Team Project - Software Market Place

Submit Assignment

Due No Due Date **Points** 100 **Submitting** a text entry box or a file upload

Team Project: Software Market Place

Assume you are a software architect now. Here is a simplified scenario of a software portal supporting online software store like Amazon. Your company developed an online software system earlier, and now you are asked to design the new service-oriented system based upon the legacy system.

[10] 1. As an architect, you are asked to provide a 4+1 View design for the system using the corresponding UML diagrams: scenario view (use case diagram); logical view (component diagram); development view (class diagram); progress view (sequence diagram); and physical view (deployment diagram).

[6] 2. Your portal will provide a collection of service including: (1) search function, allowing customers to search for interested software; (2) purchase function, allowing customers to buy software using credit card; (3) delivery function, allowing customers to select preferable delivery option (e.g., DVD, download copy, or online service). Each such service will be handled by dedicated subsystems, but you surely do not want customers to know all such details. Instead, you would like to provide an interface for customers to select from, and based on their selections, you will direct them to corresponding modules to serve for them. Two design patterns may be used here together. Briefly explain your recommendation, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 3. As discussed above, your portal will take payment through credit cards. But you do not want to implement your own credit card payment system; instead, you would leverage PayPal service for now. You decide to implement a handler locally in your code which prepares PayPal-compatible data format and forward the call to PayPal. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 4. You decide to adopt a dynamic pricing model. If customers buy multiple software at the same time, you may apply some discounts for them. Furthermore, some software might announce some discounts during a certain time period. Therefore, you will calculate the purchase fee for a customer at check-out time. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 5. You want to create a way which is extensible for you to generate various types of purchase orders. For example, a purchase order of one software for downloading, a purchase order of two software for downloading with discounts counting automatically, etc. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 6. When you create a new purchase order, for performance purpose, you will try to find whether you have already created a purchase order for the same purpose (e.g., for the same customer with the detailed address etc information). If yes, you will try to start from the existing one and modify from there. In this way, you might avoid from checking the database frequently. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 7. You want to create a way which is extensible for you to generate various types of purchase orders supporting various types of software delivery options. For example, a purchase order of one software for downloading, a purchase order of two software for downloading with discounts counting automatically, etc. For another example, a purchase order of one software as pay per use, a purchase order of two software using pay per use with discounts counting automatically, etc. In other words, for supporting each type of software delivery option, the template of the purchase order is associated with a specific type of counting algorithm. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 8. Sometimes a purchase order is complex, for example, a user purchases multiple software using various types of delivery options. You want to design your code base in a way so that you can leverage existing purchase order constructors to combine them for a dynamic need. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[9] 9. You want to build your portal as as a social network for software users and developers, in a way that various social relations can be formed based on the software they purchase. For example, all customers purchased the same software form a social circle.

[3] 9.1 A user can post a message that can be seen by all other users purchased the same software, sharing his/her comments on the software. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 9.2 A software developer may post information to all of their users (who purchased their software) about their new software. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 9.3 A user may post a question in a social circle, about another software he/she is thinking of purchasing. If nobody in his/her circle use the software, he/she might forward the question to their other social circles for answers. Briefly justify your suggestion of design pattern, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 10. Your system may periodically send coupons to all users automatically. Specially, you want these coupons to be customized based on customers' preferences which might be extracted from his/her past purchase history. Will you recommend to use any design pattern to realize this push-mode notification? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 11. All of your subsystems will be connected to a single database storing all data. Briefly justify your suggestion of design pattern, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 12. You plan to leverage some legacy code from the previous online software market function. That means you want to turn those legacy code as reusable components in your new system but with different interfaces. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate). [Hint: You can simulate a legacy function.]

[3] 13. As you are expanding your services into new countries, you may need to add some new features/options to some services. You don't want to change the code structure (because some countries may need the original version) while you add new functions. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 14. Nowadays, users may like to use various types of devices to conduct business. You want to allow user to use smartphone or pad devices to access your services. While customers may use different types of devices to view your app, which architectural style/design pattern would you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[9] 15. You have considered various ways to enhance system QoS.

[3] 15.1 You decide to use one central database to ensure concurrency control. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 15.2 You will develop a shopping cart service, so that customers can encapsulate their purchase orders with parameterization (e.g., special requests) and you can queue and log the shopping carts for optimized scheduling in the system to ensure performance. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 15.3 When customers check out, your system will create a purchase order. To ensure performance, you may decide to create the new order from an existing order. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[3] 16. Your system obviously has an inventory sub-system, where you have to monitor the lifecycle of software from when they enter your inventory until they are successfully delivered to customers' hands. At any stage, your system needs to be able to answer customer queries such as whether the software has been ordered, has been paid, has been delivered, etc. Any design pattern you see appropriate here? Briefly justify your suggestion, and draw a class diagram to explain your design (maybe with short descriptions if you see appropriate).

[8] 17. Like many startups, your software may begin its journey with a monolithic architecture, built for a single offering in a single country. Having one codebase seemed "clean" at the time, and could solve your core business problems, which included connecting customers, billing, and payments. It was reasonable back then to have all of your business logic in one place. As you rapidly expand into more countries and introduced new products, this architectural style should be changed. As a software architect, which architectural style(s) will you recommend to? Briefly explain why you recommend such architectural styles, any advantages? Disadvantages? Draw a component diagram to explain your design (maybe with short descriptions if you see appropriate). You need to show the way in your implementation.

[14] 18. Your app is becoming very popular. In order to support rapidly increasing user base while quickly developing new functions and deploying it, you decide to deploy and run your app in cloud at a data center. In order to smooth and facilitate this process, you have decided to enforce DevOps in your project team.

[3] 18.1 Briefly explain your motivation of using DevOps.

[8] 18.2 You want to use container management technique. Briefly explain your reasons and briefly explain how you will do it using the latest container management technique. You will use docker to wrap up your

code.

[3] 18.3 In order to save budget, you plan to use multitenancy architectural style. Briefly explain your reasons and how you will make it work.

[8] 19. Use ATAM methodology to analyze the project requirements and deliver an ATAM document.