Module: Network Software Engineering

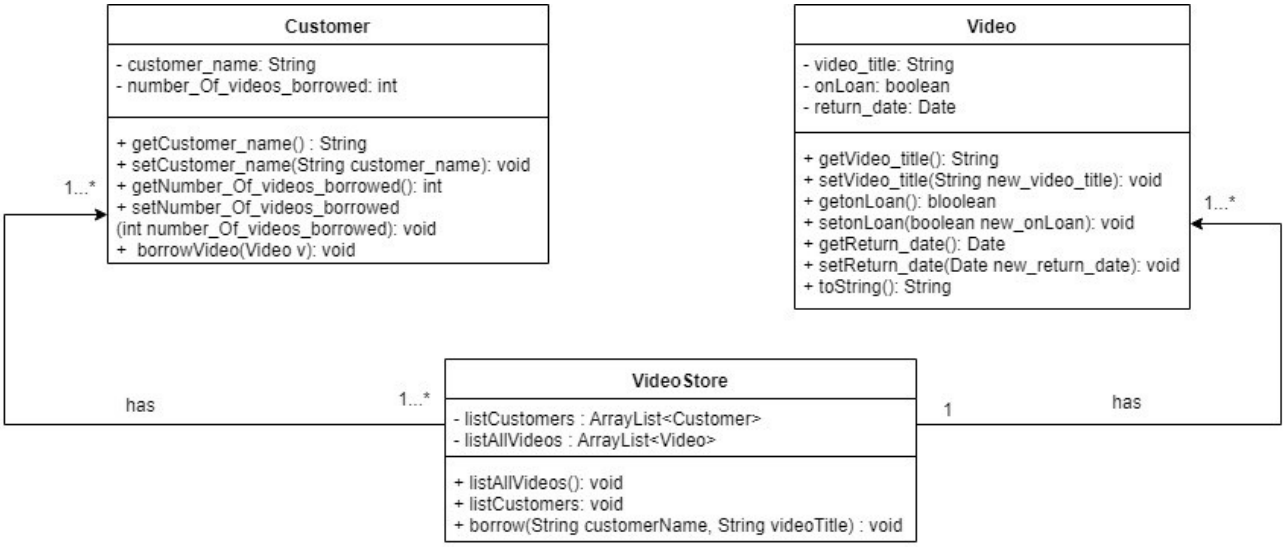
Video Store

Student ID: W1669150

Name: Valdir Santos Fonseca

Date:07/01/2019

Class Diagram



Class Customer

Fields	private String customer_name	The name of the customer that is registered in the Video Store database
	private int number_Of_videos_borrowed	Number of videos that the customer borrowed from the Video Store
Constructor	public Customer(String new_customer_name, int new_number_Of_videos_borrowed)	create a new customer with a new name and set the number of videos borrowed from the Video Store
Methods	public String getCustomer_name()	Return the current name of the Customer
	public void setCustomer_name(String customer_name)	Receive a new Customer name
	public int getNumber_Of_videos_borrowed()	Return the current number of videos borrowed from the Video Store
	public void setNumber_Of_videos_borrowed(int number_Of_videos_borrowed)	receive a new video borrowed from a customer
	public void borrowVideo(Video v)	receive a Video and check if it's on Loan or not, if it's on Loan it set the return date and if the number of borrowed it´s greater than 2 the customer receives a warning that indicate that the maximum was reached. The same method updates the number of video borrowed if the customer borrows one or more

Class Video

Fields	private String video_title	Name of the Video available on the shelf at the Video Store
	private boolean onLoan	It set if a video is on Loan or not
	private Date return_date	Date that the video borrowed has to be return
Constructor	public Video(String new_video_title)	create a new Video with the Title of the video without being in Loan and hasn't the return date
Methods	public String getVideo_title()	return the current Title of the Video
	public void setVideo_title(String new_video_title)	set a new Video title
	public boolean getonLoan()	Return true or false if a video is on Loan or not
	public void setonLoan(boolean new_onLoan)	Set true or false to a Loan of a Video, if it's going to be on Loan or not
	public Date getReturn_date()	return the video date when is due to be return
	public void setReturn_date(Date new_return_date)	set the date to return the video
	public String toString()	return if the video is on Loan or not and if it's on Loan it set the due date to be return

Class Video Store

Fields	private ArrayList<Customer> listCustomers ->	List of the customers on the Video Store's database
	private ArrayList<Video> listAllVideos	ArrayList of the Videos available in the Video Store
Constructor	public VideoStore()	Initialise the List of the Customers and the Videos available in the video store using an ArrayList
Methods	public void listAllVideos()	show the List of all videos available or not in the Video Store
	public void listCustomers()	Show the List of all Customers available
	public void borrow(String customerName, String videoTitle)	it receives a name and a title of a video then it checks if they exist already in the database or not

Class VideoTest

Methods	public static void main(String [] args)	At the Main method I created a new Video Store, and from there using a for loop to present the Menu with the option to List the Videos, borrow a video and Quit the menu. To borrow a video, we need to introduce the customer name already in the Video Store database and which video we wish to borrow.

Print out of the Java source code

Class Customer

```
package videostore;
import java.util.*;
import java.io.*;
import java.text.*;
import java.time.*;
import java.time.format.DateTimeFormatter;
/**
 * @author w1669150 - Valdir Fonseca
 */
public class Customer
{
   /*
   *
   */
   private String customer_name;
   private int number_Of_videos_borrowed;

   /**
    * Constructor create a new customer with the  name and number of videos
    * borrowed
    * @param new_customer_name
    * @param new_number_Of_videos_borrowed
    */
   public Customer(String new_customer_name, int new_number_Of_videos_borrowed)
   {
      this.customer_name = new_customer_name;
```

```
    this.number_Of_videos_borrowed = new_number_Of_videos_borrowed;
}

/**
 * @return the customer_name
 */
public String getCustomer_name() {
    return customer_name;
}

/**
 * @param customer_name the customer_name to set
 */
public void setCustomer_name(String customer_name) {
    this.customer_name = customer_name;
}

/**
 * @return the number_Of_videos_borrowed
 */
public int getNumber_Of_videos_borrowed() {
    return number_Of_videos_borrowed;
}

/**
 * @param number_Of_videos_borrowed the number_Of_videos_borrowed to set
 */
public void setNumber_Of_videos_borrowed(int number_Of_videos_borrowed) {
    this.number_Of_videos_borrowed = number_Of_videos_borrowed;
}

        /**
```

```
        * borrowVideo method checks if a video is on Loan or not and if the
     * customer reached is limit of borrowing
        */
   public void borrowVideo(Video v)
   {
      if(v.getonLoan())
        {
          SimpleDateFormat dateReturn= new SimpleDateFormat("dd-mm-yyyy");
          System.out.println("The Video"+v.getVideo_title()+ " is on Loan and the return date is
on"+dateReturn.format(v.getReturn_date()));
        }
        else if(getNumber_Of_videos_borrowed()>=2)
        {
          System.out.println("The Customer "+getCustomer_name()+" achieve the maximum limite of
borrowing");
        }

      LocalDate localDate = LocalDate.now();

      v.setonLoan(true);
     // v.setReturn_date(LocalDate.now());

      SimpleDateFormat dateReturn= new SimpleDateFormat("dd-mm-yyyy");
      System.out.println("The return date is on "+dateReturn.format(v.getReturn_date()));
      setNumber_Of_videos_borrowed(getNumber_Of_videos_borrowed()+1);
      System.out.println(getCustomer_name()+" borrowed the video "+v.getVideo_title());

   }
}
```

Class Video

```
package videostore;
import java.util.*;
import java.text.*;
import java.io.*;

/**
 *
 * @author w1669150 - Valdir Fonseca
 */
public class Video {

   /**
    * @param title the Video's title
    * @param onLoan the date of the loan
    * @param return_date the date of return of the video
    */
   private String video_title;
   private boolean onLoan = false;
   private Date return_date;

   /**
    * Constructor of the class Video
    * @param new_video_title
    */
   public Video(String new_video_title)
   {
      this.video_title = new_video_title;
      this.onLoan = false;
```

```
    this.return_date = null;
}

/**
 * @return the video_title
 */
public String getVideo_title()
{
    return video_title;
}

/**
 * @param video_title the video_title to set
 */
public void setVideo_title(String new_video_title)
{
    video_title = new_video_title;
}

/**
 * @return onLoan state
 */
public boolean getonLoan()
{
    return onLoan;
}

/**
 * @param onLoan the onLoan to set
 */
public void setonLoan(boolean new_onLoan)
{
```

```
        this.onLoan = new_onLoan;
    }

    /**
     * @return the return_date
     */
    public Date getReturn_date()
    {
        return return_date;
    }

    /**
     * @param return_date the return_date to set
     */
    public void setReturn_date(Date new_return_date)
    {
        return_date = new_return_date;
    }

    /**
     * method toString() return weather the video is available or not
     */
    public String toString()
    {
        SimpleDateFormat myFormat= new SimpleDateFormat("dd-mm-yyyy");
        if(getonLoan())
            return "The Video "+video_title+" is on loan and the return date is on
"+myFormat.format(return_date);
        else
            return "The Video "+video_title+"\n is available on the shelf\n";
    }
}
```

Class Video Store

```
package videostore;


import java.util.*;
import java.text.*;
import java.io.*;
/**
 *
 * @author w1669150 - Valdir Fonseca
 */
public class VideoStore
{
   private ArrayList<Customer> listCustomers = new ArrayList<Customer>();
   private ArrayList<Video>  listAllVideos = new ArrayList<Video>();


   /**
    *
    */
   public VideoStore()
   {
      listAllVideos.add(new Video("The Kung Fu Panda"));
      listAllVideos.add(new Video("The Kung Fu Panda 2"));
      listAllVideos.add(new Video("The Kung Fu Panda 3"));
      listAllVideos.add(new Video("Resident Evil"));
      listAllVideos.add(new Video("Jumanji"));
      listAllVideos.add(new Video("Conjuring"));
      listAllVideos.add(new Video("The Lion King"));
      listAllVideos.add(new Video("Despicable me"));
      listAllVideos.add(new Video("Tarzan"));
```

```
    listCustomers.add(new Customer("Jonh",0));
    listCustomers.add(new Customer("Susie",0));
    listCustomers.add(new Customer("Peter",0));
    listCustomers.add(new Customer("Jessica",0));
    listCustomers.add(new Customer("David",0));
    listCustomers.add(new Customer("Natalie",0));
    listCustomers.add(new Customer("Andrew",0));
    listCustomers.add(new Customer("Brian",0));
    listCustomers.add(new Customer("Steve",0));



}

/**
 * List All Videos available
 */
public void listAllVideos()
{
    System.out.println("List of Videos: ");
    for(int i = 0; i < listAllVideos.size(); i++)
    {
        System.out.println(i+1+" - "+listAllVideos.get(i));
    }
    System.out.println();
}

/**
 * List all Customers
 */
public void listCustomers()
{
    System.out.println("List of Customers: ");
```

```
    for(int i = 0; i < listCustomers.size(); i++)
    {
        System.out.println(i+1+" - "+listCustomers.get(i).getCustomer_name());
    }
    System.out.println();
}


public void borrow(String customerName, String videoTitle)
{

    Customer cust = new Customer(" ",0);
    Video v = new Video(" ");
    for(int i = 0; i < listCustomers.size();i++)
    {
        for(int j = 0; i < listAllVideos.size();j++)
        {
            if(listCustomers.get(i)==null)
            {
                System.out.println("Customer "+customerName+" doesn't exist!!");
                return;
            }
            else if(listCustomers.get(i).getCustomer_name().equalsIgnoreCase(customerName))
            {
                cust = listCustomers.get(i);
                break;
            }

            if(listAllVideos.get(j).getVideo_title().equalsIgnoreCase(""))
            {
                System.out.println("Video "+videoTitle+" doesn't exist!!");
            }
```

```
            else if(listAllVideos.get(j).getVideo_title().equalsIgnoreCase(videoTitle))
            {
               v = listAllVideos.get(j);
               break;
            }
         }
      }
      cust.borrowVideo(v);
   }
}
```

Class VideoTest

```
package videostore;
import javax.swing.*;
import java.util.Scanner;
/**
 *
 * @author w1669150 - Valdir Fonseca
 */
public class VideoTest
{

   public static void main(String [] args)
   {
      VideoStore v1 = new VideoStore();
      Scanner in = new Scanner(System.in);

      String answerMenu = " ";
      for(;;)
      {
         // Show the Menu
         System.out.println("Menu:\n L-List\n B-Borrow\n Q-Quit:\n");
         System.out.print(" --> ");
         answerMenu = in.nextLine();
         System.out.println();
         try
         {
         if(answerMenu.equalsIgnoreCase("L"))
            v1.listAllVideos();
         else if(answerMenu.equalsIgnoreCase("B"))
         {
```

```
            System.out.println("Please Introduce the customer's name: ");
            String answerName = in.nextLine();



            System.out.println("Please Introduce the Title of the Video you wish to Borrow: ");
            String answerVideo = in.nextLine();


            v1.borrow(answerName, answerVideo);
            System.out.println();
        }
        else if(answerMenu.equalsIgnoreCase("Q"))
              System.exit(0);
        }catch(Exception e)
        {
           System.out.println("["+e+"]");
        }


      }
   }
}
```

References

- MKYONG. JAVA – HOW TO GET CURRENT DATE TIME. [ONLINE] 2016. AVAILABLE FROM: HTTPS://WWW.MKYONG.COM/JAVA/JAVA-HOW-TO-GET-CURRENT-DATE-TIME-DATE-AND-CALENDER/ [ACCESSED: JAN 6TH 2019].
- Ambisoft Inc. Uml 2 Class Diagrams. [Online] 2003. Available from: http://www.agilemodeling.com/artifacts/classDiagram.htm#ConceptualClassDiagrams