Q1 Notification System Design

12 Points

You work for a news outlet and your employer wants you to design a notification system with a front end that allows you to subscribe to different topic areas such as Sports, Weather, Entertainment, etc. Users should be able to pick whether they want to have notifications sent to them via the following methods: as soon as possible, at a certain time of day, or have notifications paused. This news system is special and allows the user to choose whether they receive the long version of the news or a shorter summary of the news. Your objective is to come up with at least four design patterns that you would use. Make sure to justify which design pattern(s) solves each part of the system design.

1. Observer pattern for user subscriptions. Observer pattern lets you define a subscription mechanism to notify multiple objects about any events that happen to the object they are observing. In this notification system, users can choose subscribe or unsubscribe the notification system. And if the users subscribe, they will get noticed about the news of some topic areas they choose to subscribe.
2. Strategy pattern for notification strategy. Strategy pattern lets you define a class of algorithms, put each of them into a separate class, and make their objects interchangeable. In the notification system, there are different strategies for users to receive notifications, including as soon as possible, at a certain time of day, and have notifications paused. Users can also choose between long version of the news and a shorter summary of the news.
3. Decorator pattern for different topics subscriptions. Decorator pattern lets you attach new behaviors to objects by placing these objects inside special wrapper objects that contain the behaviors. In the notification system, there are different topics that the users can pick from, including Sports, Weather, Entertainment, etc. They are the concrete decorators, and they are overriding methods of the base decorator, topics.
4. Facade for front end interfaces. Facade provides a simplified interface to a library, a framework, or any other complex set of classes. In the notification system, the client side uses facade to get access to some interfaces, like subscription topic choice, notification method selection. The client side does not need to call the complex system directly, and does not need to know the detailed implementation of the subsystems, like the algorithms used to implement different ways of notifications.

Q2 TikTok (or short video sharing app) System Design

12 Points

You have been asked to design a short video-sharing app (like TikTok or YouTube). General features might include (but are not limited to) users being able to log into the app, the ability for the system to create new accounts (users and creators) on the platform, users can follow content creators on it and interact with them, the system has different ways to recommend videos on the platform to users, etc.... Your task is to solve many of these features and tell us which design patterns do you think work really well towards building this app. Justify your answer. You can also come up with your own feature(s) and respective design pattern solution(s) which would help you implement those features. You must use at least 4 design patterns in your solution.

1. Observer pattern for the new message notification feature. Observer pattern lets you define a subscription mechanism to notify multiple objects about any events that happen to the object they are observing. The users subscribe to some channels will get notifications from the channels when there are any updates. Additionally, the users can choose to subscribe and unsubscribe.
2. Abstract factory pattern for the account creation feature. Abstract factory pattern lets you produce families of related objects without specifying their concrete classes. There are two kinds of accounts (the accounts in general are abstract products), users and creators (concrete products). They can be produced all with the AccountFactory which is the abstract factory interface. And there are two concrete factories, UsersAccountFactory, and CreatorsAccountFactory, which corresponds to users and creators.
3. Facade for the login feature. Facade provides a simplified interface to a library, a

framework, or any other complex set of classes. In the TikTok system, the client side interact with the interfaces, and it does not need to know the detailed implementations that is invisible to the client side. For the login feature, the client side does not need to know the implementation details of user authentication, but it can still use the login feature with the interface for user authentication.
4. Strategy pattern for the video recommendation feature. Strategy pattern lets you define a class of algorithms, put each of them into a separate class, and make their objects interchangeable. The system has different strategies to recommend videos on the platform to users, and the implementation of these recommendation algorithms are isolated from the code that uses it.

Q3 Favorite Color

1 Point

What's your favorite color?

My favorite color is pink!

Quiz 03

● GRADED

STUDENT
Ruijun Ni
View or edit group

TOTAL POINTS
25 / 25 pts

QUESTION 1
Notification System Design — **12** / 12 pts

QUESTION 2
TikTok (or short video sharing app) System Design — **12** / 12 pts

QUESTION 3
Favorite Color — **1** / 1 pt