Bilkent University

Department of Computer Engineering

CS 319 - Object-Oriented Software Engineering
Fall 2023

Deliverable 5

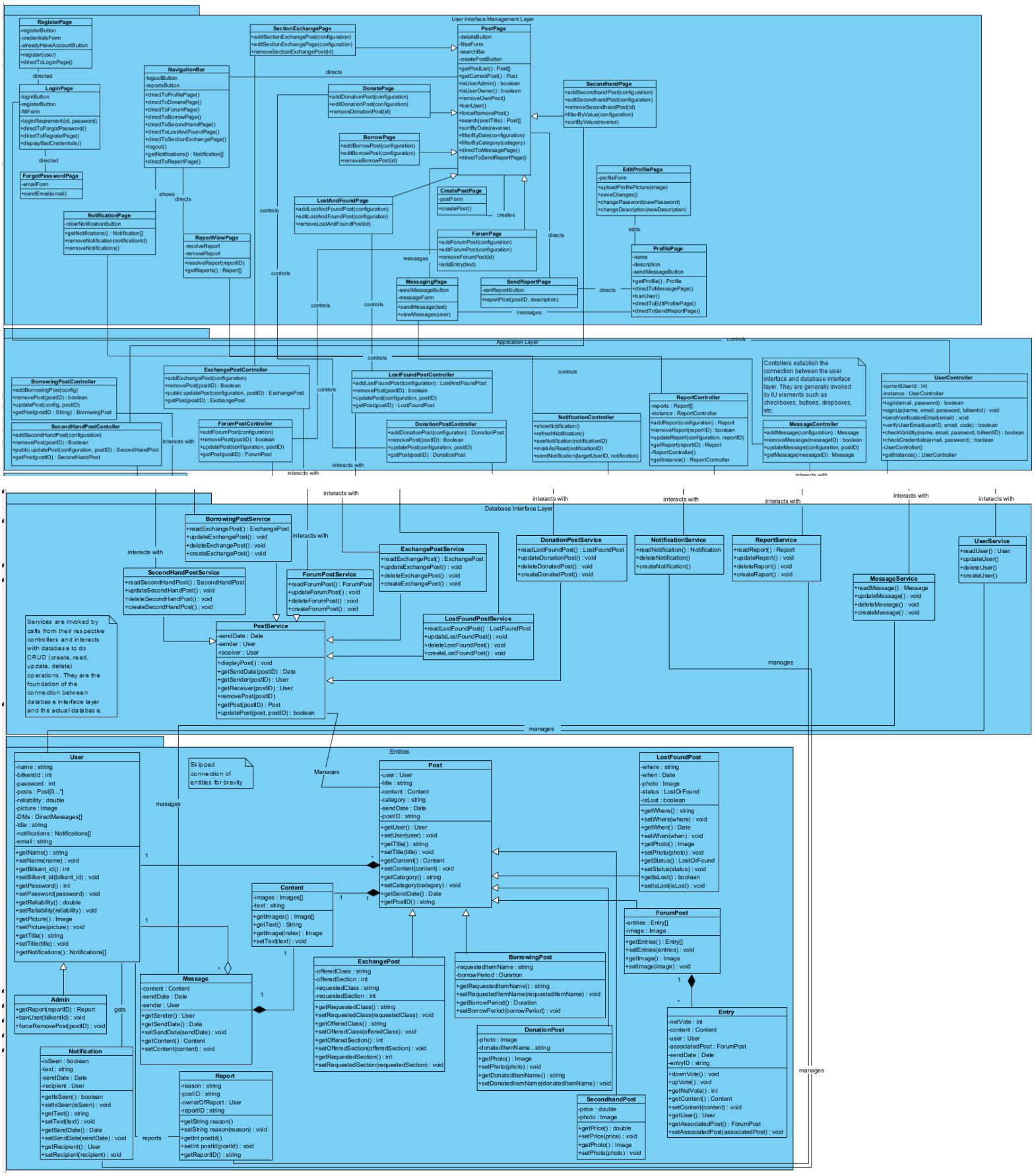
Class Diagram

Team 5

Cahit Ediz Civan (22003206)
Efe Kaan Fidancı (22102589)
Emir Tuğlu (22003165)
Görkem Kadir Solun (22003214)
Mete Enes Yılmaz (22102849)

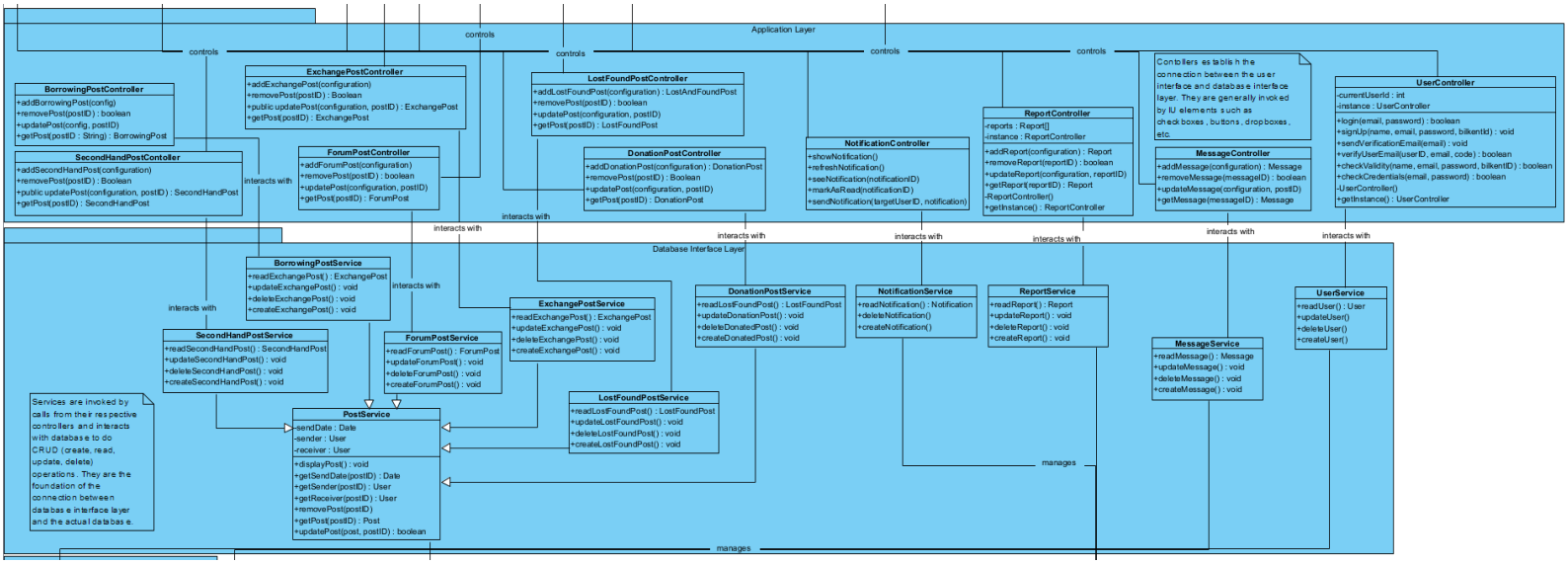
Table of Contents

Solution Level Class Diagram



Design Patterns

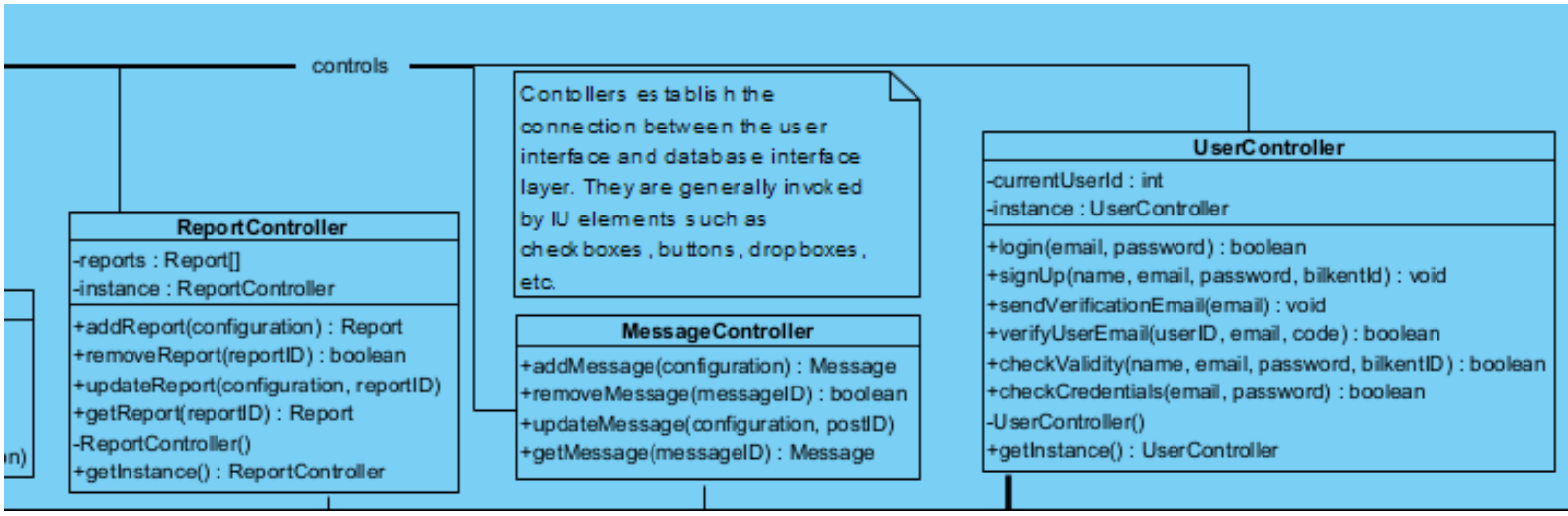
Facade Pattern



In essence, the Facade pattern acts as a "facade" or a front-facing interface that shields the user from the intricate workings of the system behind it. It allows users to interact with the system through a simpler interface without needing to understand its internal intricacies, thus promoting easier usage and maintenance of the codebase.

We apply the Facade design in two distinct places. One is located in the application layer, and the other is in the database interface layer. The database interface layer consists of several classes known as services, such as SecondHandPostServices, ForumPostServices, etc. Services decouple each endpoint's implementation of the requests. In this way, in the event that the implementation changes, controllers such as SecondHandPostController, ForumPostController, etc., will still function with, at best, minor adjustments. The communication with the database is managed via services' classes. As a result, controller classes don't use any database-related logic.

Singleton Pattern



Singleton pattern plays a crucial role in ensuring that a class has only one instance and provides a global point of access to it. The Singleton pattern is particularly beneficial when there is a need for a single, shared instance of a class that controls actions, coordinates activities or manages resources. By enforcing a single instance, the Singleton pattern helps maintain consistency and prevents unnecessary duplication of resources. In our implementation of the Singleton pattern, we have encapsulated the instantiation process within the class itself through a private constructor, ensuring that clients cannot create additional instances of the respective class. The two classes that utilize the creational design pattern of Singleton are ReportController and UserController. In ReportController, we use it in reporting the posts to prevent all users from accessing the report content. UserController, we will use it to evaluate the user activities. Only a single instance of them is necessary in our system. Thus, we implemented them in such a manner.