COLLEGE OF COMPUTING TECHNOLOGY - DUBLIN
BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

SYSTEM ANALYSIS & DESIGN

Assessment 1 – Model a Video Rental Machine using UML and CASE tools

Adelo Vieira
Student Number: 2017279
Lecturer: Mr. Michael Weiss
March 25, 2018

Contents

List of Figures

1 Overview of the XtraVision Video project

In this project we describe and carried out some phases of the process of developing a software for a Video rental Machine.

The idea behind this project is model, using UML, the initial phase of the Development process.

Ir order to do so, we are going to start collecting all the information we need to create a List requirements for the new XtraVision Video Rental Machine software. The documentation of the current computer system available on the XtraVision web site.

Having created the List of requirements, we will start with a Use Case Modeling, which is a good starting point to understand the system.

Then, we will generate a Class diagram using a **domain model** approach. Since we are using an Object-Oriented approach, this phase is crucial to have an initial idea of our classes and the relationship between them.

The last part of our modeling will be the creation of a Sequence diagram. This would allow us to understand better the messaging between a collaboration of objects.

2 Problem description

Due to a growth in movie rentals, XtraVision Video has problems with long queues at the counters in their stores. That is why, they have decided to implement a Video rental machine that will take some of the burden from the staff.

XtraVision also expects to increase sales because the machine will be installed in SuperValu Store (a business partner).

3 Requirements engineering

3.1 Elicitation

In order to collect the information we need to create a List of requirements for the new XtraVision Video Rental Machine software, we are going to study the documentation of the current computer system available on the XtraVision web site:

- From the XtraVision Renting a Movie Video (`https://www.youtube.com/embed/s6vPrcAWvws?rel=0`) we will build the **Scenario for the rent of a movie**.
- From the XtraVision Returning a Movie Video (`https://www.youtube.com/embed/dvXoOrw-znA?rel=0`) we will build the **Scenario for the return of a movie**.
- We will also use the «FREQUENTLY ASKED QUESTIONS» (`https://xtra-vision.ie/how-it-works/`) to complement the information obtained from Scenarios.

3.1.1 Scenario for the rent of a movie

1. A customer arrives at the Video rental Machine. The Touchscreen shows two main options:
   - **Touch to RENT**
   - **Touch to RETURN**
2. If the customer Selects the Rent option on the screen → The available movies will be shown on screen.
3. The customer selects the movie it wants to rent by touching on it → The movie selected will be shown on the entire screen with a description of the movie and three main buttons:
   - **Rent**
   - **Back** → To return to the screen where available movies are shown
   - **Cancel**
4. In case the customer press the Rent button → the screen will show the movie along with the rent price and three main options:
   - **Check out** → To go to the payment screen:
   - **Back**

- **Cancel**

5. In case the customer press the Check out button → the screen will show a message saying «Please insert the card into the reader».

6. When the customer inserts the card into the reader, it will need to follow the indications of the card reader:

    - Enter PIN, then press enter

7. If the banking operation is successful, the movie will be addressed and available on the «Pick Up» panel of the machine

3.1.2 Scenario for the return of a movie

1. A customer arrives at the Video rental Machine. The Touchscreen shows two main options:

    - **Touch to RENT**
    - **Touch to RETURN**

2. The customer selects the RETURN option on the screen (it has to make sure the case is property closed before returning).

3. The customer inserts the disc into the return slot.

4. The customer has to wait for the message confirming the disc has been successfully returned.

5. If the customer is registered, it will receive a return receipt by email.

3.1.3 Frequently asked questions

- What do I need to to rent from an Xtra-vision Xpress Machine?
    - No membership is required, just a credit or debit card. You can provide your email address if you require receipts and details of new movie releases.

- How many discs can you rent at one time?
    - While your first transaction is limited to two discs, after that you can rent up to four discs at one time.

- Will I be charged if I am late returning a disc?

  - If you want to hold onto your disc for longer there is no problem. For each additional day you keep it past 8pm, you'll be charged €1.50. If you hold onto your disc for the maximum rental period – 10 days, you will be charged a maximum fee of €15 and the disc is yours to keep.

If the customer holds the disc for longer than the defined rental period, for each additional day the customer keeps it past 8pm, it'll be charged certain amount of money (currently €1.5). If the customer holds the disc for the maximum rental period (currently 10 days), it will be charged a maximum fee (currently €15) and the customer can keep the disc.

- How does the billing process work?
  - The charge is processed as soon as you rent your disc. Any additional charges are charged when a disc is returned.

- **Will receipts for rentals and returns be provided?**
  - Printed receipts will not be issued, however you will receive email confirmations for rentals and returns listing all charges.

- **What if my disc does not play?**
  - If problems persists, please contact us for further assistance.

3.2 Specification

Source	**Description**	**Nouns (possible objects)**
Scenario for the rent of a movie Scenario for the return of a movie	The system will have two main functions: - Rent a movie - Return a movie When a customer arrives to the Rental Machine, the Main Screen of the machine will show two main options: - Touch to rent - Touch to return	The System Movie Rental Machine Screen of the Machine
Scenario for the rent of a movie Scenario for the return of a movie	**Keep a list of movies:** Keep a list of available movies and their details including ID, Title, Director, Release date, Description and Price. In order to do that, the system has to be able to communicate with a bar code reader device of the Machine. Through this device the system will keep track of discs. Keep track of rented movies details: Movie ID, Rental date, Customer bank details, Customer email address (if entered). The list will be automatically update when a disc is rented or returned	**List of available movies**: ID, Title, Director, Release date, Description and Price. Bar code reader device of the Machine Discs Rented movies details: Movie ID, Rental date, Customer bank details, Customer email address
Scenario for the rent of a movie Scenario for the return of a movie	**Show available movies and allow customers to select them:** In order to do that, the system has to be able to communicate with the TouchScreen of the Machine.	Customers TouchScreen of the Machine.
Scenario for the rent of a movie Scenario for the return of a movie	**Issue a disck:** The system has to be able to communicate with the device of the Machine responsible for Picking the movie to giving to the customer	Device of the Machine responsible for Picking the movie to giving to the customer
Scenario for the rent of a movie Scenario for the return of a movie	**Take a disck back:** The system has to be able to communicate with the device of the Machine responsible for take the movie back when the customer is returning the movie	Device of the Machine responsible for take the movie back when the customer is returning the movie
Scenario for the rent of a movie Scenario for the return of a movie	**Process payments:** The system has to be able to communicate with a credit card company system through the card reader of the Machine (only card payment will be accepted).	**Payments** Credit card company system Card reader of the Machine
Frequently asked questions	The charge is processed as soon as the customer rent the disc. Therefore, the system has to calculate the charge for all rented movies. Any additional charges are charged when a disc is returned	The charge Rented movies Additional charges
Frequently asked questions	If the customer holds the disc for longer than the defined rental period, for each additional day, it'll be charged certain amount of money (currently €1.5). If the customer holds the disc for the maximum rental period (currently 10 days), it will be charged a maximum fee (currently €15) and the customer can keep the disc.	Rental period Additional day Amount of money Maximum rental period Maximum fee
Frequently asked questions	Receipts: The system has to be able to create an electronic receipt and sent it to the email address entered for the customer	Receipts Electronic receipt Email address
New requirement	The new system has to be able to print a paper receipt if the customer asks for it. To do that the system has to communicate with the Printer of the Machine	Paper receipt Printer of the Machine

Table 1: Problems and requirements list

3.3 Validation

4 Use case diagrams

One of the first things that is useful to do when starting the development process is to model how the **System** interact with the **Users**. In System analysis this is call a **Use case model**, which more important part is the **Use case diagram**.

4.1 Identifying actors

- The Machine:
  - The bar code reader device of the Machine: Through this device the system will keep track of discs.
  - The TouchScreen of the Machine: To show available movies and allow customers to select them.
  - The device of the Machine responsible for Picking the movie to giving to the customer.
  - The device of the Machine responsible for take the movie back when the customer is returning the movie.
  - The printer machine that is going to print the paper receipts.
- The credit card company system (It includes the card reader): This system will communicate with our system in order to precess payments.
- The administrator of the Machine: the person who, among other things, is responsible of:
  - Enter new movies to the machine
  - Replace damage movies
  - Update list of movies
- The customer

4.2 Identifying Use cases

- Maintain movie list
- Show available movies
- Handle movie selection
- Issue movie

- Handle movie return
- Process payment
- Issue receipt:
  - Print paper receipt
  - Sent e-receipt by email

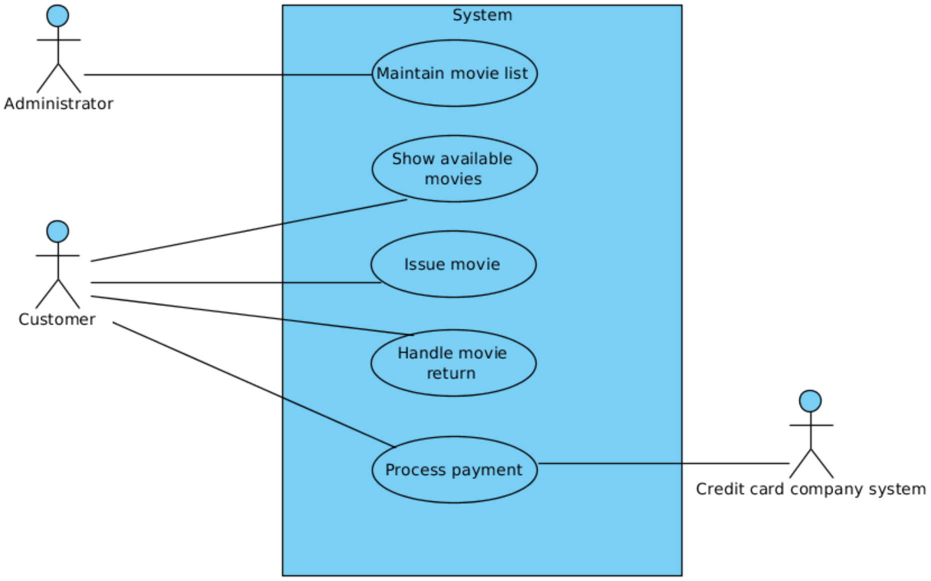

Figure 4.1: Use case diagram

5 Class diagram

We use the Class diagram to model classes and the relationships between classes, and also to model higher-level structures comprising collections of classes grouped into packages. [Britton and Doake (2004)]

To build our Class diagram, we are going to use a **domain model** approach. The stages of this approach are:

- Identify the objects and derive classes
- Identify attributes
- Identify relationships between the classes
- Write a data dictionary to support the class diagram
- Identify class responsibilities using CRC cards

- Separate responsibilities into operations and attributes
- Write process specifications to describe the operations

5.1 Identify the objects and derive classes using noun analysis

After organizing the nouns in different categories, we obtained the following list:

- **Movie:** ID, Title, Director, Release date, Description
- **Hire:** Movie ID, Rental date, Initial Rental period, Maximum rental period
- **Payment:** Initial charge, Additional charges, Maximum fee, bank details, Customer email address
- **Credit card company system (Card reader of the Machine)**
- **The Machine:**
  - TouchScreen of the Machine
  - Bar code reader device of the Machine
  - Device of the Machine responsible for Picking the movie to giving to the customer
  - Device of the Machine responsible for take the movie back when the customer is returning the movie
  - Printer of the Machine
- **Receipts:**
  - Electronic receipt
  - Paper receipt
- **Customers**

As explained by Britton and Doake (2004), Hire could be an association between Customer and Movie. However, «the general rule in this situation is that if there is data associated with the relationship (and in this case there is), then we probably want to model it as an object» [Britton and Doake (2004)]

Receipt doesn't qualifies because, as explained by Britton and Doake (2004), *is too tied up with physical inputs and outputs.*

Customers doesn't qualify because we don't want to store customer details. The only attributes related to customers are *Bank details, and Email address* but they are related to the payment process.

The candidate objects that result from picking out all of the nouns and noun phrases of the list of requirements are:

- **Hire:** movieID, rentalDate, initialRentalPeriod, maximumRentalPeriod
- **Movie:** id, title, director, releaseDate, description.
- **Payment:** initialCharge, additionalCharges, maximumFee, bankDetails, customerEmail
- Machine
  - TouchScreen
  - BarCodeReader
  - PickingMovieDevice
  - TakebackMovieDevice
  - Printer

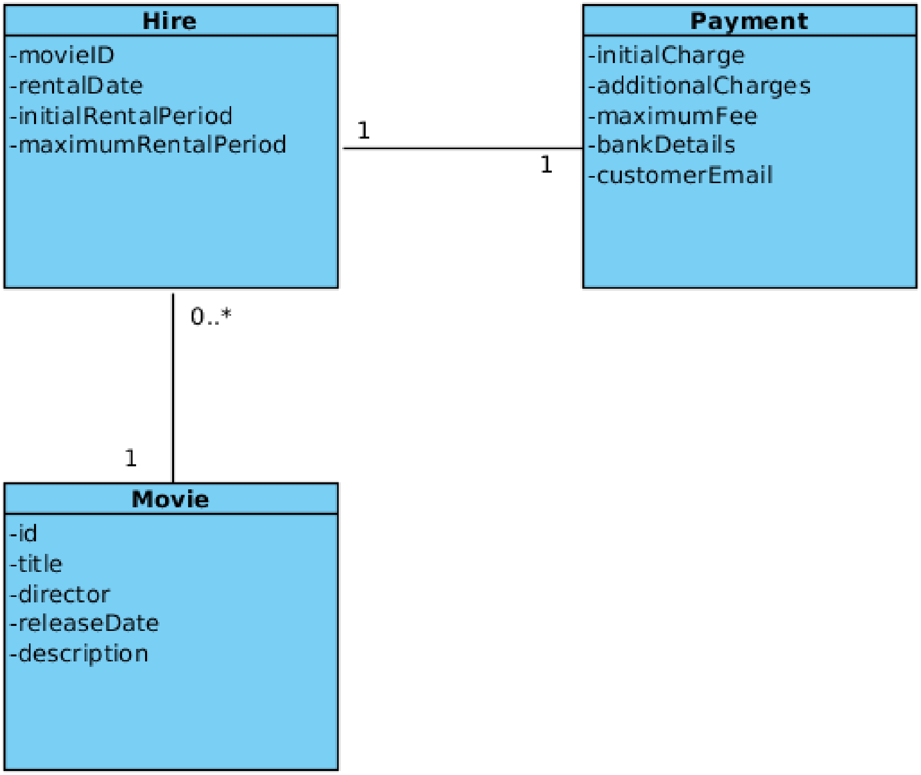

Figure 5.2: Class diagram

We will also need to create some classes that will allow the communication between the software and the different devices that make up the machine. A first approximation of these classes is shown in the Figure 5.3.

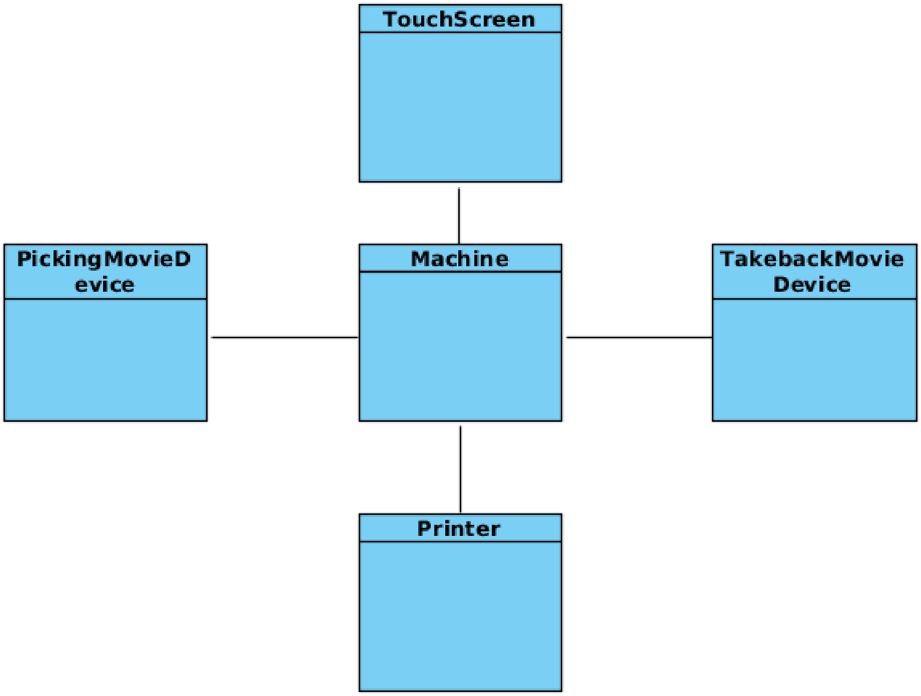

Figure 5.3: Class diagram

6 Sequence diagram

7 Research question

Software systems can start off simple and quickly grow to be quite complex and therefore when developing a computer system, the developer will often be dealing with a certain amount of complexity. **Explain (in your own words) how modelling techniques manage complexity in the development of computer based systems.** Your answer should include a description of how models can support software developers who are striving to understand the complexity of the systems that they are developing. Be sure to include factors that are typically used to deal with complexity when analysing a system during the UML modelling process.

To answer this question, let's start from the beginning. Even if the questions refers to modelling techniques, we would like to start by giving a picture of how a software system is developed.

7.1 The System Life Cycle

As described by Britton and Doake (2004), every large project needs «some kind of **Framework** in order to help identify milestones, structure activities and monitor deliverables.»

«In software system development, a framework has traditionally been known as a **System Life Cycle Model**». The traditional system life cycle is composed by:

- Requirements,
- Analysis,
- Design,
- Implementation,
- Installation. [Britton and Doake (2004)]

The **Object-Oriented Life Cycle Approach** considers the following phases:

- Inception,
- Elaboration,
- Construction,
- Transition. [Britton and Doake (2004)]

7.2 The Development Method

However, «a life cycle provides a high-level representation of the stages that a development project must go through to produce a successful system, but it does not attempt to dictate how this should be achieved, nor does it describe the activities that should be carried out at each stage» [Britton and Doake (2004)]. That is why the concept of **Development Method** has been developed. «A development method is much more prescriptive, often setting down in detail the tasks, responsibilities, processes, prerequisites, deliverables and milestones for each stage of the project». [Britton and Doake (2004)]

Nowadays, the most popular development method is the **Rational Unified Process (RUP)**. This method allows the development process to be automated and structured, which is crucial in large and complex projects.

One of the most important tools in the RUP is the incorporation of **Modelling Techniques** from the **Unified Modelling Language (UML)**. As said by Britton and Doake (2004), in our context, modelling refers to the use specialized diagrams to model the system.

The object-oriented approach also recognizes fully the reality of iterative development.

Having said that, we are now going to explain how modelling techniques manage complexity when a software system is developed.

To understand the functionality and complexity of a system, UML propose a series of diagrams that allow the developers to have a picture of the whole system. The developer who is trying to understand the system usually start with the simplest diagrams. In our case, for example, the simplest diagrams is the Use case diagram in Figure 4.1. However, it is possible to generate many other diagram in order to understand a particular aspect of the system with more details. [Erickson and Siau (2004)]

The use case model, for example, allows to model how the **System** interact with the **Users**. This is very useful to understand the functionality of the system and to detect problems.

The Class diagram provides an excellent starting point to visualize classes and the interaction between then.

The UML defines many ways to model a system. However, the models we have to carried out in a particular project depends on the complexity of the system. In many cases, complexity depends on interaction between objects. That is way, the sequence and collaboration diagrams is a very useful tool when complexity is controlled by interaction between objects. [Britton and Doake (2004)]

7.3 The reality of iterative development

One of the most important thins we have learned while doing this exercise is the reality of iterative development. Even for this little exercise, we realized that is impossible to carried out in a ordered way without having to go back to previous. As mentioned by Britton and Doake (2004), «the developer may have to revisit a range of workflows several times during one phase of development».

Declaration

I hereby declare that all of the work shown here is my own work.

Student's Name: Adelo Vieira

Student Number: 2017279

Date: March 25, 2018

Bibliography

Carol Britton and Jill Doake. *A student guide to object-oriented development*. Butterworth-Heinemann, 1 edition, November 22 2004. 7, 8, 11, 12

John Erickson and Keng Siau. *Theoretical and Practical Complexity of UML*. Americas Conference of Information Systems (AMCIS), 2004. 12

Lenovo Web site. *Rack Servers*. URL `https://www3.lenovo.com/ie/en/data-center/servers/racks/c/racks`.

Wikipedia. *RAID*. URL `https://en.wikipedia.org/wiki/RAID`.