PLANNING AND IMPLEMENTING AN OBJECT-ORIENTED SOFTWARE SYSTEM

Pairs Practical Project

Abstract

The Xtra-Vision company has signed a contract with your team to design and write the software for their Kiosk. Your team will be combining skills to design and implement a Java program that will provide the functionality of the system described.

BSc of Science in Information Technology – Year 2

Minh Truong Van– 2018420
Gisele de Lima Lopes – 2018076

Contents

1. Reflection about the tasks

The pair study the system and report requirements and with virtual meeting, they sharing the impressions and perceptions about the tasks. Each member shares their own notes about what should de done and a big picture of the project, beginning of the problem definition and list of requirements were drawing.

About the OOC activities division, the creation of the GitHub repository, part of the Java Classes planning, the database connection, exclusion and changes in some classes and creation of the Views and database tables were done by Gisele. The part of the Java Classes planning, creation of the folders, basis Java classes in NetBeans and constructors were done by Minh.

Related to SAD division, after the problem definition and list of requirements be done in time the tasks were 2 use cases diagram to each team member and correction to adequate of one narrative. Class and sequence diagrams done by Gisele, state diagram done by Minh. Activity and User story diagrams and research 2 were done in collaborative way, with previous drawings of each member and later review.

The team has problems to deliver a functional Java program, but we decided don't postponed the deliver and we expect that the part submitted could be partially graded. Was a challenge to put in use what we planned to do and what we learned during the course.

2. SAD class requirements

2.1. Problem definition

Client: Xtra-Vision

Current Scenario:

Xtra-Vision would like a software system to be used by their rent points to automate the activity of rent and return movies cases. For that, the customers will interact with the software in the rent points to select the movie and do the payment.

Objective of the system:

- The rent point should be autonomous and easy operation, making possible to the customers the operation without assistance.
- The system should not require complex subscription, it should be optimized for the customers.
- The system should be effective for the customers satisfaction.

Scope of the system:

The system should incorporate the overall operations:

- The customer chooses a movie from a list.
- The customer rent a selected movie and collect it on the indicated slot.
- The customer returns the movie on the indicated slot.
- Receipt by e-mail communication.
- Payment by card.

2.2. List of requirements

List of requirements regarding the Xtra-Vision system:

- Welcome screen with the services options rent and return.

- Rent screen with a list of movies available.
- Movie details screen with movie code, picture, synopsis and rent button.
- Selection movie screen, with selected movie, unit price, total price, movie selection screen button and proceed payment button.
- Payment screen, with field to insert the card number and name, with e-mail receipt option.
- The system must keep e-mail and the card data for billing and requirements check.
- The system should validate by the card records if it is the first rent and limited the rent to two movies, if not it is up to four movies simultaneous.
- The system should proceed the billing process at 8pm daily. If more or equal to fifteen euros the billing process should be stopped, and receipt purchase should be sent.
- Return screen should inform the customer to check the movie case and insert it on indicated slot.
- Thank you and return confirmation screen.
- Return e-mail receipt should be sent.
- The system must stop the billing process for the movie and return it to the available list.

2.3. Use Case Diagrams

This section presents four Use Case Diagrams that describe four different scenarios in which a user will use the Xtra-Vision movie kiosk.

- **Use Case 1**

Use-Case Name: First time rent and one movie.

Trigger: The customer goes to rent point and interact with the system.

Preconditions: The rent point should be on, connected and available to use.

Primary Actor: Customer

Secondary Actor: -

Main Success Scenario (MSS):

1. The customer selects the rent option on screen.
2. The customer selects the movie.
3. The customer checks the movie details.
4. The customer clicks on rent button to proceed.
5. The customer inserts the e-mail, card number and name and click to proceed the payment.
6. The system checks if the e-mail and card data are stored before and the quantity requirements.
7. The system stores the e-mail and card data.
8. The customer takes the movie case.
9. The system removes the movie from available movies list.
10. The system sends by e-mail the receipt end proceed the first day billing process.

Extensions:

6a. Alternative case, if data was stored before the process is not the first time rent, skip the step 7 and goes to step 8.

Guarantee: The customer gets the movie, the daily rate rent is debited on the card, the movie is not available from the movies list.
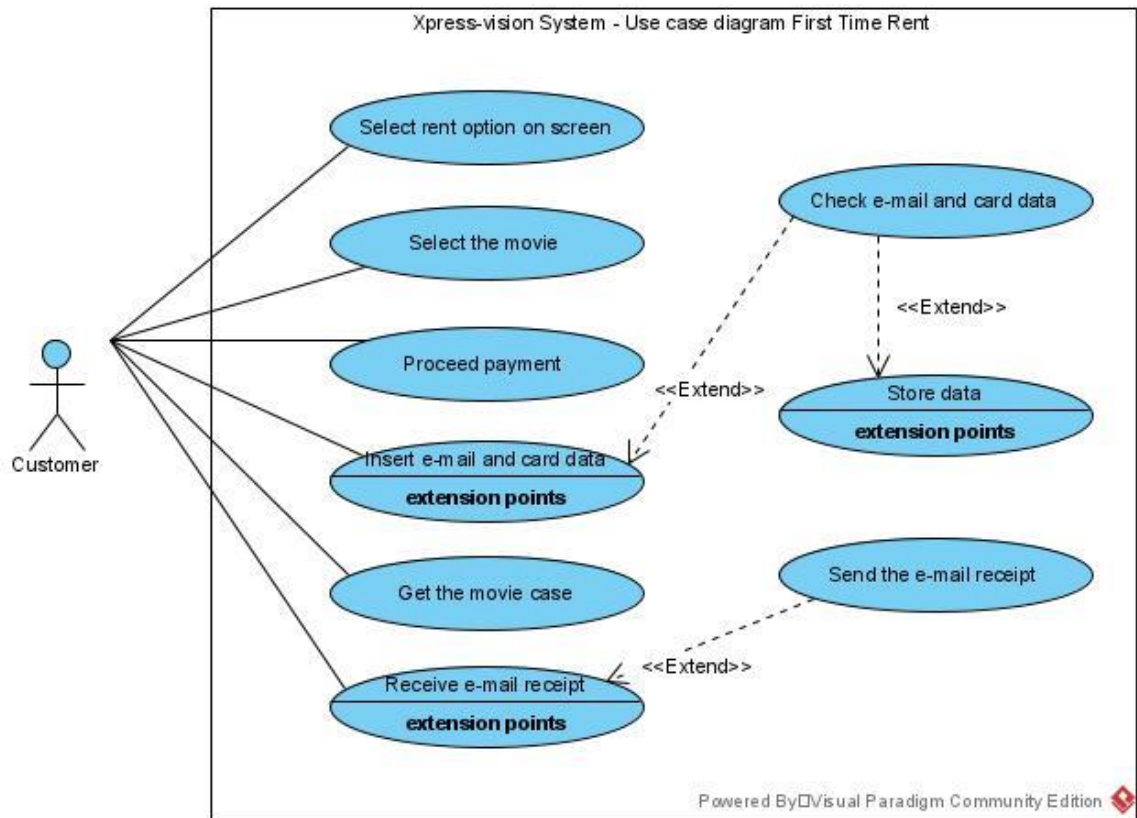
Figure 1.1 Use case diagram First time rent and one movie

- **Use Case 2**

Use-Case Name: First time rent and three movies.

Trigger: The customer goes to rent point and interact with the system.

Preconditions: The rent point should be on, connected and available to use.

Primary Actor: Customer

Secondary Actor: -

Main Success Scenario (MSS):

1. The customer selects the rent option on screen.
2. The customer selects the movies.
3. The customer checks the movies details.

4. The customer clicks on home to proceed another movie choice.
5. (*Repeat steps 2 and 3 three times) The customer clicks on rent button to proceed.
6. The customer inserts the e-mail, card number and name and click to proceed the payment.
7. The system checks if the e-mail and card data are stored before and the quantity requirements.
8. The system informs only two movies are allowed on first rent. The system returns to movies details screen.
9. The customer deletes one movie from the selection and clicks on rent button to proceed.
10. The customer inserts the e-mail, card number and name and click to proceed the payment.
11. The system checks if the e-mail and card data are stored before and the quantity requirements.
12. The system stores the e-mail and card data.
13. The customer takes the movies cases.
14. The system removes the movies from available movies list.
15. The system sends by e-mail the receipt end proceed the first day billing process.

Extensions:

6a. Alternative case, if data was stored before the process is not the first time rent, skip the steps 7, 8, 9, 10, 11 and goes to step 12.

Guarantee: The customer gets the movies, the daily rate rent is debited on the card, the movie is not available from the movies list.
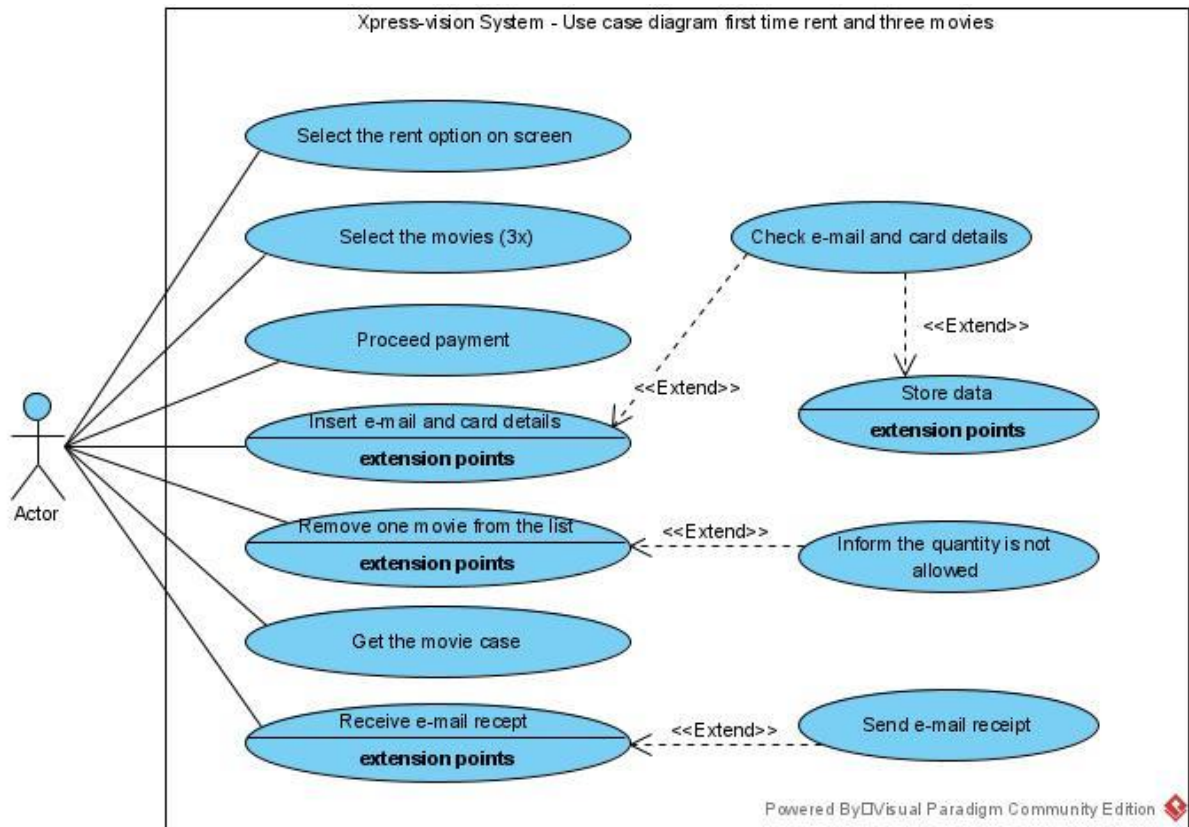
Figure 1.2 Use case diagram First time rent and three movies

- **Use Case 3**

Use-Case Name: Customer does not proceed the rent process.

Trigger: The customer goes to rent point and interact with the system.

Preconditions: The rent point should be on, connected and available to use.

Primary Actor: Customer

Secondary Actor: -

Main Success Scenario (MSS):

1. The customer selects the rent option on screen.
2. The customer selects the movie.
3. The customer checks the movie details.

4. The customer clicks on rent button to proceed.
5. The customer inserts the card number and name but does not insert the e-mail and click to proceed the payment.
6. The system checks the e-mail is null and inform in screen alert message the e-mail needed be filled.
7. The customer leaves the rent point without proceed the rent process.
8. The system returns to Welcome screen after 30 seconds.

Extensions:

5a. Alternative case, if all the fields are filled, the process will proceed to rent, case 1 or 2.

Guarantee: The customer will not get the movie, the daily rate rent will not be debited on the card, the movie still available from the movies list.
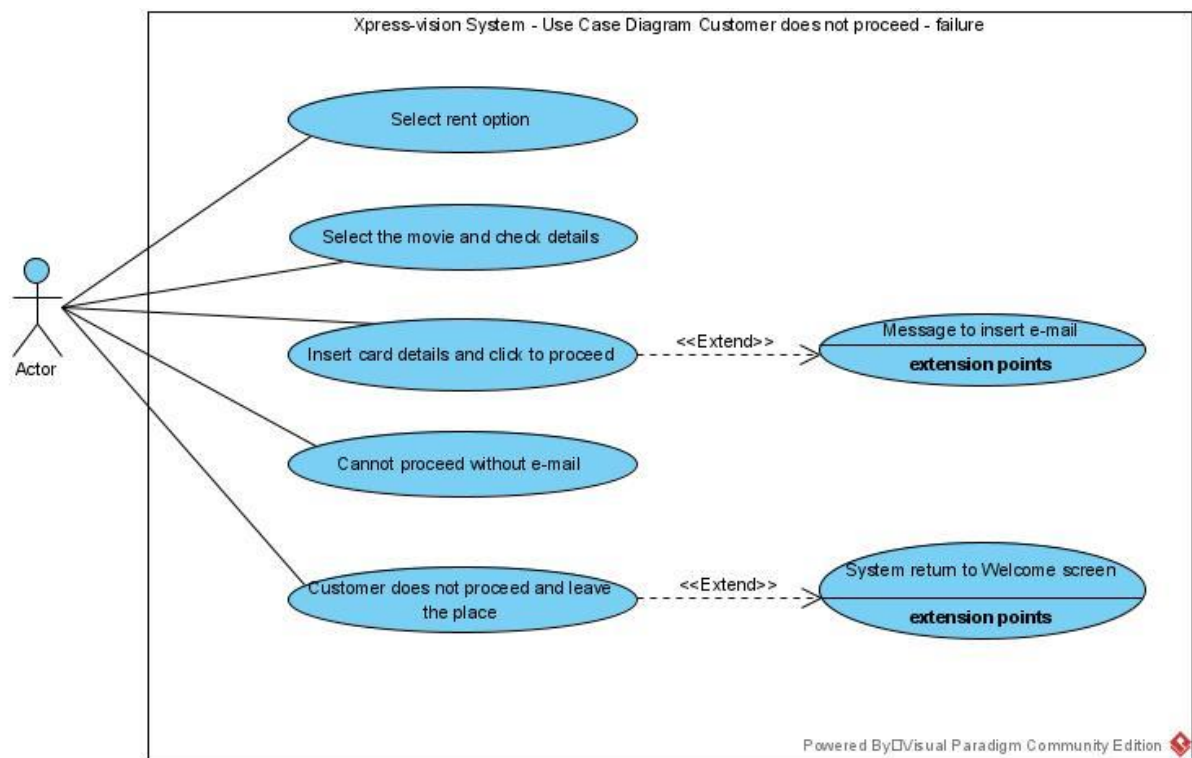

Figure 1.3 Use case diagram Customer does not proceed the rent

- **Use Case 4**

Use-Case Name: Return movie case process.

Trigger: The customer goes to rent/ return point and interact with the system.

Preconditions: The rent/ return point should be on, connected and available to use.

Primary Actor: Customer

Secondary Actor: -

Main Success Scenario (MSS):

9. The customer selects the return option on screen.
10. The customer types the movie cod.
11. The customer clicks on proceed returns button.
12. The system received the movie case.
13. The movie returns to available movies list.

Extensions:

11a. Alternative case, if the movie was rented for more than 10 days the system will show the message "Maximum rental period – 10 days, you were charged a maximum fee of €15 and the disc is yours to keep" and return to main screen.

Guarantee: The customer will return the movie, the daily rate rent will no more be debited on the card and the movie will be available on the movies list.
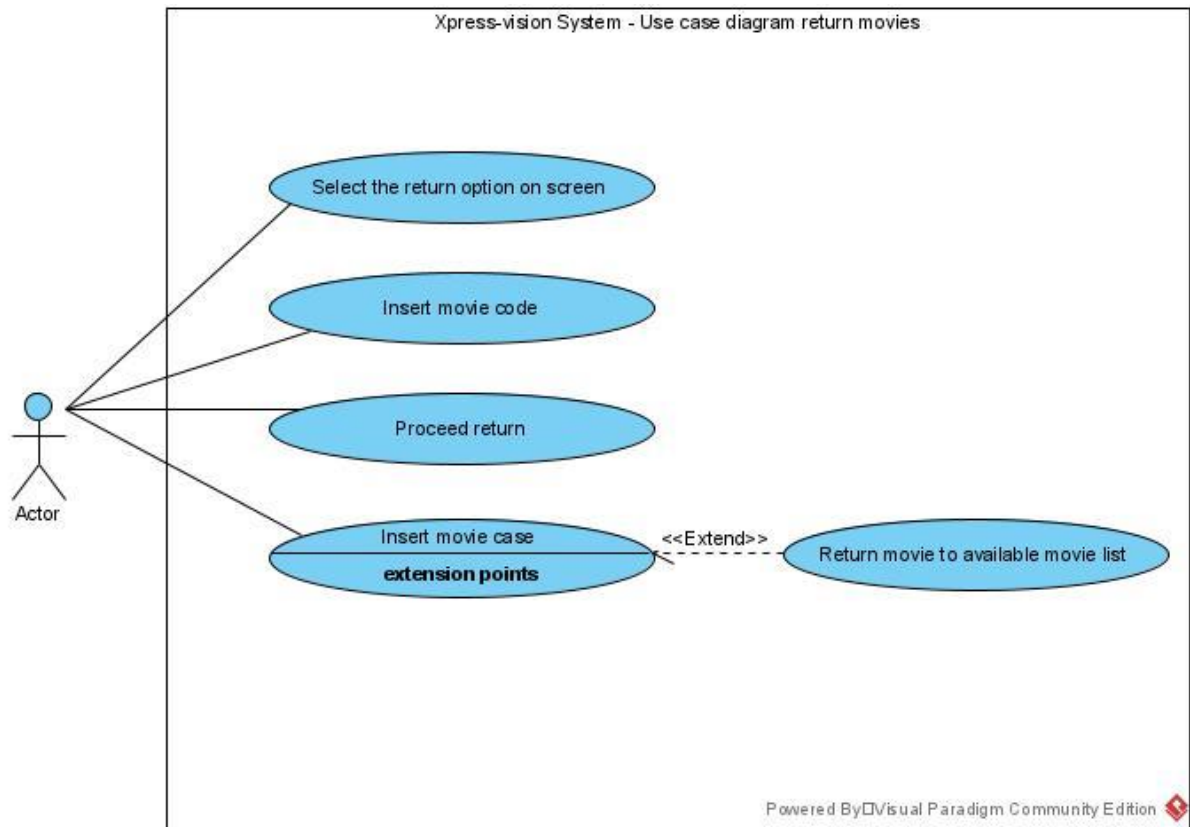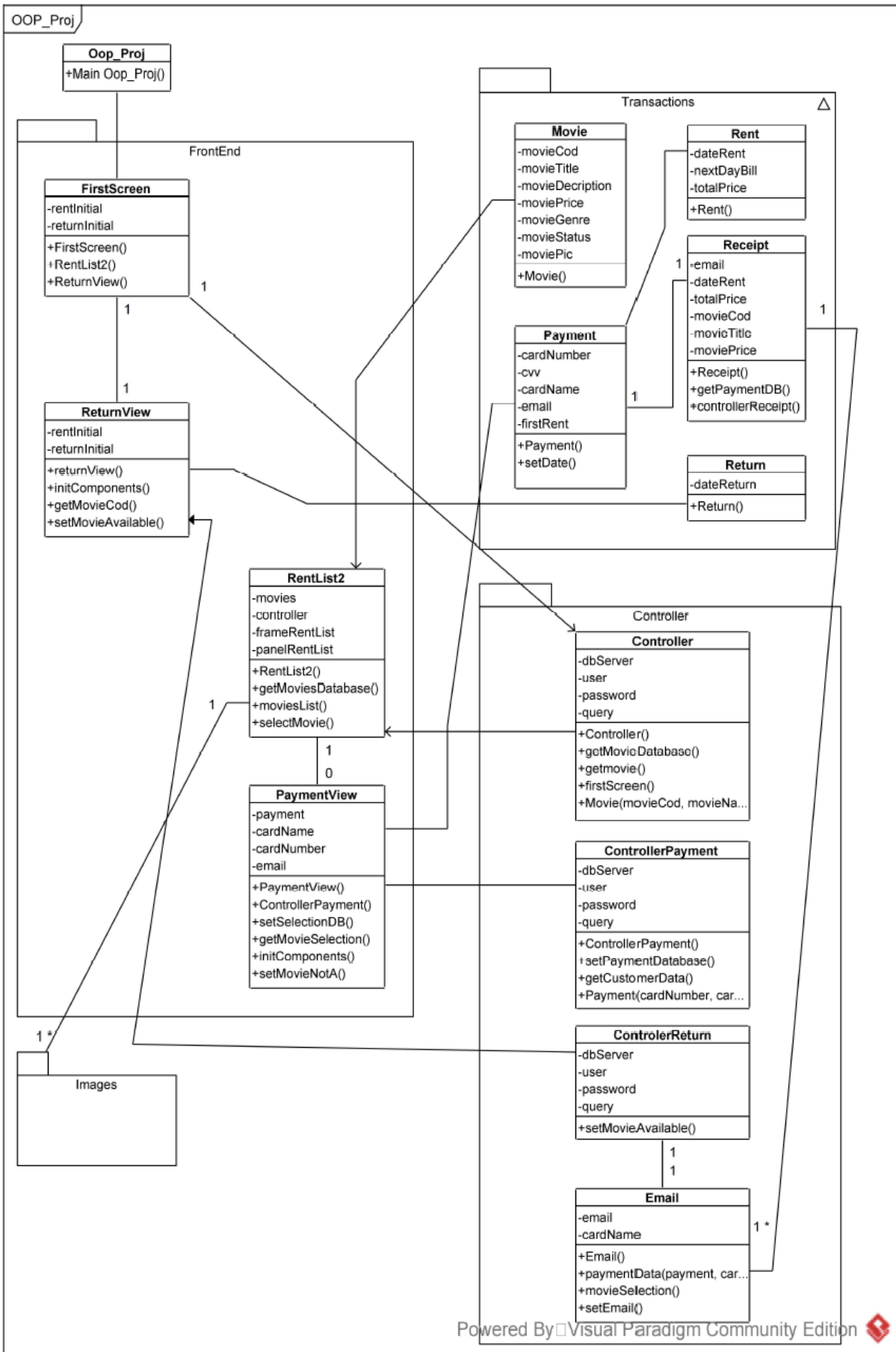
Figure 1.4 Use case diagram return movie

2.4. Class Diagram

A Class diagram that will properly illustrate the relationships between classes.

OOP_Proj

Oop_Proj
+Main Oop_Proj()

FrontEnd

FirstScreen
-rentInitial
-returnInitial
+FirstScreen()
+RentList2()
+ReturnView()

ReturnView
-rentInitial
-returnInitial
+returnView()
+initComponents()
+getMovieCod()
+setMovieAvailable()

RentList2
-movies
-controller
-frameRentList
-panelRentList
+RentList2()
+getMoviesDatabase()
+moviesList()
+selectMovie()

PaymentView
-payment
-cardName
-cardNumber
-email
+PaymentView()
+ControllerPayment()
+setSelectionDB()
+getMovieSelection()
+initComponents()
+setMovieNotA()

Images

Transactions

Movie
-movieCod
-movieTitle
-movieDecription
-moviePrice
-movieGenre
-movieStatus
-moviePic
+Movie()

Rent
-dateRent
-nextDayBill
-totalPrice
+Rent()

Receipt
-email
-dateRent
-totalPrice
-movieCod
-movieTitle
-moviePrice
+Receipt()
+getPaymentDB()
+controllerReceipt()

Payment
-cardNumber
-cvv
-cardName
-email
-firstRent
+Payment()
+setDate()

Return
-dateReturn
+Return()

Controller

Controller
-dbServer
-user
-password
-query
+Controller()
+getMovieDatabase()
+getmovie()
+firstScreen()
+Movie(movieCod, movieNa...

ControllerPayment
-dbServer
-user
-password
-query
+ControllerPayment()
+setPaymentDatabase()
+getCustomerData()
+Payment(cardNumber, car...

ControlerReturn
-dbServer
-user
-password
-query
+setMovieAvailable()

Email
-email
-cardName
+Email()
+paymentData(payment, car...
+movieSelection()
+setEmail()
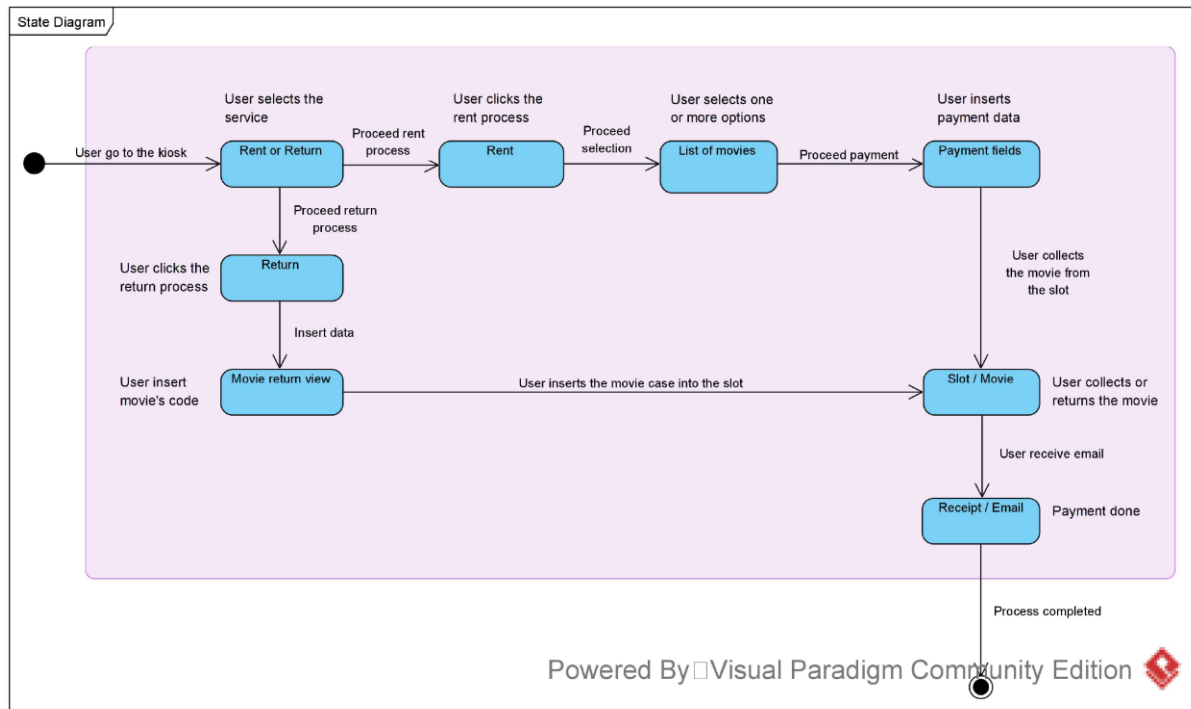
1   1   0   1   *   1   1   1   *

2.5. Sequence diagram

A Sequence diagram that will properly illustrate the flow of the system functionality.
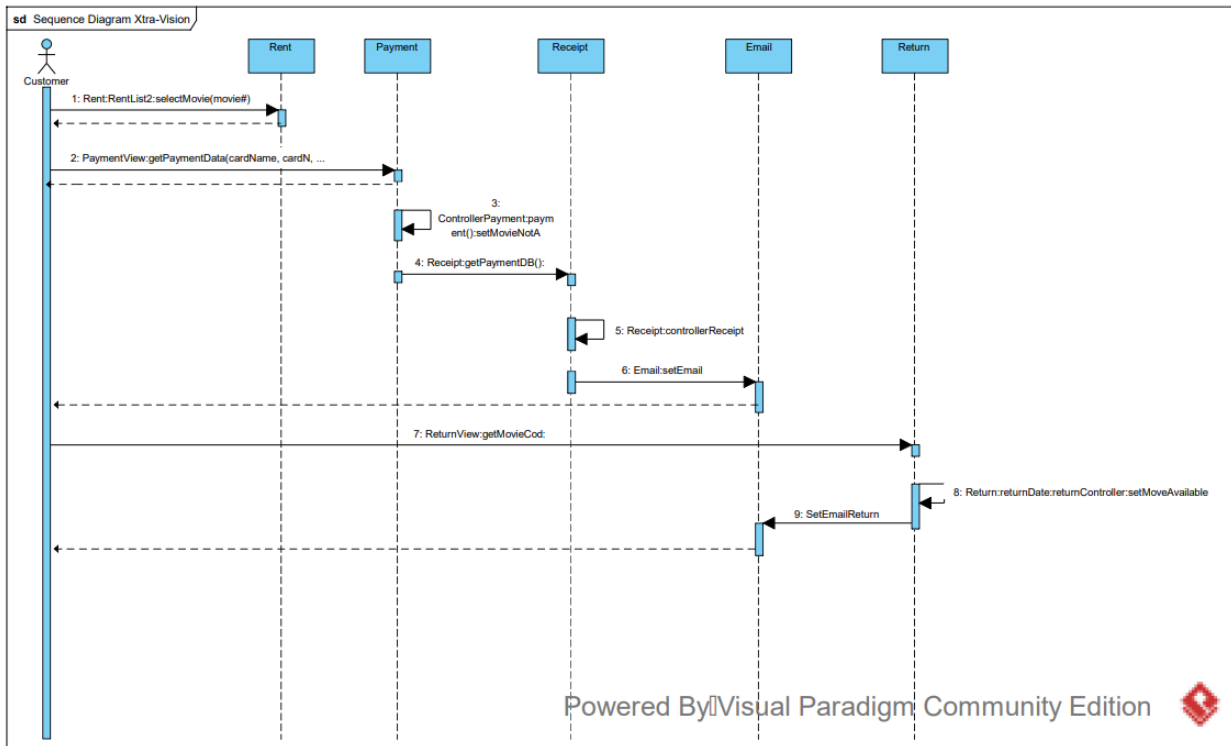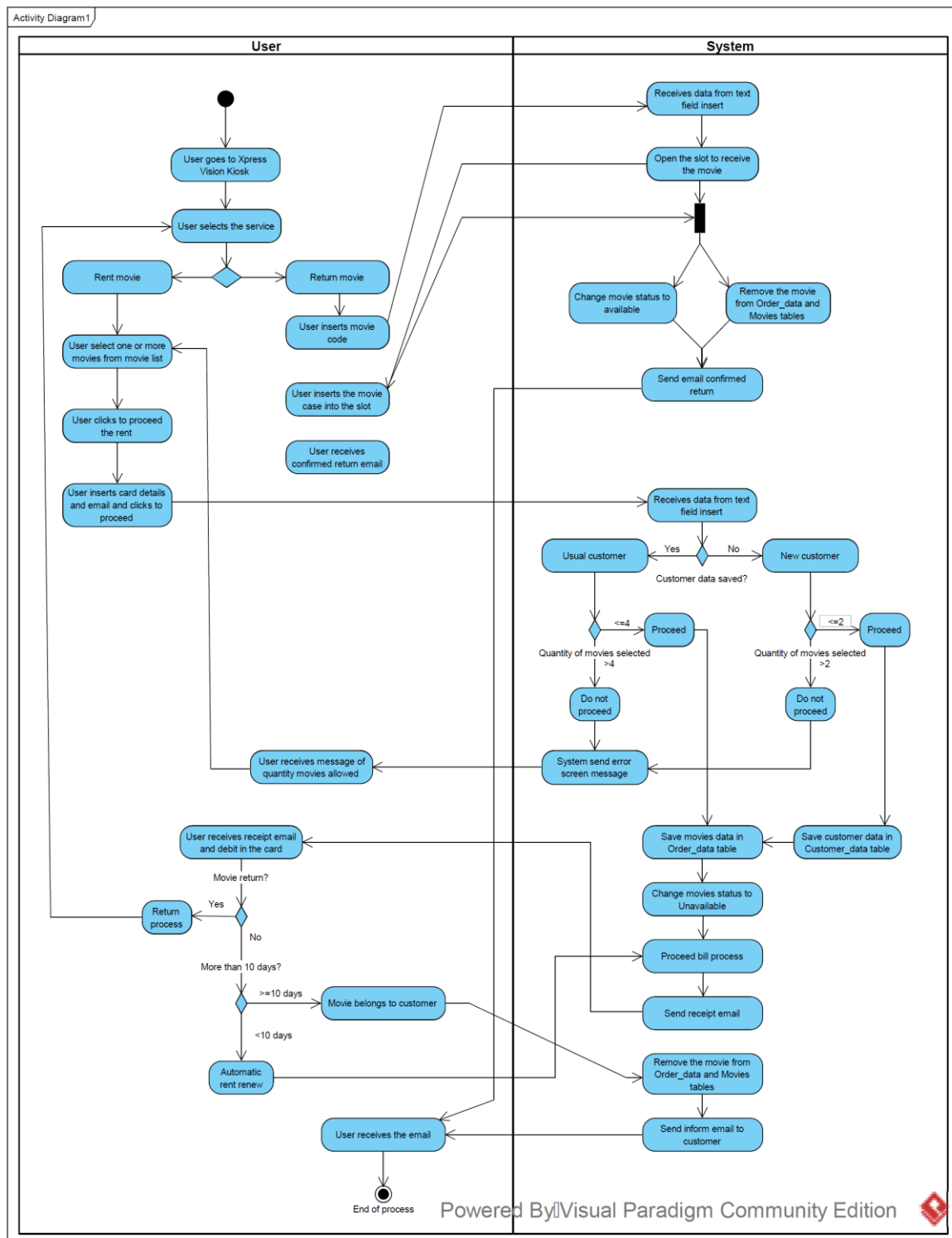
2.6. State Diagram

A State Diagram modelling the state of the system as a movie gets rented.

2.7. Activity Diagram

An Activity Diagram which models the entire flow of activity. Include 'swim-lanes'.

2.8. User Story

User interaction feature of the Xtra-Vision system.

Title: Movie Release Email	Priority: Low	Estimate: September 2021
User story: **As a** Customer **I want** to be informed when the new movies will be available in the kiosk to rent. **So that** should be include in the system a check option to send by email the new movies included in kiosk.		
Acceptance criteria: • Create a check box option and a function in the Payment View screen. • The development team should finish the test template until September 2021. • The new function will be available after the Test Team approval.		

3. Research

3.1. Research item 2

The software development can be a complex task even for big companies. As the system design area are more competitive and offer a range of services, tools and functionalities, the teams became multifunctional and interactive. This situation pushes the companies to use and adopt tools and systems to manage the development and improve the teams' communication.

The competitive scenario demands more efficiency from de development companies as states by Cellier (2021), the object-oriented modelling brings the reusability and the abstraction main factors to reduce costs and optimise the time of a project. The way as an object-oriented software is designed contribute to divide it into functional parts of the dense job. It allows the multifunctional teams works completely, with programmed partial delivers, tests, and changes.

One methodology used to develop the process and the interaction of the team is the Agile Methodology. As stated by Cprime.com (2021), it "refers to any development process that is aligned with the concepts of the Agile Manifesto", normally uses Kanban and Scrum frameworks. The Agile Methodology has the objective to develop solutions where needs interactive development and cross-functional teams by its application, the discipline, inspection, adaptation, and leadership is promoted offering a high business response to the project impacting positively the customer satisfaction and company profit.

The Scrum framework increases the productivity if compare of a classical Waterfall methodology, with focus in rapidly-changing requirements, proximity with the client and better control and estimates of the process and high visibility into the state of a development project, on a daily basis (Cprime.com, 2021).

Agile Alliance (2021) states that Kanban is a well-known tool to "design, manage, and improve flow systems for knowledge work". If the objective of the Scrum is to reduce time and smooth changes, on the other hand the Kanban lets the

organization improve their workflow using a visual mechanism to control the progress. The team can visualize the flow and limiting the progress, realizing the restrictions of the process.

Basically, what can be said is the project management skills and tools, leadership, the uses of technology communications are the ways to manage the complexity of software design projects. With the variety range of services and models the companies need to set their objectives and define a framework that is satisfactory to the company and the team. Once the tools are defined, the correct use will be the key for the goals achievement, the lack of engagement, misuse or partial implementation can affect negatively the process.

4. Attachments

Link GitHub: https://github.com/giselellopes/OOP.git

Java file: OOP_Proj.zip

MySQL Workbench tables: Database_CA.zip

Information connection:

$host="52.50.23.197";

$port=3306;

$socket="";

$user="Gisele_2018076";

$password="";

$dbname="Gisele_2018076";

References

Agile Alliance. Kanban. Available at: https://www.agilealliance.org/glossary/kanban/ . Accessed 27/04/2021.

CELLIER, F. E. Object-oriented modeling: Means for dealing with system complexity. Department of Electrical and Computer Engineering, The University of Arizona. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.1808&rep=rep1&type=pdf . Accessed in 15/05/2021.

Cprime.com. Available at: https://www.cprime.com/resources/what-is-agile-what-is-scrum/ . Accessed in 15/05/2021.