COLUMBUS STATE

UNIVERSITY

Object Oriented Development

Term Project

# E-Learning Platform

Asst Professor: Dr. Md. Amjad Hossain

Dheer Madhav Balusu

Venkat Ramana Reddy Kuppi Reddy

Shashank Kammanahalli Chandra Sekhara

Date: Dec 2nd,2024

2. **Final State of System Statement 1. A paragraph on the final state of your system: what features were implemented, what features were not and why, what changed from Project 5 and 6**

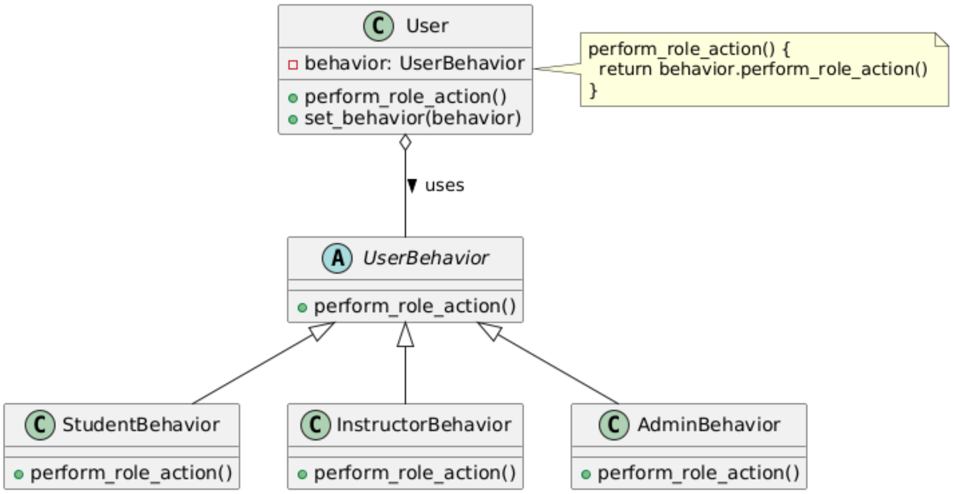
In the final state of our system, we implemented the core features of the proposed E-Learning Platform: user registration and authentication, role-based access of students, instructors, and administrators, course management-creation, enrollment, and tracking, access to course resources, progress tracking, and generation of a certificate upon completion. The system provides a user-friendly interface that allows students to search for courses, enroll in them, view assignments, and receive notifications for deadlines and new resources. An instructor can stipulate and manage classes, add material, and track student progress, while an administrator ensures smooth operations on the platform.

Some features, advanced notification customization and real-time chat functionality between students and instructors, were foreseen but had not been implemented due to time constraints and prioritizing of the most critical functionality. These features will remain as future enhancements for further enrichment of the platform.

Key changes between Project 5 and 6 were the further fine-tuning of role-based access control to ensure the system would work effortlessly for all the user types, optimization for performance in the course management system, and implementing feedback from the interim reviews. The platform has acquitted itself well and achieved its goals by providing a robust yet friendly environment for online education.

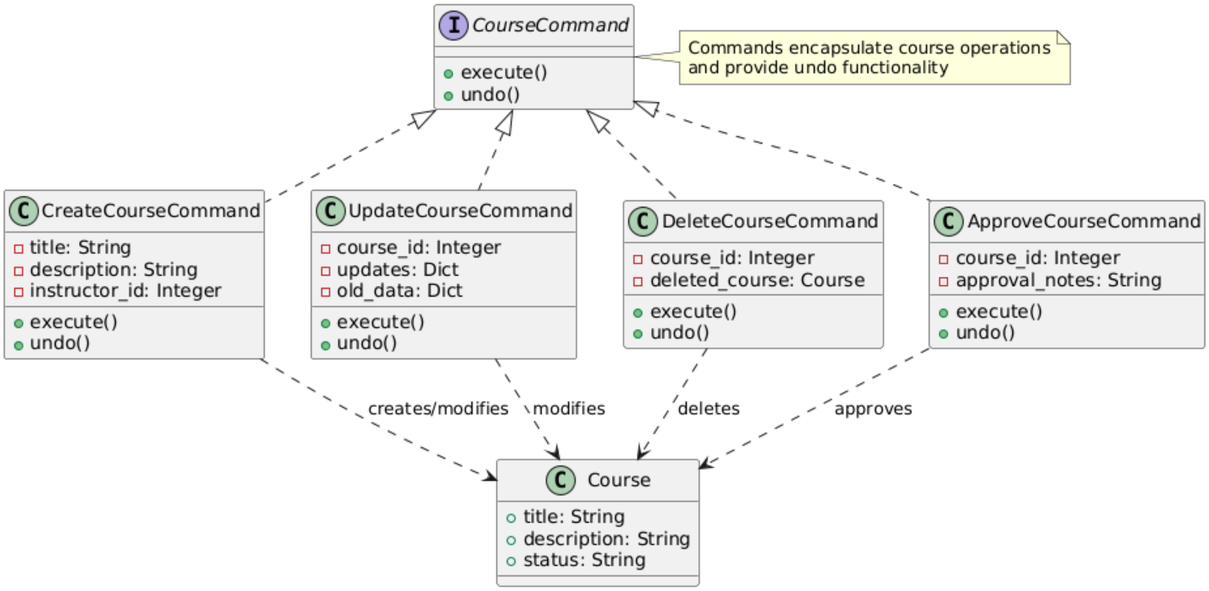
3. **Final Class Diagram and Comparison Statement 1. A thorough UML class diagram representing your final set of classes and key relationships of the system 2. Highlight and document in that diagram any patterns that were included (in whole or part) in your design 3. Include the class diagram submitted in Project 5, and use it to show what changed in your system from that point into the final submission 4. Support the diagrams with a written paragraph identifying key changes in your system since your design/work was submitted in Projects 5 and 6**

**Strategy Pattern - User Behavior**



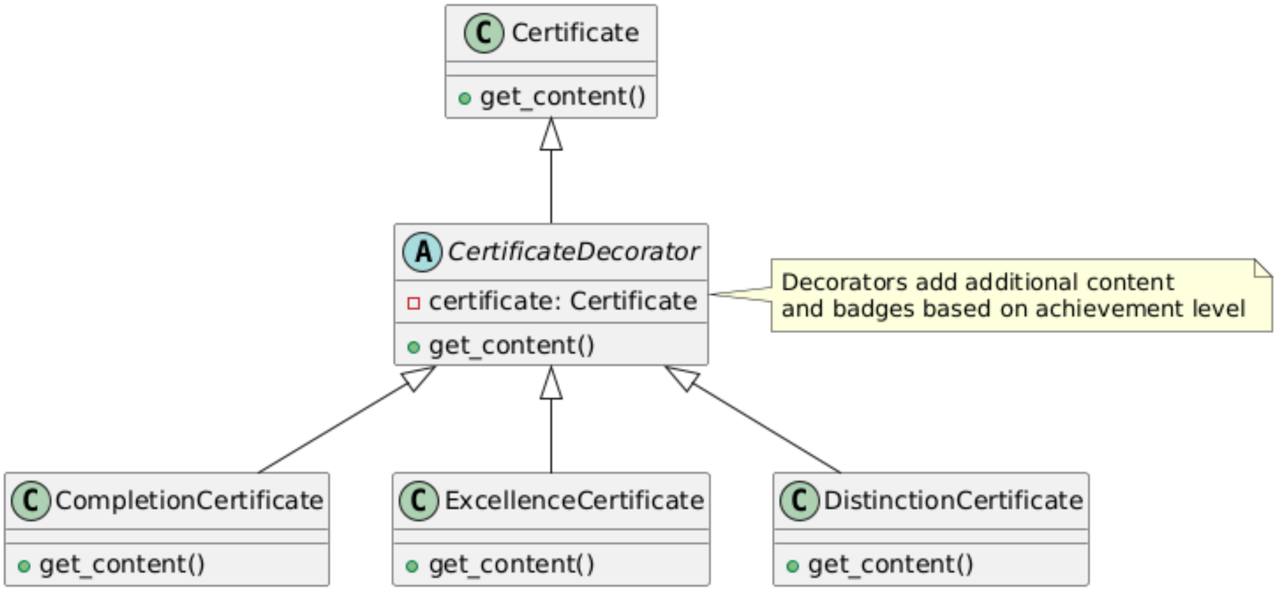
perform_role_action() {
 return behavior.perform_role_action()
}

**Command Pattern - Course Operations**



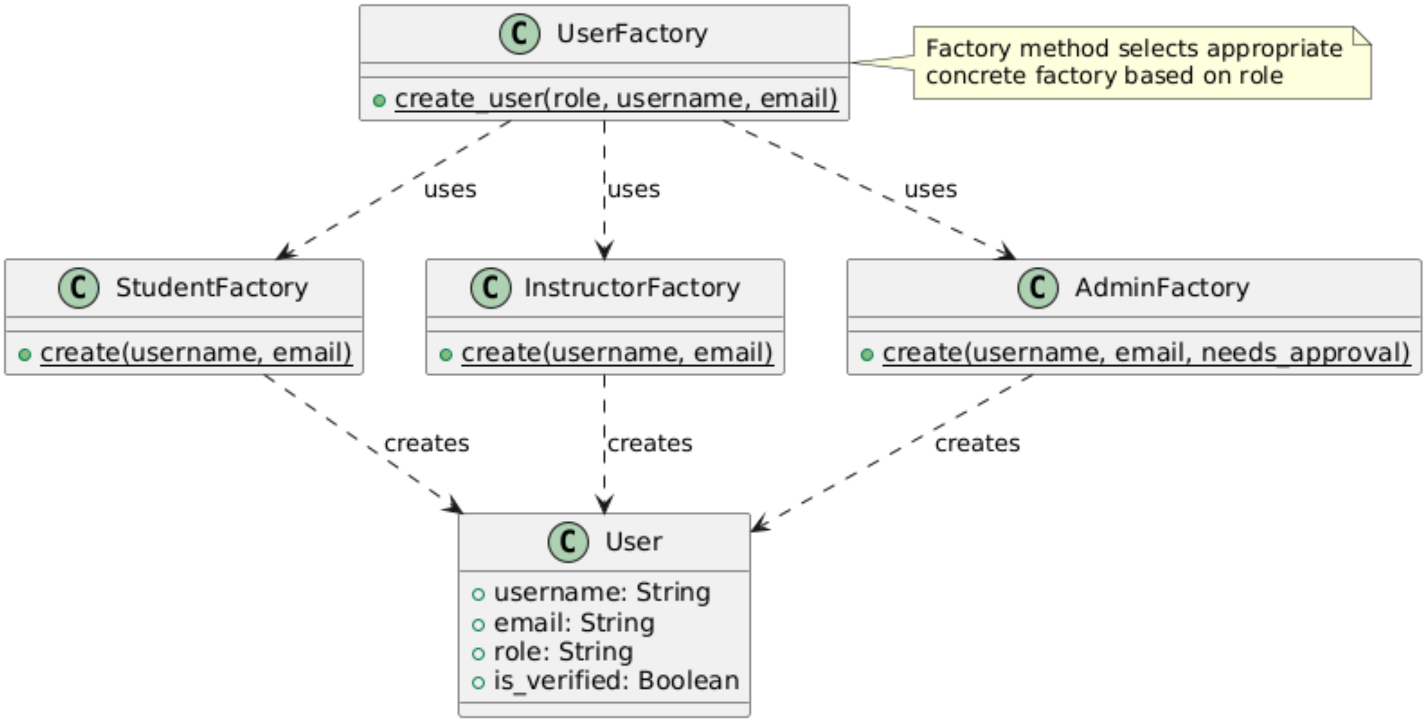
Commands encapsulate course operations
and provide undo functionality

**Decorator Pattern - Certificate Types**



Decorators add additional content
and badges based on achievement level

**Factory Pattern - User Creation**



Factory method selects appropriate
concrete factory based on role

4. **Third-Party code vs. original code Statement 1. A clear statement of what code in the project is original vs. what code you used from other sources – whether tools, frameworks, tutorials, or examples – this section must be present even if you used NO third-party code - include the sources (URLs) for your third- party elements.**

**Third-Party Code/Frameworks**

Flask Framework and Extensions (https://flask.palletsprojects.com/en/stable/extensions/)

SQLAlchemy ORM (https://docs.sqlalchemy.org/orm/)

Werkzeug Security(https://werkzeug.palletsprojects.com/en/stable/utils/)

Faker Library (for generating test data)( https://fakerjs.dev/)

All code not explicitly listed under Third-Party Code is original work, developed specifically for this e-learning platform. The implementation follows best practices and patterns from software engineering but with original implementations tailored to the application's needs.

**5. Statement on the OOAD process for your overall Semester Project 1. List three key design process elements or issues (positive or negative) that your team experienced in your analysis and design of the OO semester project – three things that went well or not so well as you worked through your deliverables.**

**1. Pattern Implementation and Integration (+)**

Positive Element:

The implementation of multiple design patterns (Strategy, Factory, Decorator, and Command) was well-executed and provided clear benefits:

**Benefits:**

- Clear separation of concerns
- Easily extensible for new user roles
- Improved maintainability
- Reduced code duplication

**Process Success:**

- Started with UML diagrams to plan pattern integration
- Implemented patterns incrementally
- Tested each pattern implementation separately
- Successfully integrated patterns without conflicts

**2. Database Schema Evolution (-)**

**Challenges Faced:**

- Frequent schema changes as requirements evolved
- Complex relationships between models
- Migration management in development
- Data integrity maintenance

**Learning Points:**

- Need for better initial database planning
- Importance of considering future requirements
- Value of proper migration testing
- Need for better documentation of schema changes

**3. API Design and Documentation (+/-)**

**Successes:**

- Clear endpoint structure
- Role-based access control

**Challenges:**

- Initial lack of API documentation
- Authentication complexity