StudyBoard

Software Design Presentation

Team Members

Hawon Park	PM	hawon.park@stonybrook.edu
Jeong Ho Shin	LP	jeongho.shin@stonybrook.edu
Pyungkang Hong	UI/UX	pyungkang.hong@stonybrook.edu

Table of Contents

Problem Summary

Product Solution

UI Mockup

System Architecture

UML Sequence Diagrams

Data Design

API Design

Schedule

Problem Summary

Premise

- People often turn to the internet to answer their questions

Issue

- Time → Might be able to find an answer in five minutes or five hours
- Reliability → Answers might be vague, factually incorrect, or poorly explained

Problem Solution

StudyBoard

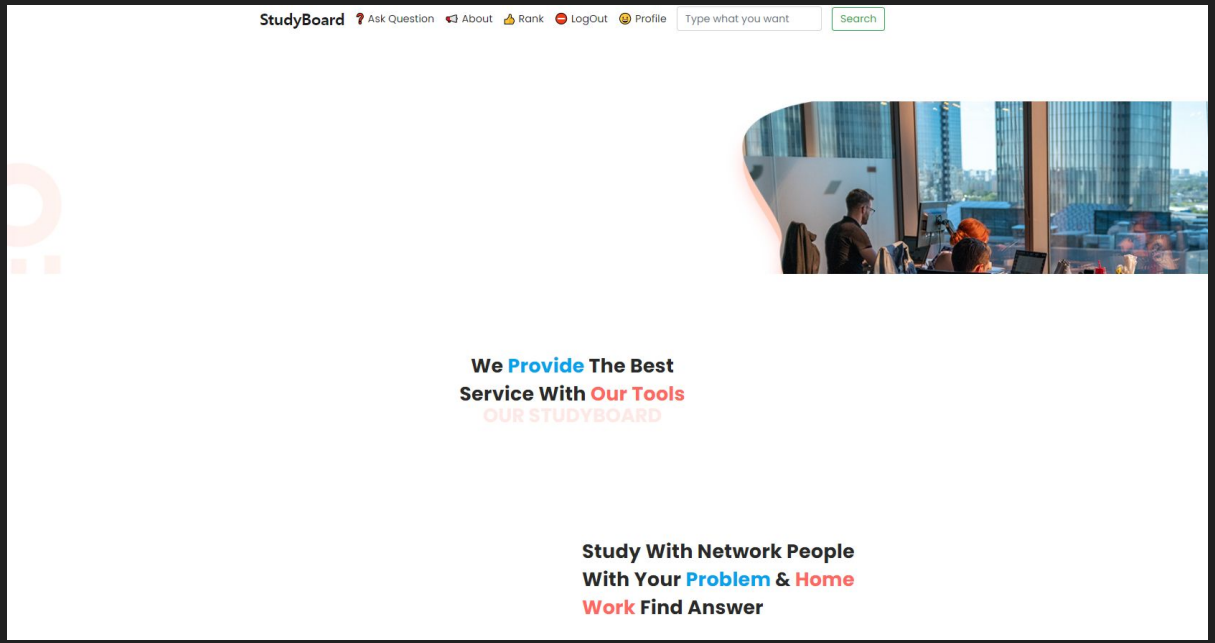
- Web platform where users can post and answer questions freely.

StudyBoard - Major Features

- Sign In with Google
- Post a question
- Reply to a question/reply
- Search for a question
- Interact with a question (Like, Favorite, Share, Report)
- Interact with a reply (Like, Report)
- Ranking System
- Moderators (Flagged posts, replies ; Users)

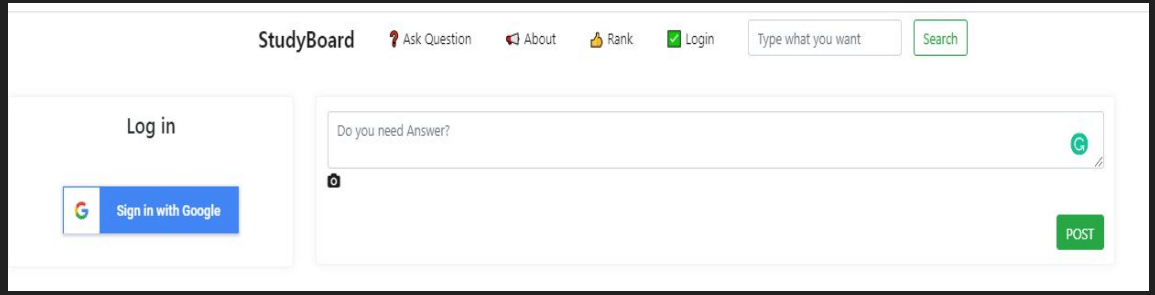
User Interface

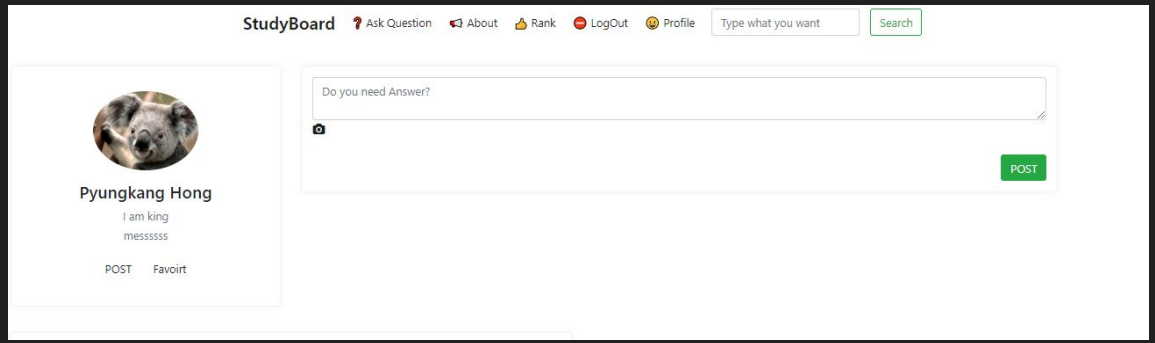
Home Page



User Interface

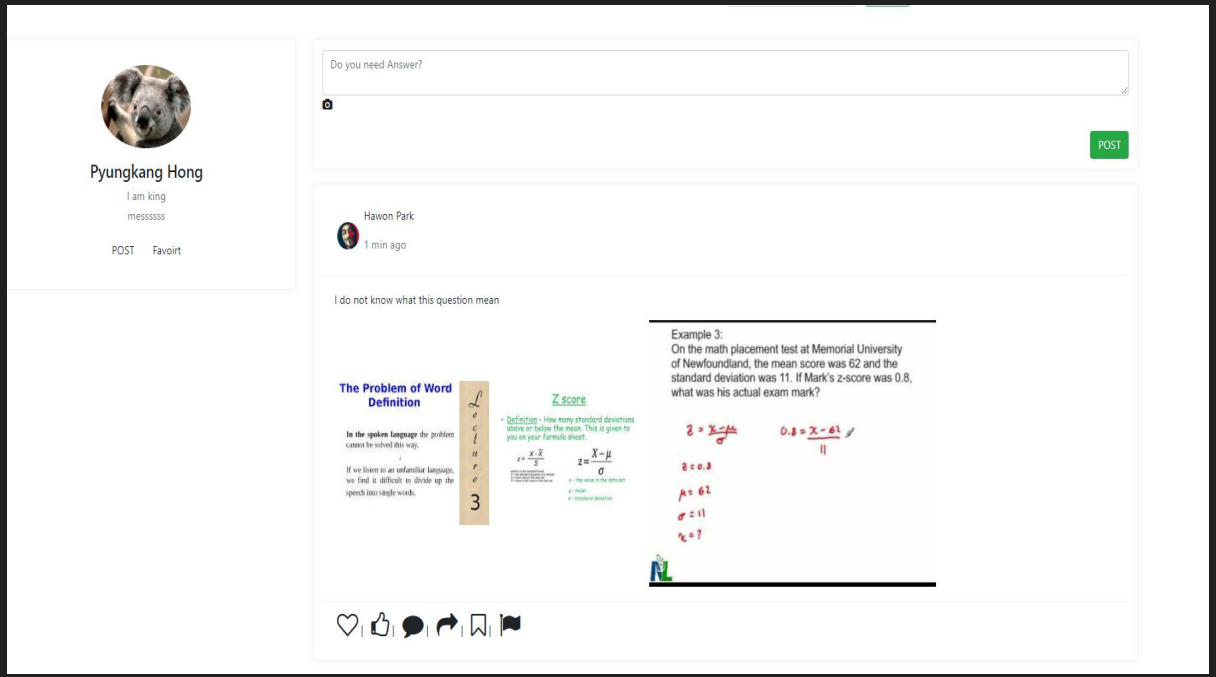
Sign In





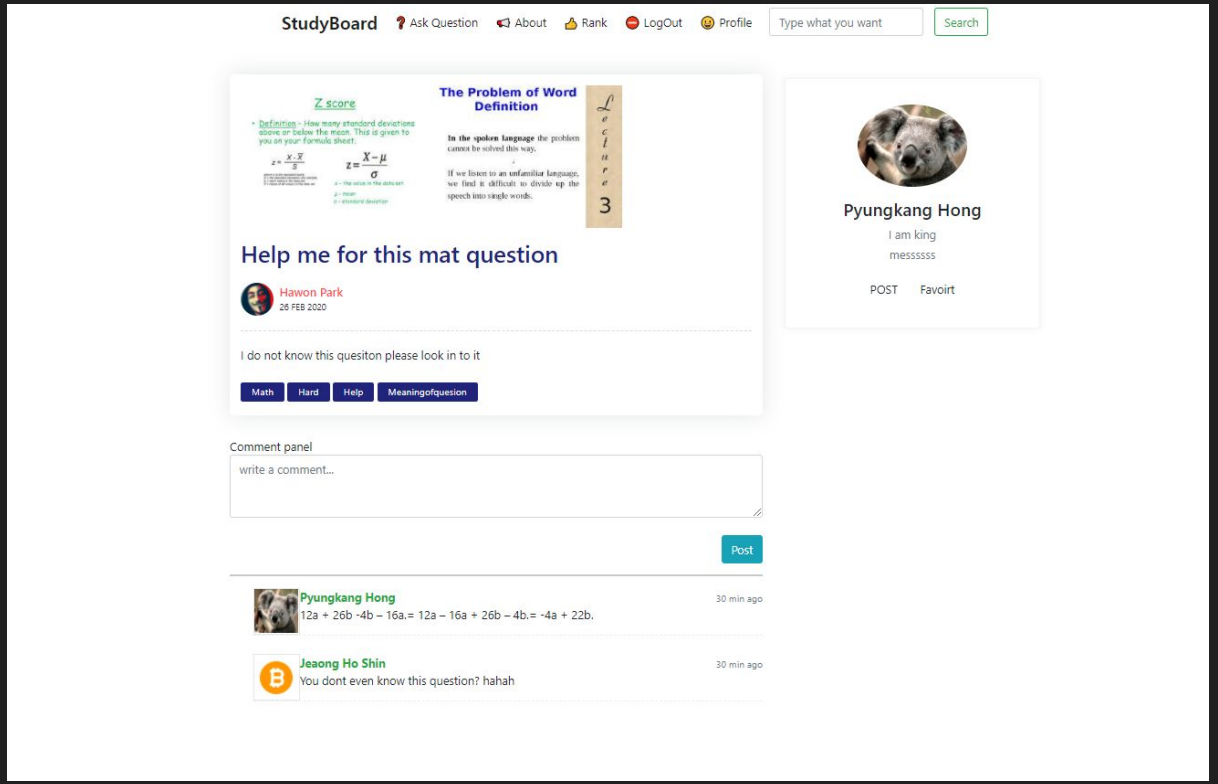
User Interface

Post a Question



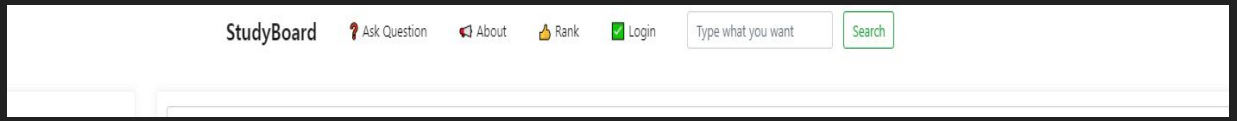
User Interface

Reply to Question



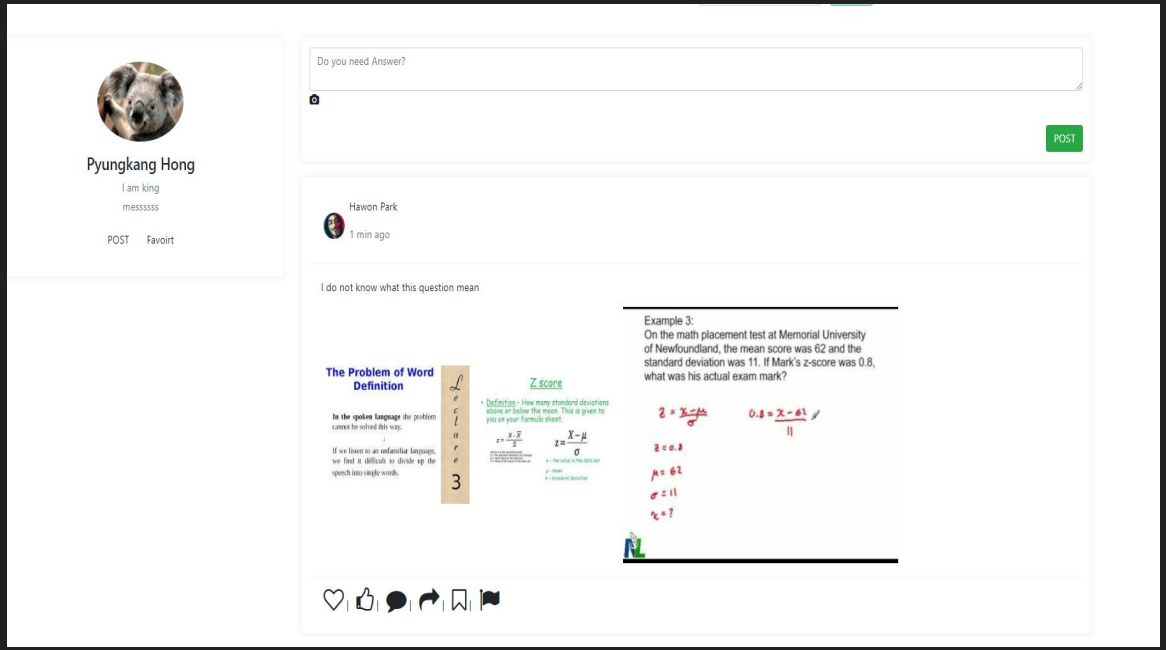
User Interface

Search for Question



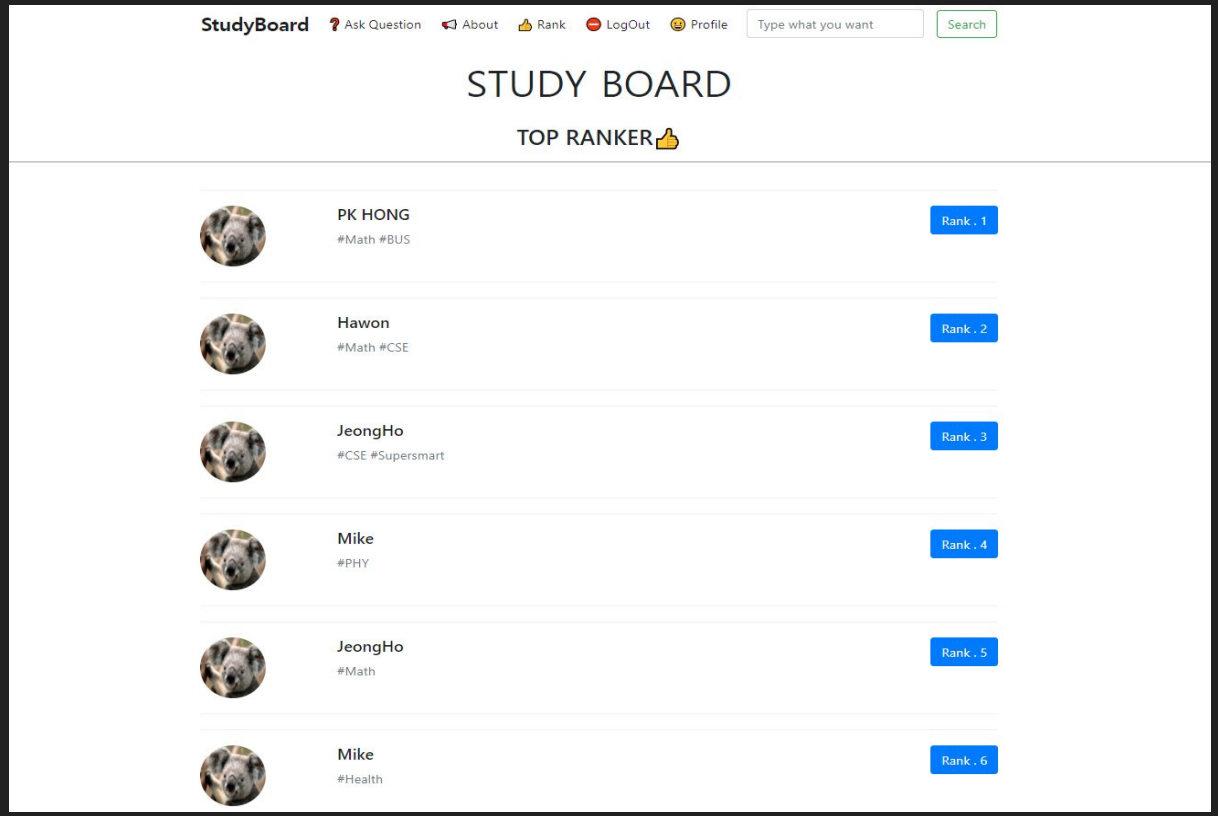
User Interface

Interact with Question



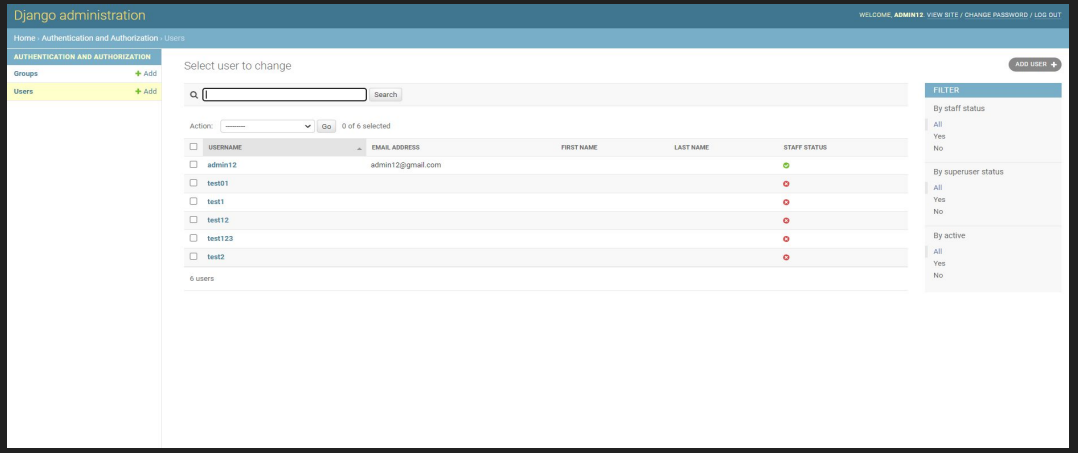
User Interface

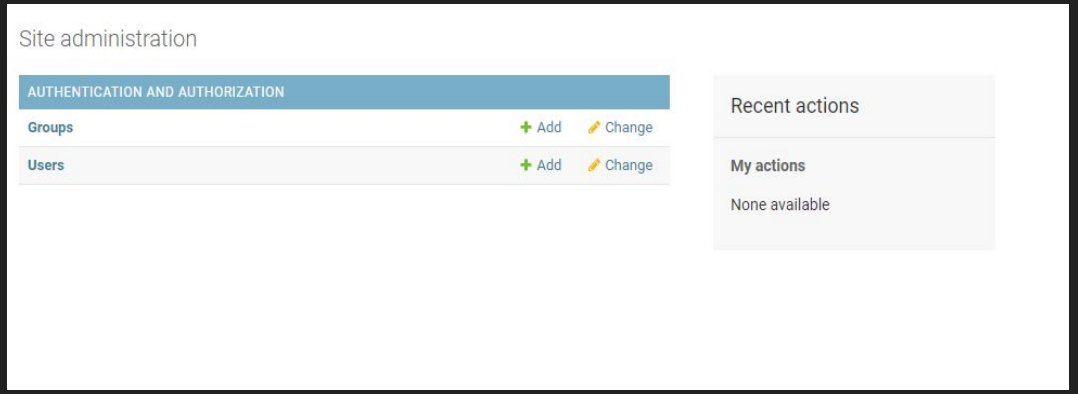
Ranking System



User Interface

Moderators





System Architecture - Key Technology Choices

Django

- Python based web framework based on the MVT design pattern
- High Scalable, Highly popular w/ many libraries

Bootstrap

- Open source HTML/CSS framework (frontend)
- Extensive documentation on UI/UX design

AWS Cloudfront

- Secure and scalable CDN
- Fast data transfer, Global service

System Architecture - Key Technology Choices

Google OAuth 2.0

- Used in conjunction with Django-Auth for easy sign up / login

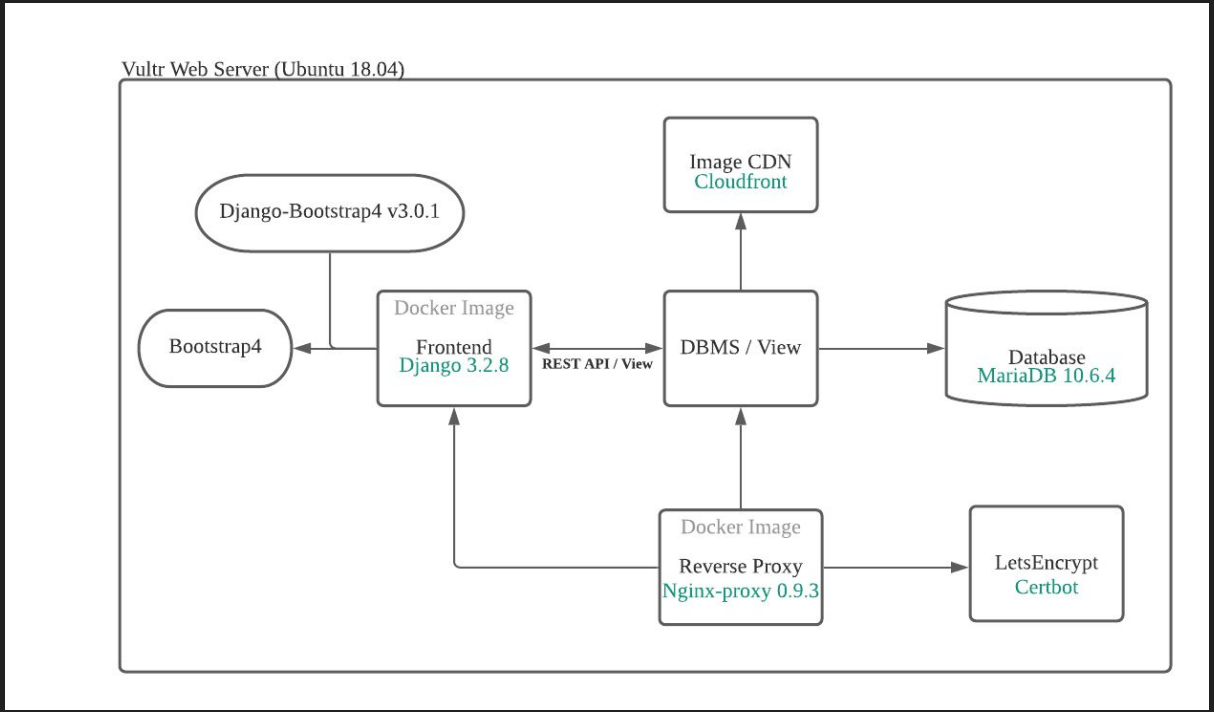
Docker

- Consistent and isolated environments
- Flexible, Scalable, Modular

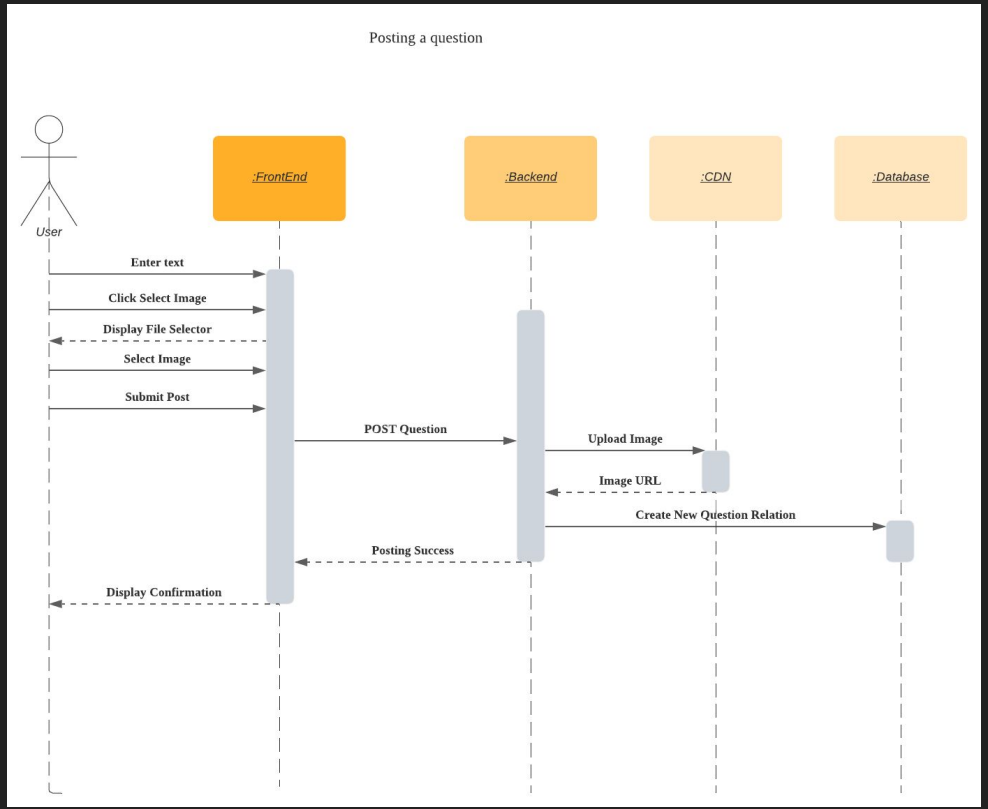
Nginx-Proxy

- Contains Nginx and Docker-gen
- Automated reverse proxy configurations whenever containers are started and stopped

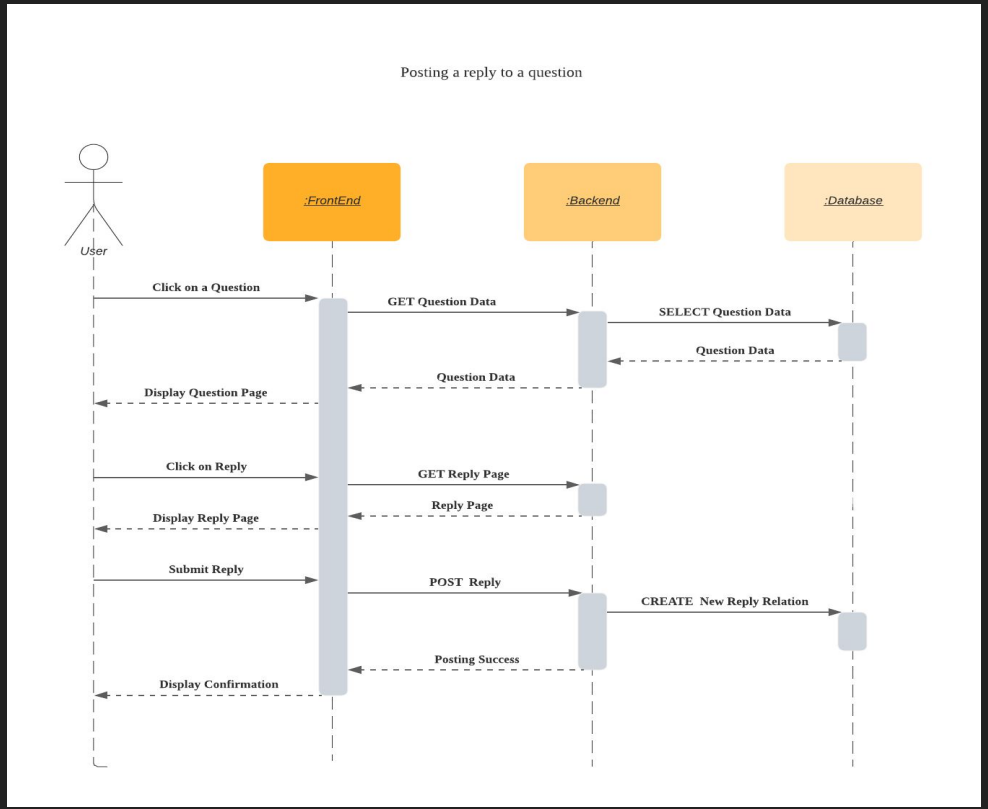
System Architecture - High Level Overview / Deployment



UML Sequence Diagrams - Post



UML Sequence Diagrams - Reply



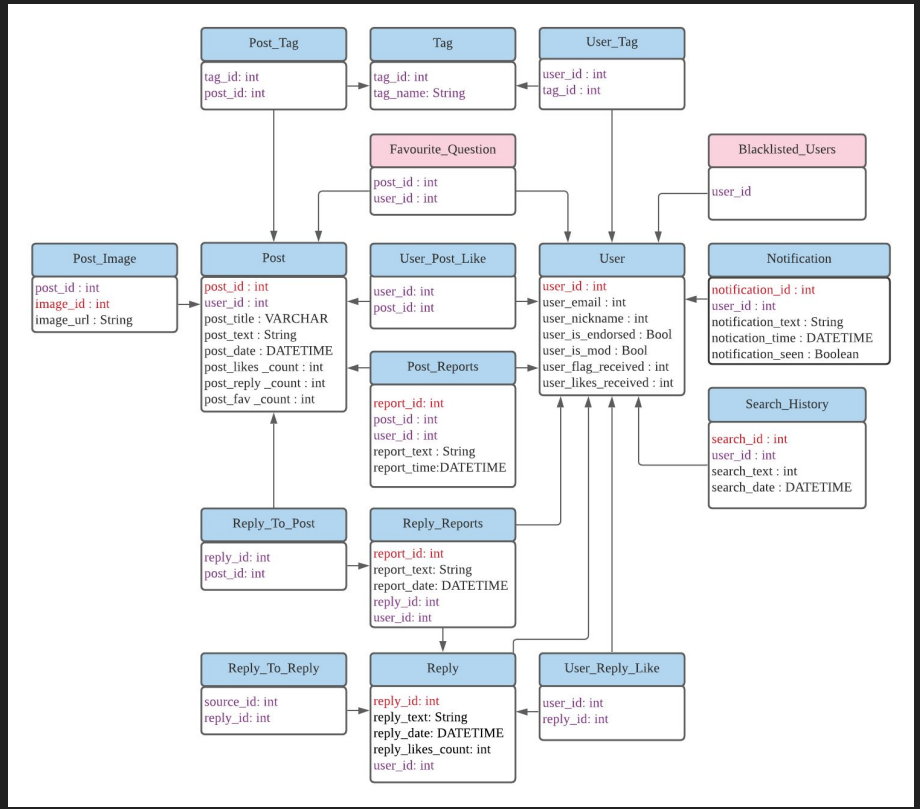
Simplified Data Design

Updates/Changes in Data Design

+ Favorites Function
+ Blacklist User Function
- Shared Function

Simplified Data Design

Arrows indicate foreign keys
Red Text → Primary Key
Purple Text → Foreign Key



API Design

+ Consolidate Profile into one api call
+ Add missing API calls
+ Consistent Naming Scheme (collections in plural, singular item)
- Deleted unnecessary API calls

Link

Summarized Schedule

Milestone 1	Milestone 2	Milestone 3	Milestone 4	Milestone 5	Milestone 6	Final
Nonfunctional Web Views	50% of Functional Requirements	100% of Functional Requirements	User Feedback	Feedback Changes; Bug Fixes	Goal is to finish by this date	Polish kinks & Submit
Initial DBMS Server Setup	Populate DBMS w/ Test Data	Alpha Test	Beta Test	Reverse Proxy Deployment		

Thank you!

Any questions?