



CS-319

Deliverable 5

Team 06

Group Members

Muhammad Rowaha - 22101023

Maher Athar Ilyas - 22001298

Mehshid Atiq - 22101335

Ghulam Ahmed - 22101001

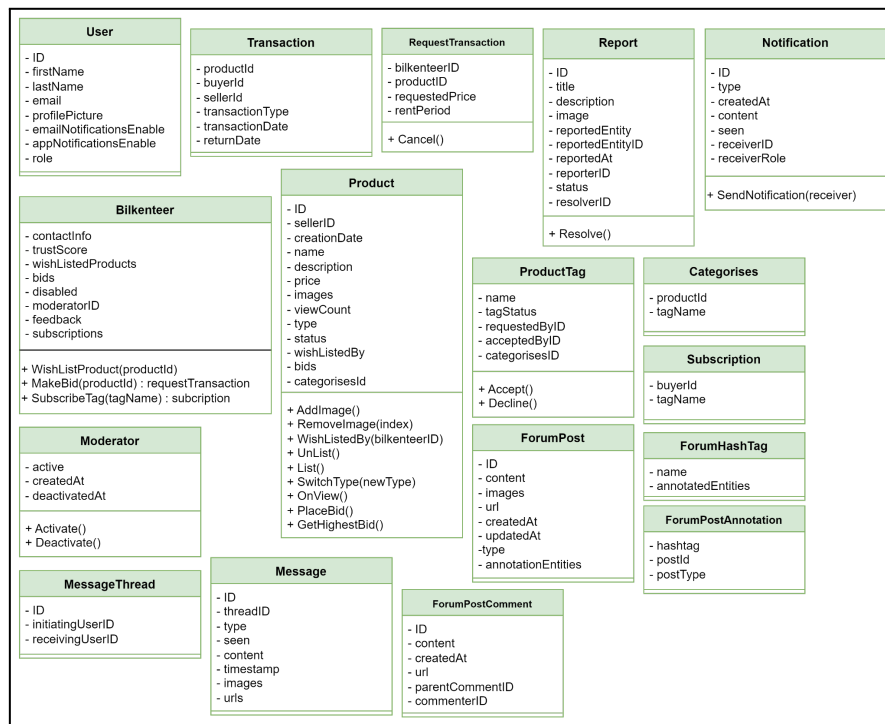
Ismail Özgenç - 22001648

Date: 03/12/2023

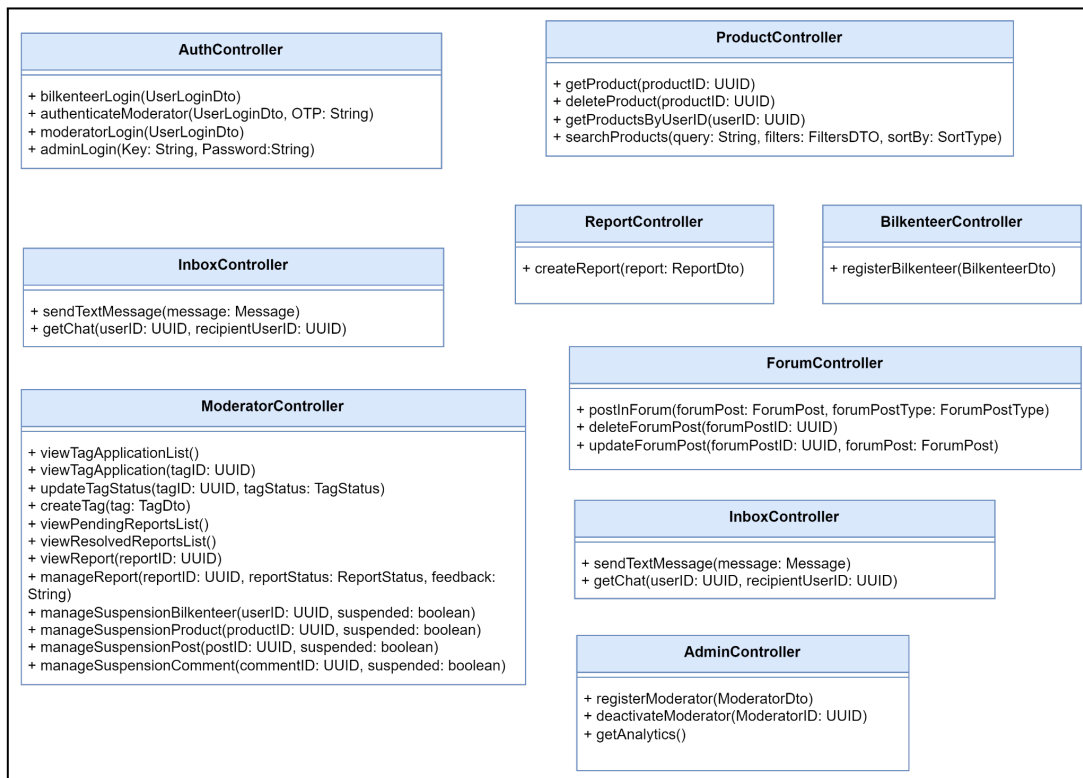
1. Class Diagram

1.1 Class Definitions

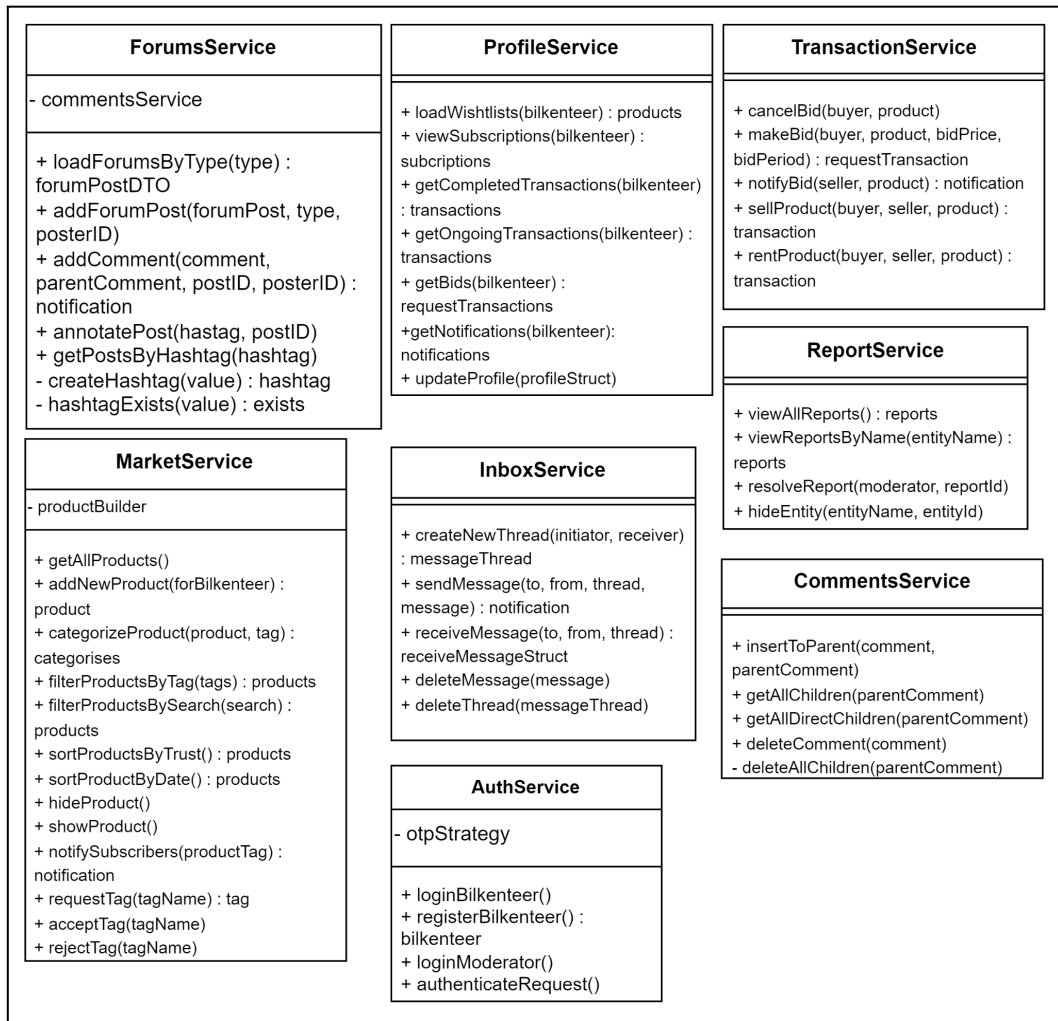
Will be replaced with entities. E.g. UserEntity, ReportEntity, etc



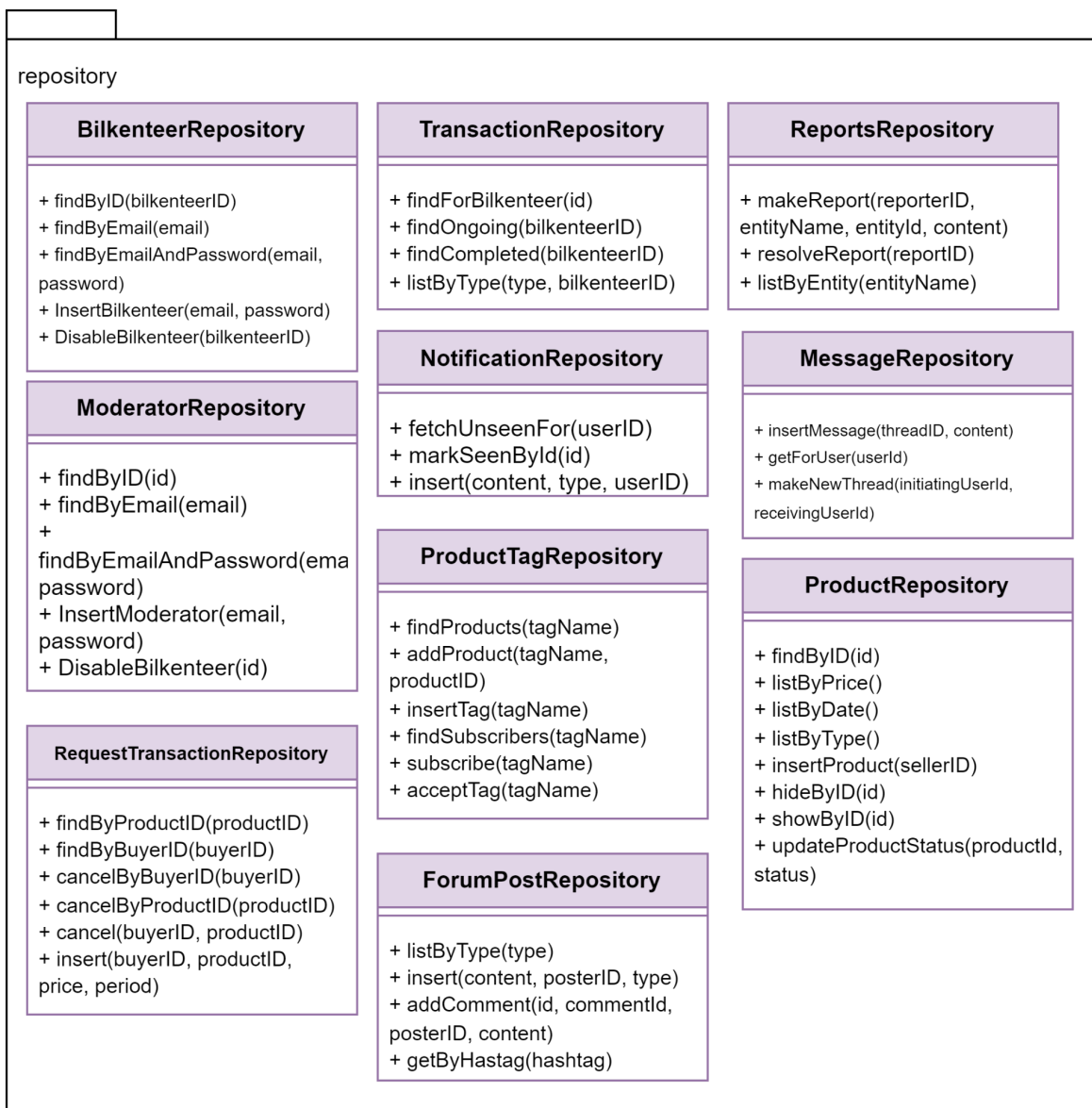
1.2 Controller Definitions



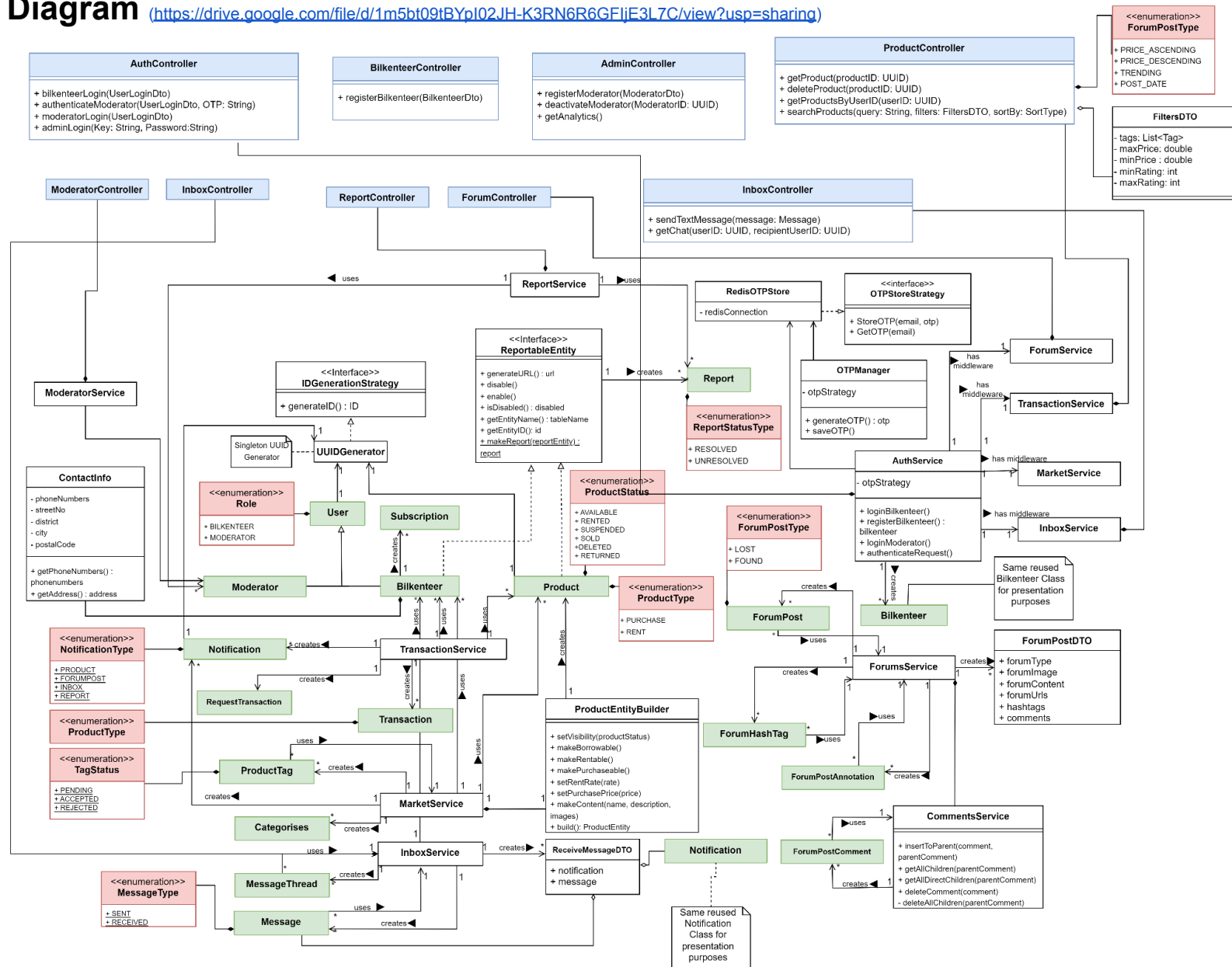
1.3 Service Manager Definitions



1.4 Repositories Definitions



<https://drive.google.com/file/d/1m5bt09tBYpl02JH-K3RN6R6GFliE3L7C/view?usp=sharing>



2. Design Patterns

2.1 Builder Pattern

The Product class internally can have different states, such as purchasable, rentable, or borrowable for the transaction type, hidden or listed for the visibility on the marketplace, sold or rented for the product's status, and disabled or enabled depending on the violation of the community guidelines. The ProductEntityBuilder class exposes a simple interface to build a product of any of the types and states above. Furthermore, the builder pattern is commonly used as an alternative to transmitting the whole entity across services. Instead, only the DTOs are passed as parameters for all communication between controllers and services.

2.2 Facade Pattern

The classes in the repository layer provide an abstraction over the raw SQL queries that will be performed on one or more related entity classes the repository may be dependent on. Methods like findTransaction(buyerID, productID), for example, will use the BilkenteerEntity and ProductEntity to perform SQL queries and joins before returning the Object. The implementation of all service classes is done to provide intermediate APIs to other services so that the complexity of developing individual services is reduced.

2.3 Strategy Pattern

Examples of strategy design pattern include the IDGeneratorStrategy and the OTPStoreStrategy. Concrete implementations are UUIDGenerator and RedisOTPStore, respectively. However, different strategies can be employed, such as AutoIncrementingIDGenerator and HashMapOTPStore strategies.