Chapter Four: Implementation

4.1 Implementation

Export Class Diagram into Code and Update Code and Diagram

The implementation phase translates the system design, particularly the class diagram, into executable code.

In this project, we used the UML class diagram created during the design phase to guide the development of

our backend classes and database structure.

Exporting Class Diagram to Code:

Using tools like Visual Paradigm, or Enterprise Architect, the class diagram can be exported to skeletal

code in java. These tools generate class definitions, methods, and attributes as per the design.

Updating the Code and Diagram:

During development, several enhancements and changes may occur. These changes should be reflected

both in the codebase and the class diagram to maintain consistency. This ensures both design

documentation and implementation stay synchronized.

**4.2 Steps to Generate Code from Class Diagram**

Step 1: Create the Class Diagram

- Use a UML tool ( Visual Paradigm)

- Define all classes, attributes, methods, and relationship

Step 2: Validate the Diagram

- Ensure there are no missing relationships or misnamed attributes

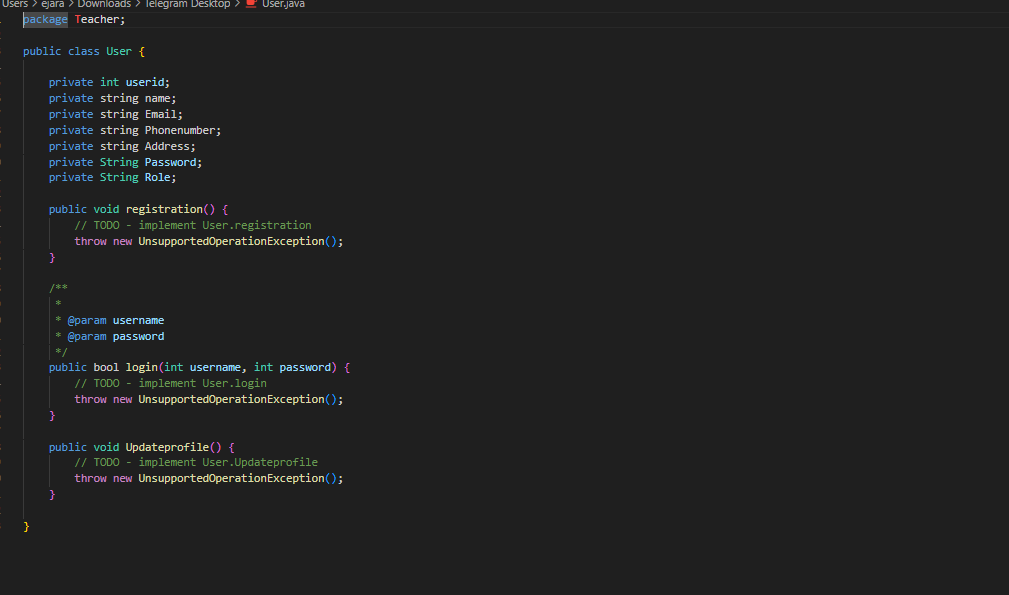
- Check inheritance and association types

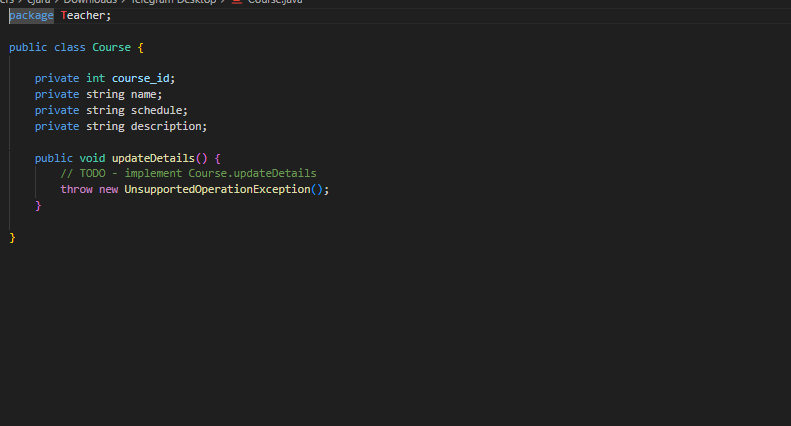
Step 3: Use the Code Generation Feature

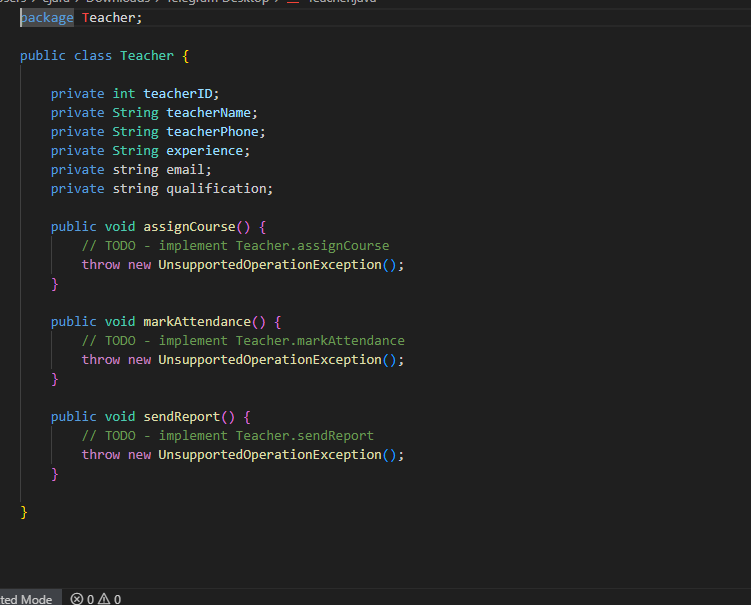
- Most UML tools have a 'Code > Generate Code' feature

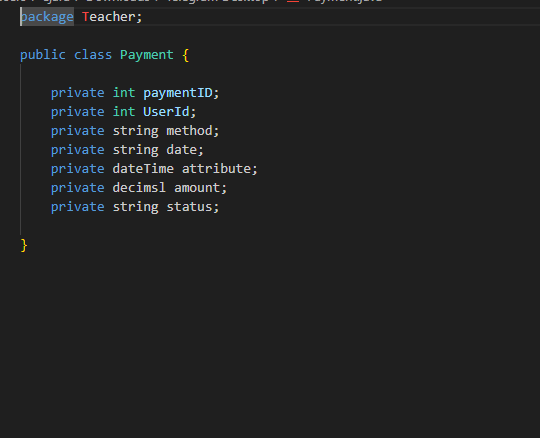
- Choose the language (e.g., java)

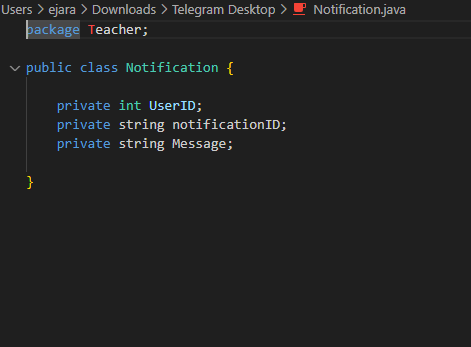
- Select destination folder and generate the files

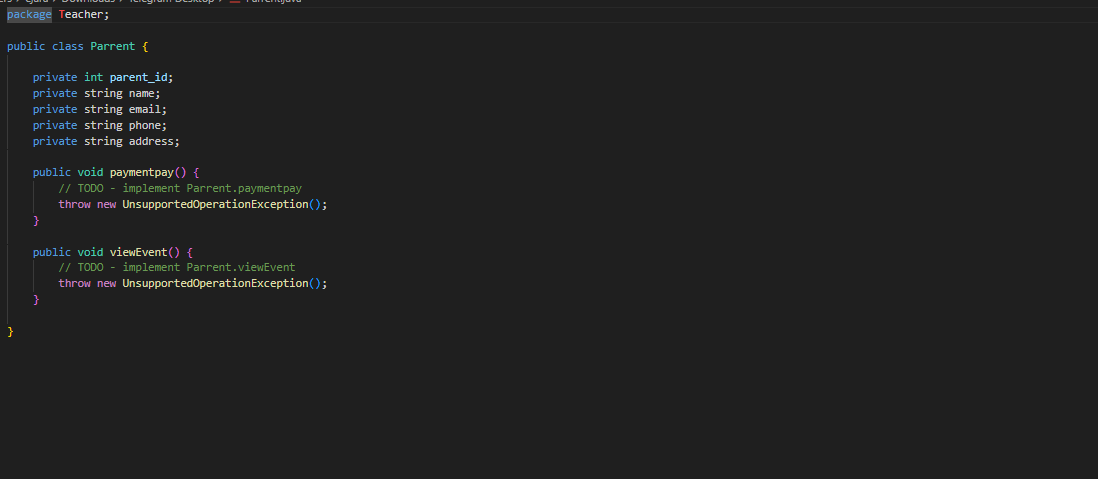


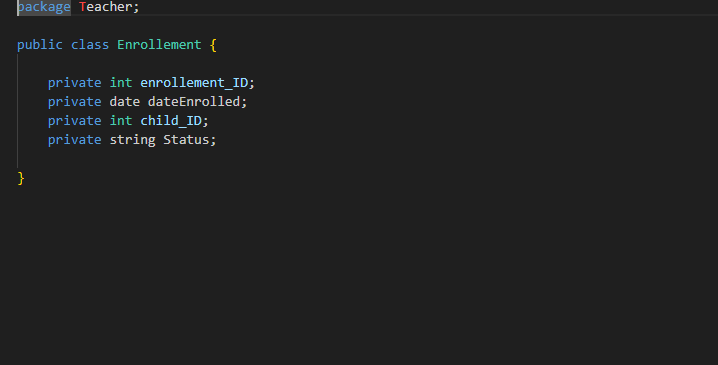


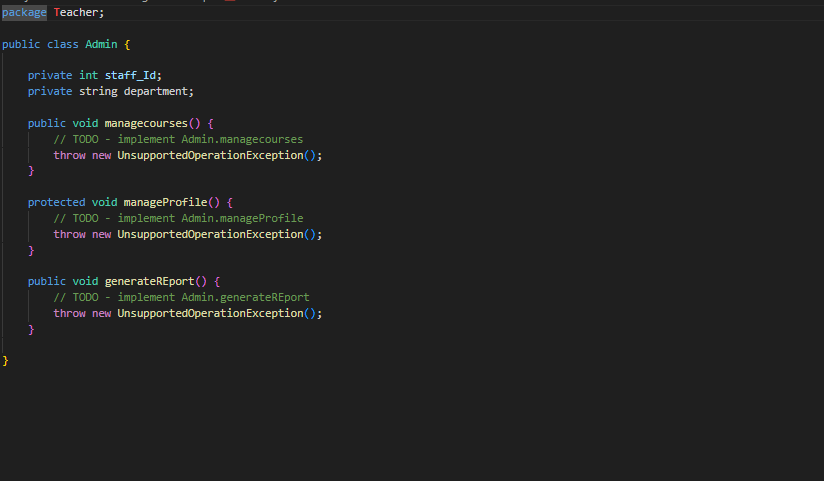


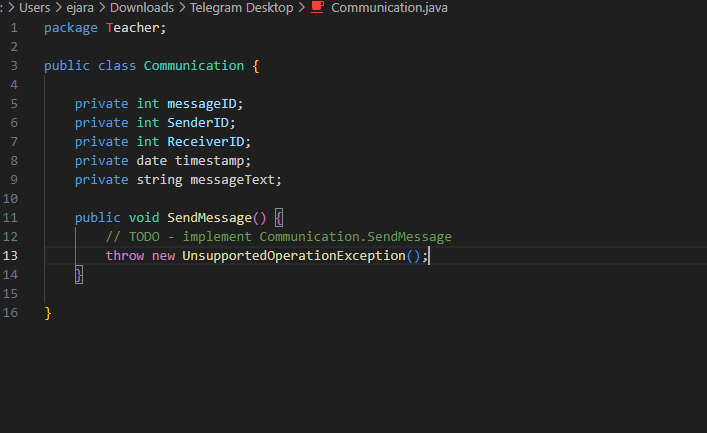


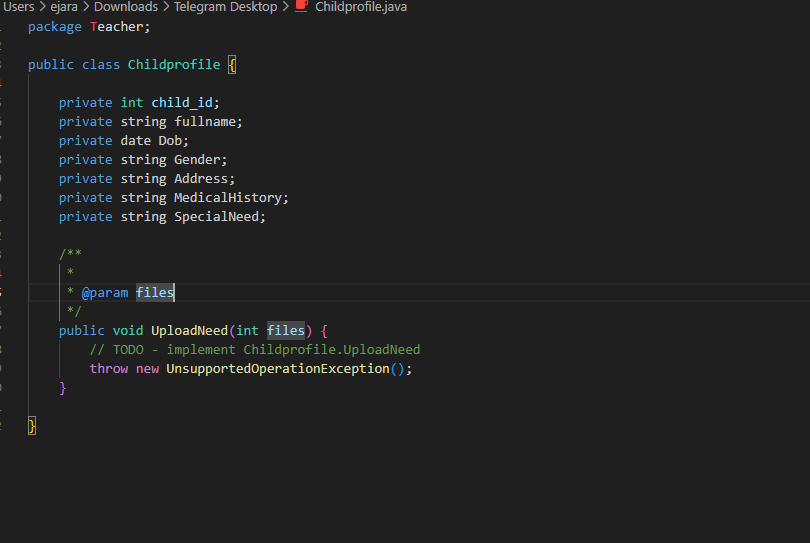












Step 4: Review and Customize the Generated Code

- Add logic to methods

- Integrate with database

step 5: Keep Class Diagram Updated

- As you add/remove methods or attributes in code, reflect those changes in the diagram

- Helps in future maintenance and understanding