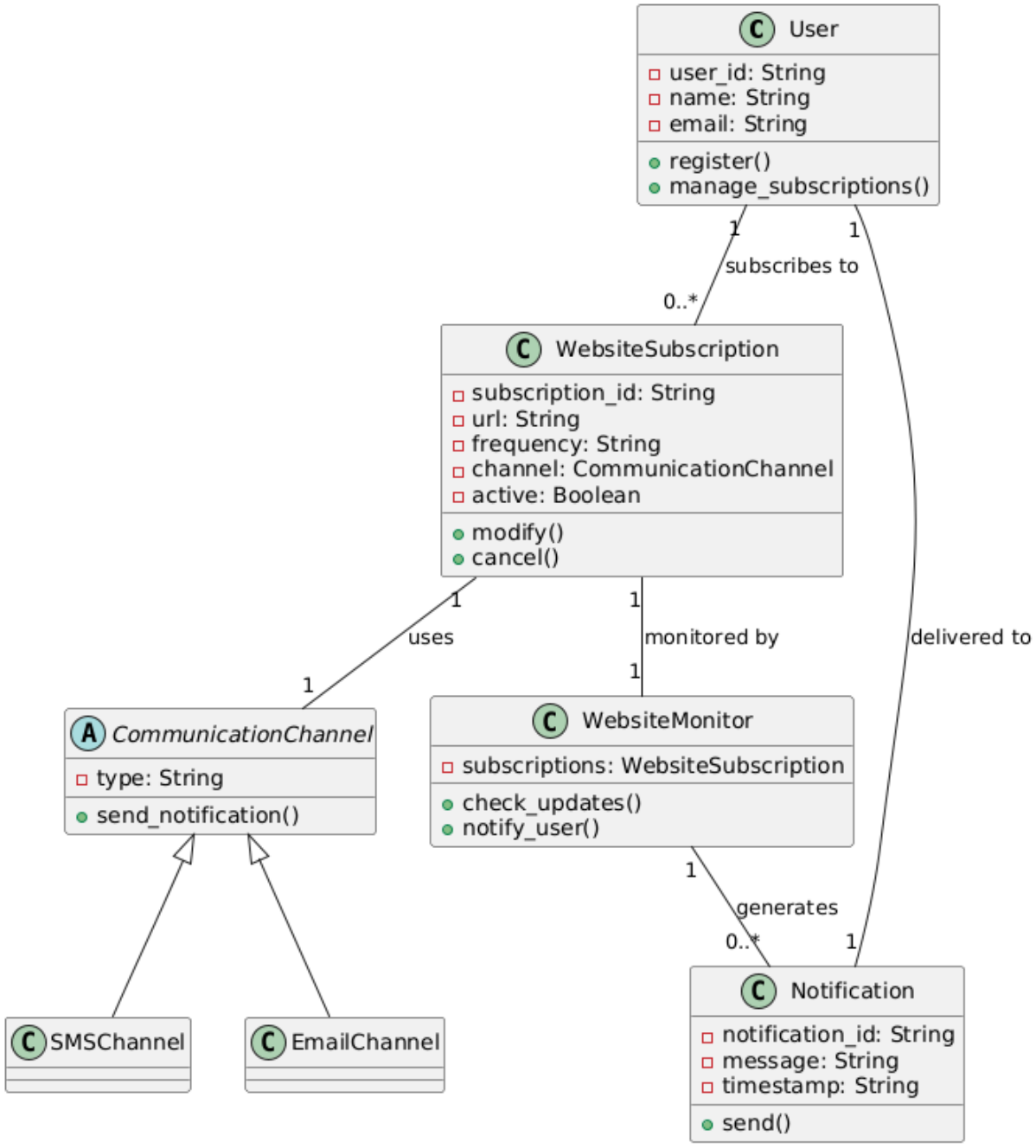

Q2: How to choose the right class for a method or object?

1. Choosing the Class for a Method

   GRASP Principle: Information Expert

   Give the job to the class that has the needed data.

   How to do it:

   Think about what the method does.

   Find which class has the needed information.

   Put the method in that class.

Example:

If you want to get the total price of an order, the Order class should have this method (because it knows the items and prices).

Other helpful GRASP rules:

Controller: For system actions (like a user clicking a button), use a controller class.

Low Coupling: Choose a class that doesn't depend too much on others.

High Cohesion: Put methods in classes where they feel like a natural part.

2. Choosing the Class to Create an Object

GRASP Principle: Creator

Let the class create another object if it uses, contains, or knows the data for it.

How to do it:

Choose the class that holds or uses the new object.

Or the one with the data needed to build it.

Example:

If an Order has OrderLine items, then Order should create OrderLine objects.

Summary Table

What You're Doing	GRASP Rule	How to Choose the Class
Writing a method	Information Expert	Give it to the class that has the needed information
Creating a new object	Creator	Give it to the class that holds, uses, or builds the data

Class: User

Responsibilities:
- Register for the system
- Manage website subscriptions (add, modify, cancel)
- Set notification preferences
- Receive notifications

Collaborators:
- WebsiteSubscription
- Notification

Class: WebsiteSubscription

Responsibilities:
- Store website URL, frequency, and notification channel
- Allow modification or cancellation
- Maintain active/inactive status

Collaborators:
- User
- CommunicationChannel
- WebsiteMonitor

Class: WebsiteMonitor (Controller)

Responsibilities:
- Periodically check all subscriptions for updates
- Detect changes on monitored websites
- Generate and dispatch notifications to users

Collaborators:
- WebsiteSubscription
- Notification
- User
- CommunicationChannel

Class: Notification

Responsibilities:
- Represent a message about a website update
- Store message content and timestamp
- Deliver itself to the user via the chosen channel

Collaborators:
- User
- CommunicationChannel

Class: CommunicationChannel (Expert)

Responsibilities:
- Define interface for sending notifications

Collaborators:
- WebsiteSubscription
- Notification

Class: EmailChannel

Responsibilities:
- Send notifications via email

Collaborators:
- Notification

Class: SMSChannel

Responsibilities:
- Send notifications via SMS

Collaborators:
- Notification

Did I change the responsibility of any method?

Answer:

No. The responsibilities of all methods are still the same as before.

Each method is in the right class based on its role and data. This follows the GRASP principles: Information Expert and Controller.

Examples:

- register() and manage_subscriptions() → stay in User
- modify() and cancel() → stay in WebsiteSubscription
- check_updates() and notify_user() → stay in WebsiteMonitor (Controller)
- send() → stays in Notification
- send_notification() → stays in CommunicationChannel and its subclasses