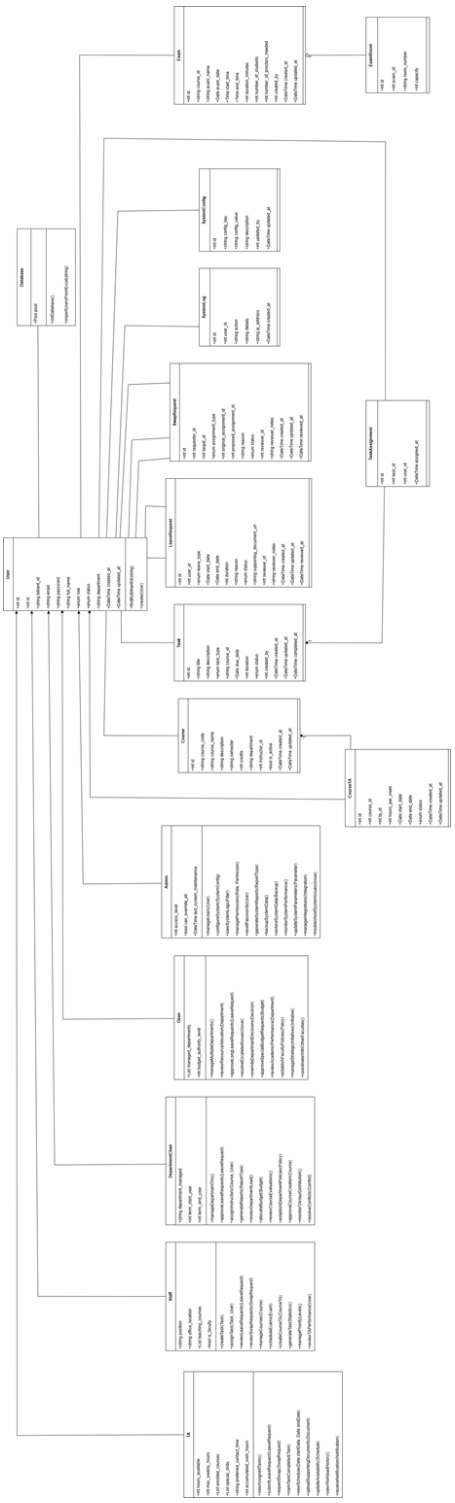
Class Diagram

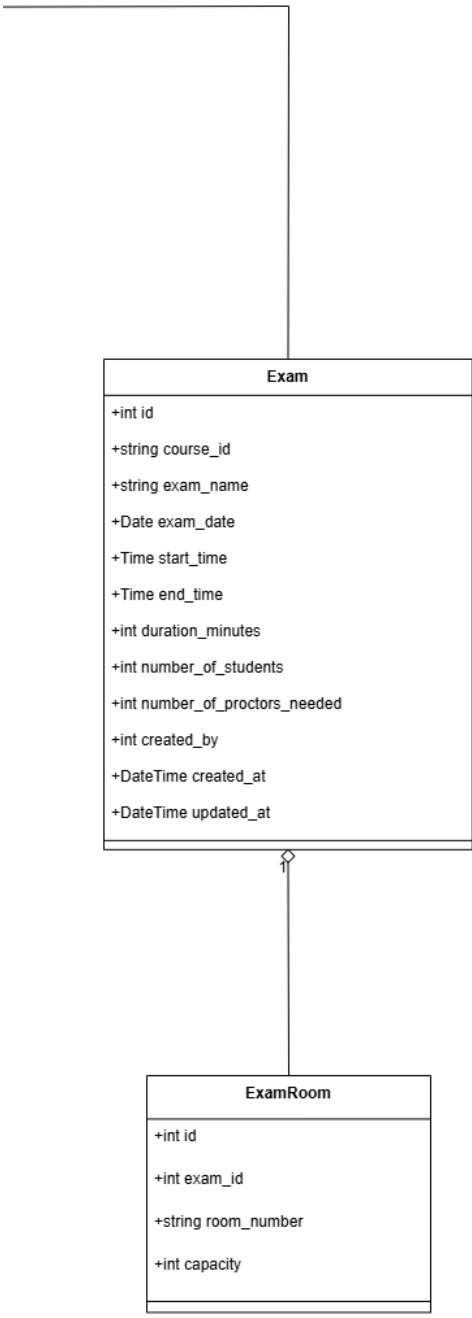


Software Design Patterns

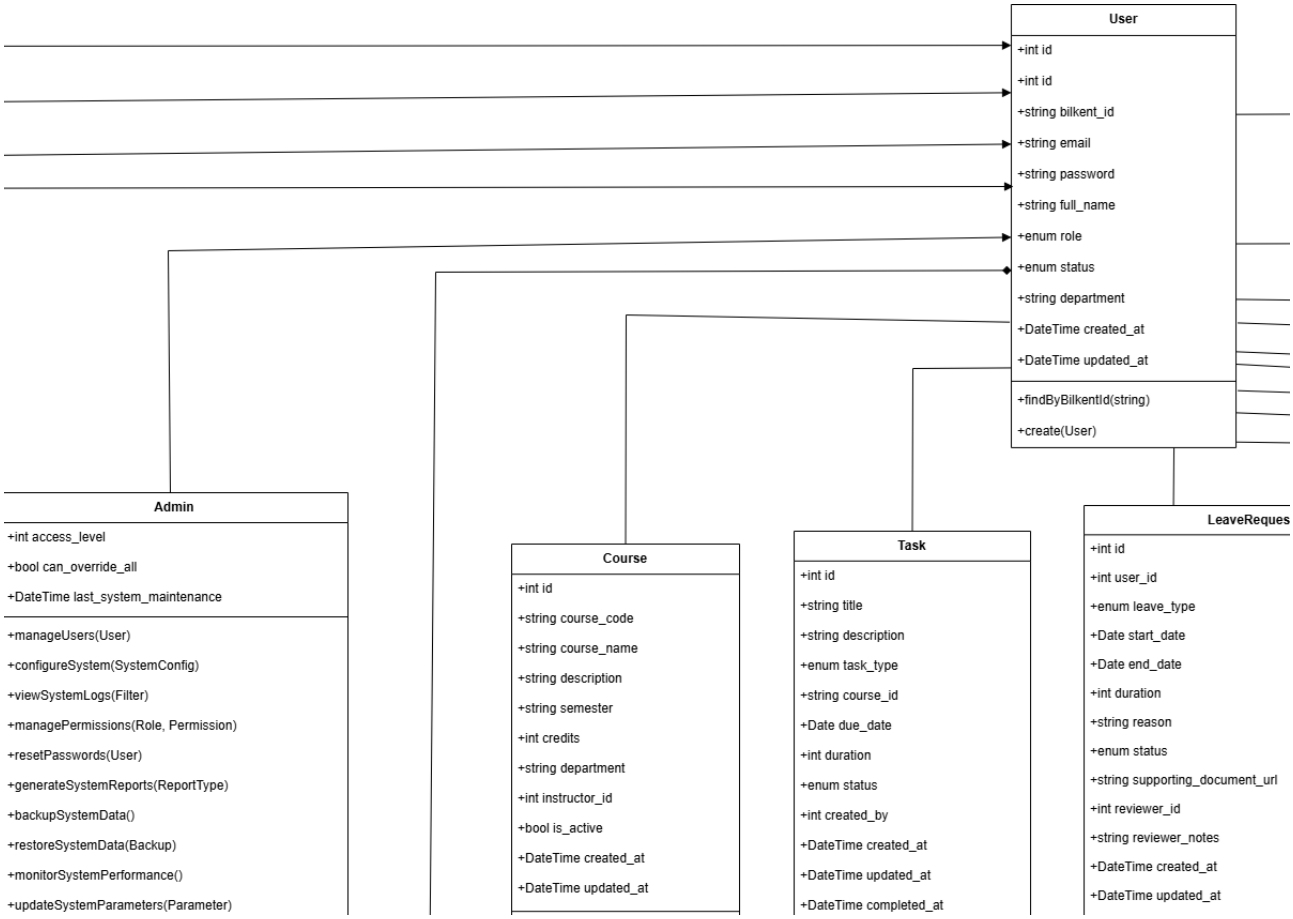
1. Aggregation:

The aggregation design pattern is represented in the class diagram by the following classes:
Exam: This is the class that holds all Exam information, including cours_id, dtae and time, and duration. It includes the ExamRoom class which can exist without an exam, since its just a room.

ExamRoom: This is a concrete class that represents a ExmRoom object and holds all information about the room an exam will be held in including room_number, and capacity. The classroom exists regardless of whether an exam will be taking place in it or not, which means it does not rely on the existence of an Exam class. It is a leaf node in the composite hierarchy with no child objects.



2. Generalization (Inheritance)



Note: This example only shows the relationship between admin and User since the diagram is too large to show other user relationships

User: This is the abstract superclass that generalizes common attributes and behaviors shared by all user types in the system. It holds general information such as ID, email address, password, and role. It also provides shared operations like setPassword and setEmail that are inherited by TA, staff, and admin subclasses. The Account class serves as the root of the generalization hierarchy.

Admin: This is a subclass of Account, representing administrative users with elevated privileges. Admins can perform actions such as viewing logs, accessing user statistics, and adding or overriding accounts. This class inherits the basic user structure from Account but extends it with admin specific functionality.