Your submission to Graded Online Exercise (Tianhao Gu)

Exam Mode: Exam

Date: Aug 8, 2022 **Time:** 15:15 - 16:55

Duration: 1h 40min 0s

Examiner: Prof. Dr. Stephan Krusche **Module number:** IN0006 and INHN0006 **Course:** Introduction to Software Engineering

Exercises: 8 **Points:** 100

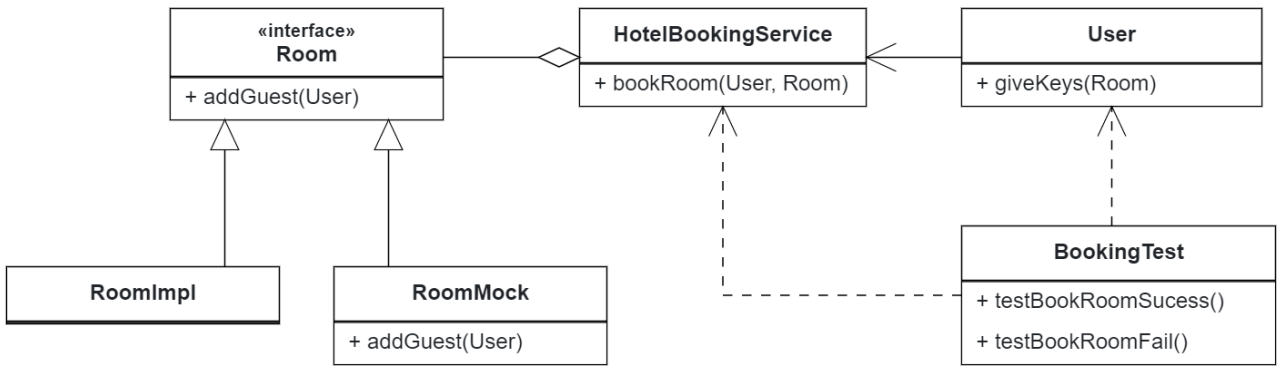
Your result will be published here as soon as the correction is finished. You can get to this page by clicking on this exam in the exam overview of this course.

Exercise 1

Quiz 4 [4 Points]

1) Testing

Consider the following UML class diagram. Which classes are part of the test model?



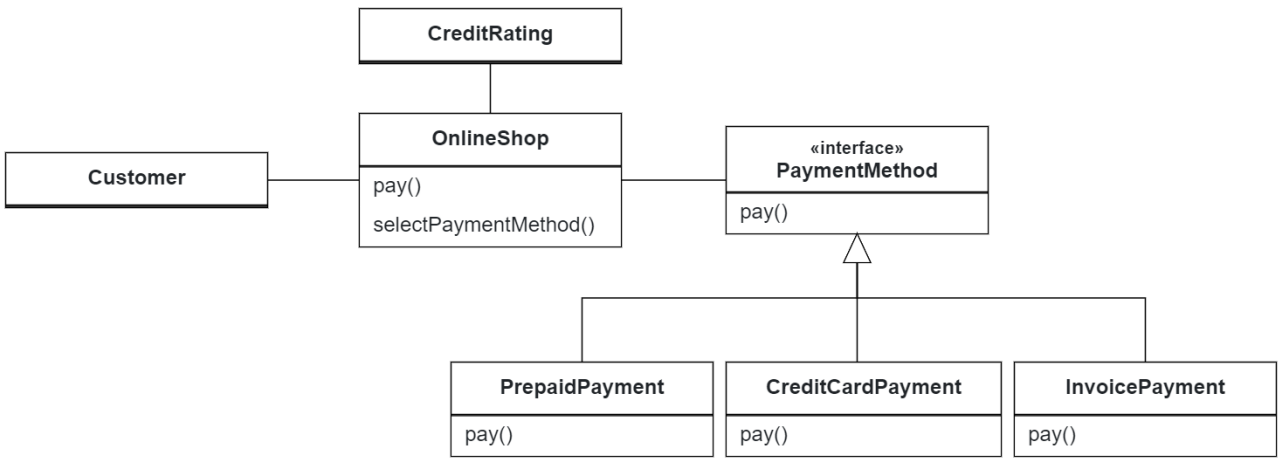
Please choose all correct answer options

Points: 2

- [ ] Room
- [ ] BookingTest
- [x] User
- [ ] RoomMock
- [ ] RoomImpl
- [x] HotelBookingService

2) Design pattern

Identify the design pattern in the diagram below



Please choose all correct answer options

Points: 2

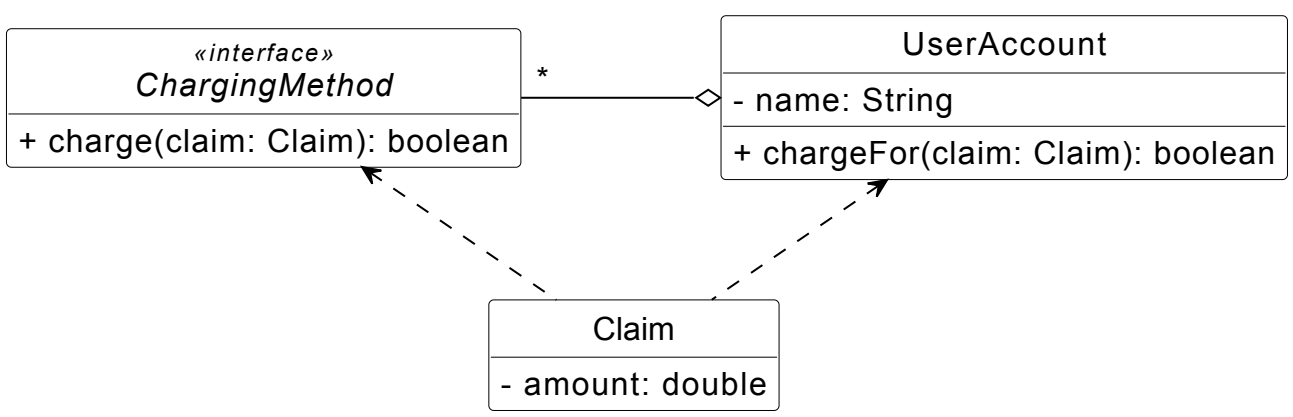
- [x] Strategy Pattern
- [ ] Proxy Patterm
- [ ] Composite Pattern
- [ ] Façade Pattern
- [ ] Observer Pattern

Exercise 2

Testing [25 Points]

The TUMShop wants to extend its online shopping services. You are working on the software responsible for handling the different payment charges. This payment service will support many charging methods. However, none of them are already implemented. The characteristic of this service is that every user can order their added charging methods in a sorted list. When a claim is made by the seller, the system will ask the user account to process the payment. The user account contains a selection algorithm that tries one method after the other in the order of the list until one works. Therefore, it can return on the first try or even walk through the whole list.

The following UML diagram shows the current implementation of the system.



Your task

Implement one JUnit 5 test case to test that the selection algorithm of `UserAccount` is working correctly for an account with three payment methods. Use the Mock Object Pattern with `EasyMock` to mock the `ChargingMethod`s.

Test that if

- the first charging method fails and
- the second charging method succeeds

then

- the third charging method is not called and
- the charging method selection algorithm succeeds

Hints

- Do not change anything in the root directory or the `src` folder!
- Make sure to use the appropriate JUnit 5 and EasyMock annotations.
- Use your local IDE to solve the exercise.

Repository Link: **https://bitbucket.ase.in.tum.de/scm/EIST22EXAMTESTINGA/eist22examtestinga-ge42tij.git**
Last Commit Hash: 30d3b98

Exercise 3

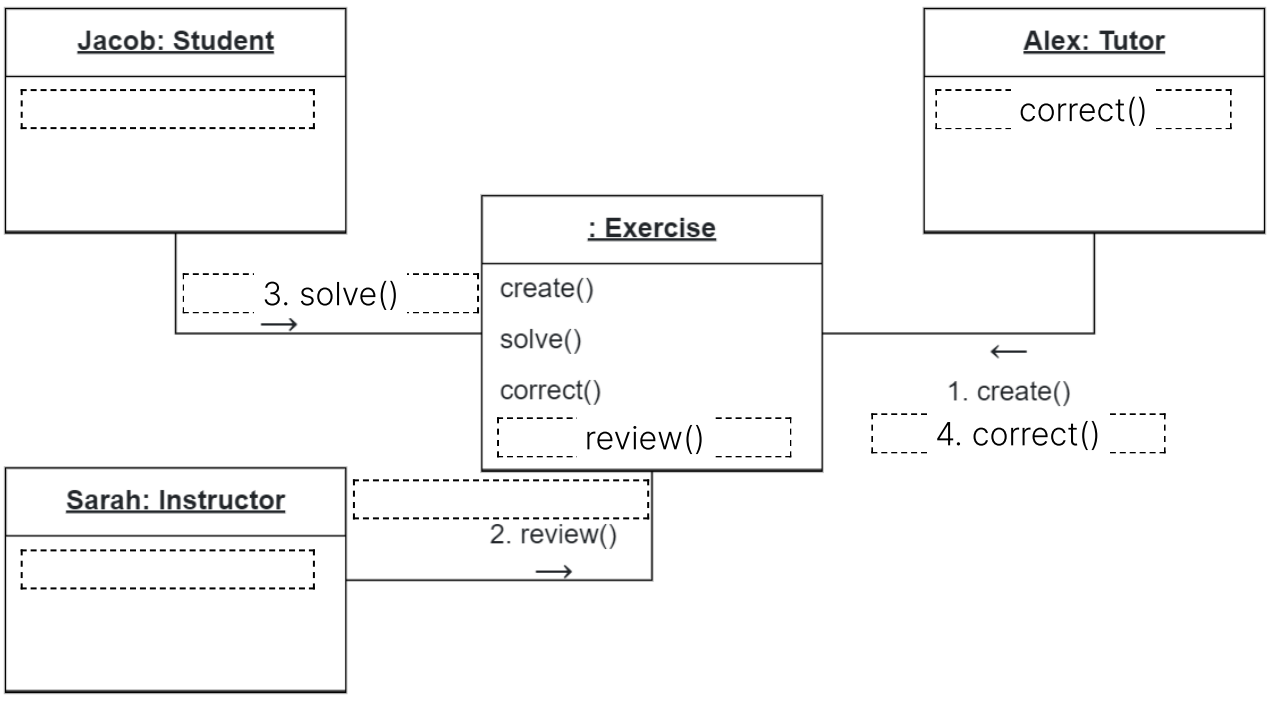
Quiz 1 [6 Points]

1) UML communication diagram

As part of the EIST course, Alex, one of the tutors, creates an exercise. This is then reviewed by Sarah, the instructor. After that, Jacob, one of the students attending the course, solves the exercise. After the submission deadline this is then corrected by Alex.

Complete the diagram by dragging and dropping the items into the correct places.

Points: 2



Drag & Drop: Place the suitable items on the correct areas.

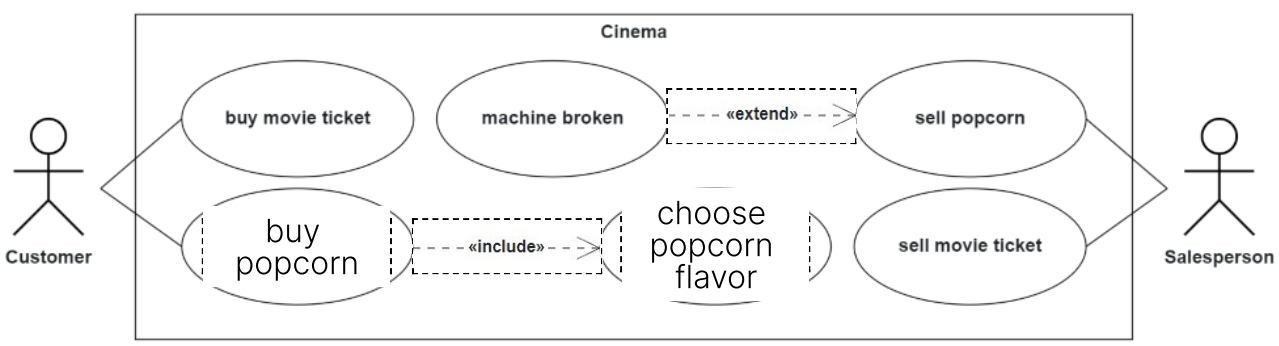
solve() | 2. review()

2) UML use case diagram

A customer buys a ticket for a movie she likes. After that she wants to buy some popcorn and asks the salesperson to give her salty ones. The salesperson informs the guest that the popcorn machine is currently not working and is offering the guest free gum as a compensation.

Complete the diagram by dragging and dropping the items into the correct places.

Points: 2



Drag & Drop: Place the suitable items on the correct areas.

watch movie trailers - - - - «include» - - - > < - - - «include» - - - -

<- - - «extend» - - - - - <- - - «include» - - - -

3) Requirements

A software developer got the task to develop a software for a car with the following requirements. Which of these are `functional requirements`?

Please choose all correct answer options

Points: 2

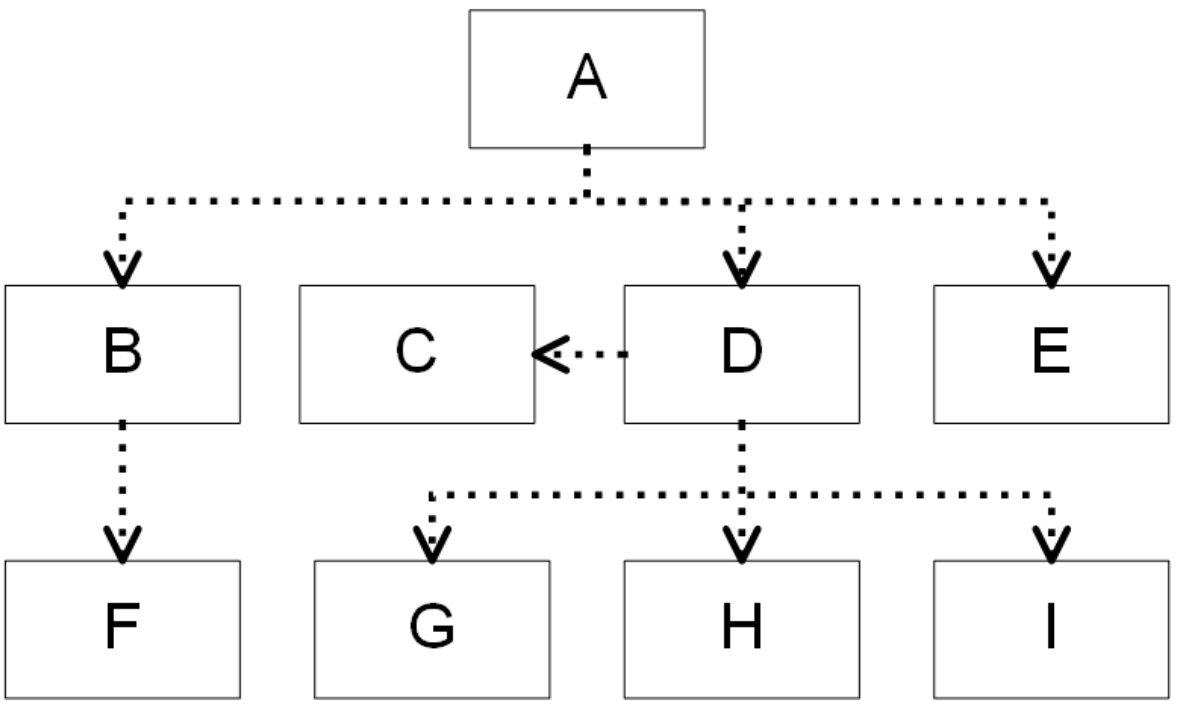
- [ ] The system has to be available 99.99% of the time
- [ ] The data saved on the system should be preserved even if the system crashed
- [x] Music can be turned on/off
- [x] The air condition temperature can be set by voice command

Exercise 4

Quiz 3 [6 Points]

1) Integration testing

According to the following diagram, how many drivers and stubs are needed to test subsystem D?



Please choose all correct answer options

Points: 2

- [ ] 2 drivers and 2 stubs
- [x] 1 driver and 2 stubs
- [ ] 1 driver and 4 stubs

- [ ] 3 drivers and 2 stubs

2) Process

A farmer sows her fields twice a year. Each time, she chooses the plants based on which demand is the highest in the respective season.

Which of the following aspects describe this process?

Please choose all correct answer options

Points: 2

- [ ] Linear, incremental
- [ ] Parallel, iterative
- [x] Iterative, adaptive
- [ ] Incremental, adaptive

3) Communication

The scrum teams at EISTCoffee GmbH decided to discard their daily scrum meetings to save time. Instead they agree to update each other only when necessary via Zulip.

Which of the following aspects describe this communication event?

Please choose all correct answer options

Points: 2

- [x] Unscheduled, asynchronous and informal
- [ ] Scheduled, synchronous and formal
- [ ] Unscheduled, asynchronous and formal
- [ ] Scheduled, synchronous and informal

Exercise 5

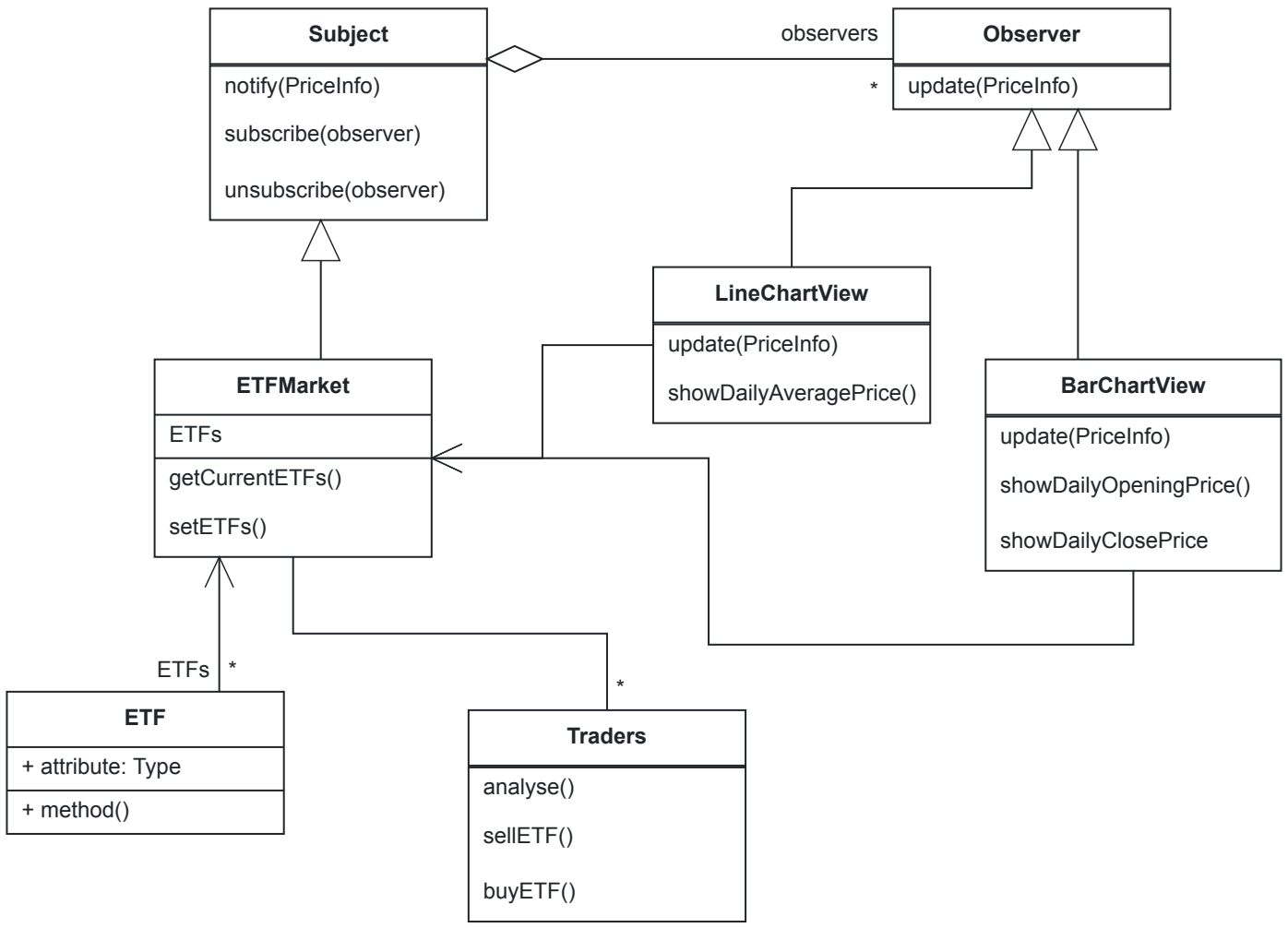
Design patterns [13 Points]

An ETF is an exchange-traded fund that allows investors to invest in many different stocks by buying a single security. Traders should first analyze the ETFs they're interested in to decide whether to sell or buy ETF. Some websites offer charts that help to make an informed decision. Assume you are working for ETF Visualization GmbH, and you participate in developing a website that offers two price chart views: the line chart view and the bar chart view, both of which traders can view in the graphical user interface. The line chart view represents the course of the daily average price. The bar chart view represents the daily opening and closing price.

In this scenario, you have a closer look at the S&P500 ETF. This ETF stores its daily opening and closing prices and the average price. As soon as there is new price information, the ETF notifies all chart views and sends them the latest course prices.

Your task

- Choose an appropriate design pattern for the problem described in this scenario and create an object design model which includes the chosen design pattern.
- Explain in 4-5 sentences which design pattern you have chosen and why you modeled the diagram like this.



Explanation

I choose observer pattern.Because the system should be sychronized,namely the two price chart views should reflect the information which is stored in ETF.When ETF (works like a model in MVC architecture,better to understand) updates,he will notify all his observers(chart views) and send the latest prices ,so that the user can get the latest information and the chart views are all synchronized.

Exercise 6

Process [12 Points]

Imagine you recently graduated from TUM, and now you start your first job at the Munich-based startup EISTCoffee. Before you can begin to apply your newly learned software development skills, your manager wants to test your ability to create UML models. He asks you to improve the developer documentation by creating a UML activity diagram for the workflow of the client acceptance test. You get introduced to Bruno, a developer currently working on a client acceptance test with your customer and who will show you the process at EISTCoffee.

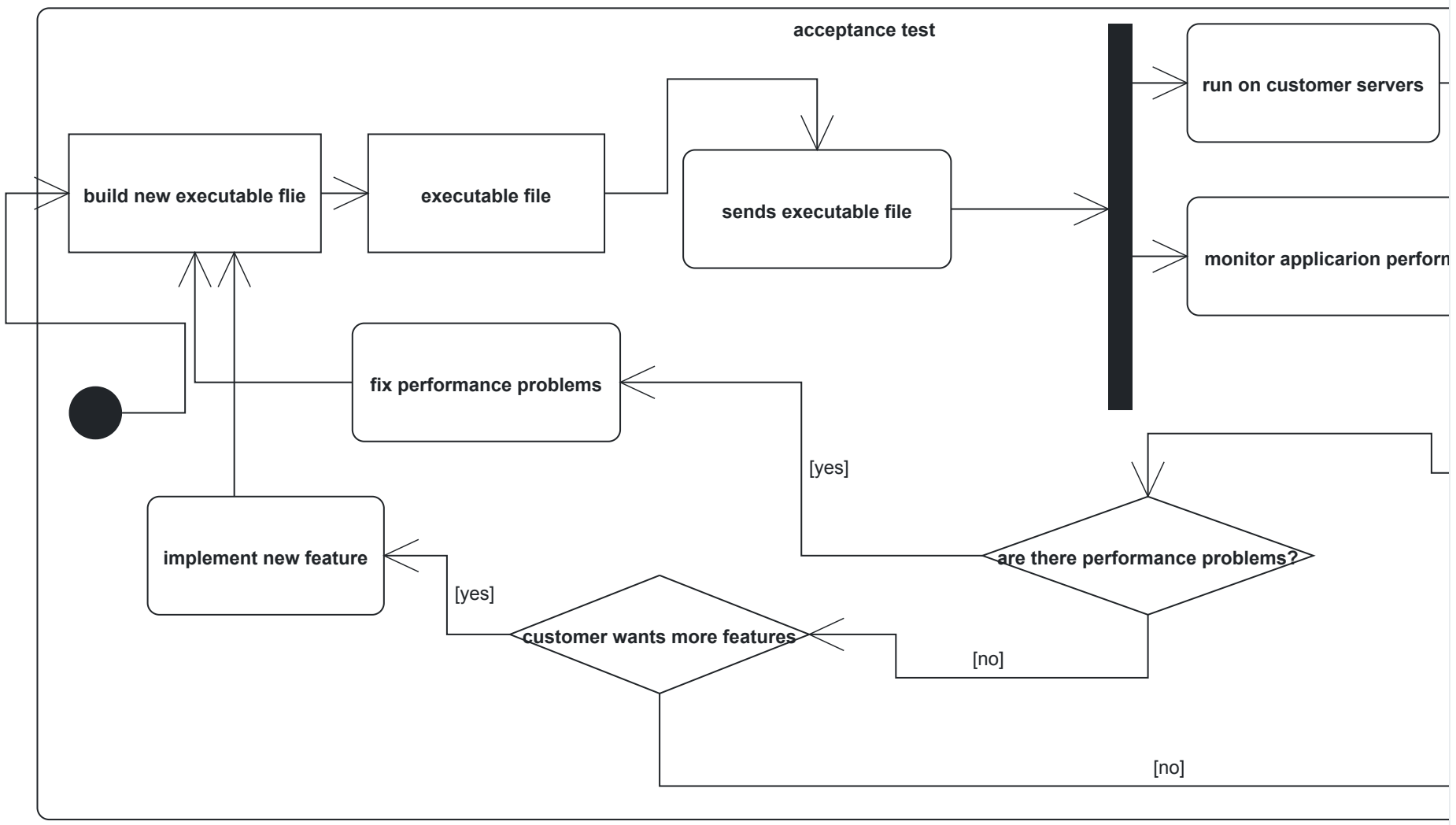
Since the development is already done, Bruno only needs to build an executable file of the application. He sends this executable file to the customer, who then runs it on their servers while monitoring the application's performance. If there are performance problems, Bruno must fix them and start the acceptance test again by creating a new executable file. If there are no performance problems, but the customer wants more features, he needs to implement the new features and start the acceptance test again by creating a new executable file. Finally, if the customer is happy with the current feature set, Bruno is done with his work.

Your task

Create a UML activity diagram for the underlying process of this scenario. Then, explain in 4-5 sentences why you chose to model the process like this.

Hint

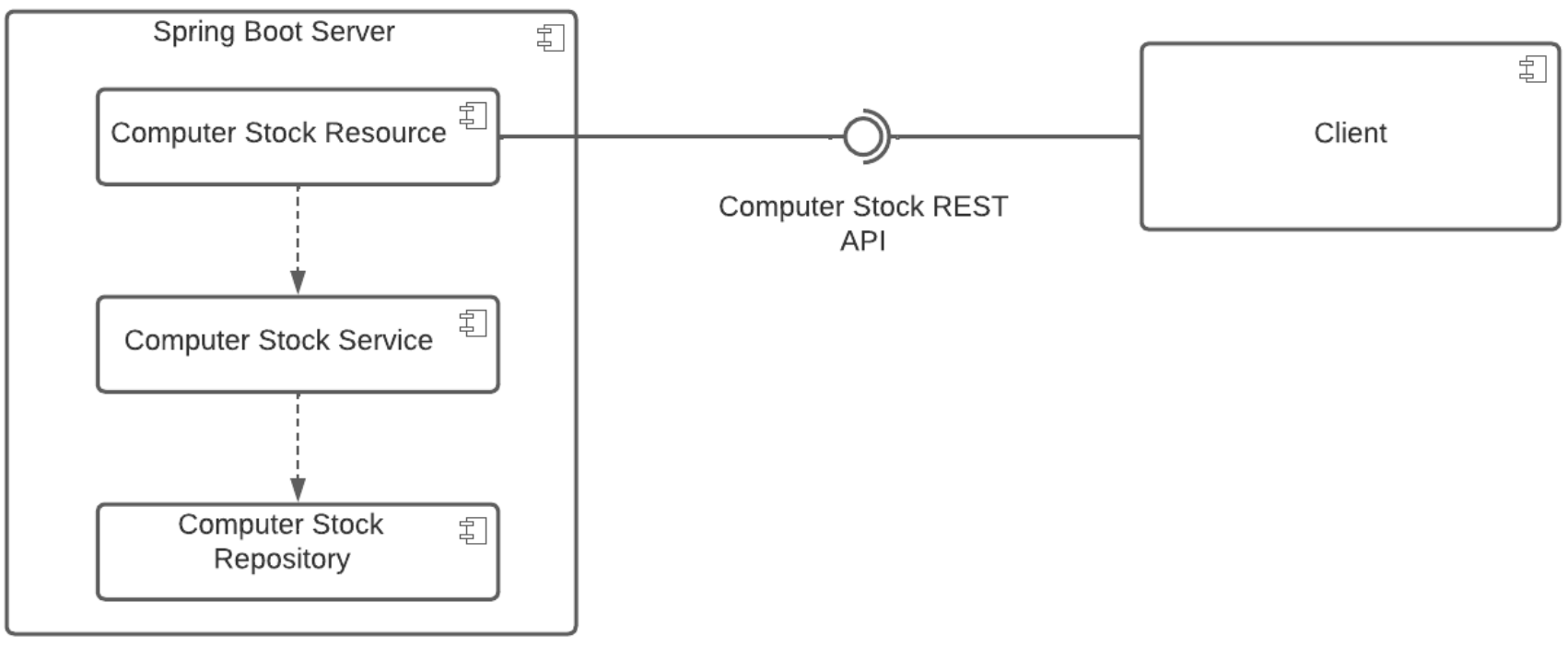
Make sure to model the development workflow as a generic process that would also work for similar scenarios.



Exercise 7

REST [30 Points]

You are working for an online shop which sells computers. Sellers can offer their computers and customers can buy these computers. This online shop also allows all end users to see all offered computers. The online shop system uses the REST architectural style and provides three endpoints in the `REST API`.



Note: This exercise only focuses on the server.

Your tasks

Finish the implementation of the server. A `ComputerStock` represents the stock of computers of a certain type (`processor`, `graphicsCard`, etc.) but additionally contains the attribute `quantity` that defines how many computers of this specific type are currently in stock. Make sure to follow the closed layered architectural style.

1. Offer a computer stock

Create a REST endpoint in `ComputerStockResource` with an appropriate HTTP method and URL that allows sellers to create one stock of computers. You have to create a method with the name `createComputerStock` in `ComputerStockResource` that handles REST calls.
If the passed `computer stock` already has an id, this method should throw a `ResponseStatusException` with code 400 (Bad Request). Otherwise, the resource should invoke corresponding methods in the service and respond with the created computer stock.

2. Buy a computer

Create a REST endpoint in `ComputerResource` with an appropriate HTTP method and URL that allows customers to buy one computer of a specific type. Hint: Include `computerStockId` in the URL to identify which computer the user wants to buy. You have to create a method `buyComputer` in `ComputerStockResource` that handles the REST calls. In addition, you have to make changes to `buyComputer` in `ComputerStockService` to decrease the quantity accordingly.
If no computer stock exists with this id, the endpoint should throw a `ResponseStatusException` with code 404 (Not found). If the computer has a quantity of 0, the endpoint should throw a `ResponseStatusException` with code 400 (Bad Request). Otherwise, the resource should invoke corresponding methods in the service and respond with the updated computer stock.

3. Retrieve all computer stocks

Create a REST endpoint in `ComputerStockResource` with an appropriate HTTP and URL that allows customers to retrieve all computers. Add an optional request parameter `onlyAvailable` to only retrieve those computers with a quantity greater than 0. You have to create a method `getAllComputerStocks` in `ComputerStockResource` and implement required functionality in `ComputerStockRepository`.
If the parameter is not defined in the request, the endpoint should use `false` as the default value. The endpoint should respond with a list of computer stocks.

Hints

- Only adjust code in the classes `ComputerStockRepository`, `ComputerStockService`, and `ComputerStockResource.`
- You can execute the main method of `ServerApplication` to check if your server application starts up and runs.
- Use your local IDE to solve the exercise.

Repository Link: **https://bitbucket.ase.in.tum.de/scm/EIST22EXAMSHOPCO/eist22examshopco-ge42tij.git**
Last Commit Hash: 5c4ac7c

Exercise 8

Quiz 2 [4 Points]

1) Object design

You have created a game that supports airplanes, boats and motorcycles. Even though they all offer a similar set of methods like `steer()`, `refule()` or `park()`, the implementation of those methods differ significantly. Which object design concept(s) would be applicable?

Please choose all correct answer options

Points: 2

- [x] Specification inheritance
- [ ] Generalization
- [ ] Specialization
- [ ] Implementation inheritance

2) Design goals

Drag and drop the correct design goal trade-off to the space next to the description.

Points: 2

With its new Smart TV series, Samsung wants to offer customers all the latest entertainment trends. They need to make sure that the devices can still be used in an easy and intuitive way.	functionality vs. usability
Your car sharing startup wants to offer three new vehicle types in their next app release that is scheduled in one week. You're running low on time and might have to postpone one of the vehicle types to a future release.	rapid development vs. functionality
To cut their cost, Uber decides to lay off two thirds of its test engineers and instead rely on their black box fuzzing system.	cost vs. robustness

Drag & Drop: Place the suitable items on the correct areas.

portability vs. efficiency

backward compatibility vs. readability