

Assignment 4

MET CS 665 - Software Design and Patterns

Using Legacy Systems

(20 points)

GitHub Invitation Link

<https://classroom.github.com/a/qlutWmNt>

1 Application Description

An online book shop company has been very successful in selling books in the past years. The company management plans to open the online shop as a marketplace to other products and third-party companies so that they can also sell their products using the online platform.

The company needs to update the backend software system of the online shop with the following requirements:

1. **Customer Data Access.** The customer data is retrieved from a large database by using the customer email address only. The main company has a lots of legacy systems that work with the existing interfaces to get the customer data.

The company engineers plan to change this system. In the new online shop (marketplace), all components including external companies have to retrieve customer data using email address and phone number. The company plans to implement the new system in a later phase. At the current stage, engineers like to expose only the new API to external companies. Consider that we have the following two APIs:

```
// Old System
interface CustomerDataOld{
    Customer getCustomer(String email);
}
```

The new system has the following API to access customer data using customer ids.

```
// New System
interface CustomerDataManagementSystem {
    Customer getCustomer(String email, String phoneNumber );
}
```

Customer data requests should include both customer email and phone number. The customer data storage system returns customer data item as an object encapsulating all customer data attribute/values.

As engineers have not yet developed the new system, all of the requests have to be converted to the old API and use the old system. The old system is only using the email address. To convert the requests to the old API, customer email address should be forwarded and customer phone number be dropped.



2. **Customer Data Processing.** The customer data has to be processed in different steps. for example like the following steps, 1. EMail address verification 2. Background Checks 3. Welcome email generation and send out 4. Rejection email generation, and others.

Depending on the situation, the company may need to run some of these data processing steps and needs the flexibility to add or remove them. Design your implementation to provide this flexibility.



3. **Order Processing. (OPTIONAL Requirement)** Orders processing is very complex and includes multiple sub-processes. The order processing includes multiple sub-process, like the following steps, 1. product pick-ups, 2. packaging (if a gift then use gift wrap) and 3. package delivery. Consider that these 3 steps are very complex and include multiple sub-processes.

The main company requires to provide a base implementation of these 3 processes and allow external companies reuse the basic implementation of these, or have the possibility to jump in specific places of the process execution and changed it as they needed.

You should provide a software design for the implementation of these processes that allows such flexibility.

1.1 Implementation Details

Your implementation should include the following functionalities:

- Create specific implementation of the given interfaces. (You can create some Mockup objects for customer objects)
- Create concrete implementations and test executions as demonstration of your implementation.

Note: In this assignment, you should apply “**at least one correct**” design pattern. We do not mention which one of the design patterns is suitable for this scenario. This would be your task as a software engineer to find it out.

Note: The given application scenario can be implemented in different ways and with different implementation details. There is not only one single correct design and implementation for the above application scenario. Each software developer might consider different assumptions and design the software based on them. In this assignment you are free to have your own detail assumptions and implement the details in your own way based on your own ideas. You should document your assumptions very well in your *README.md* file of your project and consider them in your UML diagrams.

2 Assignment Tasks

2.1 Task 1 : Implementation Description. (2 points)

Describe how would you implement this application considering software design principles. Describe what are your main software design concepts regarding this application.

For example describe:

- How is the flexibility, of your implementation, e.g., how you add or remove in future new types?

- How is the simplicity and understandability of your implementation?
- How you avoided duplicated code?

In case that you have used any design patterns, describe which design pattern you have applied and why.

Write your description in a **README.md** file, use Markdown format <https://spec.commonmark.org/current/> and add the README.md file to the root folder of your project.

We recommend writing this description after you finished the other tasks in this assignment.

2.2 Task 2 - UML Class Diagram. (5 points)

Create a class model for your application, containing 5-8 of the most important classes. It should encompass the functionality of the use case application described above. Show only non-obvious and key methods.

- You can draw your UML graph on a piece of paper and make of photo of it to add to your assignment report.
- You can use any drawing software to create your UML graph, like any vector graph program like Open office draw, MS PowerPoint Illustrator. Sketch. CorelDRAW. Affinity Designer. Inkscape. Adobe Illustrator Draw, etc .
- You need to show the main structure of your program on your UML class diagram, there is no need to draw all the details.
- **Do not use any automated UML graph generator that generates UML graph from your code.**

2.3 Task 4 -Implement your solution in Java (13 points)

- You should use the provided project templates to implement your project.
- Provide a zip file that includes your implementation package.

The zip file should include a **README.md** file that describes how to compile and run your implementation. Create a Zip file from the root of your working directory so that it includes all subdirectories of your project. Remove the binary files before creating the zip files.

- Your zip file should **not be larger than 10MB**, include only source files of your project but not the binaries that you generated.
- Document your code very well. The best way to write documentation inside your code is to write the code while you are implementing the project and not postpone it to later time.
- You should use the Google Java Style Guide (<https://google.github.io/styleguide/javaguide.html>)
- We should be compile your solution on a unix-based operating system like Linux or MacOS. Your project should include a **run.sh** that can be used to compile and run your project.

2.4 Using GitHub Classroom

For all of your assignments you should use GitHub Classroom.

You should use the provided project templates to implement your project. Login to your github account and click on the Github Invitation Link, a fresh private project with propagated template project will be created for you.

Follow the following steps:

1. You need to have your own GitHub account.
2. Login to your own GitHub Account
3. Click on the Invitation Link for this Assignment and Click on Accept to accept the assignment. You can find the GitHub Link for this assignment on the first page of the assignment.
4. You need to provide access to your GitHub Classroom in our account (This is a one time setup for the GitHub Classroom App).
5. Then a private repository will be created for you that you, your instructor and facilitator have access to it.
6. You should use "Star" the assignment repository to be able to find it easier and use it when you develop your assignment.
7. A good rule of thumb is commit your code to the repository as soon as you have implemented some small piece of code that does something and your code can compile.

2.5 Turnin

1. When you are finished with your assignment. You need to download all of your code from GitHub or have the latest version of your code on your computer.
2. Create a single document that has results for all three tasks. For example a PDF document for UML diagrams.
3. Please zip up all of your code and your document (use .zip only, please!). Remember to **remove the binary files**, these are normally in bin/ or target/ folders. The binary files can increase the size of your zip file.
4. Double check if you have uploaded correctly your zip file. You can download back your file, unzip it and check if it is the correct file and it is correctly zipped.

We will evaluate the zip file that you have uploaded to the blackboard and cannot evaluate wrong or damaged files.

You can use the green download button of GitHub, download a zip file of your repository and upload it to blackboard.

Please note that we will grade your final zip file uploaded to the Blackboard, but we will also check the history of your GitHub repository. Both versions should be the same. The main reason why we want to have zip file on blackboard is to archive a zip file of your assignment on blackboard.

2.6 Grading

Your solution should be a self-contained solution that can be compiled and executed based on the instruction given in your README.md file. We recommend to use our project templates, and add your implementation to our template and use build tools like maven or sbt. If your solution program is complete based on the requested functionalities, can compile and run then you would get the full points.

We will grade your solution, and reduce the points for each task based on the following grading policy.

- Your UML diagram does not include important components like Interfaces/Classes - 5% reduction for each component.
- We will compile all solution using "*mvn compile*" command in your project (we will download your zip file, unzip it and run the "mvn compile" - your project should compile using Java JDK 1.8). If your code does not compile for any reason it will cause 10% grade deduction for the implementation task.
- Your code includes functionality bugs - 10% deduction for each bug
- Your code should include a README.md that describes your conceptual solution, how to compile and execute. If your program does not include such README.md file, or your README file does not include all requested infos then we will reduce 10% of points.
- Your program does not implement the requested functionalities - 10% deduction for each functionality.
- We will programmatically check all solutions for plagiarism using jplag (<https://github.com/jplag/jplag>) (We will run some scripts on your codes). If your code is an exact duplicate of someone else's solution, then we can not accept your solution. We will contact you regarding the issue.

2.7 Assignment Completion and Late Work

All assignments should be submitted on time. If there is a delay, the student must be in touch with the instructor and his/her facilitator.

Late submissions without reasons will result in grade deduction.

You can turn in an assignment up to

- **24 hours late, in which case you receive a 10% penalty** (that is, 10 points are subtracted from an assignment that is worth 100 points), or
- **up to 48 hours late, in which case you receive a 20% penalty.**
- Assignments turned in after 48 hours (more than 48 hours late passed) **will not be accepted.**

We kept on saying **no exceptions, but there are exceptions** in very extreme circumstances, with proper documentation. For example, if you obtain a doctor/dentist note stating that you were so ill at the due date/time that you could not reasonably be expected to meet the deadline, it is possible to get an extension.

2.8 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between "cheating" and acceptable and beneficial interaction between peers. Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way — visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**.

As far as going to the web and using Google, we will apply the "**two line rule**". Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.