

Software Engineering

2016-17

Lab session 7

Advanced subscriptions and notification strategies

Objective

1. Implement the design based on Observer for subscriptions and notifications.
2. Implement the design based on Strategy for notifications.

Instructions

In this lab session the aim is to implement the design based on the Observer design pattern for managing subscriptions and notifications, and the design based on the Strategy design pattern for sending notifications according to the preference of clients.

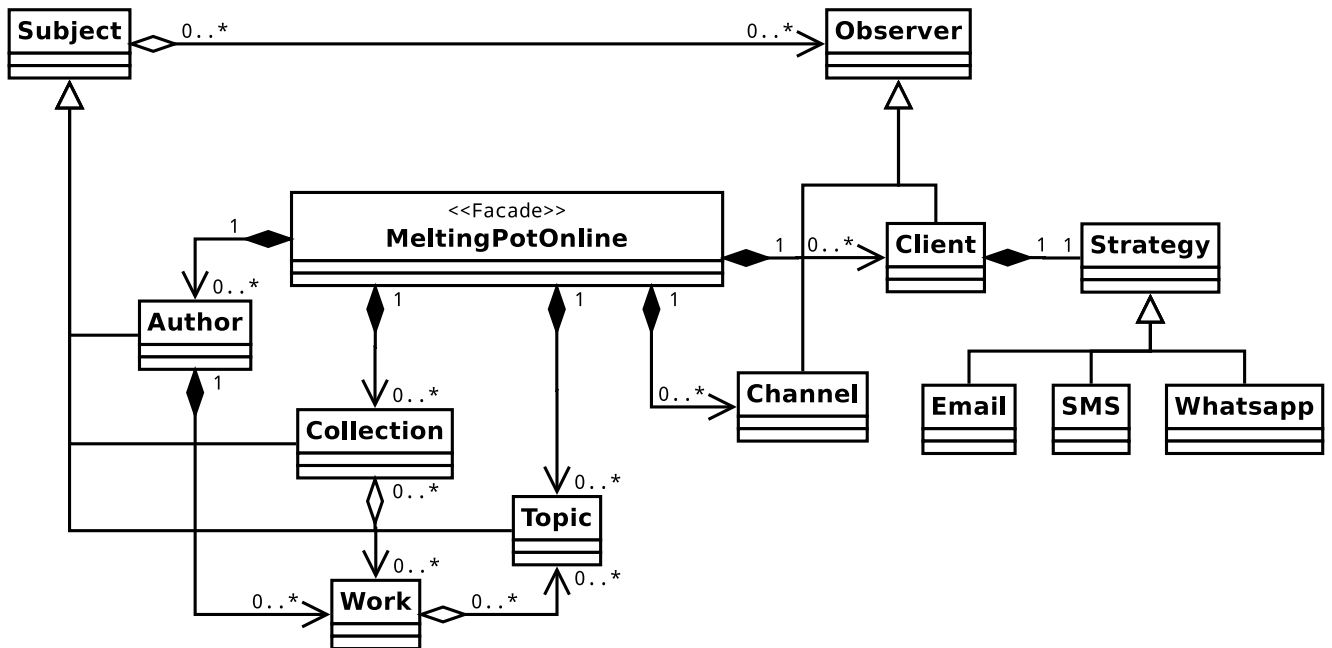
The editors have requested that apart from email and SMS, clients should also be able to receive notifications by Whatsapp. The API components in `MailStub.hxx` have been updated to include a component that simulates sending Whatsapp messages.

Tasks

Implement the necessary classes and methods to pass the functional tests in the two test sets `AdvancedSubscriptionTests.cxx` (that includes examples related to subscribing observers to subjects) and `NotificationTests.cxx` (that send notifications to clients according to their preferences). Remember to perform cycles RED-GREEN-REFACTOR.

For this lab session we will not provide a guide, so you have to deduce on your own which new classes and methods are necessary, and how the classes are related. As usual, when creating a new class you should also define new unit tests for this class.

To organize the classes you may use the following partial solution to seminars 7 and 8, extended to include notification by Whatsapp:



It is optional (but not required) to implement the design based on Factory Method for creating instances of classes representing notification strategies.

What to hand in?

The classes implemented in the `src` directory of your Git repository, and the unit tests in `src/unitTests`. You should also send an email to your lab session teacher in the format previously indicated.

When is the deadline?

Prior to the next lab session.