Simone Guidato
0279885

Thomas Santapaola
0286161

Progetto ISPW

Anno 2023/2024

Card Emporium

Simone Guidato 0279885

Thomas Santapaola 0286161

Simone Guidato **Thomas Santapaola**
0279885 0286161

1. Software Requirement Specification

1.1. 1 Aim of the document:

This document aims to illustrate the design steps to realize and create the Card Emporium software.

The software manages the trading of playing cards where different games and users are involved.

1.1.2 Overview of the defined system:

This application has been created to allow card game lovers to collect cards by trading. Users can register by entering their personal information and initially the only feature that they possess is to be able to put in the shopping cart and eventually buy the desired cards.

Some users can become sellers, other than basic users, and have the possibility to insert their cards among the cards available for purchase.

Users who don't follow the trading policies correctly are labeled as banned and denied access to the system.

Instead, the user registered as a manager have the authority to ban a user and to change the role to a base user to become a seller.

1.1.3 HW and SW requirements

Card Emporium can run on Windows (Windows 10 is the minimum version required), MacOS and Unix-like systems.

Hardware requirements are the Intellij minimum requirements.

1.1.4 Related system, pros and cons

It is a web page for trading playing cards. Some of the pros of this system are that you can sell excess cards and then buy cards that are missing from the collection or for example see when it is worth selling a different card according to the chart of the

price of the card in question. But as cons it cannot sell box of cards but only one type of card at a time.

In addition, the return policies of a card are not available now.

1.2 User stories

- *Thomas Santapaola*
  - As a non-registered user, I want to see all* types of cards that the site offers, so that i can decide to register to the website to buy them.
  - As a registered user, I want to add a card in my shopping cart, so i can buy multiple cards at the same time.
  - As a registered user, i want to chat with the seller, so that we can bargain about the cards's price.

- *Simone Guidato*
  - As a seller, i want to expose my cards with their names, their gradation** and from which set they are, so that other users can see and buy them.
  - As a seller, i want to change the price of my exposed card, so that i can sell it with the current market price.
  - As a admin, i want to change the role of a user, so that i can punish them for their bad behaviour***.

"*" = types of cards could be: pokemon card, Yu-gi-oh card, Magic card, Dragonball card, Force of will card.

"**" = the gradation can be specified as: Nearm mind, Excellent, Good, Light played, Poor.

"***" = a bad behaviour could be when: a seller refuse to send a sold card without any explanation, refuse to answer the admin or the registered user or a selle who send a card with the wrong attributes (name, gradation, set).

1.3 Functional Requirements

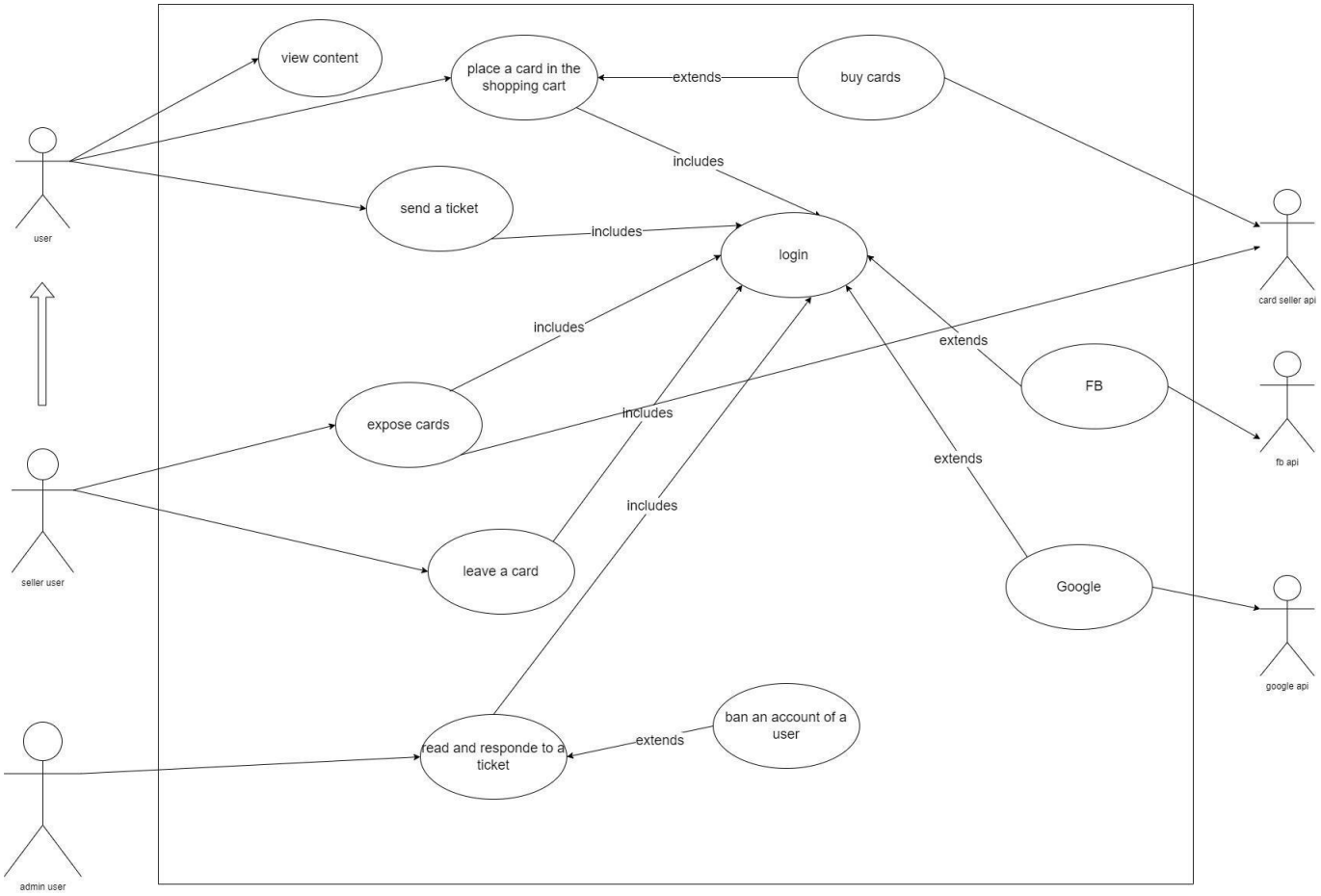
- *Simone Guidato*
  - The system shall send an e-mail to the seller (specifying the card name and the money earned) when they sell a card.
  - The system shall provide a button to customize my cards' research.
  - The system displays a graph of the card price for the last week.

- *Thomas Santapaola*
    - The system shall send an e-mail (specifying the card name and the card price) to the buyers when they buy a card.
    - The system shall display a list of the most selled cards of the week.
    - The system shall order a cards' list by their price in ascending order.

1.4 Use Cases

- *Overview Diagram*

Simone Guidato **Thomas Santapaola**
0279885 0286161



- *Internal Steps*

- *Simone Guidato*

Expose Card

1)login

2)the system gets available a button to switch on the seller page.

3)the seller clicks the button to switch on the seller page.

4)the system gets available a bar to search a card.
5)the seller researchs the card that they want expose.

6)the system lists all the cards conteining the researched name.

7)the seller selects form the list the card they want expose.

8)the system provides a form and a button to expose the card.

9)the system displays a graph with the selected card price during the current week.

10)the seller fills in the schedule form.

11)the seller clicks the button to expose the card.

12)the system checks the schedule.

13)the system adds the card on the market.

Extensions

1a) username or/and password aren't correct: the system notifies the user with a message on display.

12a) the seller has not fullfilled all prerequisites of the schedule form: the system notifies the seller and terminates the use case.

5a)tha researched name hasn't a match on card database:the system displays a message notifing the seller and terminates the use case.

- *Thomas Santapaola*

1)the user selects the type of card to buy.

2)the system displays a list of that type of cards.

3)the system gets available a bar to search a card.

4)the user digits on the research bar the card's name they want to buy.

5)the system lists all the different cards with the selected name.

6)the user chooses the cards' set from where they want to buy.

7)the system lists all the different sellers of the selected card with the amount of the selected card they sell .

[8)the system displays a graft with the selected card price during the current week.]

9)the user selects from which seller they want to buy.

10)the system **places the card in the shopping cart** and removes that card from the market.

11)the system displays the cards in the shopping cart with the full price of the sold.

12)the user buys all the cards in their shopping cart.

13)the system saves the sold and sends to the user and the seller an email about the sold.

Extensions

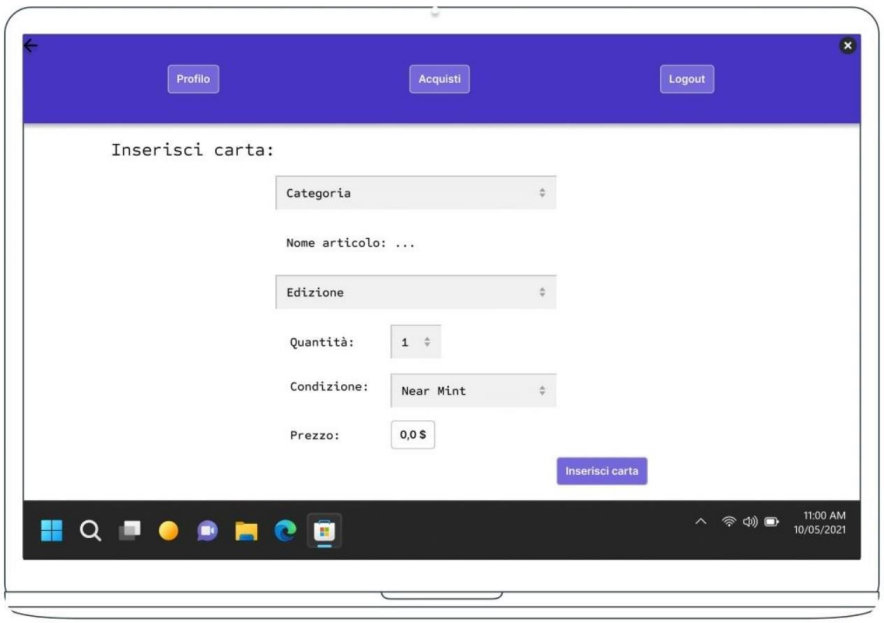
9a) the user is not logged: the system requires the login.

9b) the item is not available: the system tells the user that the item is no longer available and refresh the page.

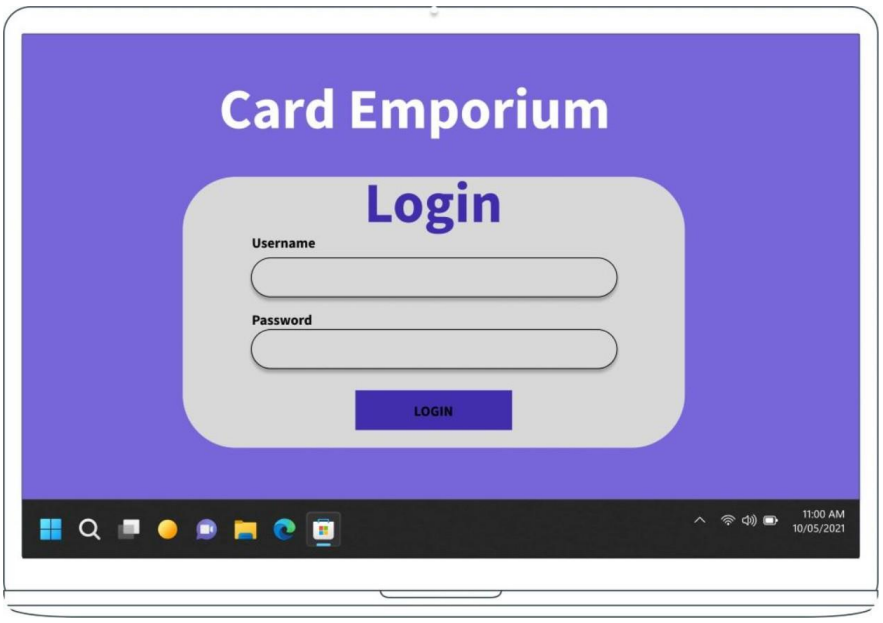
11a)the item in the shopping cart is no longer avilable:the system remove all the no longer avialable cards from the shopping cart.

2. Storyboards

- *Simone Guidato*
  - The first storyboard is the seller UI where the seller can enter the data related to the card he wants to sell. The button “Inserisci carta” allows the insertion of the card on the market.



  - The second storyboard is the Login page that an already registered user uses to access the services of the application.



- *Thomas Santapaola*
  - The first interface is the main page that a user displays as soon as he logs in, where he can see a list of new products and cards sold in the last month.

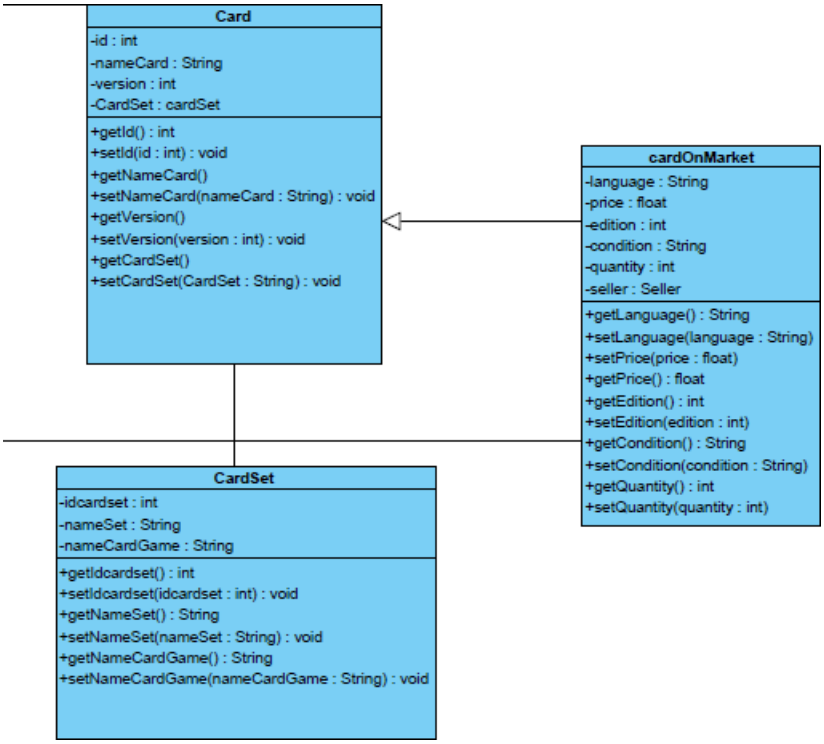
  - This is the interface used by the user to register in the application so you can start buying and eventually selling cards.

3. Design

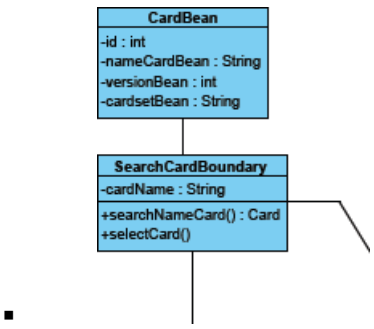
3.1 Class Diagram

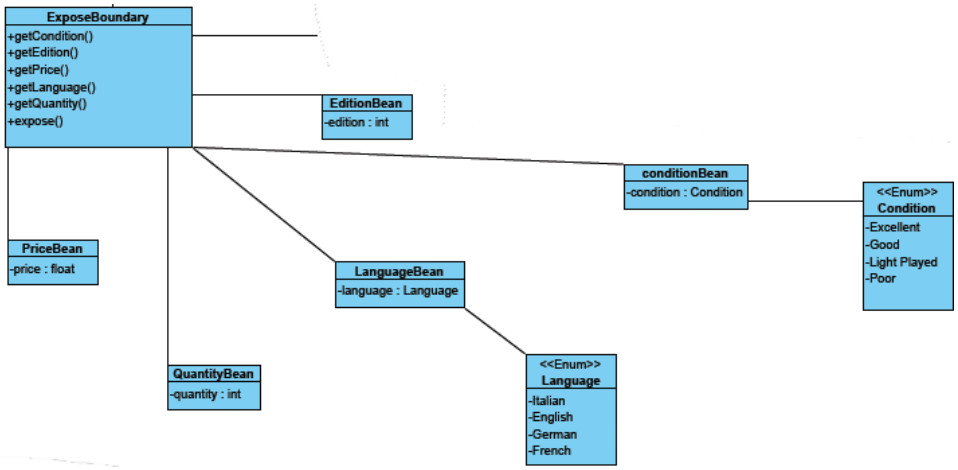
3.1.1 VOPC

- *Simone Guidato*
  - The Model part

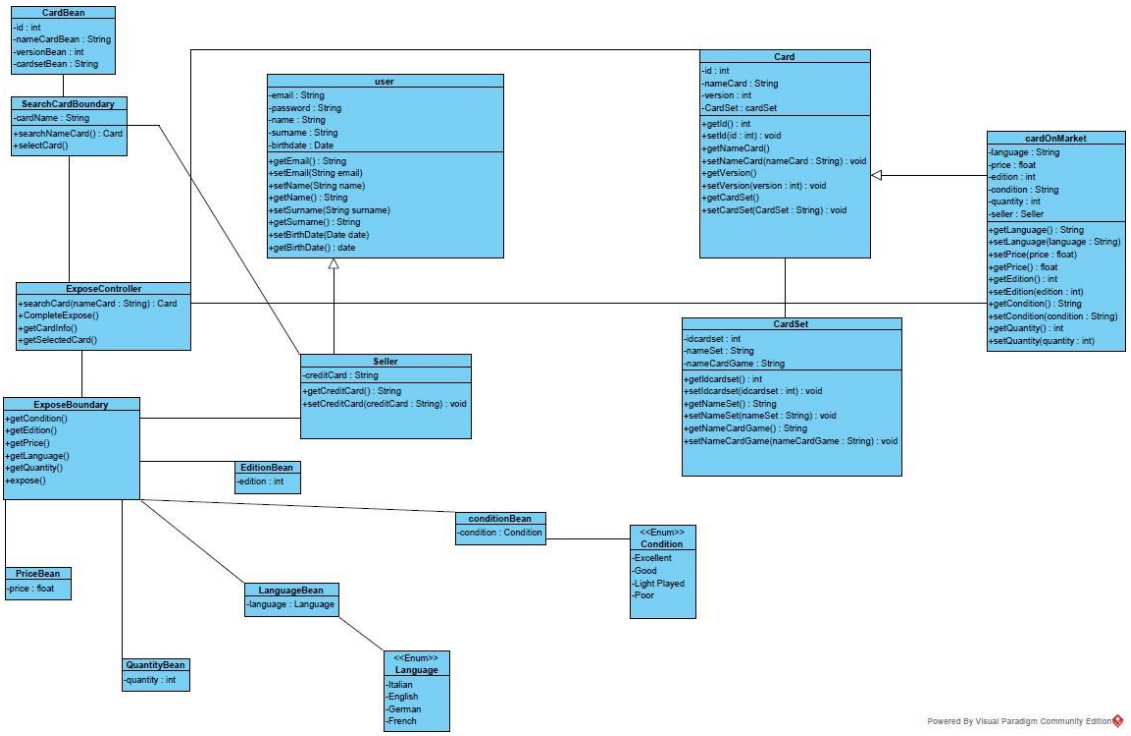


  - The Boundary part



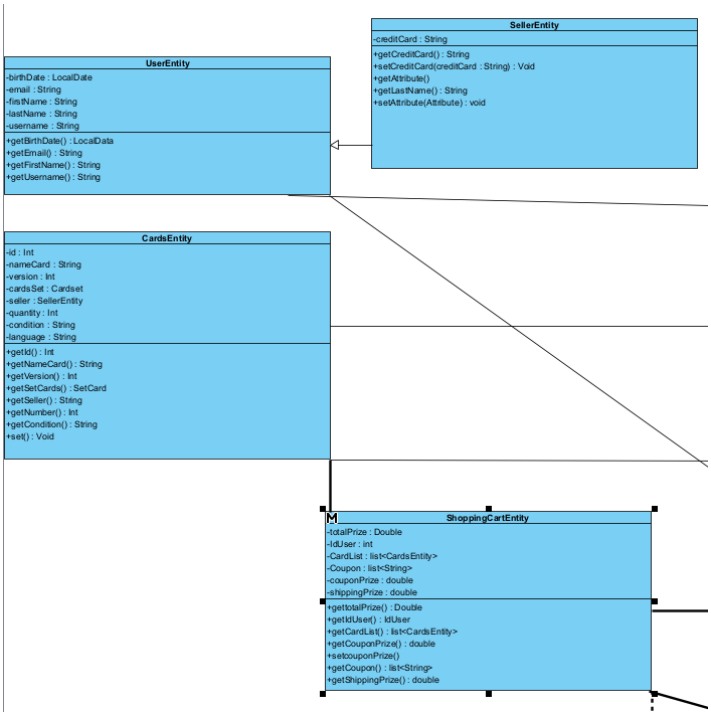


- The global view.

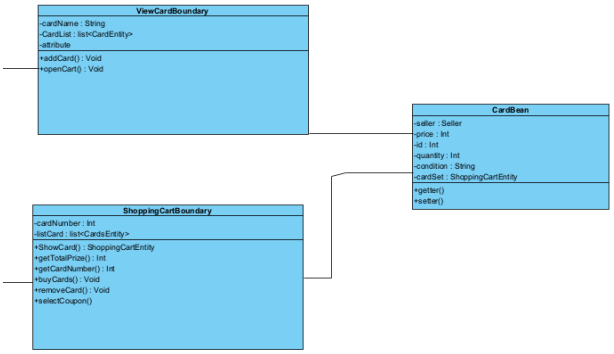


- *Thomas Santapaola*

- The model part.

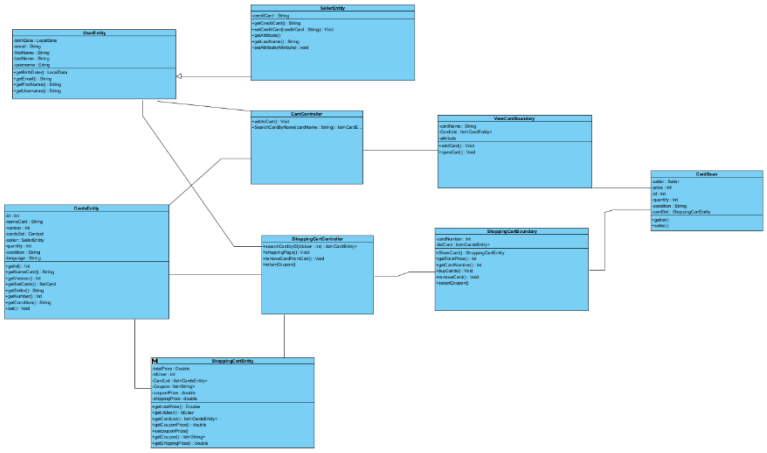


- The boundary part.



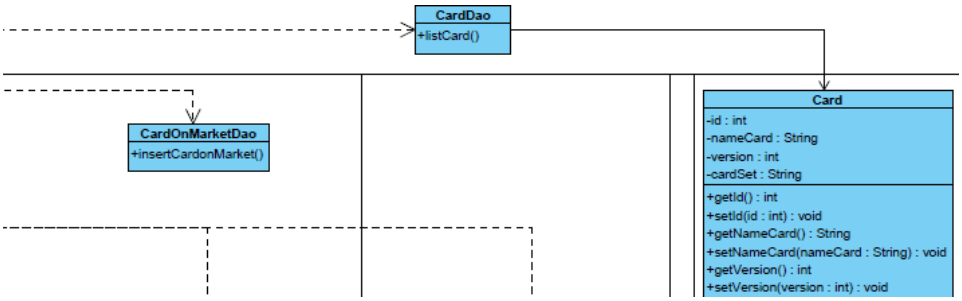
-

- The global view.

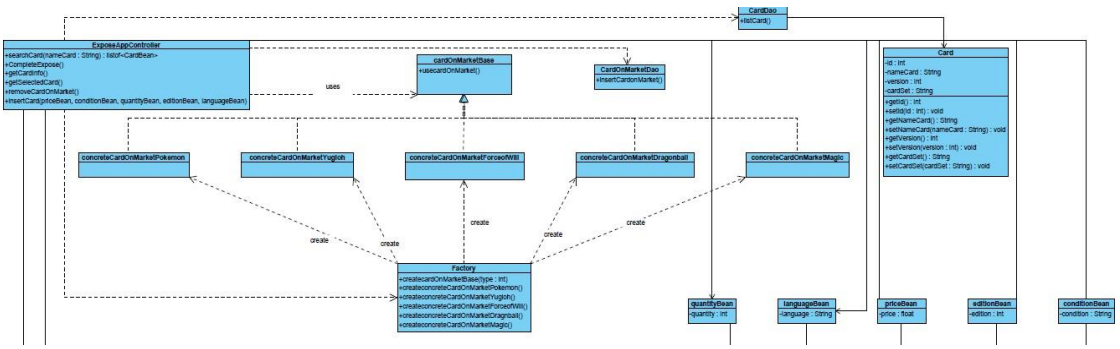


3.1.2 Design-level Diagram

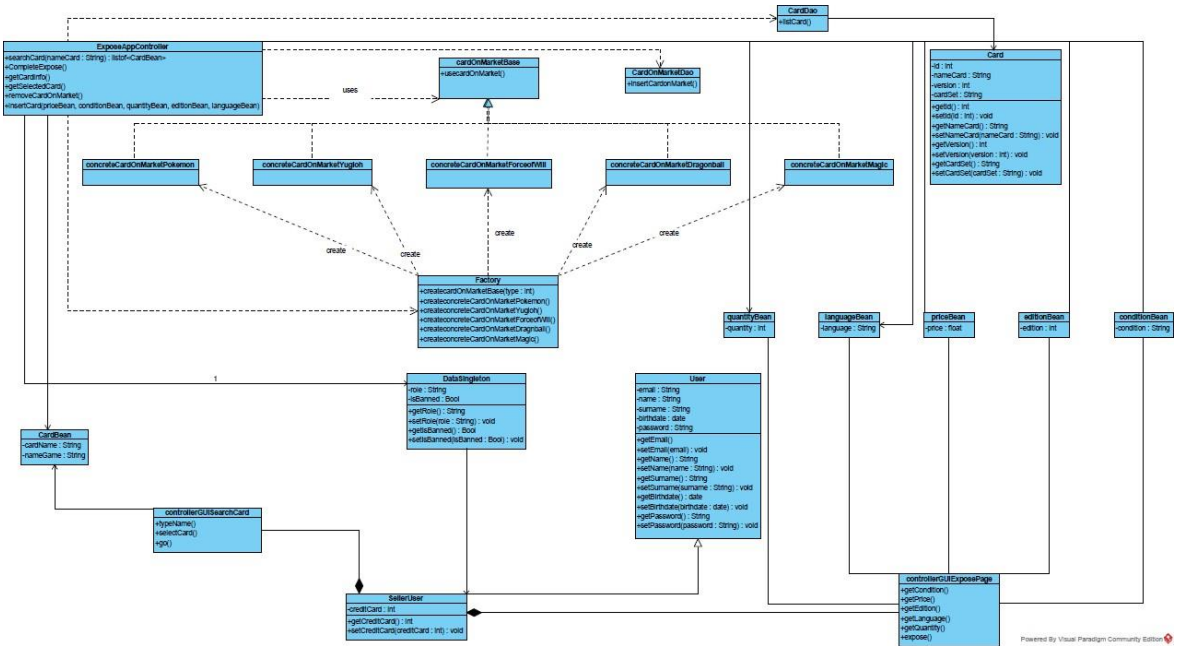
- *Simone Guidato*
  - The DAO part.



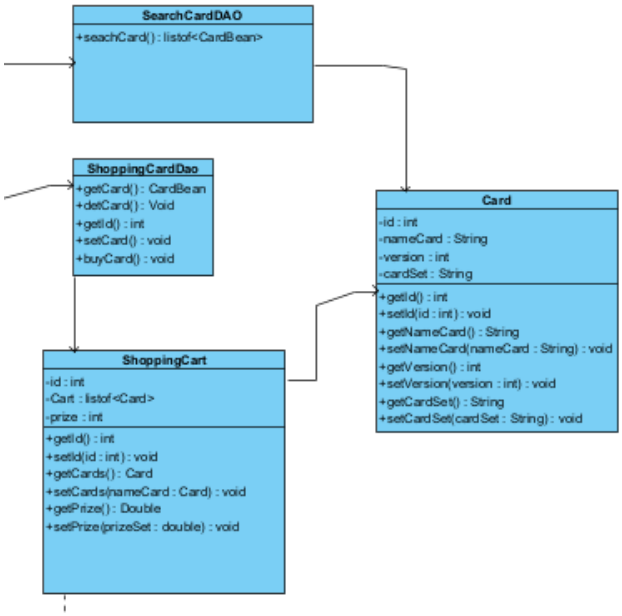
  - The controller part.



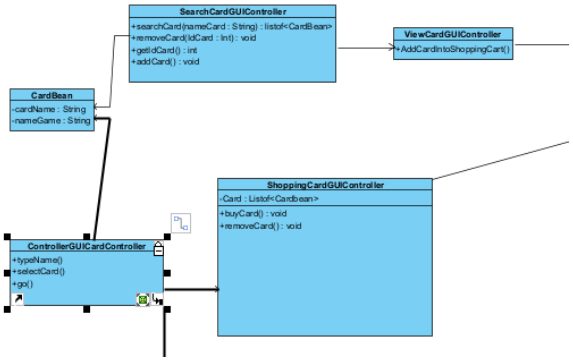
- The global view.



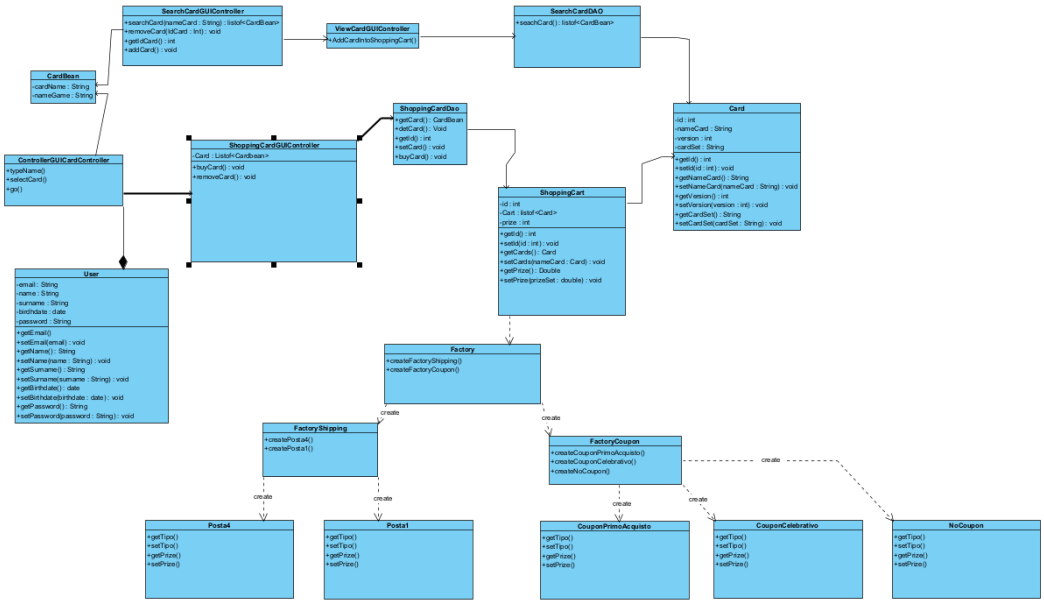
- *Thomas Santapaola*
  - The DAO part.



- The controller part.

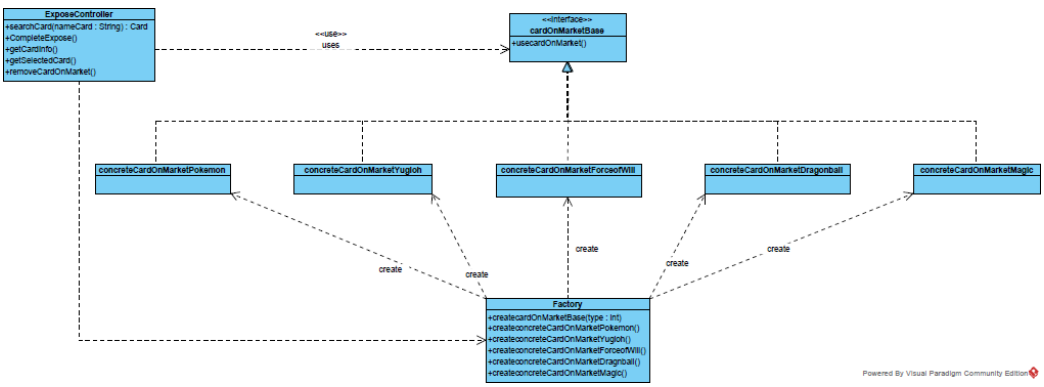


- The global view.



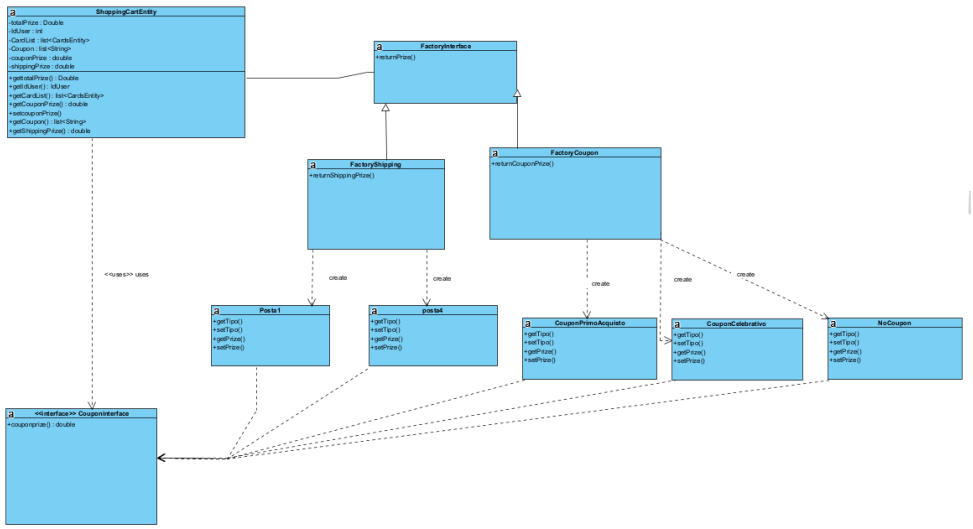
3.2 Design Patterns

- *Simone Guidato*
  - In the context of the system, it has been implemented the *Factory Method* pattern to decide at run-time which game a card belongs to and consequently create a different GUI based on the game.



-

- *Thomas Santapaola*
    - I used the "Factory Method" design pattern to decide at run-time which coupon will be used into the cart shop and consequently elaborate the final prize of the cart based on the coupon selected by the user.

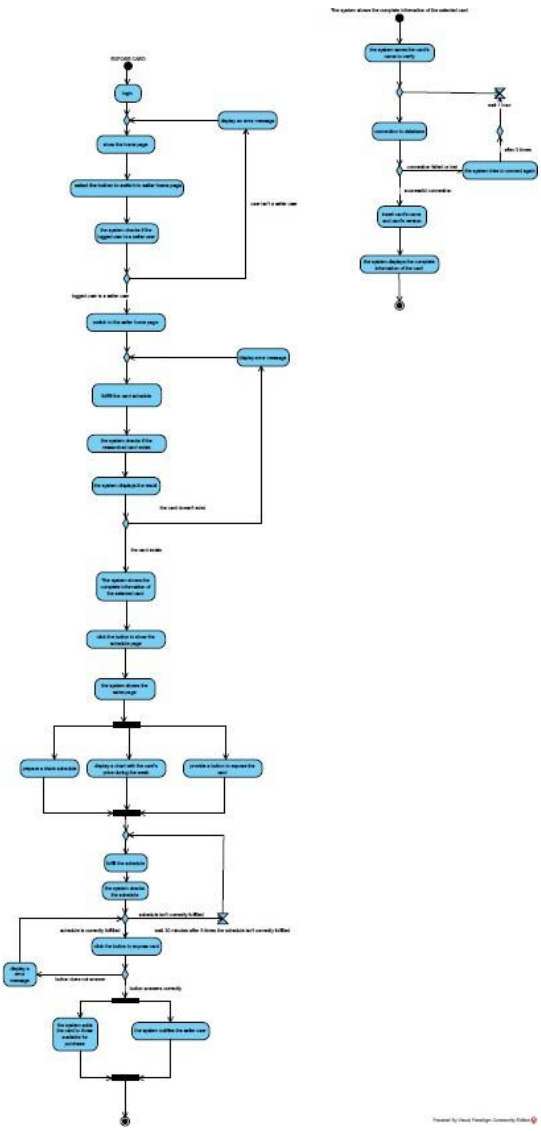


3.3 Activity Diagram

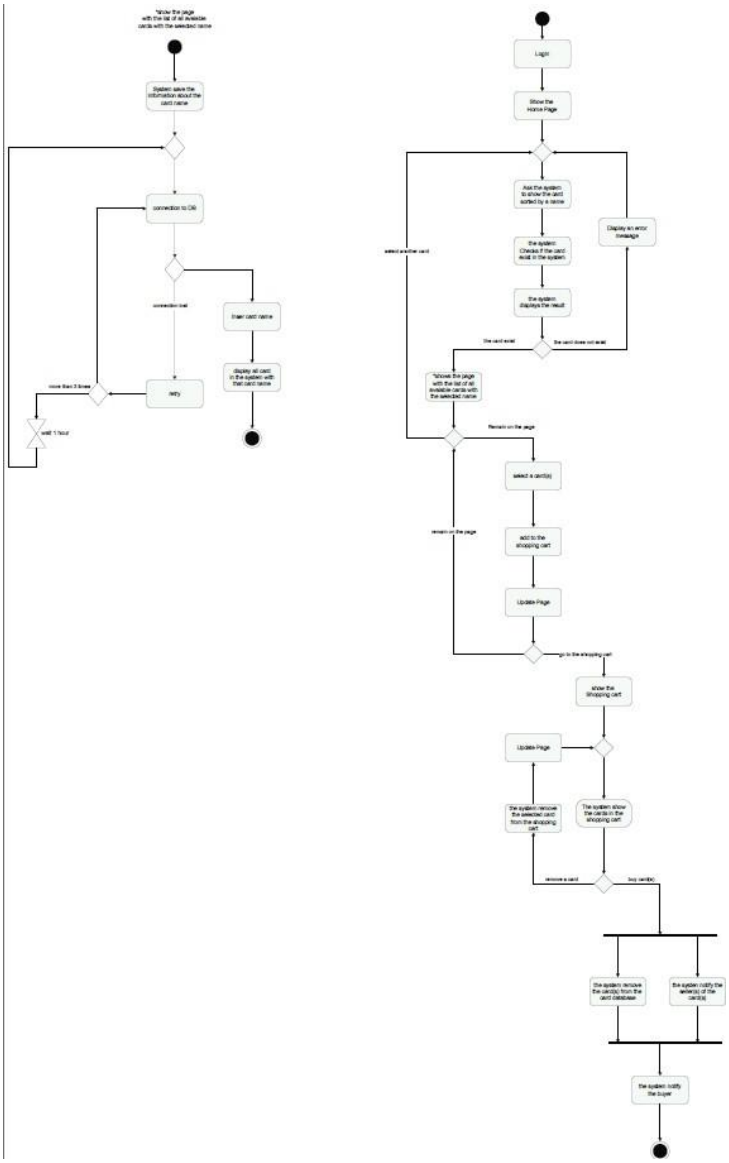
- *Simone Guidato*

Simone Guidato
0279885

Thomas Santapaola
0286161

- 

- *Thomas Santapaola*

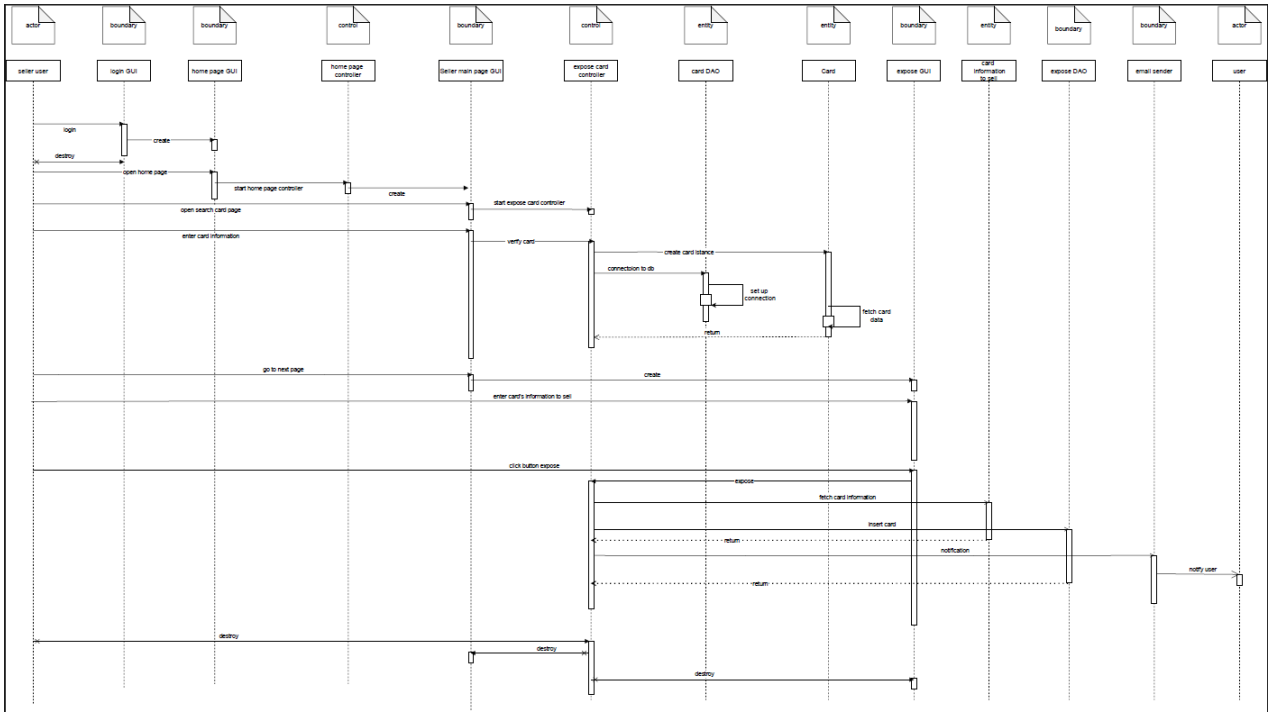


Simone Guidato
0279885

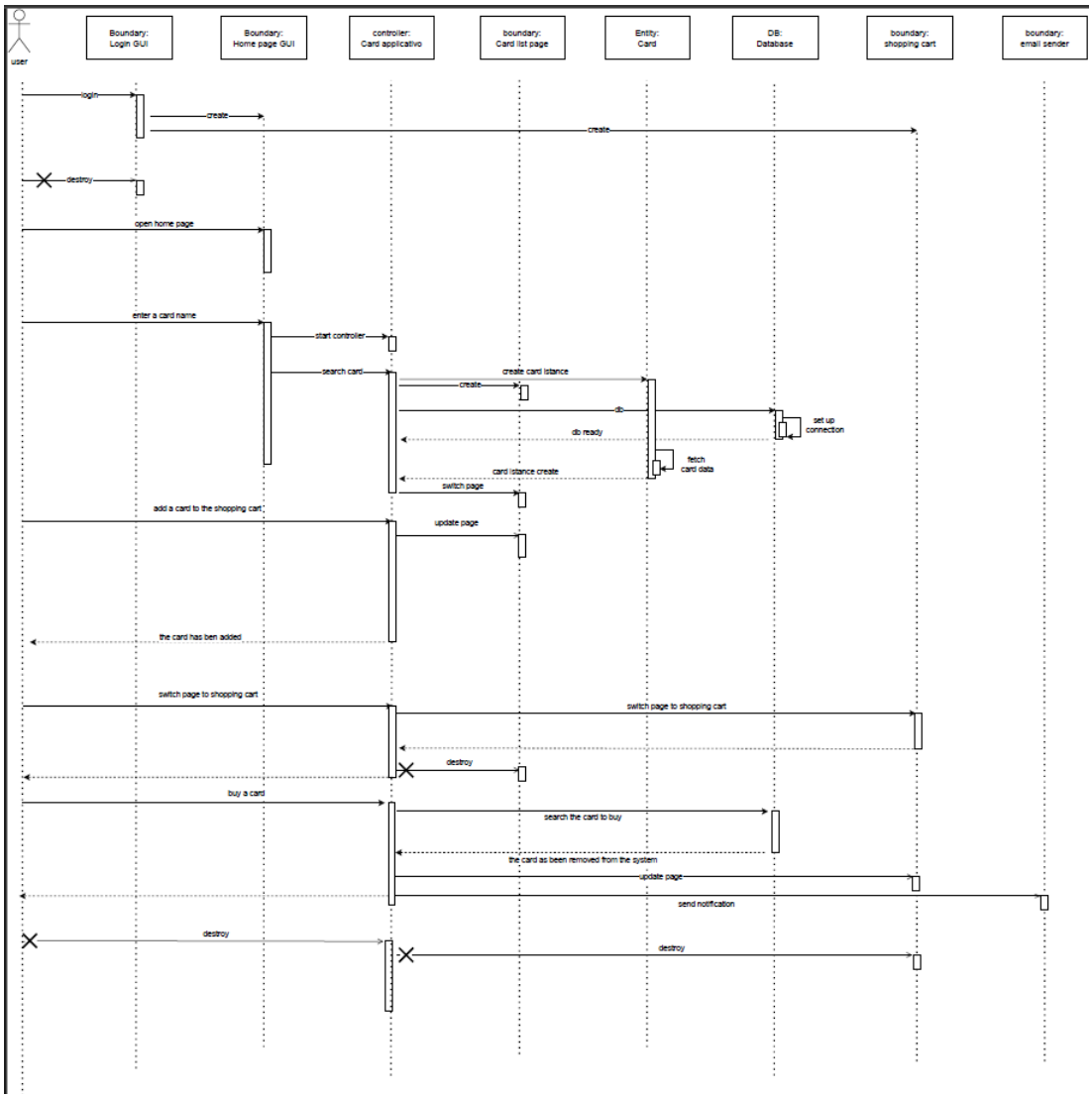
Thomas Santapaola
0286161

3.4 Sequence Diagram

- *Simone Guidato*

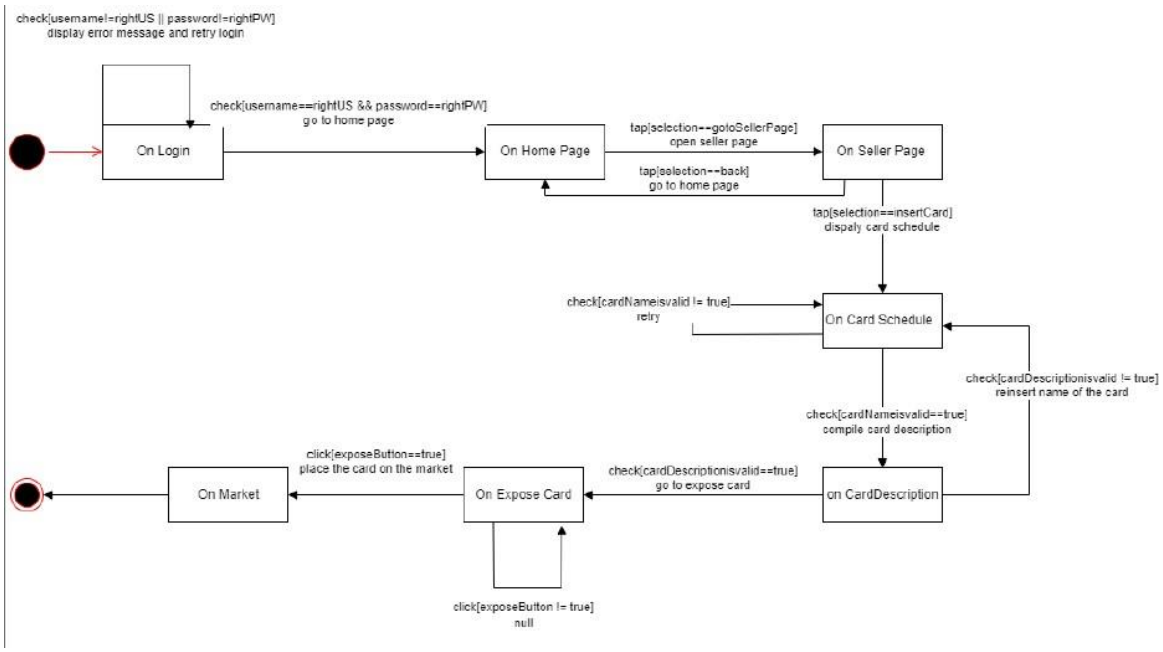


- *Thomas Santapaola*

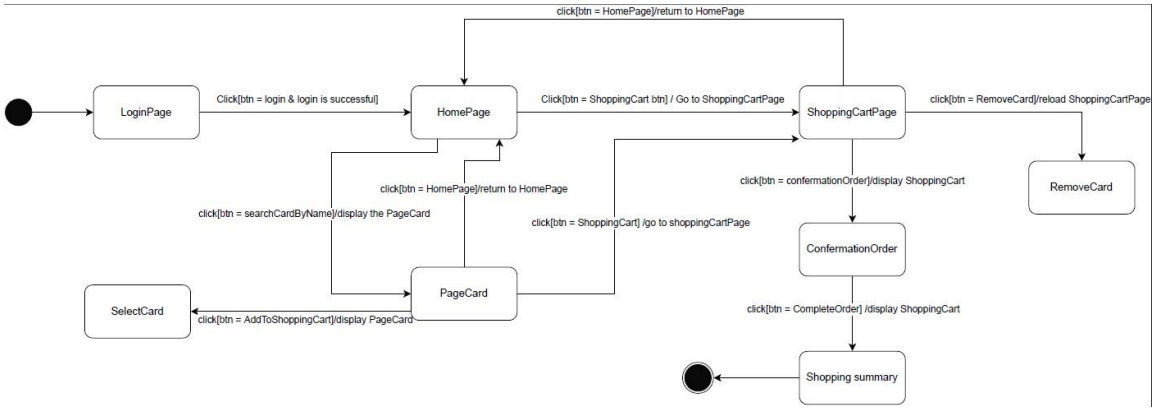


3.5 State Diagram

- *Simone Guidato*



- *Thomas Santapaola*



4. Testing

- *Simone Guidato*
  - *ExposeCardTest,* that oversees testing the ExposeController class.
  - *LoginTest,* that oversees testing the Login function.
  - *RegisterTest,* that oversees testing the Register function.
- *Thomas Santapaola*
  - *UserNotSellerTest,* which tests whether a user is a seller or not.
  - *SearchCardTest,* that oversees testing the *BuyCardApplicativo* class.
  - *BannedTest,* that tests whether a user is banned.

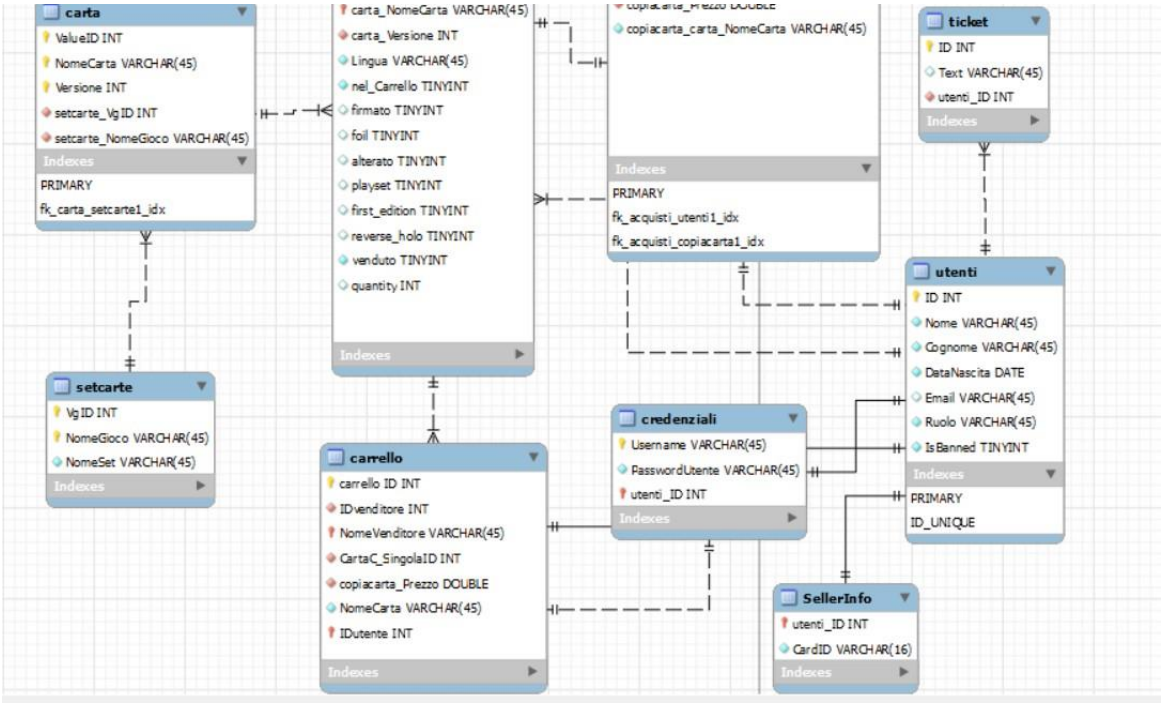
Simone Guidato
0279885

Thomas Santapaola
0286161

5. Exceptions

- *Simone Guidato*
    - *CardGameException*
    - *InvalidChoiceException*
    - *ExceptionQuantity*
- *Thomas Santapaola*
    - *ExceptionCardNotExist*
    - *ExceptionBannedUser*
    - *ExceptionDBError,*

6. Databases

- *File System:*
    - *Credenziali* that is a file that contains user's credentials.
    - *Utenti* that is a file that contains the anagraphic data of a user.
    - *Shop* that is a file that contains all cards in the shopping cart.
- *MySQL*
    - 

    - Internally this SQL must follow the struct specified in the file *MYSQL_CardEmporium.sql*

7. Sonarcloud

To check the correct behavior of this project, it was used SonarCloud system. This system points out the possible bugs, possible vulnerabilities, and review that the functions are well implemented. You can check the results of the project in the following link:
https://sonarcloud.io/summary/new_code?id=ThomasBread20_CardEmporium.