Software Architecture Documentation

"Picturesque" System

SOEN 6441

ADVANCED PROGRAMMING PRACTICES - FALL 2022

PROJECT

Github Repo: https://github.com/mikasa511/APP-Project

Team Members:

Raveena Choudhary (40232370)

Umangkumar Maheshbhai Patel (40228475)

Faculty Coach:

Dr. Constantinos Constantinides, P.Eng.

Table of Contents

Introduction

In the era of digitalization, everyone is luring customers by providing them with various discounts to register on their websites, whereas customers always want something they can easily find online and free of cost. “**Picturesque System**” intends to fulfill customers' requirements without paying unnecessary hefty amounts to download pictures to use in their blogs, websites, and possibly where they want to use those pictures.

This system is a platform to download free photos online in any size i.e., medium, large, and original as per the end-user requirements. Furthermore, end users can use the downloaded photos on their websites, blogs, as wallpaper, etc.

1.1 Purpose

The purpose of this document is to provide a detailed architectural overview of the new Picturesque System, using several different architectural views to depict different aspects of the system.

1.2 Scope

This Software Architecture Document incorporates many views of the system, and the technologies used to build the system.

Architectural Representation

This section details the architecture design used to build the system.

The system is divided into layers: The database layer, the Application layer and the Presentation layer(also known as the UI layer). Each of these divisions are vital for the system to operate.

2.1 Architecture Diagram

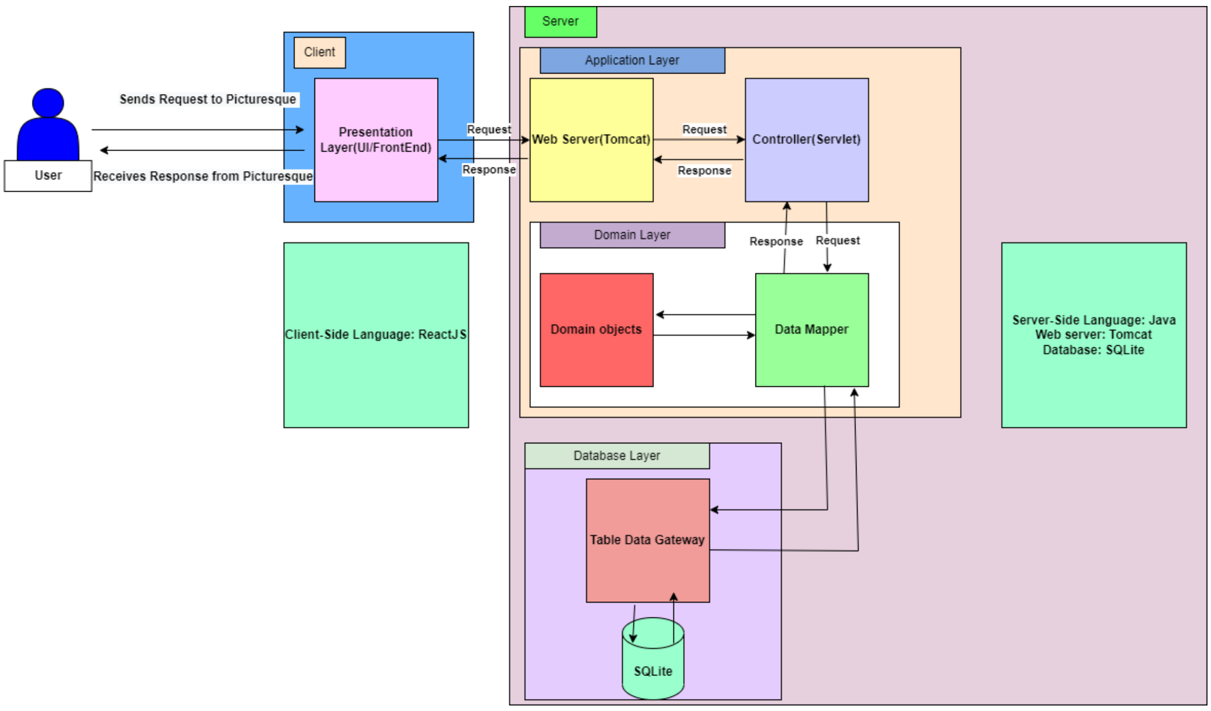
Request handling at client-side:

Here we are using a client-server architecture where the user will send a request (for example: likes a photo) to the picturesque system, that request will be sent to the tomcat server(webserver) which calls the servlets to process the incoming request and receives a response and sends that response to the client.

Request processing at server-side:

As mentioned above, tomcat will call servlet, then servlet will verify request parameters if any and sends call to the data mapper which in turn will call table data gateway and domain object. Once, the data mapper receives any result, with the help of the domain object it maps the data in the relevant form(here we are using JSON). The data mapper will then send that to servlet and servlets sends that back to the tomcat server.

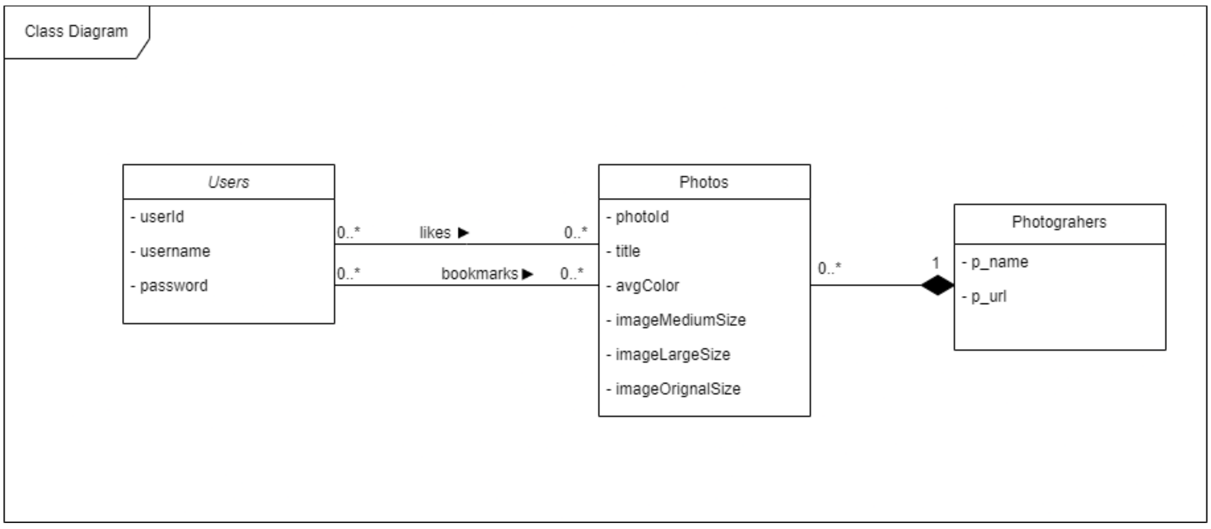
In the end, the user will be able to see the liked photos.



2.2 Class Diagram:

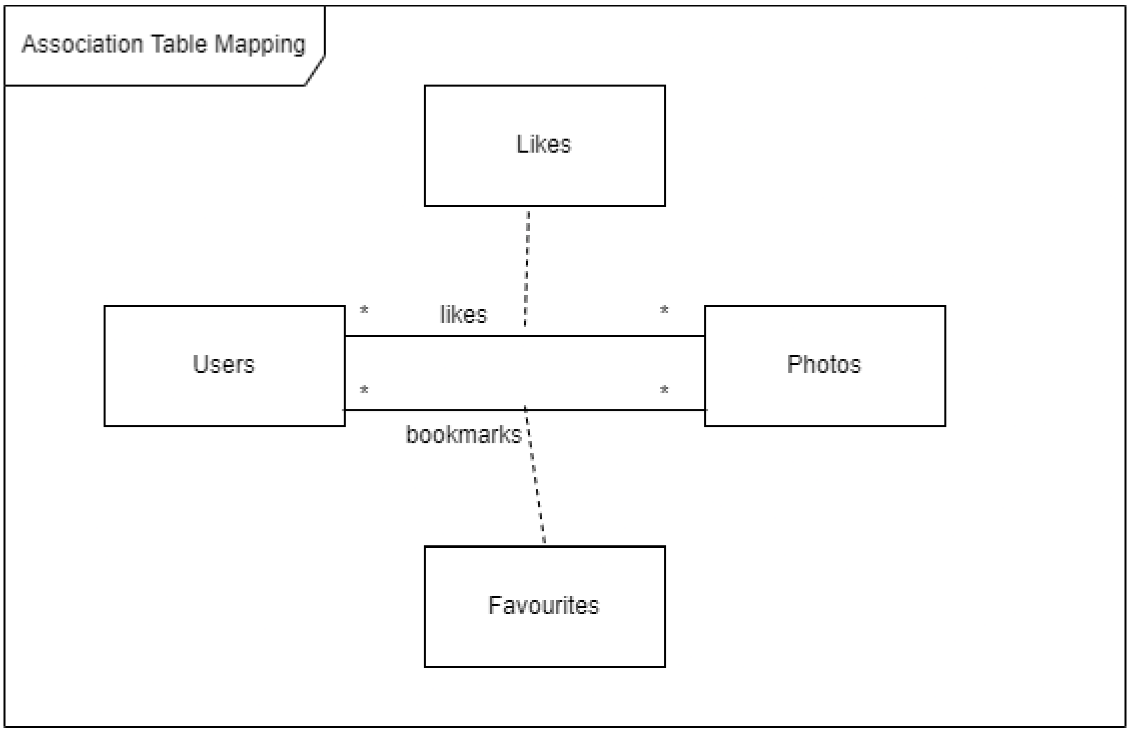
Users can like photos and also bookmark photos to store in their collection. A single photo can be liked or bookmarked by multiple users. That is why there is a many-to-many relationship between Users and Photos classes.

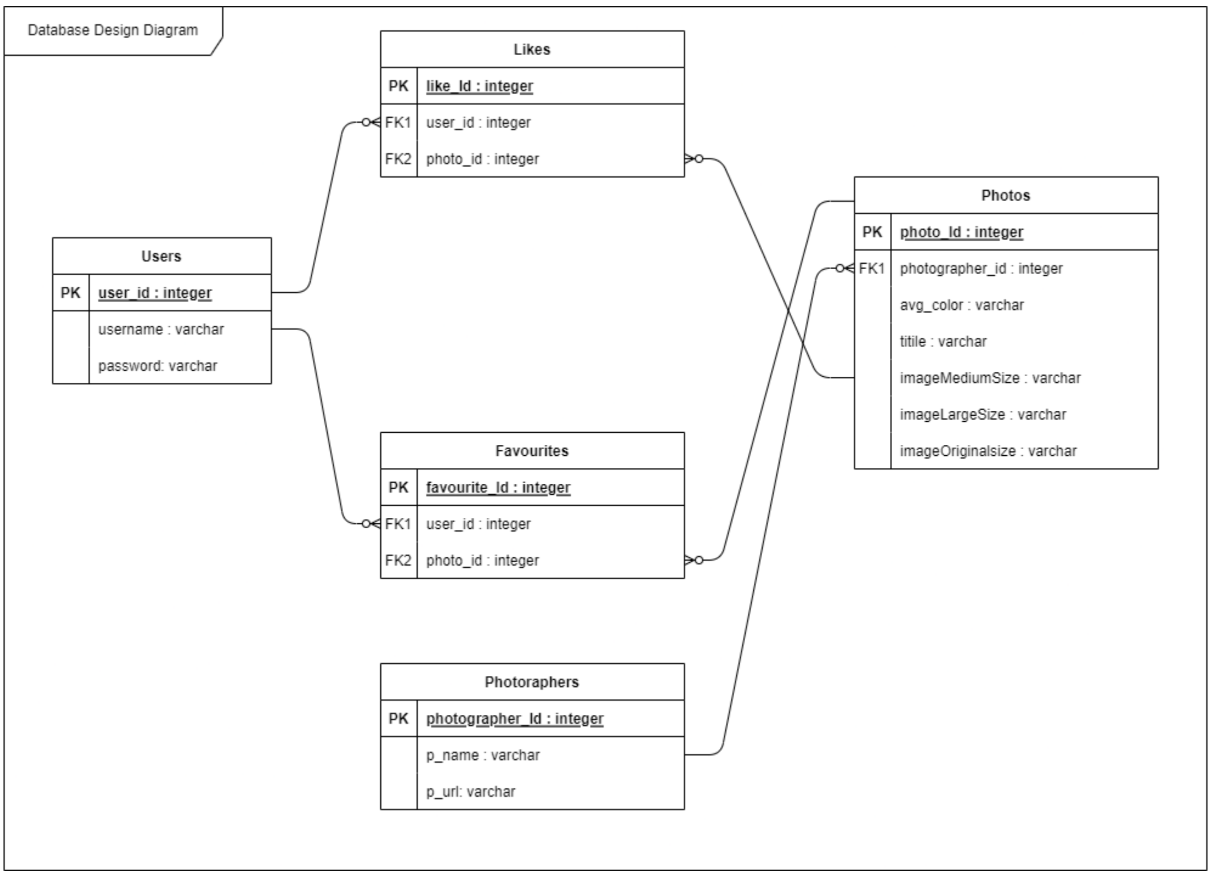
As a photo can not logically exist independently without having a photographer, there is a composition relationship between Photos and Photographers classes.



2.3 Database Modelling Diagram:

In data modelling, as there is a many-to-many relationship between Users and Photos classes, association tables namely Likes and Favourites will be required.





References

http://www.cs.toronto.edu/~wl/teach/407/2002/rup-sad.html#:~:text=%5BThe%20introduction%20of%20the%20Software,of%20the%20Software%20Architecture%20Document.%5D

https://www.se.rit.edu/~co-operators/SoftwareArchitectureDocumentation.pdf

https://www.slideshare.net/PasinduTennage/sample-software-architecture-document

https://www3.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_HowTo.html

https://semaphoreci.com/community/tutorials/testing-rest-endpoints-using-rest-assured

https://rest-assured.io/

https://tomcat.apache.org/index.html

https://reactjs.org/

https://www.javatpoint.com/servlet-tutorial

https://codeburst.io/understanding-java-servlet-architecture-b74f5ea64bf4

https://www.baeldung.com/jpa-many-to-many