Software Maintenance & Evolution

Assignment

As part of the maintenance process for an e-learning platform such as Blackboard, you are requested to add a new feature that allows the system to notify users of critical information. The newly added feature should be implemented as a "Notification" module.

The "Notification" module should support different types of messages, typically: TaskAdded, GradesAnnouncement and Daily-News messages. There is a need for flexibility of the application. Therefore notifications as sent as either a mobile SMS, an email or both.

Each message has a *prepare message* function which prepares the message by replacing placeholders used in this Message. Place holders refer to functionality such as *Task Added* which has a task type, due date place holders. Another announcement type is *Grades Announcement* which has properties a *task name* and *student grade*. For example as message with format: " Dear {x}, The grades of task {y} is announced and you can find it at {z}."

The system should support two gateways, EmailGateway and SMSGateway, each of them accepts a message and sends it to user.

Three types of users are supported:
1- Professor (Name, Department, CurrentCourses as strings list, HirringDate, PhDTopic: string)
and has notifyMe function that should be called when any updates happens to course he is interested in.
2- TA (Name, Department) and have updateMe function that should be called when any updates happens to course he is teaching.
3- Student (Name, Id) and have updateMe function that should be called when any updates happens to course he is registered in.

Course class has (Name, Code) and AddAssignment, AddExam, PostGrades, PostAnnouncement functions
that Professors and TAs can use to organize communicate with students.

Project Implementation
You are expected to use the attached java project as your baseline for fulfilling the described maintenance requests. You should work in teams of no more than 4 and no less than 2 From the same lab. You are requested to do the following actions:
1- Create a **GitHub** repository and all team members should be contributors.
2- Run static code analysis tool and generate an analysis report (You may use **SonarQube** as we did in the lab)
3- Refactor the code and apply design patterns, suggested patterns are (Factory, Abstract Factory, Strategy, Observer)
4- Run analysis again and generate the new analysis report
5- All of your work should be pushed to **GitHub** as pull requests and reviewed by another team member.
6- Register some issues or comments from your latest analysis report to a free ticketing system or software such as (Bugzilla, Jira, Agiloft, …).

Deliverables:

1- Static Analysis Report for current code
2- Class Diagram for project after refactoring
3- Refactored Code
4- Static Analysis Report For refactored code
5- Opened Tickets for Issues

Notes:

- All of the required deliverables will be pushed to your **Github** repository, all team members **must** use **Github** to push their tasks.
- You should submit a text or pdf file with your names and github repo (don't push anything after deadline), the file should have the following format: **file name: Group_1stStudentId_2ndStudentId_3rdStudentId**