

CoffeeHouse

Technical documentation

Table of contents

1. Software requirement specification	3
1.1. Introduction	3
1.1.1. Aim of the document	3
1.1.2. Overview of the defined system	3
1.1.3. Hardware and software requirements	3
1.1.4. Related systems, pros and cons	3
1.2. User stories	4
1.3. Functional requirements	4
1.4. Use case	5
1.4.1. Overview diagram	5
1.4.2. Internal steps	6
2. Storyboards	7
3. Design	10
3.1. Class Diagram	10
3.1.1. VOPC (analysis)	10
3.1.2. Design-level diagram	11
3.2. Design patterns	12
3.3. Activity diagram	14
3.4. Sequence diagram	15
3.5. State diagram	16
4. Testing	16
5. Exceptions	17
6. Discrepancies	17
7. Sonar cloud	17

1. Software requirement specification

1.1. Introduction

1.1.1. Aim of the document

The main point of this document is to provide a general overview of the system and its functionality, using diagrams and other tools to focus on various aspects, from the system's design and architecture to the user requirements.

1.1.2. Overview of the defined system

The system allows users to order food and beverages from a café with a scheduled pickup time. Users can select items from the menu, customize beverages, pay online or upon pick up and leave a note to the barista.

The system helps users track their caffeine intake by tracking their beverage consumption and providing various information.

Cafés can customize their menu, track and manage orders in real-time, and accept or reject orders based on factors such as allergen concerns.

Users can use coupons within the system and gift them to other users, offering discounts or free items.

1.1.3. Hardware and software requirements

For hardware requirements, there are no specific recommendations. Software requirements include:

- **Java Runtime Environment:** JRE 23 or higher.
- **Database:** MySQL 8 or higher.

1.1.4. Related systems, pros and cons

Starbucks: Starbucks lets users customize their beverages, but it's restricted to their own chain and isn't available in Italy. In contrast, my system allows users to order and customize drinks across multiple cafés, providing a broader range of options and greater flexibility for users in different locations.

Deliveroo: Deliveroo allows users to order food and drinks from a variety of cafés and restaurants. However, it does not include features for tracking beverage consumption or monitoring caffeine intake. In contrast, my system not only offers a convenient way to order but also helps users keep track of their caffeine levels, making it great for anyone who wants to stay healthy while enjoying their favorite drinks and foods.

1.2. User stories

1. As a user, I want to order beverages and food from a café by specifying the pickup time, so that I don't have to waste time standing in line and waiting for the preparation.
2. As a barista, I want to receive orders in advance and have the option to accept or reject them, so that I can prepare the orders more efficiently and ensure only manageable orders are handled.
3. As a user, I want to create coupons* and send them as gifts to my friends, so that I can give them a nice surprise, like offering a coffee at the vending machine.

*Coupons: There are two types of coupons: a discount coupon that applies a discount to the final price, and an item coupon that adds the items listed in the coupon to the order for free.

Not to be evaluated:

4. As a user, I want to customize my beverage by modifying its options, so that I can personalize it according to my taste and preferences.
5. As a user, I want to monitor my caffeine levels by recording the beverages I consume, so that I can take advantage of the benefits of caffeine without compromising my health.

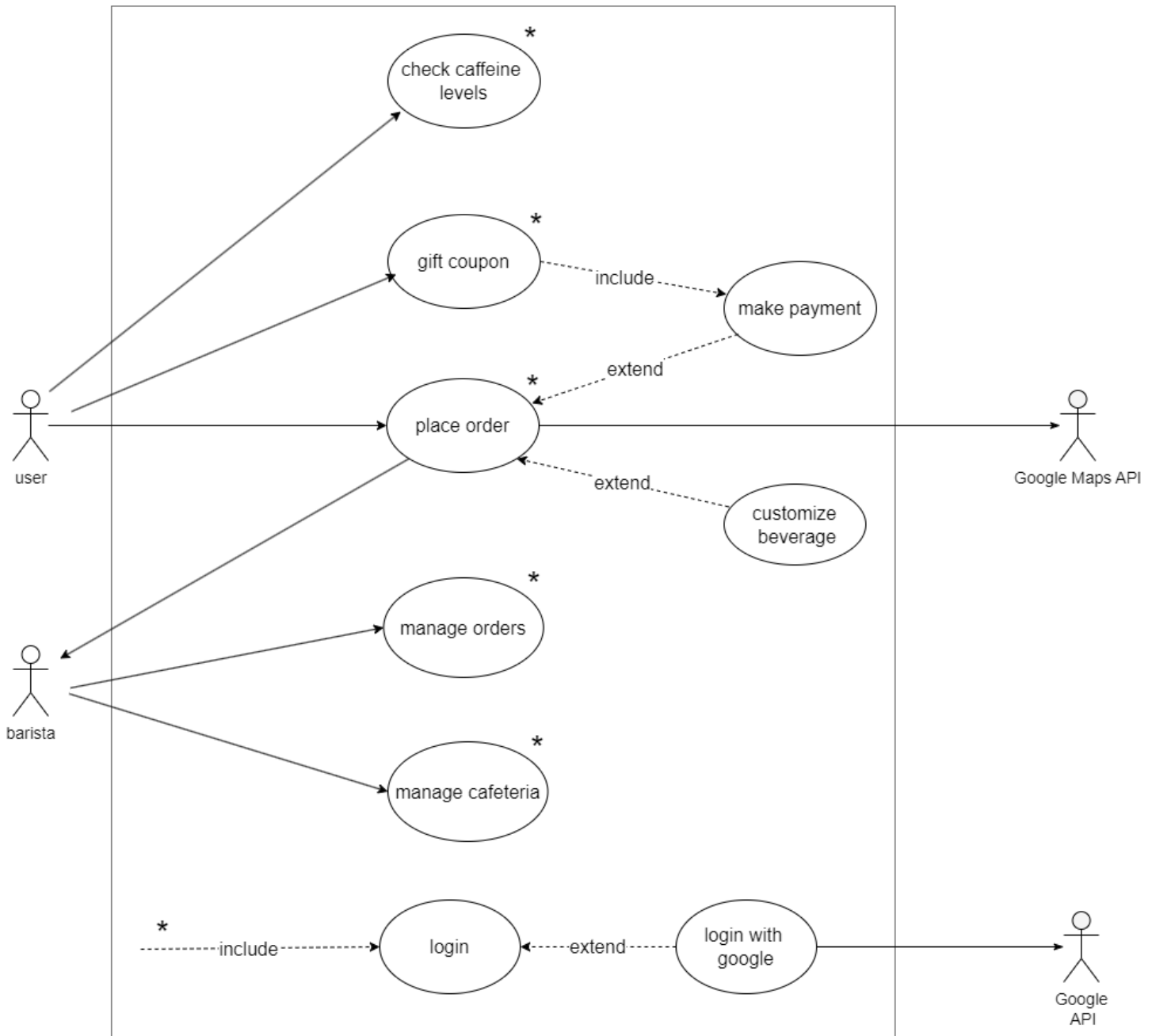
1.3. Functional requirements

1. The system shall provide an order summary at the end of the order showing:
 - added items.
 - pickup time and date.
 - total price.
 - Note.
 - café address.
2. The system shall provide a list of cafeterias filtered by name or distance (based on a city or address) and show the selected café information, including:
 - name.
 - image.
 - city.
 - address.
 - opening hours.
 - description.

3. The system shall provide information on caffeine levels in the body, based on the registered consumptions, showing:
 - The Current caffeine level.
 - The amount consumed in the last 12 hours.
 - The remaining time until ideal caffeine levels for sleep are reached (<50mg).

1.4. Use case

1.4.1. Overview diagram



Annotation:

- Make Payment is an extend for Place Order as there is the possibility to pay upon pickup.
- Google Maps API are used during the search for cafés filtered by distance

1.4.2. Internal steps

Use case: place order

1. The system verifies that the user is logged in.
2. The system gets available cafeterias.
3. The user selects a cafeteria.
4. The system retrieves the menu of the selected cafeteria.
5. The user selects an item to add to the order.
6. The system asks the user if he wants to customize the item.
7. The user declines.
8. The system asks the user if he wants to add another item.
9. The user declines.
10. The system prepares a form to enter information about the pickup date and time, additional notes for the barista, payment method and coupon.
11. The user inserts the information.
12. The system shows the order summary.
13. The system asks the user if he wants to confirm the order.
14. The user accepts.
15. The system sends the order to the barista.
16. The barista evaluates the order.
17. The system shows the response to the user.

Extensions:

- 1a. *The user is not logged in:* Notify the user and end use case.
- 2a. *The system doesn't retrieve any cafeteria:* Notify the user and end use case.
- 6a. *The user accepts:* The system prepares a form to let the user customize the item, go to step 8.
- 8a. *The user accepts:* go to step 5.
- 13a. The user declines: end use case.
- 14a. *The user has chosen to pay online:* The system prepares a form to pay, go to step 15.

2. Storyboards

Use case: Place order

Account login

username

password

[register](#)

LOG IN



caffeine
levels



place
order



my
orders



profile



logout



settings

PLACE ORDER - Search cafeteria



Q Tor vergata



Bar di ingegneria

📍 2,8 km - Via del Politecnico, 1, 00133 Roma RM - 5



Tram depot

📍 20,5 km - Via marmorata, Roma RM - 3





caffeine
levels



place
order



my
orders



profile



logout



settings

PLACE ORDER - Search cafeteria



Bar di ingegneria

📍 2,8 km - Via del Politecnico, 1, 00133 Roma RM

opening hours ▼

today 08:00 - 19:00

description

The Bar of Engineering at Tor Vergata University offers a variety of drinks, snacks, and quick meals. It's a casual spot where students and staff can relax between classes, grab a coffee, or have a bite to eat

continue



caffeine
levels



place
order



my
orders



profile



login



settings

PLACE ORDER - add items



🔍 Search



mocha cocoa

3\$

customize



creamy latte

5\$

customize



caffè espresso

1\$

customize



iced coffee

7\$

customize



creamy latte



iced coffee



total: 12\$

continue



caffeine
levels



place
order



my
orders



profile



logout



settings

PLACE ORDER - Order details



pickup time:



15/11/24



13:00

select coupon ▼

leave a note:

payment method



pay now



pay at pickup

continue



caffeine
levels



place
order



my
orders



profile



logout



settings

PLACE ORDER - Order summary



items

creamy latte 5\$

iced coffee 7\$

café

Bar di ingegneria

pick up address

Via del Politecnico, 1, 00133 Roma RM

date and time

15/11/24 13:00

payment method

pay at pickup

total price

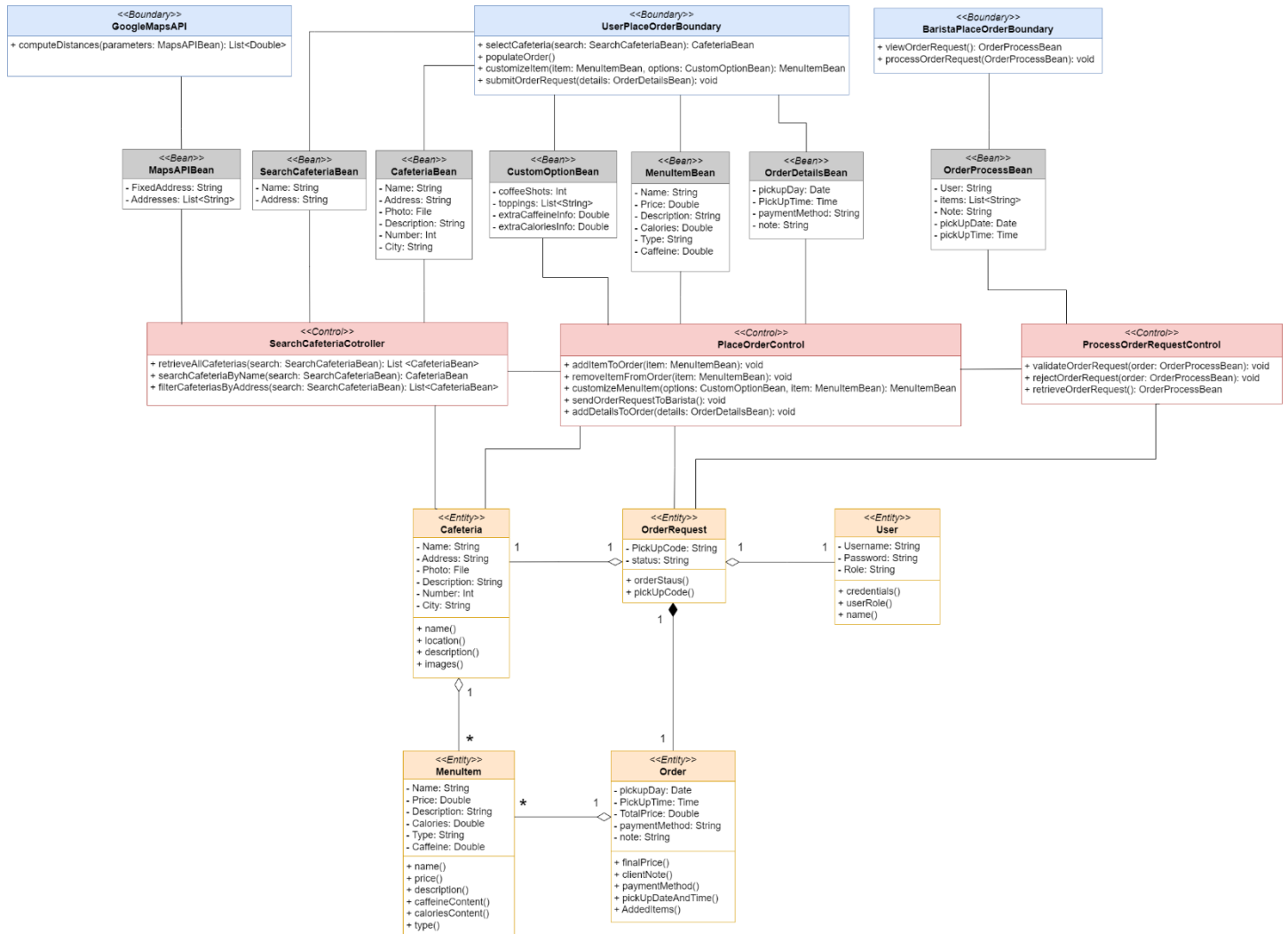
12\$

confirm

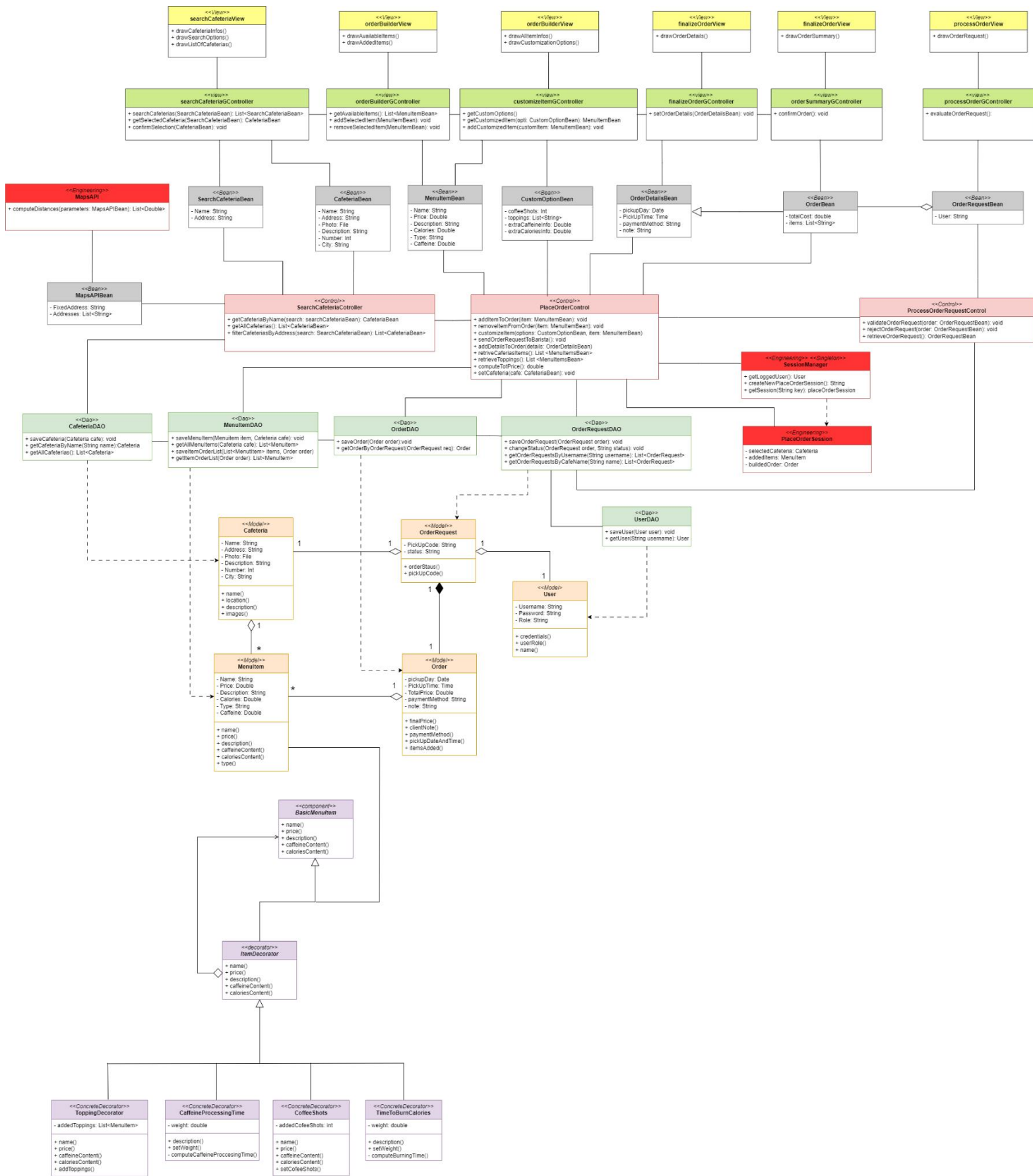
3. Design

3.1. Class Diagram

3.1.1. VOPC (analysis)



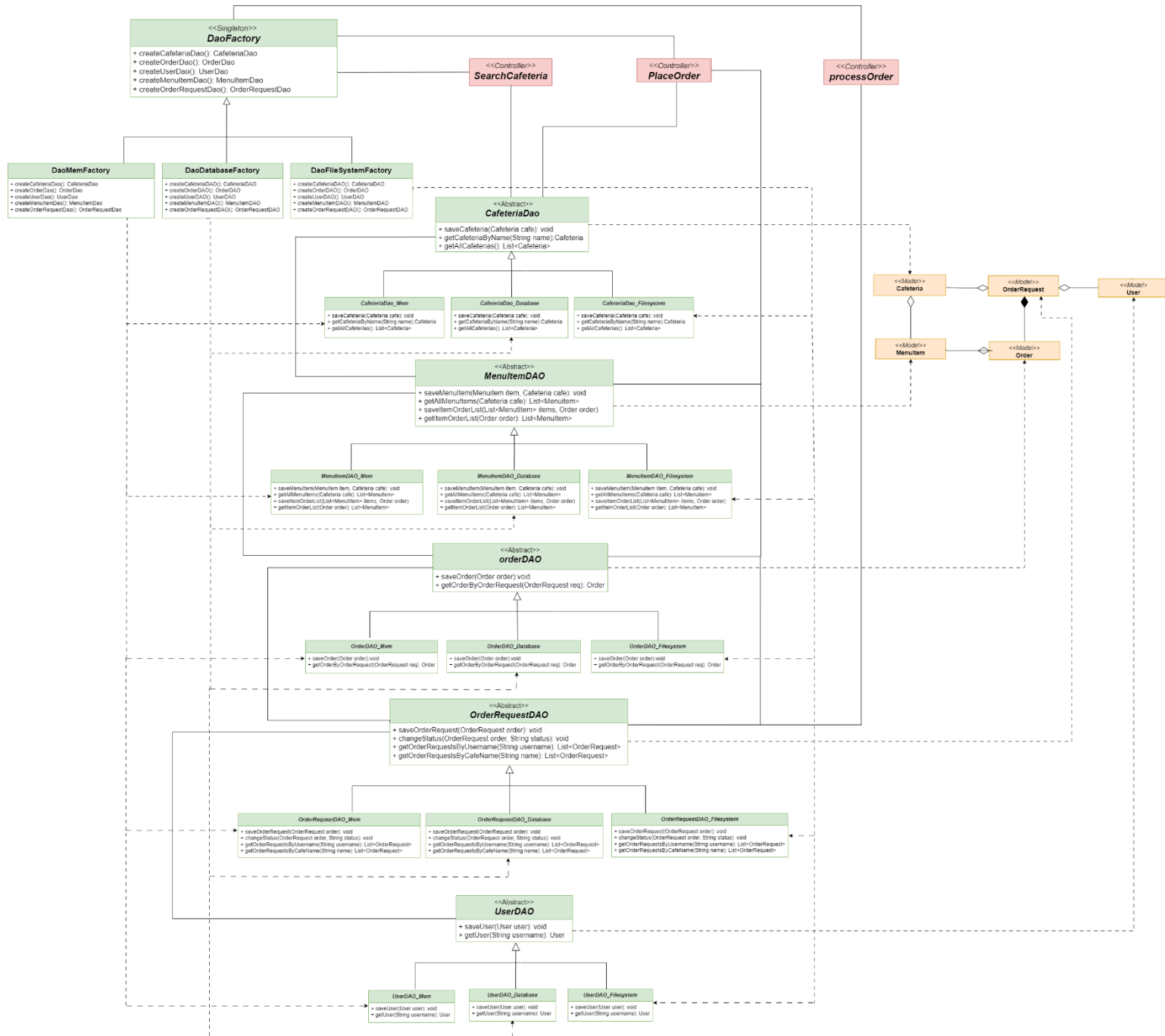
3.1.2. Design-level diagram



3.2. Design patterns

Abstract Factory

The Abstract Factory pattern has been used to manage the creation of different DAOs. The pattern fits perfectly with our case, where the persistence type represents the different families, while the various DAOs related to the models are the different products.



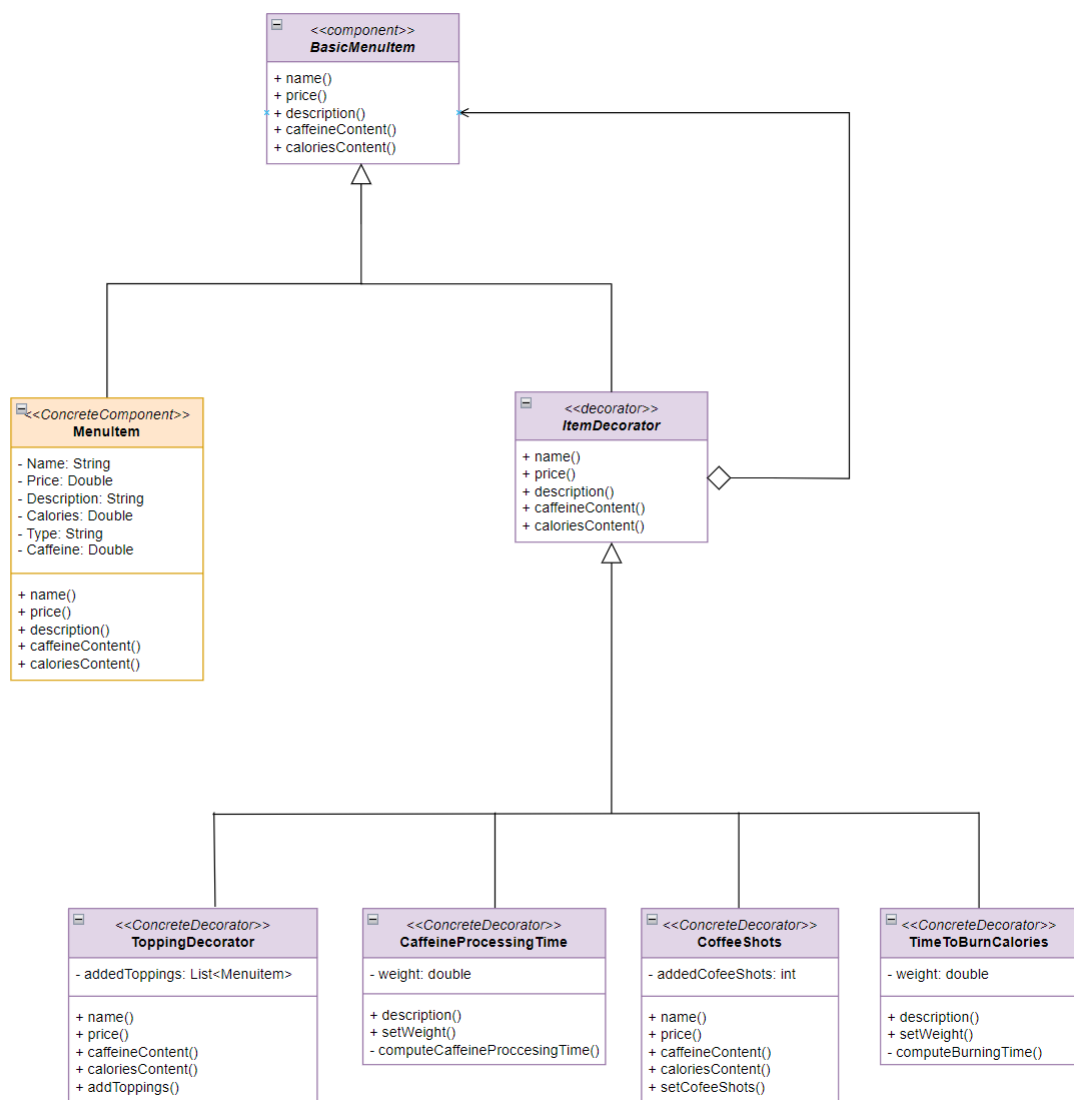
Singleton Pattern

The Singleton Pattern has been used to ensure a single instance of the database connection, avoiding the need to reopen it repeatedly and preventing possible duplications and inconsistencies across various accesses.

It has also been applied in some in-memory DAOs to maintain references and have access to a list of “saved in persistence” models.

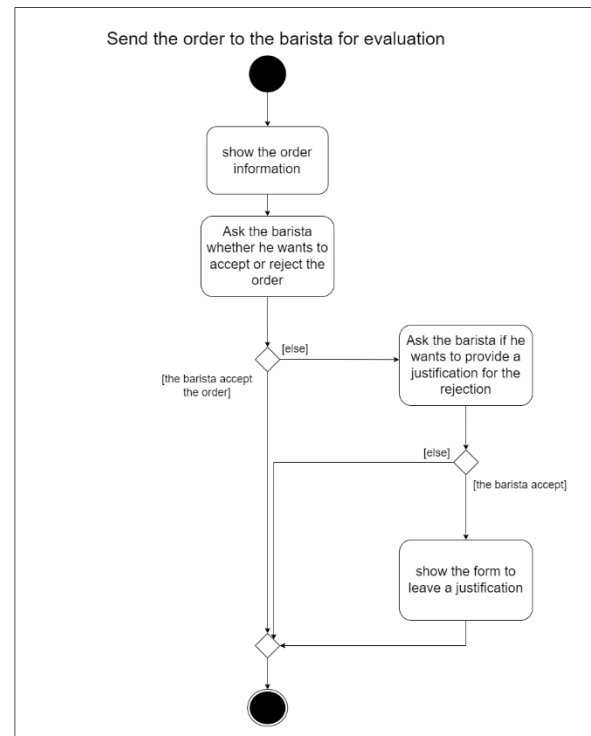
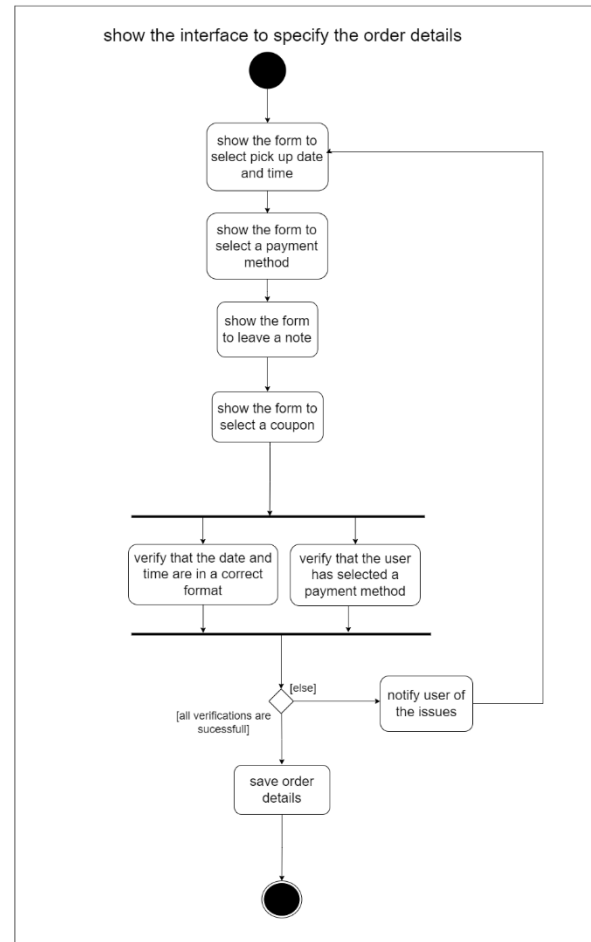
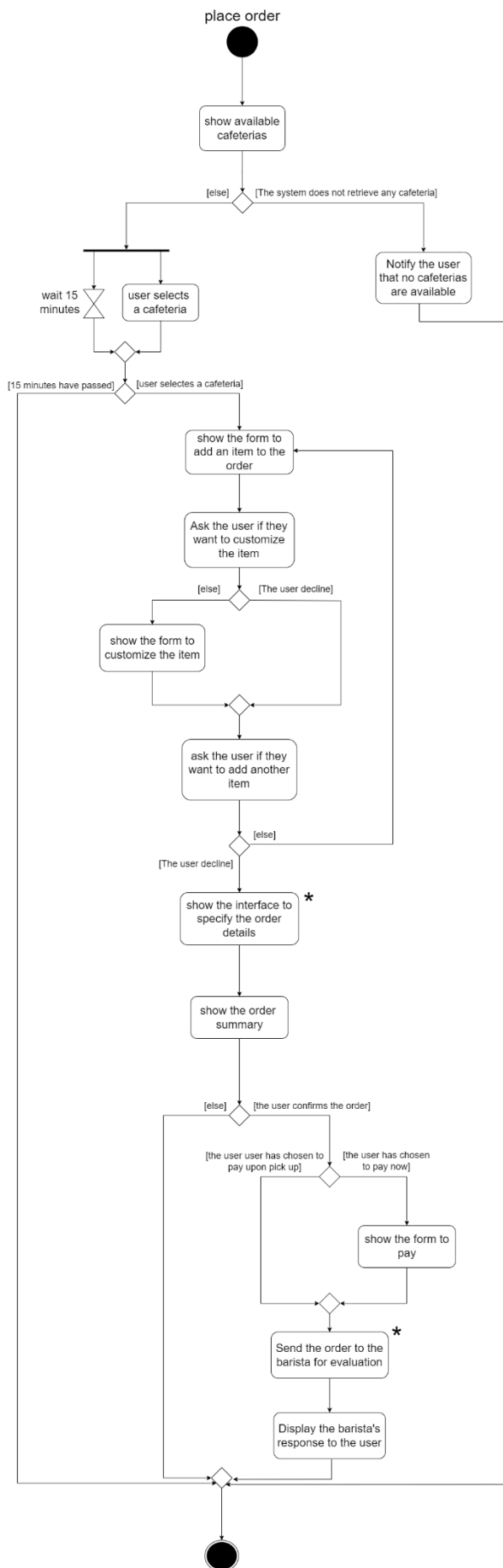
Decorator Pattern

The Decorator Pattern has been used to customize menu items, adding toppings, coffee shots, and extra information about the items, such as the time to metabolize caffeine and the amount of running needed to burn the calories contained in the item.



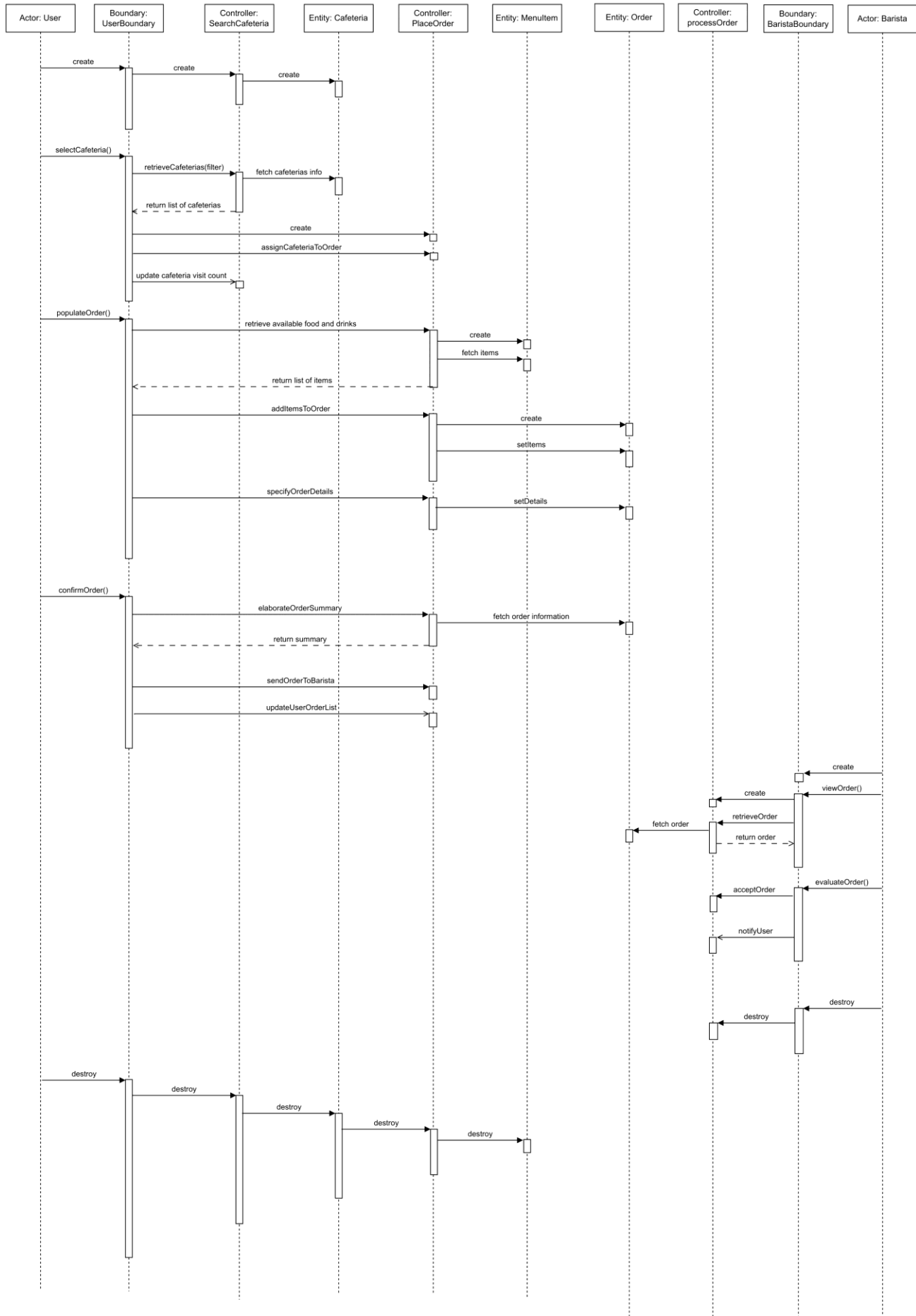
3.3. Activity diagram

Use case: Place order



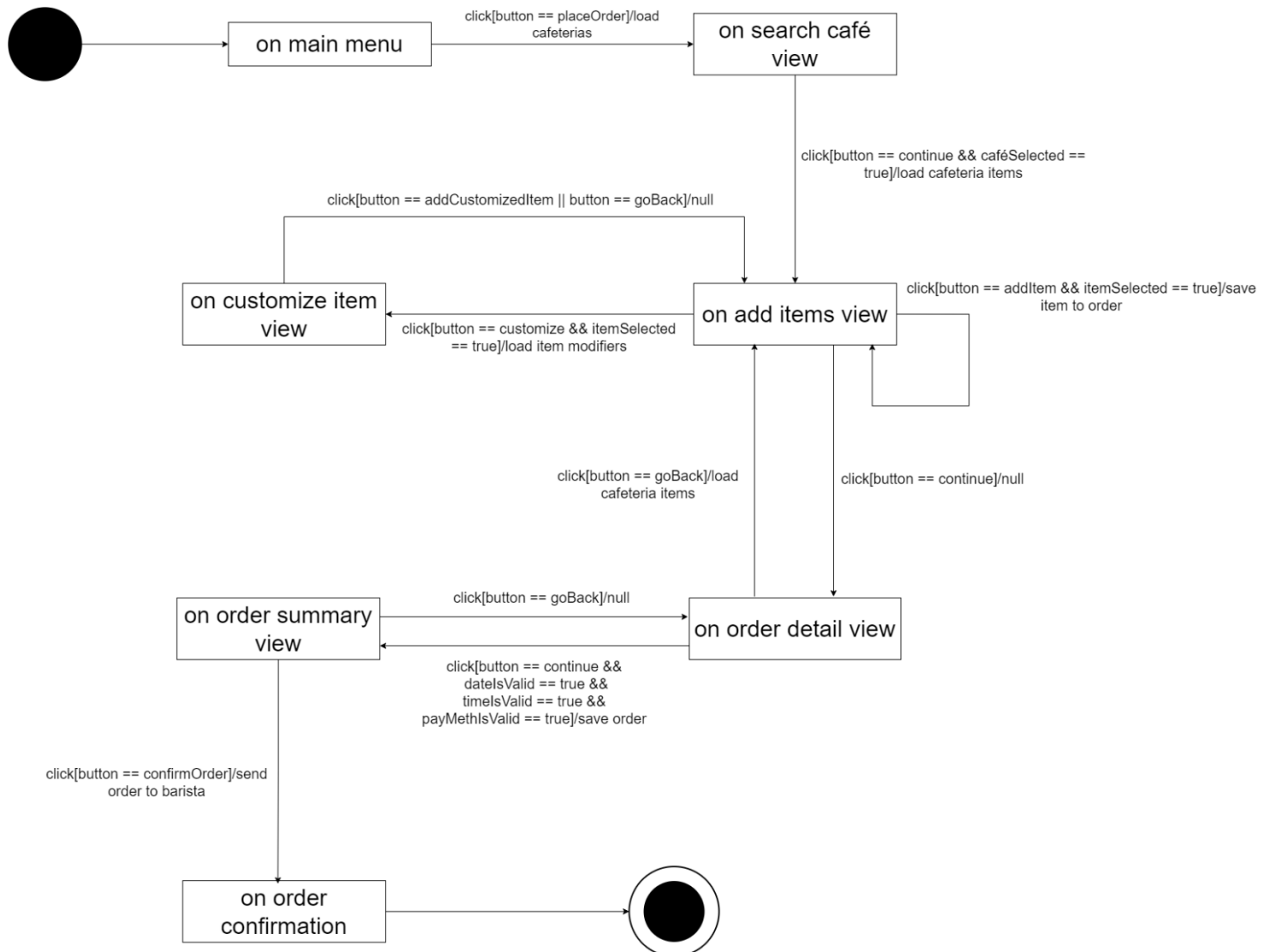
3.4. Sequence diagram

Use case: Place order



3.5. State diagram

Use case: place order



4. Testing

- **TestSearchCafeteria:** Used to test the search feature for finding cafeterias in the system, ensuring it retrieves the correct cafeteria based on the name. The test also verifies that the correct exception is thrown when no cafeterias are found matching the search criteria.
- **TestItemCustomization:** Used to verify that the decorator pattern is correctly applied to modify the beverage, making sure that the changes are properly reflected in the modified beverage as expected.
- **TestDbConnection:** Used to test the connection to the database.

5. Exceptions

- **NoCafeteriasFoundException:** This exception is thrown when the café searched by the user is not present in the system.
- **WrongFormatException:** This exception is thrown when the user enters an input that does not conform to the expected format, such as an incorrect date or time format.
- **SystemErrorException:** This exception is thrown when a critical system issue occurs, such as database failures, service unavailability, or various exceptions that prevent normal operation and do not depend on the user input.
- **NoUserFoundException:** This exception is thrown when the user attempts to log in but is not found in the system.

6. Discrepancies

coupon and online payment not implemented in the use case place order.

7. Sonar cloud

https://sonarcloud.io/project/overview?id=MarmoreSimone_ISPWproject

<https://github.com/MarmoreSimone/ISPWproject>