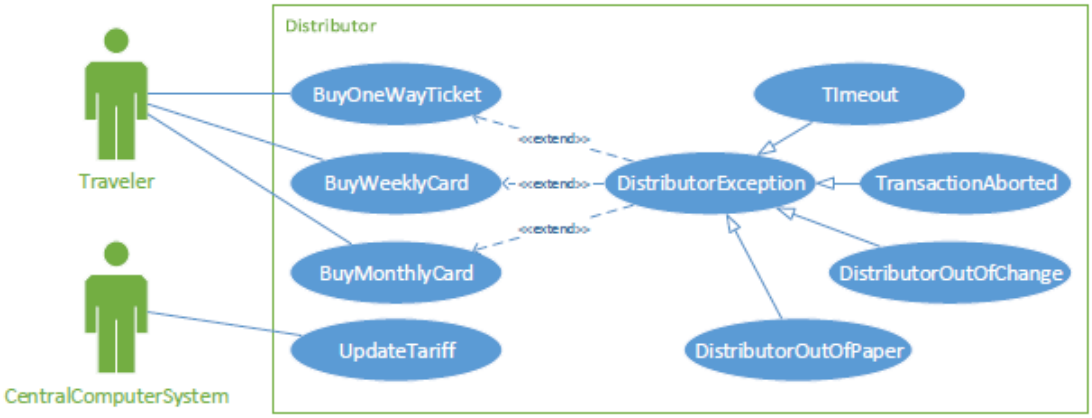
Assignment 36

Tróndur Høgnason Kristian Mohr Nielsen

12. september 2014

1 Use-case diagram

Our use-case diagram (figur 1) has an extra exceptional use-case, which we chose to include to make it more organized. There is two actors: the Traveler who invokes the three use-cases: BuyOneWayTicket, BuyWeeklyCard and BuyMonthlyCard, and the CentralComputerSystem who invokes the use-case UpdateTariff. Each one of the three use-cases invoked by the traveler extend DistributorException which generalises the other exceptional use-cases. This means that the three use-cases can invoke all of the exceptional use-cases



Figur 1: Use-case diagram

2 Use cases

Below the nine use cases are listed. Quality requirements are left blank, since there were no requirements stated in the exercise and we didn't want make something up for all of the use-cases. But some examples could be that the payment transaction needs to be completed in under xx seconds after the pin has been written.

2.1 Buy one way ticket

Participating actors:
Initiated by Traveler

Flow of events:

- The Traveler chooses to buy one way ticket
- The Traveler pays
- The ticket is printed

Entry conditions:
The ticket distributor has change and paper

Exit conditions:
The Traveler has received the ticket, **OR** the Traveler has received an explanation indicating why the transaction can't be completed.

Quality requirements:
–

2.2 Buy weekly card

Participating actors:
Initiated by Traveler

Flow of events:

- The Traveler chooses to buy weekly card
- The Traveler pays
- The card is printed

Entry conditions:
The ticket distributor has change and paper

Exit conditions:
The Traveler has received the card, **OR** the Traveler has received an explanation indicating why the transaction can't be completed.

Quality requirements:
–

2.3 Buy monthly card

Participating actors:
Initiated by Traveler

Flow of events:

- The Traveler chooses to buy monthly card
- The Traveler pays
- The card is printed

Entry conditions:
The ticket distributor has change and paper

Exit conditions:
The Traveler has received the card, **OR** the Traveler has received an explanation indicating why the transaction can't be completed.

Quality requirements:
–

2.4 Update tariff

Participating actors:
Initiated by CentralComputer

Flow of events:

- The CentralComputer updates the tariff on the distributor.
- Distributor confirms the tariff has been updated

Entry conditions:
There has been changes to the tariff

Exit conditions:
The Distributors tariff has been updated

Quality requirements:
–

2.5 Distributor Exception

Participating actors:
Communicates with Traveler

Flow of events:

- This use case extends BuyOneWayTicket, BuyWeeklyCard and BuyMonthlyCard use cases. It is initiated whenever a problem occurs, as the Traveler is trying to buy a ticket.

Entry conditions:
This use case extends BuyOneWayTicket, BuyWeeklyCard and BuyMonthlyCard use cases. It is initiated whenever a problem occurs, as the Traveler is trying to buy a ticket.

2.6 Timeout

Participating actors:
Inherited from DistributorException use case

Flow of events:

- Cancel transaction
- Notify Traveler that too much time has been spent at one stage of the transaction

---

Entry conditions:
Inherited from DistributorException, **AND** XX time has been spent on one stage of the transaction.

Exit conditions:
User has be notified

2.7 Transaction aborted

Participating actors:
Inherited from DistributorException use case

Flow of events:

- Cancel transaction
- Notify the traveler that the transaction has been aborted

Entry conditions:
Some event has aborted the transaction

Exit conditions:
Traveler has been notified

Quality requirements:
–

2.8 Distributor out of paper

Participating actors:
Inherited from DistributorException use case

Flow of events:

- Notify the traveler that the distributor is out of paper

Entry conditions:
The distributor is out of paper

Exit conditions:
The distributor has been refilled with paper

Quality requirements:
–

2.9 Distributor out of change

Participating actors:
Inherited from DistributorException use case

Flow of events:

- Notify the traveler that the distributor is out of change

Entry conditions:
The distributor is out of change

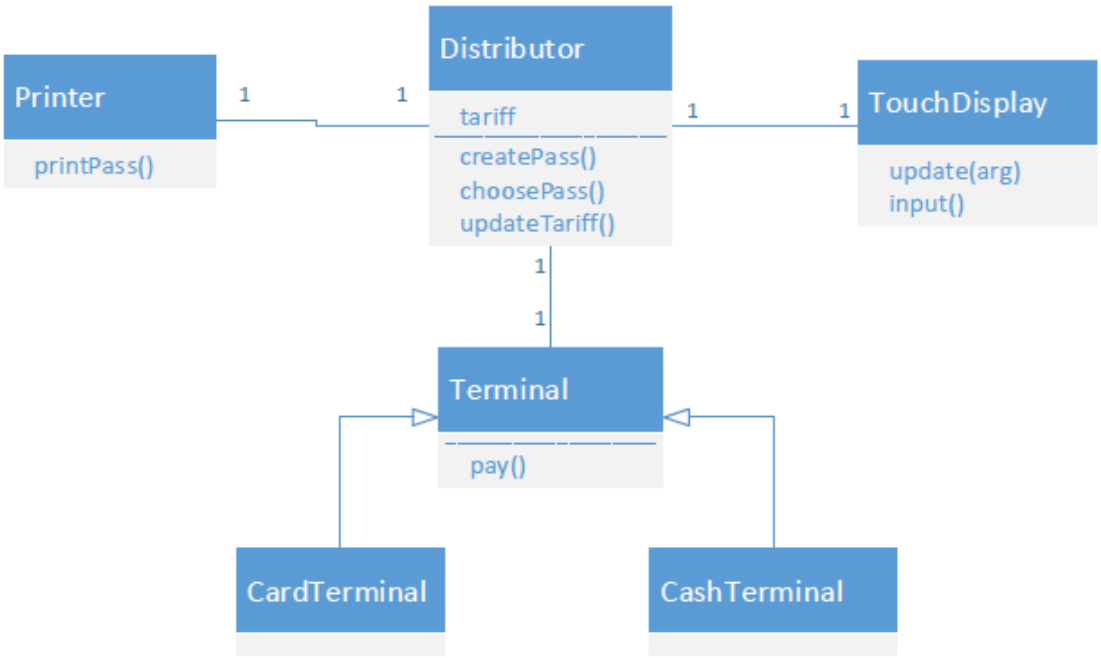
Exit conditions:
The distributor has been refilled with change

Quality requirements:
–

---

3 Class diagram

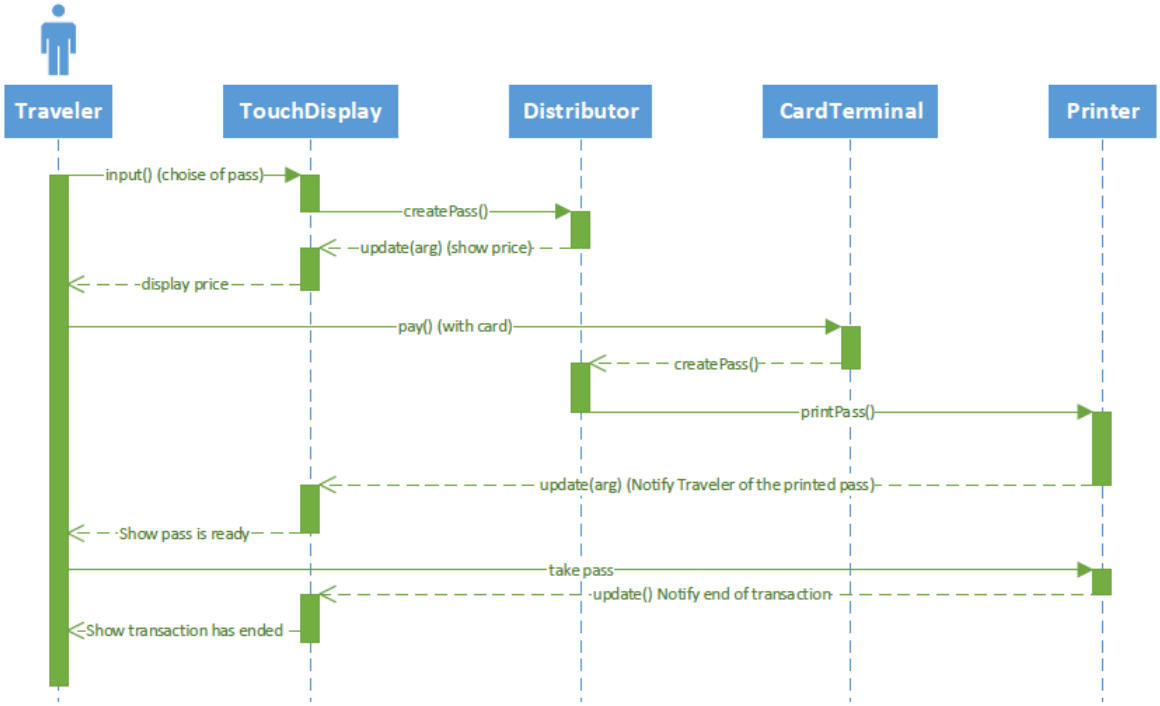
Our class diagram (figur 2) is of the distributor and its parts, although simple. We've chosen that it has: a touch display, for input and output of information, two terminals - one for paying with card and another for paying with cash and finally it has a printer to print the pass (a ticket or a card).



Figur 2: Class diagram

4 Sequence diagram

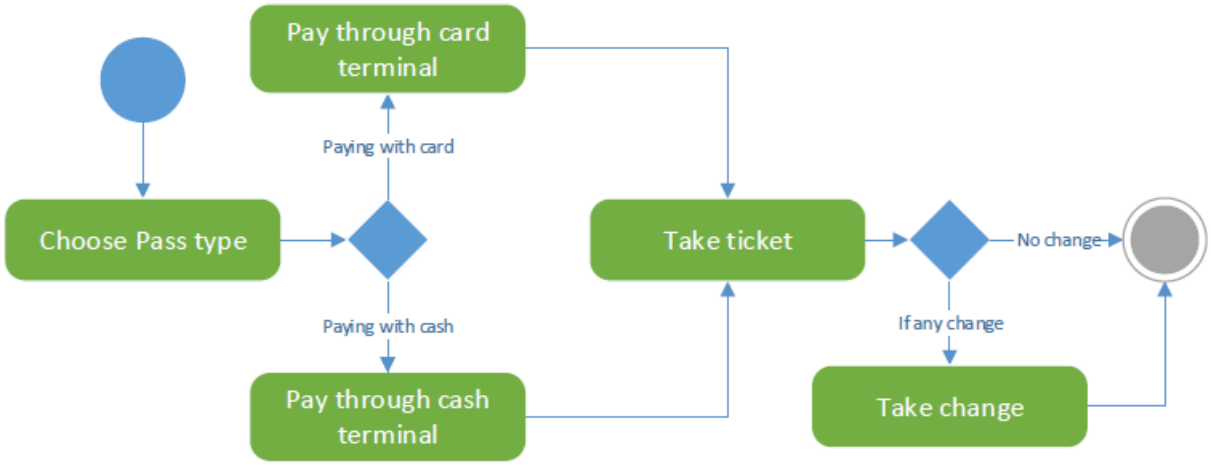
We've chosen to make a sequence diagram for the interaction between a traveler, the distributor and parts of the distributor (figur 3), when the traveler buys a pass (a ticket or a card).



Figur 3: Sequence diagram

5 Activity diagram

Figure 4 shows our activity diagram of a traveler buying a pass. First the traveler has to choose which kind of pass he wants to buy, then the traveler can choose to pay with either card or cash. After paying the ticket is printed and the traveler can take it. If there is any change the traveler picks it up (at least in a perfect world) and the transaction is over.



Figur 4: Activity diagram