets.
A: Yes.
Q: And how should people pay for these tickets.
A: We got a great deal on second hand coin machines from IKEA. One of our hardware guys also gave me information about the current interface for our card readers (Appendix C and Appendix D). When do you think the project will be ready?
Q: I really can't say that yet.
A: We need to roll out before 2016. If you cannot guarantee that, we will need to hire another design team.
Q: It's too early to give any kind of guarantee. But I wanted to ask you...
A: I had to fire the last team we had working on a big IT project. Did you hear about the headaches the recent server migrations have been causing? We tried outsourcing some of our support, but that was a complete disaster. Instead, we now have an Oracle team on-site for our database support. The drawback is that we have already overspent on our IT budget for this year. I can move some funds around, but I will need an outstanding design and highly very competitive price from you.
Q: We're still working on the initial requirements. It's too early to say anything concrete about this project just yet.
A: It sounds like you cannot commit to the 2016 deadline.
Q: I'd like to, but like I said, it's still too early to say. I did have a few other questions for you though. What tickets should the vending machines sell?
A: Any tickets that can be bought at our ticket desks.
Q: Including rail cards and discount cards?
A: No. Those will only be available from ticket desks at our main train stations.
Q: And the tickets should cost the same as at the ticket desk?
A: I don't know yet. We're still doing market research to see how people would react to different pricing models.
Q: So how much should a ticket cost from a ticket vending machine?
A: The same as at the ticket desk, I suppose. But we may want to change that later. We're also exploring introducing new rail cards and ticket options. We will make the final decision next year. I am not going to pay you extra for this kind of work.
Q: Any other changes I can expect?
A: One of our concerns is that ticket vending machines in other countries sometimes get vandalized. We want to avoid storing too much cash in the machines. We may want to

replace cash payments with a *chipknip* option in the future.

Q: So, apart from that you don't know the pricing of tickets from the machine, the payment methods that are permitted, the types of tickets the machine will able to issue, it seems like you have a pretty clear idea of what you want.

A: Great talk. I'll see you out there. Sorry, I have a business lunch in Paris. Call me later once you have your software implemented, and we can discuss the price.

4. Exercises

Part 1. Write a fully-dressed description of the ticket purchasing use case. What exceptional scenarios could occur? Be sure to adhere to Larman's guidelines for fully-dressed use cases in Chapter 6.

Part 2. Use cases are a great way to document the functional requirements of the system. What other (non-functional) requirements can you come up with? Give a supplementary specification for the ticket machine. Use the example from section 7.2 of (the Second Edition of) Larman's book as a template. (Note: this is Section 7.4 in the more recent Third Edition).

Part 3. Based on the information given in these interviews, there may still be requirements that are unclear. What issues would you like to clarify in future discussions with these stakeholders? Explicitly state any assumptions that you make about the requirements of the proposed system.

Part 4. Based on these interviews, scenarios and use cases, draw a domain model. Be as explicit as possible about how the entities in your model are related.

Part 5. Use this domain model and the GRASP patterns to assign responsibilities associated with a ticket sale.

Part 6. Design the software for the vending machine by giving a UML class diagram. You do not need to worry about the graphical user interface at this point. Instead focus on identifying the classes and their relations relevant to the price calculation and printing of tickets. For each class, specify:

- the attributes associated with the class;
- the methods the class supports;
- give a brief explanation of the abstract responsibilities assigned to every class;
- also briefly describe what the methods and attributes of every class should do.

Based on this UML class diagram, it should be clear how the system you have designed will function, even if you have not implemented it yet. Be as precise as possible about the types of the various methods and attributes.

Part 7. Explain the different kinds of *variation* that you have encountered in this domain. How does your design handle this variation?

Part 8. Explicitly identify any design patterns that you have used. For each such pattern, explicitly state:

- the problem it addresses;
- how the design pattern has been applied. Be specific about the classes in your design that are associated with the design pattern.
- which alternatives did you consider?

Part 9. Evaluate your design. Explain how it is capable of handling changing requirements. Identify at least three likely future changes to the specification. How will your design handle such changes? What kind of requirement changes would be difficult to incorporate?

Submission

Details about submission will follow. Deadline can be found on BB.

Taal

De opdracht mag zowel in het Engels als Nederlands worden gemaakt.

A. Appendix: Tariefeenheden

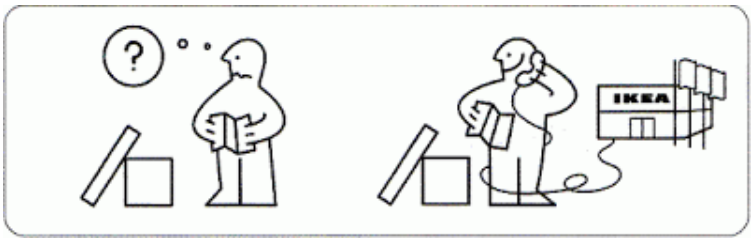
B. Appendix: Pricing table fragment

0 t/m 8	€ 2,10	€ 1,70	€ 1,30	€ 3,60	€ 2,90	€ 2,20
9	€ 2,20	€ 1,80	€ 1,30	€ 3,70	€ 3,00	€ 2,20
10	€ 2,40	€ 1,90	€ 1,40	€ 4,10	€ 3,30	€ 2,50
11	€ 2,50	€ 2,00	€ 1,50	€ 4,30	€ 3,40	€ 2,60
12	€ 2,70	€ 2,20	€ 1,60	€ 4,60	€ 3,70	€ 2,80
13	€ 2,90	€ 2,30	€ 1,70	€ 4,90	€ 3,90	€ 2,90
14	€ 3,10	€ 2,50	€ 1,90	€ 5,30	€ 4,20	€ 3,20
15	€ 3,20	€ 2,60	€ 1,90	€ 5,40	€ 4,30	€ 3,20
16	€ 3,40	€ 2,70	€ 2,00	€ 5,80	€ 4,60	€ 3,50
17	€ 3,60	€ 2,90	€ 2,20	€ 6,10	€ 4,90	€ 3,70
18	€ 3,70	€ 3,00	€ 2,20	€ 6,30	€ 5,00	€ 3,80

C. Appendix: Card reader

```
public interface ICard
{
    string[] ListReaders();
    void Connect(string Reader, SHARE ShareMode,
                 PROTOCOL PreferredProtocols);
    void Disconnect(DISCONNECT Disposition);
    APDUResponse Transmit(APDUCommand ApduCmd);
    // Charge amount to a credit card or debet card
    TransactionID BeginTransaction(int amount);
    // Returns true if transaction is successful.
    bool EndTransaction(DISCONNECT Disposition, TransactionID id);
    void CancelTransaction(TransactionID id);
}
```

D. Appendix: Coin machine



IKEAMyntÄtare2000
- köttBullar : int + status : int + summa : int # sittaPoang : bool
+ bool ätaMynt() + betala(int pris) + starta() + stoppa()

E. Report format

SOM Lab Exercise II: Design of the NS ticket machine

Student names:
Student numbers:

E.1. Use case: Ticket Purchase

E.2. Supplementary requirements

E.3. Clarification and assumptions

E.4. Domain model

E.5. Responsibilities

E.6. Design

E.7. Variation

E.8. Design patterns

E.9. Evaluation