Ingegneria del Software e Progettazione Web

Progetto A.A. 2023/2024

StudySwap

0293218

George Eduard Jugrin

INDEX

Introduction

Aim of the documentation

Provide a full description of the software system developed following an approach based on practices of software engineering.

Little overview of the application

SduySwap is a system that facilitates the exchange and purchase of study notes and materials among students.

The system primarily involves two types of users: **Buyers** and **Sellers**.

The main key features are:

1. Note Upload: the seller can easily create and update his notes, they need to specify the data path of the file so it can be stored properly, the subject and the price.
2. Note Purchase: buyers can purchase notes of their interest using the integrated payment system.
3. Leave a review: after purchasing the notes, users can leave reviews feedback on the received material.

The user can interact with the software through two different interfaces:

1. **Graphical User Interface (GUI)**: users can access the functionalities thanks to a user-friendly GUI, offering a clear view of available, purchased and uploaded notes.
2. **Command-Line Interface (CLI)**: the system also offers a well-structured and user-friendly CLI, allowing the user to perform the same operations through text-based commands.

HW and SW requirements

The software and hardware requirements:

- RAM: 2GB of free RAM

- CPU: any modern CPU

- Disk Space: 2GB

- Monitor resolution: 1024x768

- Operating System: Microsoft 8 or later, macOS 10.14 or later, any Linux distributions that supports Gnome, KDE or Unity DE

- Internet Connection: moderate speed to allow a smoother experience

Related systems, Pros and Cons

There are many related systems:

- **Online Learning Management Systems**: platform that provides for the delivering of educational content.
- **Digital Libraries**: provide repositories for storing and accessing academic resources, including research papers, textbooks and lecture notes.

Some pros about *StudySwap* are:

- **Access to diverse study materials**: StudySwap provides student with access to a wide range of study materials such as notes, practice exams, etc.
- **Convenience**: users can easily brows , search for, and purchase study materials from the comfort of their own home.
- **Cost-effective**: offers a cost-effective alternative to purchasing expensive textbooks or study guides. Students can find affordable study materials.

While some cons are:

- **Quality control**: ensuring the accuracy and reliability of users-generated study materials can be challenging. There may be instances of incomplete, outdated, or incorrect content that users need to be wary of.
- **Limited selection**: Depending on the user base and participation, StudySwap may have a limited selection of study material available, especially for niche subjects or specialized topics.
- **Dependence on user contribution**: The effectiveness of StudySwap relies heavily on user contributions. If there is a lack of active users or contributors, the platform's content may become stagnant or insufficient.

User Stories

1. As a buyer, I want to see the reviews of the material I want to buy, so that I can ensure the quality of the material I'm buying.
2. As a seller, I want to manage my inventory of notes, so that I can add new material like exercises.
3. As a buyer, I want to add many notes to the cart, so that I can buy multiple notes at the same time.

User Stories

Functional Requirements

1. The system shall provide a form to upload the notes with the possibility to insert the notes details*.

2. The system shall allow the buyer to visualize* the purchased notes

3. The system shall allow the seller to change the notes details.

visualize = open on the user device with the predefined app to read the pdf file

Use Case

Diagram
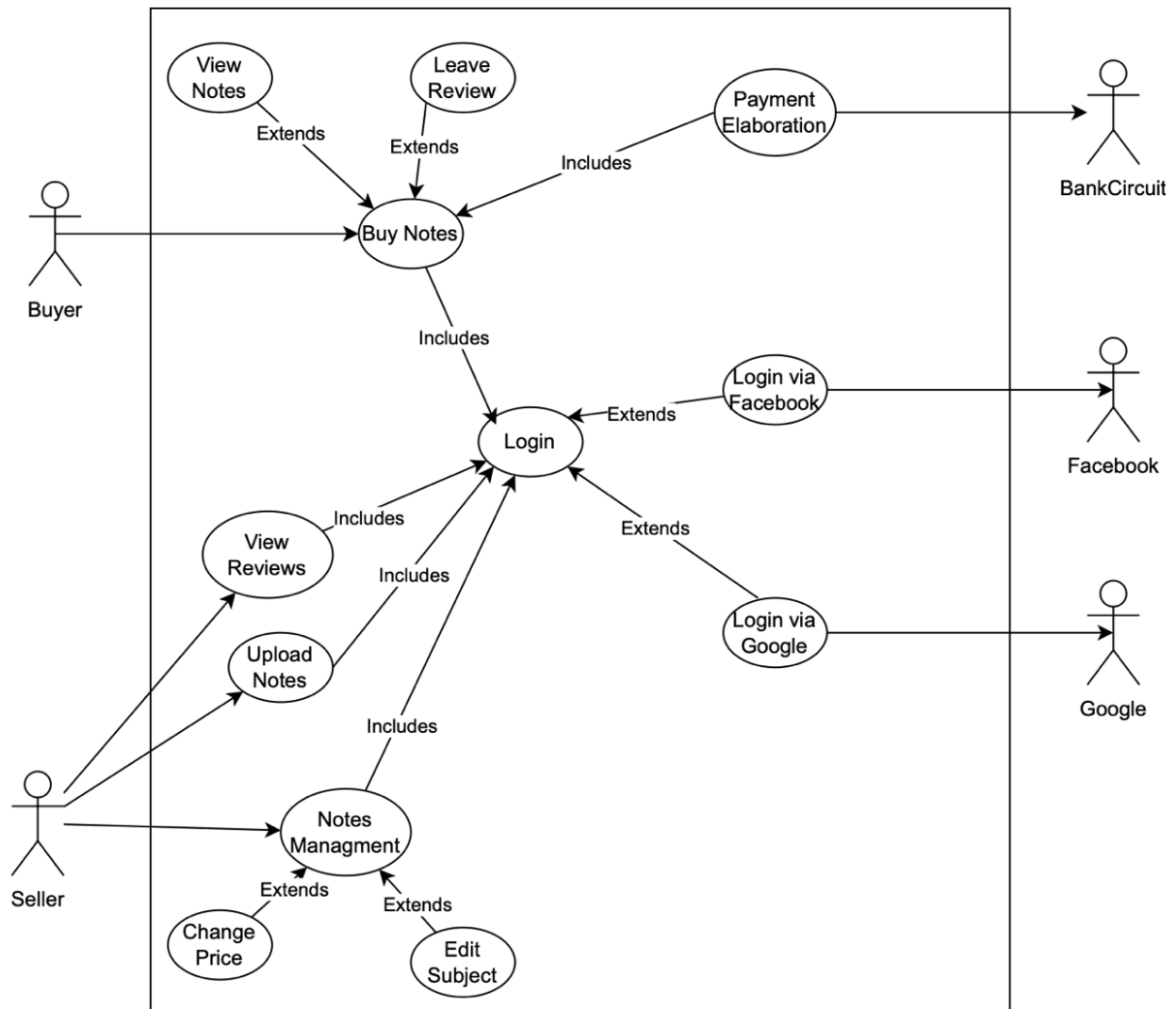
NOTE: Use Cases *View Reviews, Notes Management, Login via Google, Login via Facebook* are NOT implemented.

Internal steps

Name: buy notes (Buyer)

1. The system authenticates the buyer <u>via the use case *Login*</u>.
2. The buyer requests to view the notes that are available for selling.
3. The systems will show all the nots that are available for selling.
4. The buyer will send the request to buy the notes.
5. The buyer will select the payment method.
6. The system will start to elaborate the payment with the designed option selected.
7. The system will verify that all the payment details.
8. The system will confirm the payment was elaborated with success.
9. The system will generate the receipt with all the details.
10. The system will update the state of the order.
11. The buyer will be notified with the outcome of the payment.

Extensions

4a. *The notes are already purchased*: display an error message ("You have already purchased these notes").
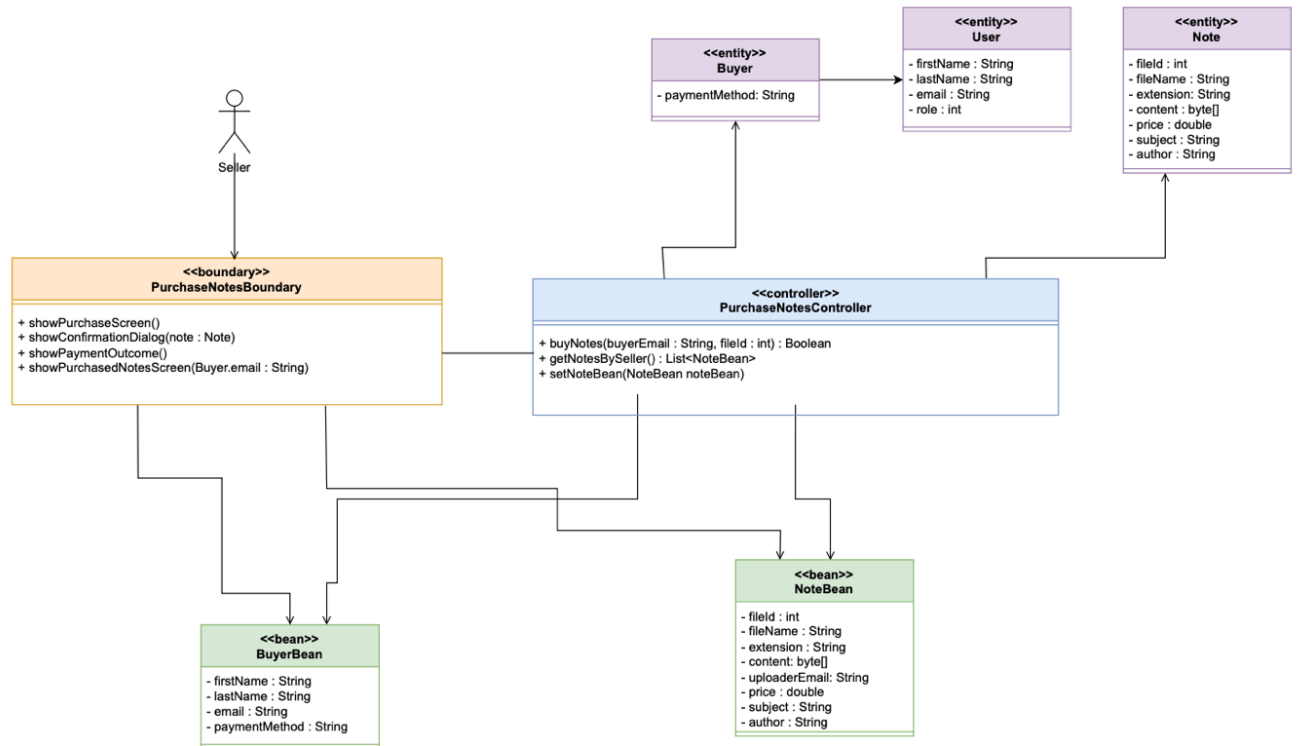
7a. *The payment went wrong*: display an error message with the all the details given by the payment system.

NOTE: these internal steps do not correspond at all to the final version of the application.
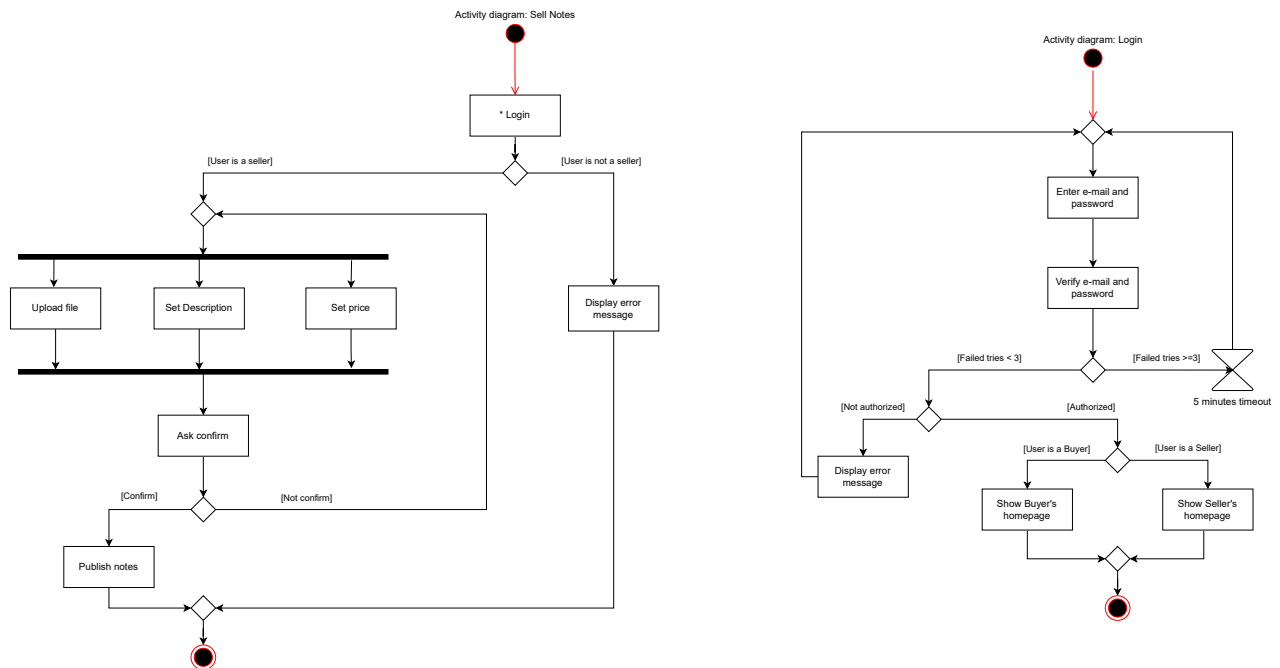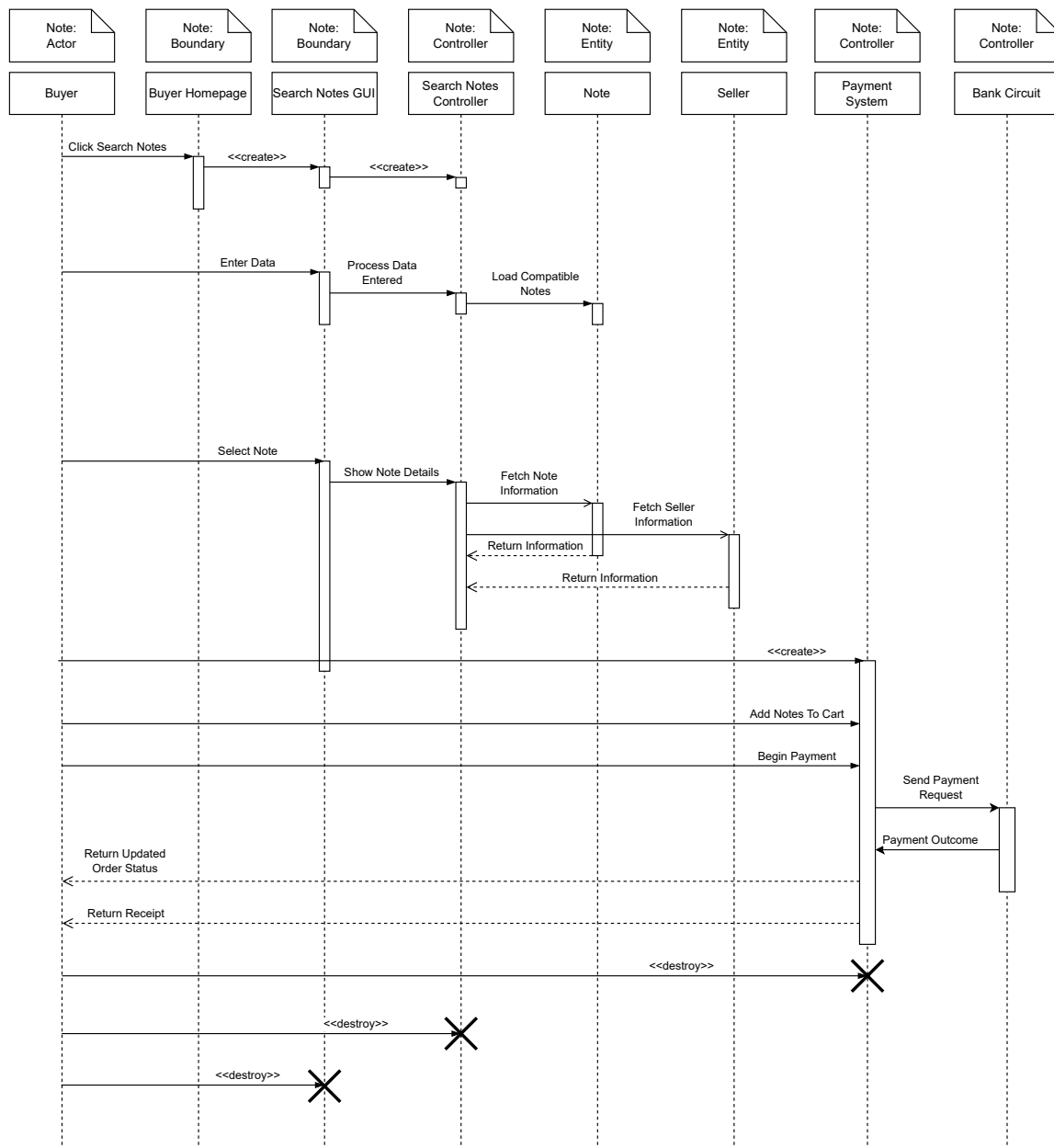
Class Diagram

BCE

MVC and Design Pattern

Activity Diagram

NOTE: this Activity Diagram does not correspond at all to the final version of the application. Many of the actions are written just to meet the requirements of the deliverable and are not implemented.
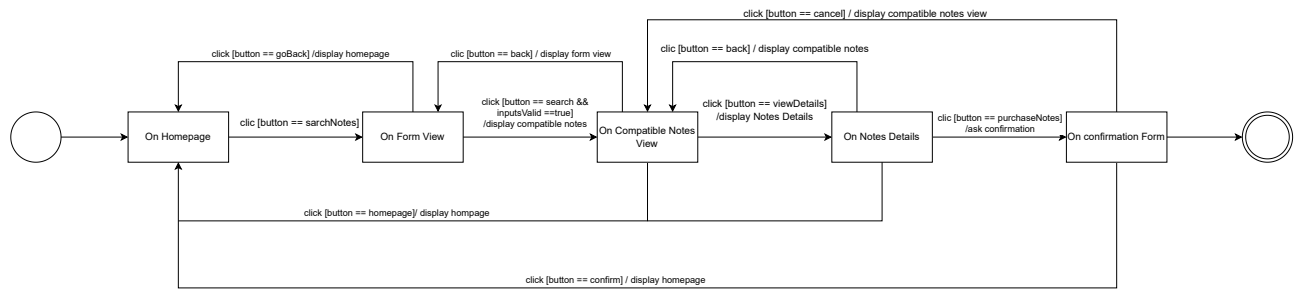
Sequence Diagram

Use Case: Buy Notes (Buyer)

NOTE: The `<<destroys>>` are not implemented since the Garbage Collector oversees the deallocation in Java.

State Diagram

Name: *Buy Notes (Buyer)*

NOTE: this State Diagram does not correspond at all to the final version of the application. Some of the states are written individually just to meet the requirements of the deliverable, but in the application, they are implemented on the same page or not implemented.