Bonus Deliverable

Behavioral Design Patterns

**Behavioral Design Patterns:**

Behavioral Design Patterns focus on the interactions and communication between objects. They help define how objects collaborate and distribute responsibility among them.

**Strategy Design Pattern:**

* It captures a family of algorithms, enabling the interchangeability of strategies without altering - the client code.
* Pattern Components: Context, Strategy Interface, Concrete Strategies.

**Observer Design Pattern:**

* It establishes a one-to-many dependency relationship where changes to a single object (Subject) notify all dependent objects (Observers).
* Pattern Components: Subject, Observer Interface, Concrete Subject, Concrete Observers.

**Use Cases Used in Each Pattern:**

* ***Strategy Design Pattern:***
  + Notify Learners of Scheduled Session
  + Learners need to receive notifications regarding upcoming sessions using different methods, including email, SMS, or push notifications.
* ***Observer Design Pattern***:
  + Notify Learners of Grades
  + Instructors update grades, and learners need to be notified immediately about the updates through multiple channels.

**UML Class Diagram**

**I. Strategy Design Pattern UML**

A screenshot of a computer

Description automatically generated

**Description**:

* *Context Class (Notification):*
  + It holds a reference to the NotificationStrategy interface and delegates strategy execution through the notify(sessionDetails: string) method.
* *Strategy Interface (NotificationStrategy):*
  + Defines the method notify(sessionDetails: string) that all concrete strategies implement.
* *Concrete Strategies (EmailNotification, SMSNotification, PushNotification):*
  + Implement the notify(sessionDetails: string) method with specific notification logic.
* *Client:*
  + Creates the context and assigns concrete strategies using main(String[] args).

**I. Observer Design Pattern UML**

A screenshot of a computer program

Description automatically generated

**Description:**

* *Subject Interface:* 
  + Defines methods to add, remove, and notify observers.
* *Concrete Subject GradeNotifier):* 
  + Maintaining a list of Observers and notifying them when grades are updated.
* *Observer Interface* 
  + Defines the method update(gradeDetails: String) for Observers.
* *Concrete Observers (EmailNotification, SMSNotification, PushNotification).*
  + Implements the update() method to handle grade notifications.

**Skeleton of Code:  
Observer Pattern:**

**A screenshot of a calendar

Description automatically generated**

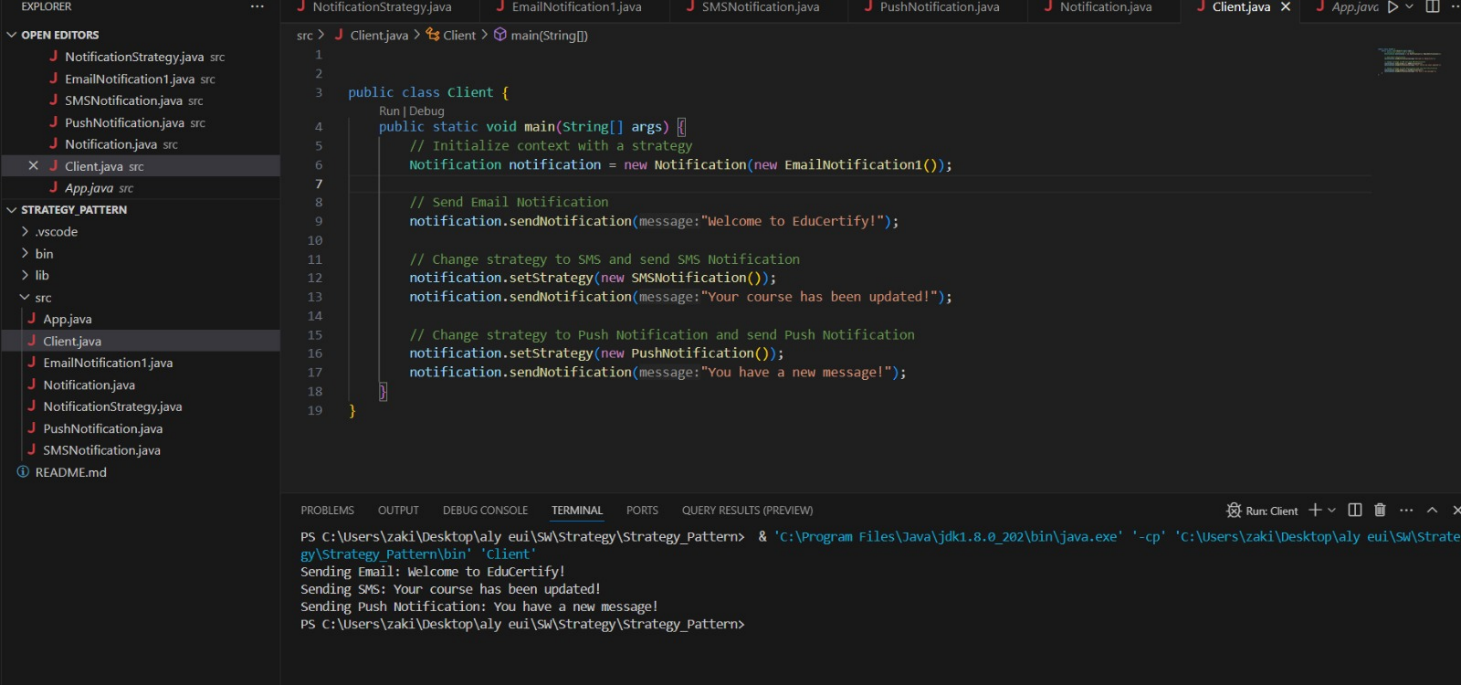
**Strategy Pattern:**

A screenshot of a computer

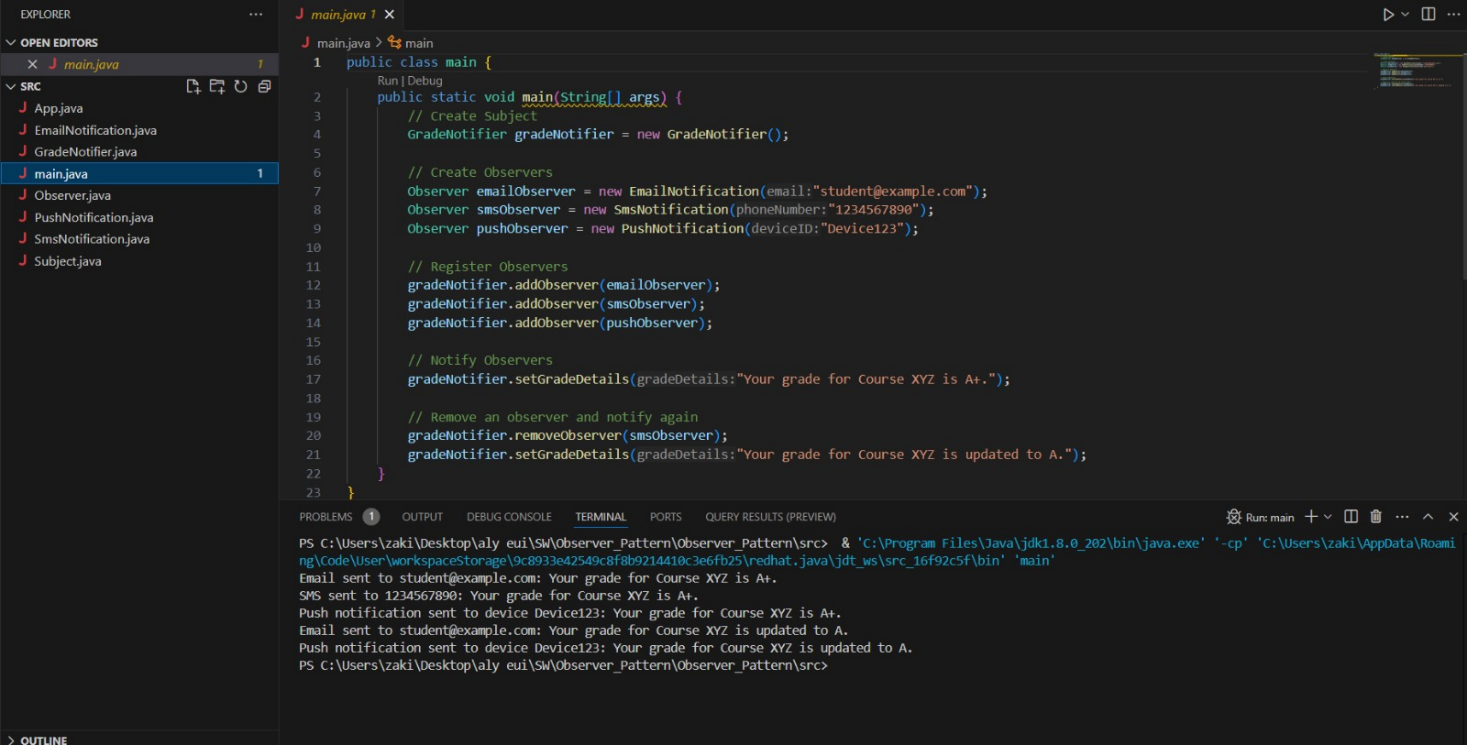
Description automatically generated

**Note: The actual code implementation is in the file not in this report.**

**Output of Strategy Pattern:**

****

**Output of Observer Pattern:**



By: Educertify Team

* Nada Ashraf 22-101043
* Aly Zaki 22-101096
* Ahmed Waleed 22-101258
* Omar Bayoumi 22-101022