

Name: Ennio da Silva Vitor

date:15-03-2017

Model a video rental machine using UML and case tools

Introduction

This assignment have the objective to identify all the problems that the Xtra-Vision company have to solve by create a software that will allow customer to purchase movies in vending machines.

Before the process of documentation that will allow us to start develop the software, is necessary to visualize how the old steps were made in order to complete the customer order.

High level description

When the customer arrives at the shop, he or she goes to the front counter, with the movie that they had chose, the staff check the status of the customer membership, if the status are ok the staff print the recipe and charge the total amount.

The return procedure is similar, the staff check the status of the membership and charge any extra fee if applicable.

Problem definition

Problems with the current system:

- Customer goes to the front counter to rent a movie.
- High volume of customers.
- The process to rent or return a movie is slow.
- The membership is necessary.
- The company accept cash then extra fees if any had to be charged later.

- A receipt is printed to the customer.

Objectives of new system:

- To reduce the volume of customers at the front counter.
- To optimize the process to rent or return a movie.
- To do not request membership.
- To accept only credit cards and debit cards.
- To E-mail a receipt to the customer.

Scope of the new system:

The project will encompass the following areas of the business:

- Hire procedures.
- Return procedures.
- Recording movie details.
- Payment of rental and return.

Preliminary idea:

- Design a user friendly movie rental software.
- Provide facilities to handle total amount of discs as new user or regular user.
- Add the software to vending machines.

Recommended actions:

- Investigate information to be stored about customers.
- review design of the movie shelf to be displayed on the software.

Requirements for the proposed system

R1 - Keep a list of all movies and their details, movie name, daily charge rate.

R2 - Keep a record of all credit/ debit cards and their past transactions.

R3 - Work out automatically how much it will cost to rent a selection of movies.

R4 - Record the details of a rent transaction including the start date, card number, movie(s), in such a way that it is easy to find the relevant transaction details when a movie is returned.

R5 - Keep track of how many movies are in a single transaction, so a receipt can be E-mailed if requested.

R6 - Work out automatically, on the return of a movie, how long it has been hired, how much extra is due (if applicable).

R7 - Record the total amount due to be taken from the account of the credit/ debit card.

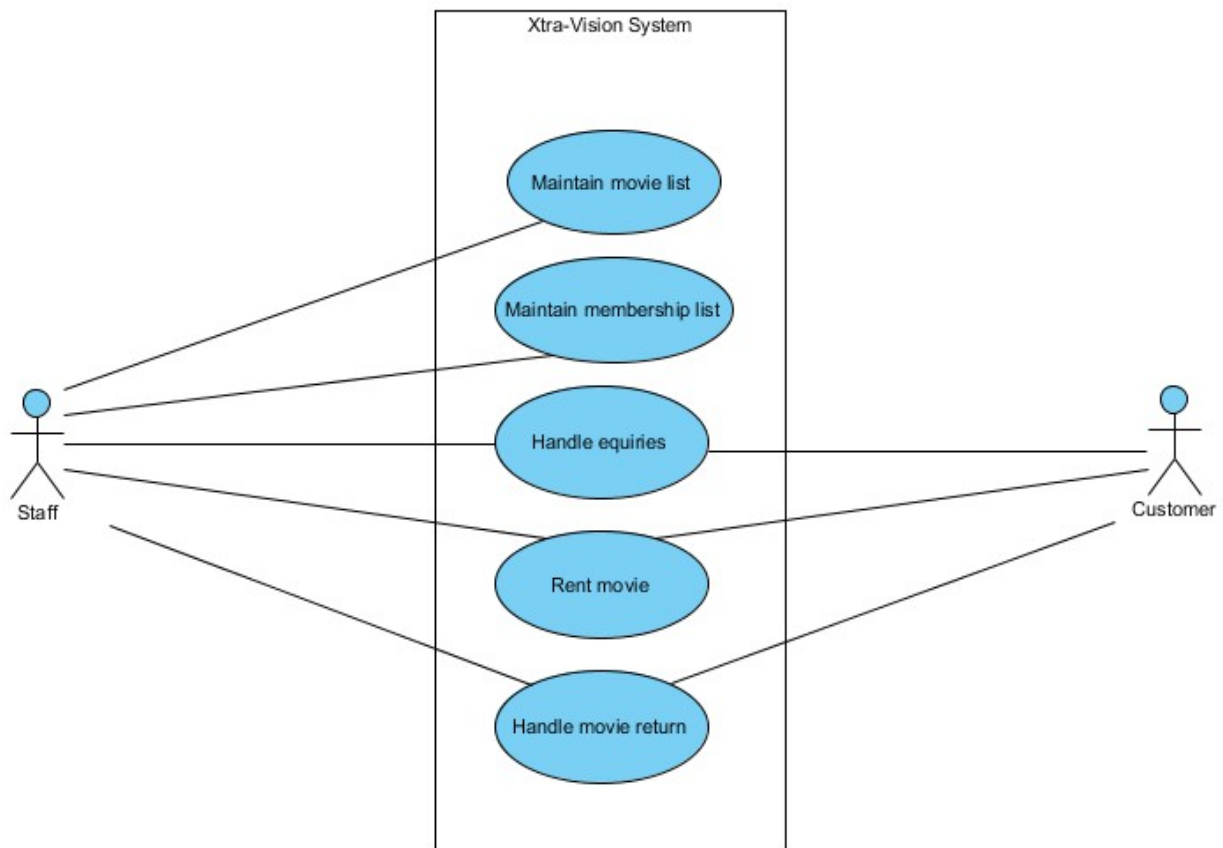
R8 - Give the option to E-mail a receipt.

R9 - Keep track of the state of each movie, e.g. Whether it is in stock, hired and position.

System analysis

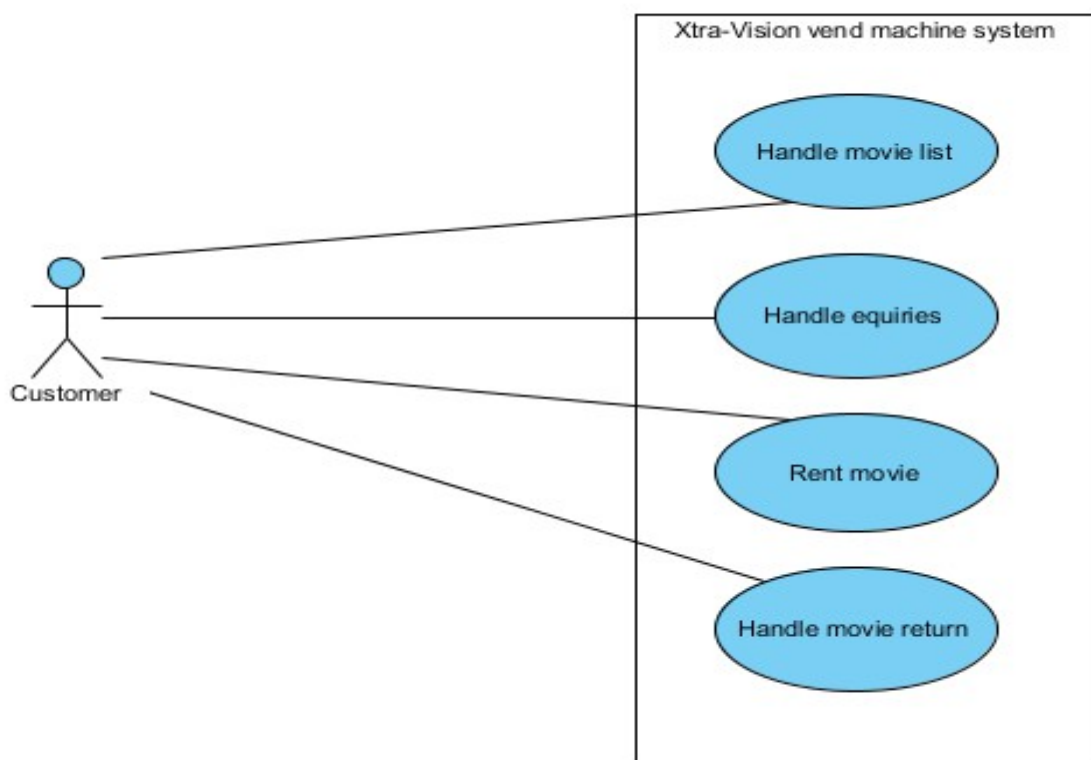
Use case diagrams:

The system where the Xtra - vision company runs can be described as bellow, the staff have to handle the movie list, maintain membership list, handle enquiries, rent movies and handle movie returns. The system is quite slow and the customer also is included into it, the customer at the front counter, handle enquiries, rent movie and handle movie return.



By creating the vend machines the Xtra - Vision company will reduce the work of the staff, as we can see in the diagram bellow the new system will give the task of handle enquiries, rent movies and handle movie returns.

The new system will not have membership. So any customer with a credit/debit card will be able to rent a movie.



Class diagram:

According with the requirements identified the class diagram bellow have been created to describe the relationship between the class and their attributes.

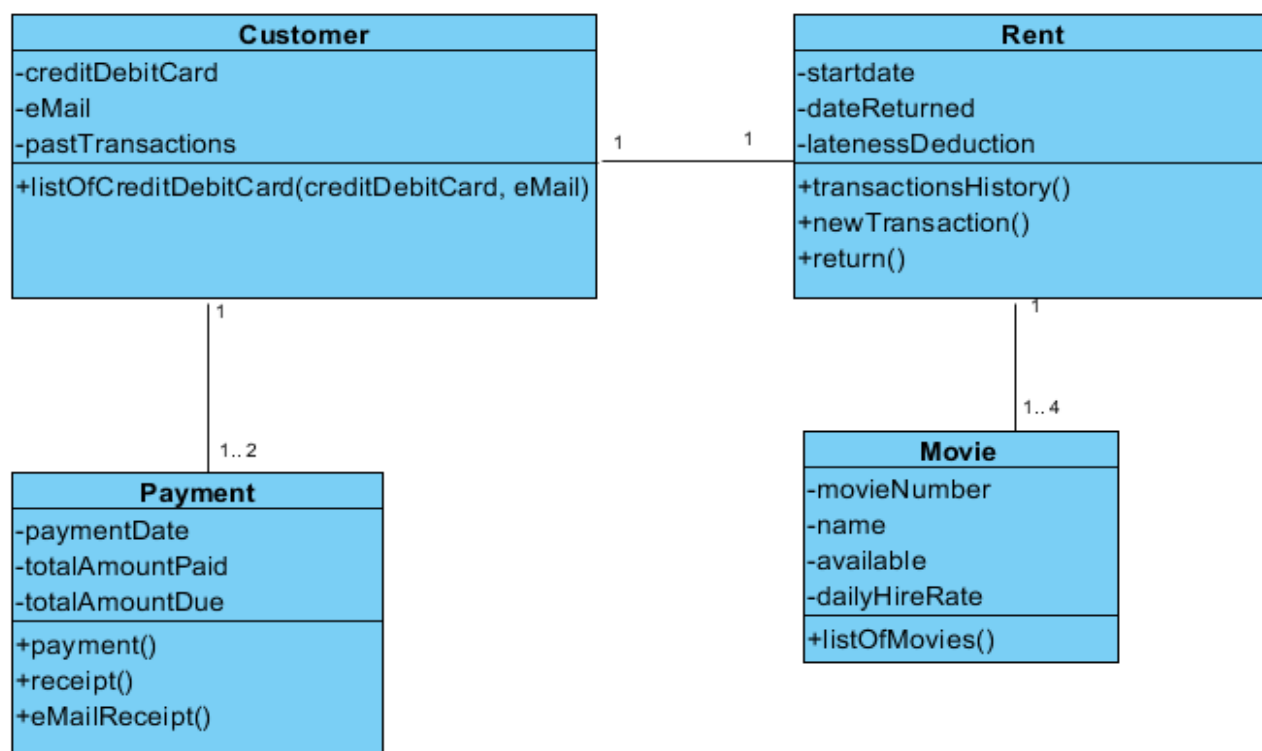
The new system will no long require a membership to track the activities of the

customers. for now the customers will be identified by their credit/ debit cards, so the class customer exists but their attributes are creditDebitCard, email, PastTransactions. This class also will be generation a list of the customers by identifying their credit/ debit card number.

The next class that have been identified is the Rent class. And the attributes are: startdate, dateReturned and latenessDeduction. This class also will work out the transactions history, new transactions and return of movies.

The Payment class will contain the payment date, the total amount paid, the total amount due (in case a lateness return occur). This class also will handle the payment process, generation of receipt and process of E-mail the receipt.

The last class that have been identified is the Movie class, this class will handle the list of movies, and will store the movie number, name, available and daily hire rate.



Class diagram for the Xtra-Vision vending system

CRC cards:

The next step that we take is to identify the responsibility of each class and their collaboration to other classes, at this moment we are able to identify and balance the responsibility between classes. Bellow we find the proposed set of CRC cards to the Xtra-Vision vending machines system.

CRC cards to the Xtra-Vision vending machine

Customer	
Responsibility	Collaborator
Provide customer information Keep history of rent	Rent

Payment	
Responsibility	Collaborator
Keep record of all payments Calculate total to be paid Generate receipt E-mail receipt	

Rent	
Responsibility	Collaborator
Provide details of rent transaction Provide details of transaction history Deal with the return process and calculate any extra fee	Customer Movie

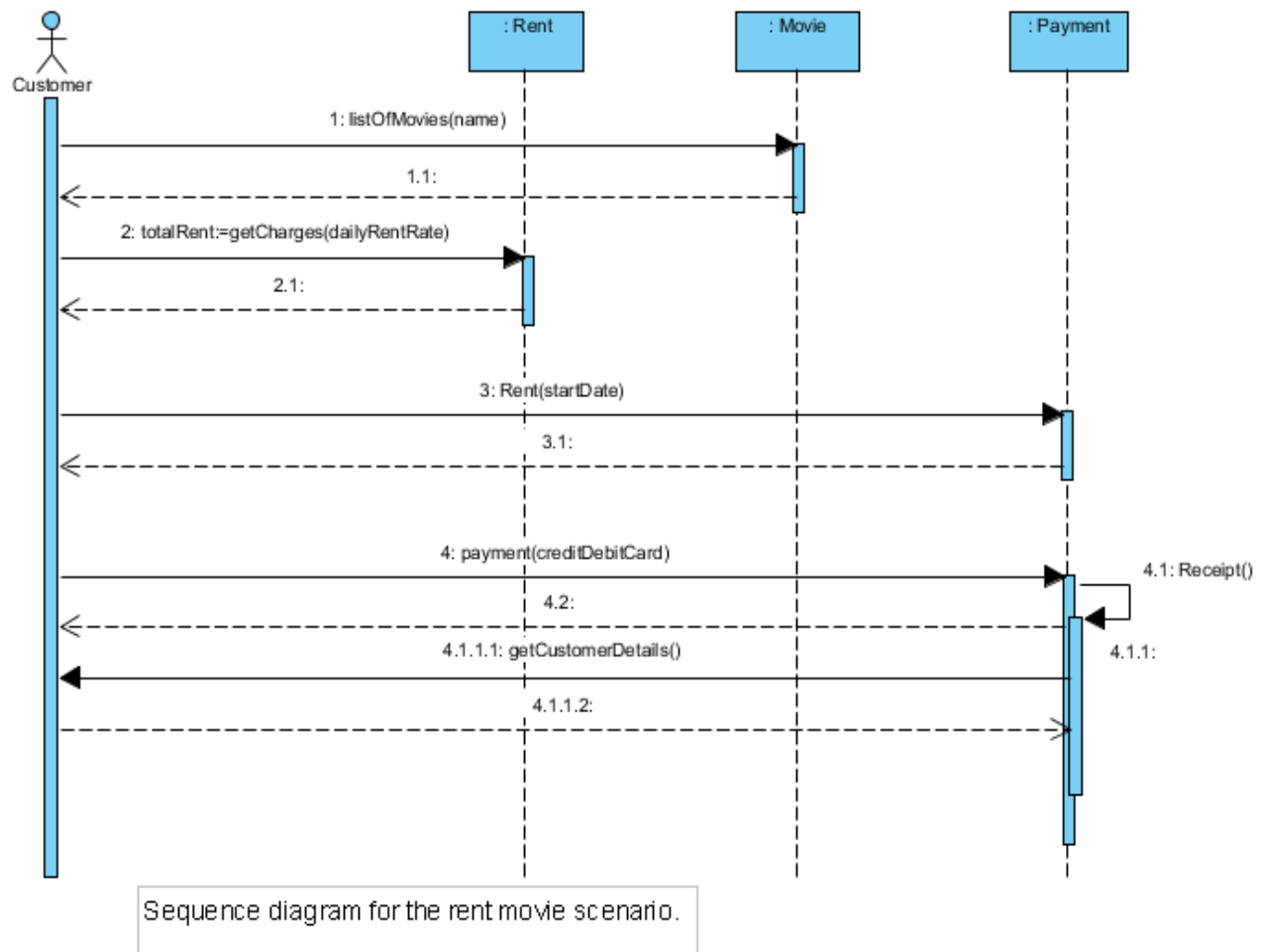
Movie	
Responsibility	Collaborator
Keep record of all movies	

Sequence diagram:

The sequence diagram will show each message and return message between the customer and the system. And to describe how the system will process all the customer requests, we need to visualize how the process of rent a movie at the vending machine happen and also cover all the requirements for the new system.

Based in a successful scenario for the use case rent a movie we can define a process where:

- 1 - The customer arrives at a vending machine and select the option to rent a movie.
- 2 - The screen shows the list of movies.
- 3 - The system confirms that this movie is available and displays the daily rate and number of the movie.
- 4 - The system ask if the customer wants to rent more movies, up to 2 for a first rent and up to 4 for a regular customer.
- 5 - The customer select "no" and the system displays the total rent.
- 6 - The customer agree and the system enter the start date.
- 7 - the customer insert the credit/debit card into the machine and pay the amount.
- 8 - The system ask if the customer needs a receipt. If "yes" the customer should type the E-mail address.



Research question

Modelling techniques have been used for many year, it can be applied in many areas of knowledge, for example: to engineers construct a airplanes, houses, cars and so on.

Basically modelling techniques are an abstract way to represent the final product that a professional is intended to create. This is the highest level of description of a system. By modelling a system the project becomes cheaper and more secure than it would be if the system was created without the modelling bases.

For this continuous assessment we describe modelling techniques to aid the

development process of a software using the Unifies Modeling Language (UML).

We decide to model a system before start the development process, because a given system is complex enough or contain risks that have to be controlled. For example, now a day many systems are very complex or with risks, even a mobile phone contain complex applications that manages or give sources to other applications, and those applications also have information about users that should not be available to third parts. Then it is normal to make the developer lost between the lines of code, and by consequence the project starts to be expensive and insecure, and takes to long time to be completed. Also if the project is not modeled, the developer can make many mistakes that will be hard to be fixed at the end, those mistakes are often very expensive to be fixed and the project can be unusable.

By the other hand when we model a system, we can have a high level of abstract view of every part that have to be built, so, when we have all the documentation of the model process, we can analyse and make changes on it before start coding.

Modelling techniques also can tell us if the system can be built or not, or if we are in the right way or if we should change some concepts before initialize the development process. Some times it is possible to recreate all the model again in order to correct any mistake of include any extra requirement that could appear during the modeling process.

The "use case model" is a example of a basic model technique to break a system into manageable parts, this model can describe the real world in order to apply the specific needs into a computer system. It contains a set of solutions that describes the real world step by step, and those information goes to diagrams that will model all the class that will be used to develop the system. Those class will be identified with their responsibilities and their contribution to other classes. So when the team of developers are creating all the algorithms to produce the system each member is able to follow the requirements more accurately.

The modelling techniques can help the developer to design the entire software before

start code. Ensure that the system is been built in the right path, and also, the developer is able to follow up all the steps of the development process, so if any part of the project needs to be changed it is the right time to do, before the system goes to the final user.

It also help to integrate system, for example, when a project will be part of a existent or bigger system. then, it will be easy to be developed. Years ago, developers were spending a lots of time to integrate systems, and some times it was impracticable because the systems were so difficult to be read or because the functions were not matching with the functions of the new system.

Models also makes the developer understand better the business or engineering system, so the software system will be as better as possible to represent the real world . The design of the system will be also improved and the visualization of the codes and implementations will be improved too.

When we consider to model a system we are identifying the factors of complexity and risk that a system have. For example: a software for a bank, is very complex and contain high level of risk, so the team of developers that is taking the project have to spend the necessary time to model all the requirements to complete the project. A other example is a medical software. This type of software use to have a extreme high level of risk, and any computation that the software will do have to be well planed and tested.

The modelling techniques can not spend to log time to be completed, because the project still have a specific time to be finished. So, models are tools that aid the developers to work on the project more accurately and possibly integrate any project with a old or future project.

References

IBM (2017) The value of modeling. Available at: <https://www.ibm.com/developerworks/rational/library/6007.html> [Accessed on 17th March 2017]

Microsoft (2017) Modeling for Software Development and Operations. Available at: <https://msdn.microsoft.com/en-us/library/ms954611.aspx> [Accessed on 17th March 2017]

ISTQB Exam Certification (2017) What are the Software Development Models?. Available at: <http://istqbexamcertification.com/what-are-the-software-development-models/> [Accessed on 17th March 2017]

I hereby declare that all of the work is done by my own.

Ennio da Silva Vitor

ID: 2015197