Debias

The Agile Alliance

Members

Siddharth Bhardwaj
Michael Brog
Ruizhe Liu
Arshdeep Saini
Hengchao Xiang

Presentation Structure

Introduction
What is Debias?
01

Requirements and Use Cases
Users, Functional and non-functional requirements
02

Overall System Architecture
High level overview of the architecture and alternatives
03

Subsystems
Functionalities of the subsystems and interactions between them
04

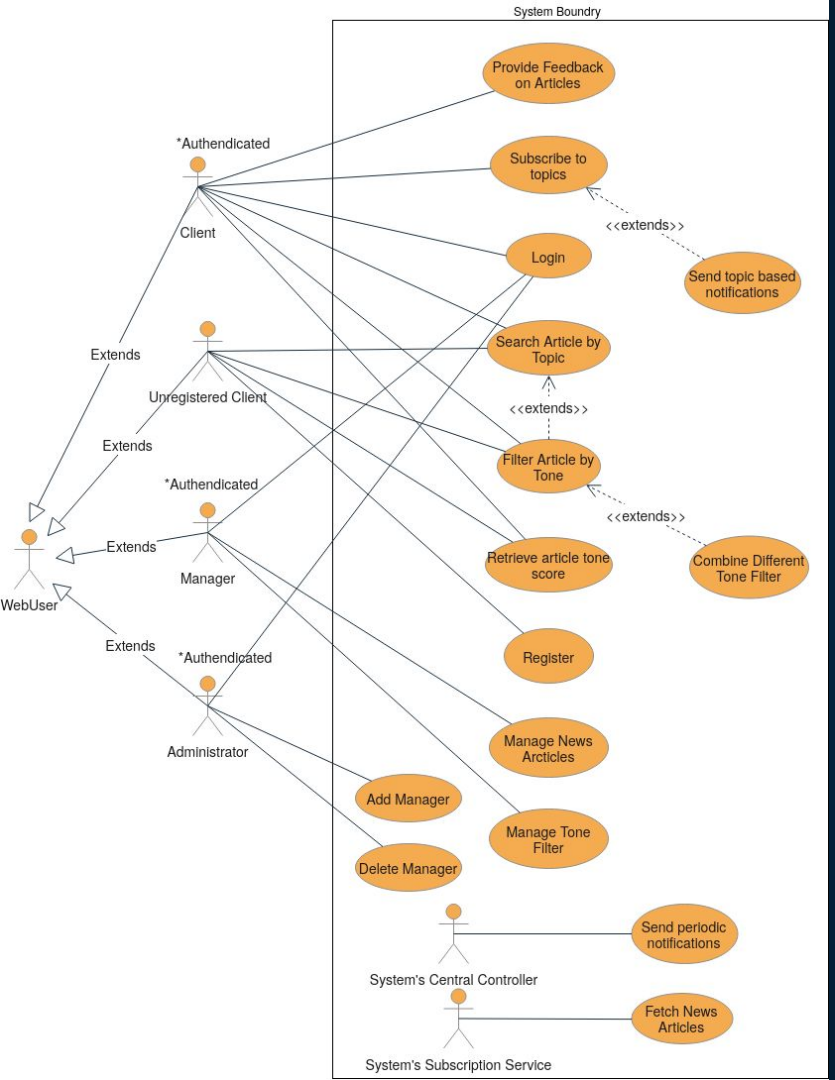
Activity/Sequence diagrams
Overview of the activity and sequence diagrams
05

Quality requirements and Concurrency Control
Functionalities of the subsystems and interactions between them
06

Lessons learned
Functionalities of the subsystems and interactions between them
07

What is Debias?

- News aggregation tool.
- Aims to solve the difficulty in finding articles with different viewpoints.
- Empower the Customer to obtain balanced perspective on news topics, which may be riddled with biased rhetoric.



Use Case Model of Debias

Based on set of all use cases specifying the complete functionality of the system.

Detailed information on requirement elicitation and use case modeling can be found from: https://viloil.github.io/EECS4314-Team-Project/

Overall System Architecture

Model - View - Controller

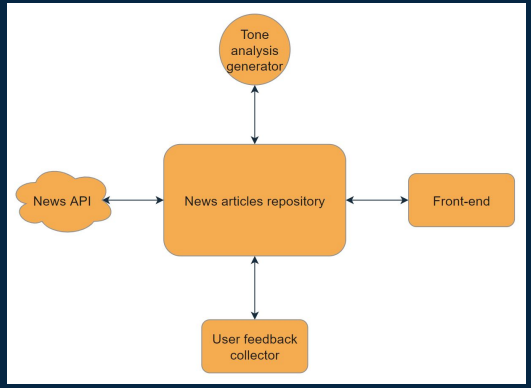
- Not suitable for small scale applications
- Higher complexity and imbalanced workload, especially the model
- Requires specialized developers
- Hard to implement synchronously
- Event-driven nature results in more complex debugging

Three Tier

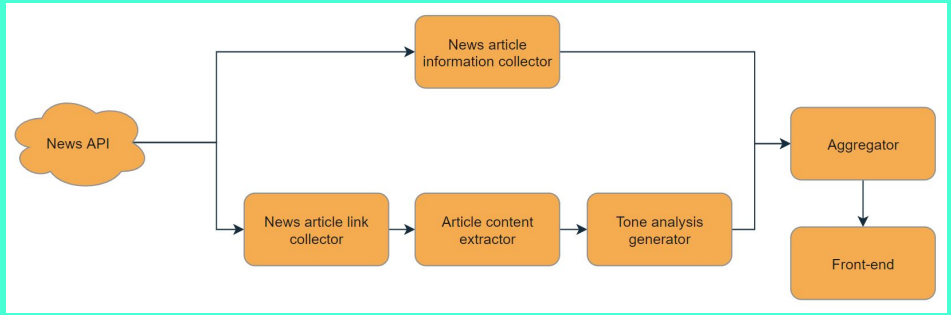
- User interface, Business logic and database
- Component independence and organized development
- Scalability and reusability of business logic
- High availability during periodic updates
- Fast synchronous development

News Processing Pipeline

Repository Architecture (Alternative)

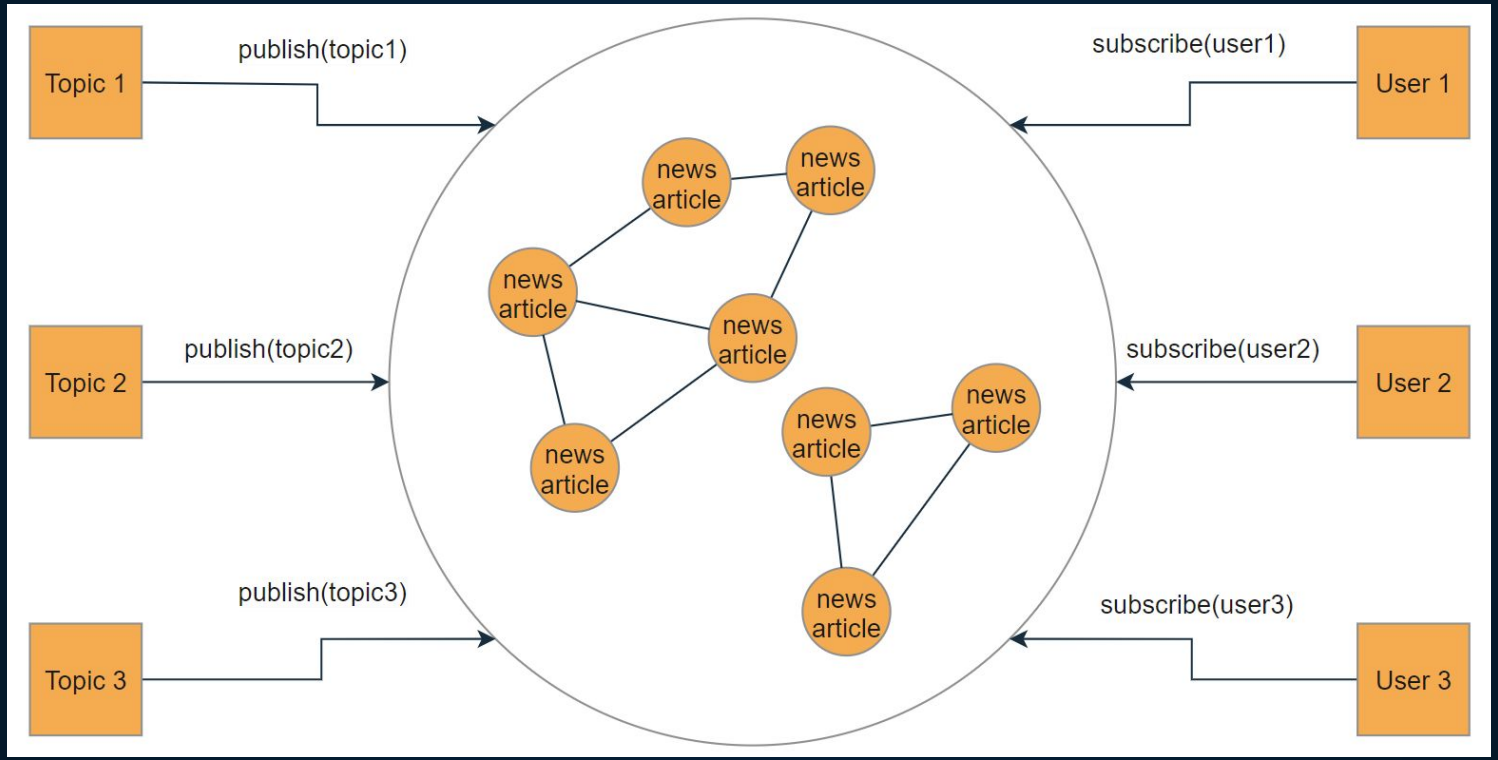


Pipe and Filter Architecture

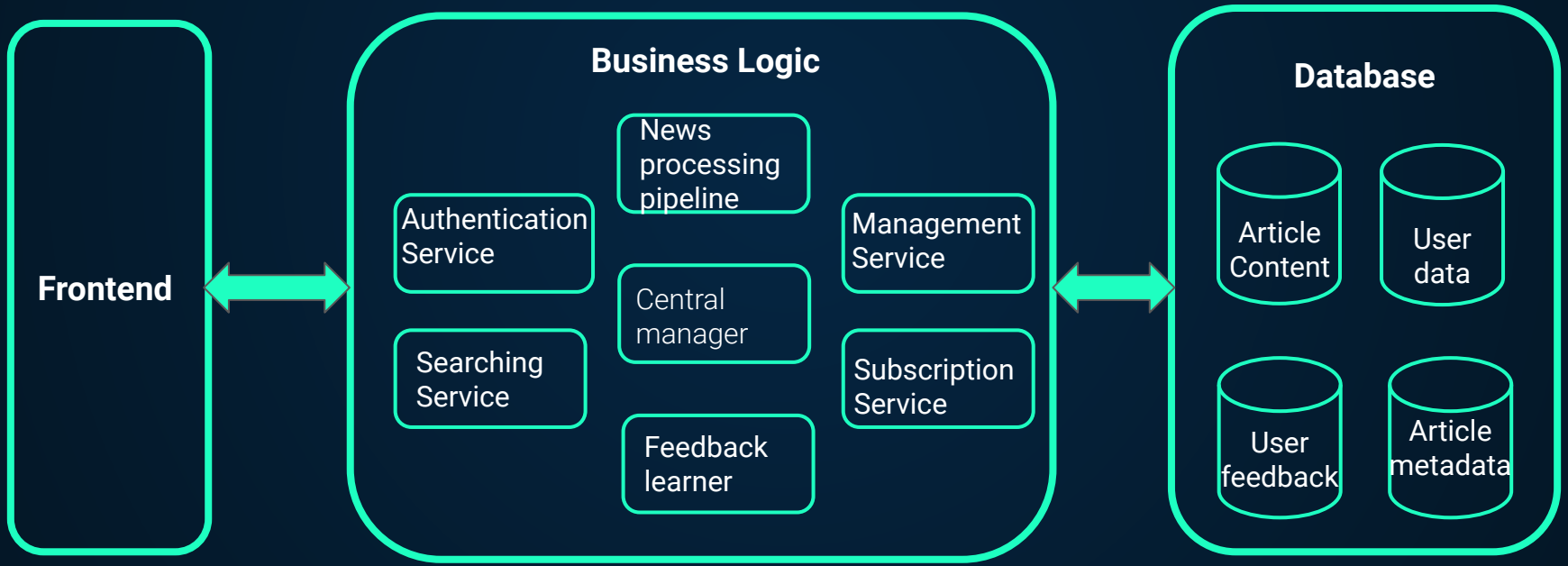


Subscription management

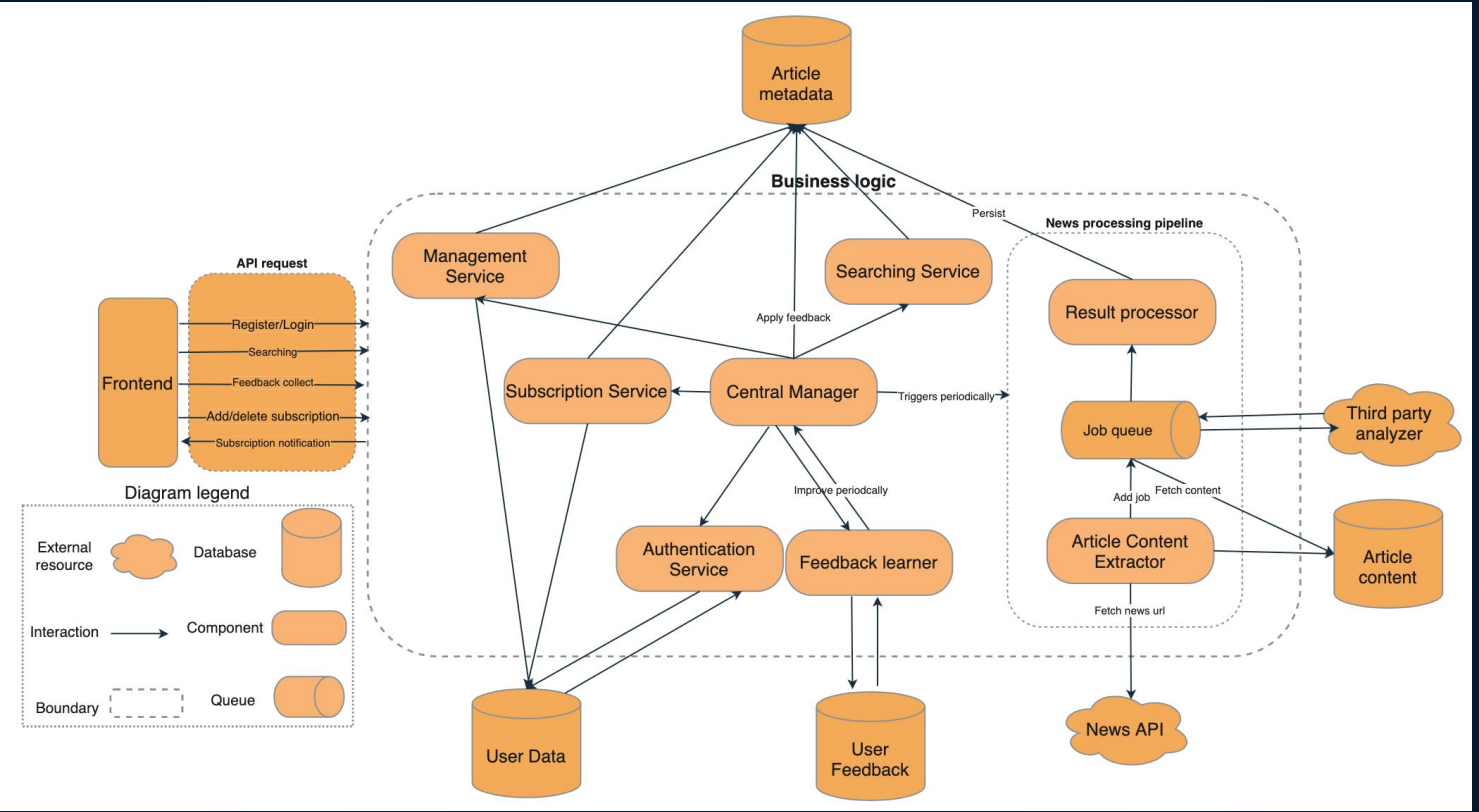
Implicit Invocation Architecture



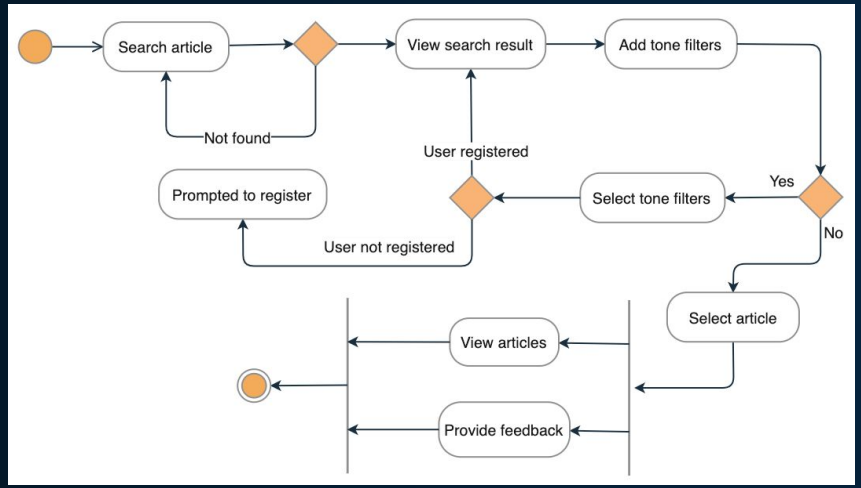
Subsystem

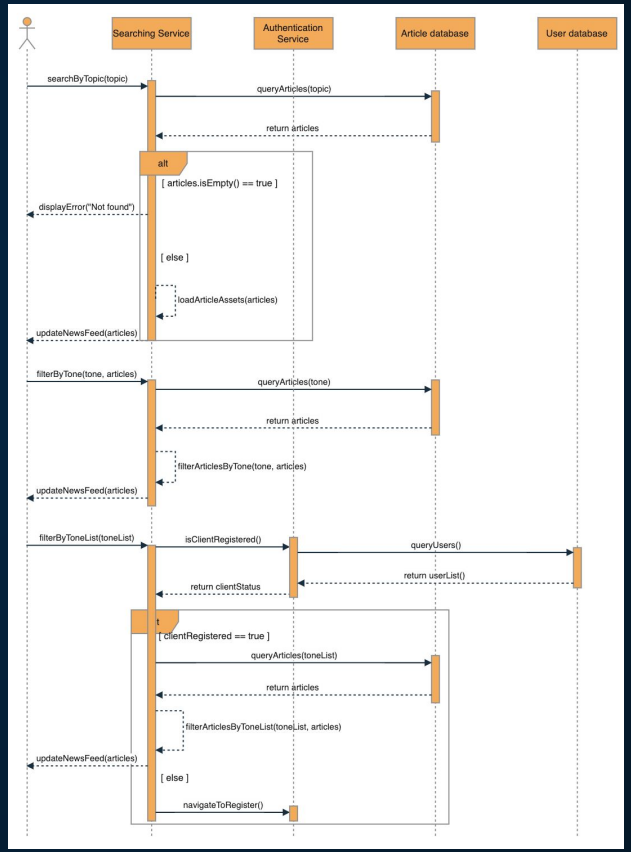

Frontend
Business Logic
Authentication Service
News processing pipeline
Management Service
Central manager
Searching Service
Subscription Service
Feedback learner
Database
Article Content
User data
User feedback
Article metadata

Interactions between subsystems

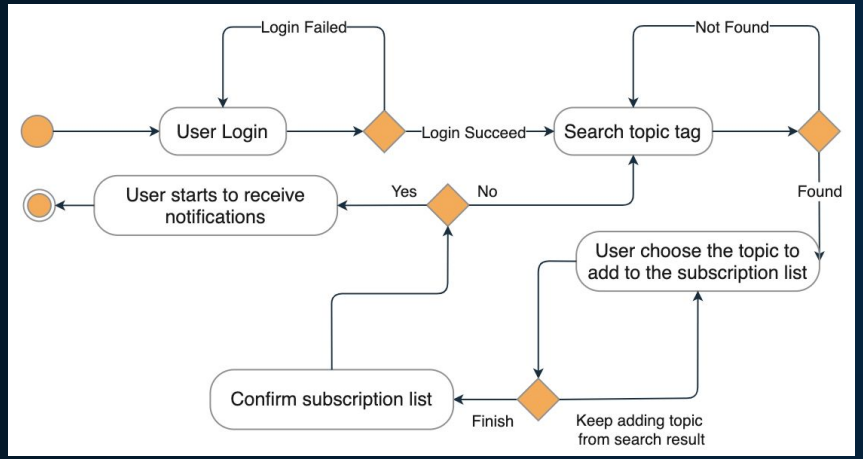


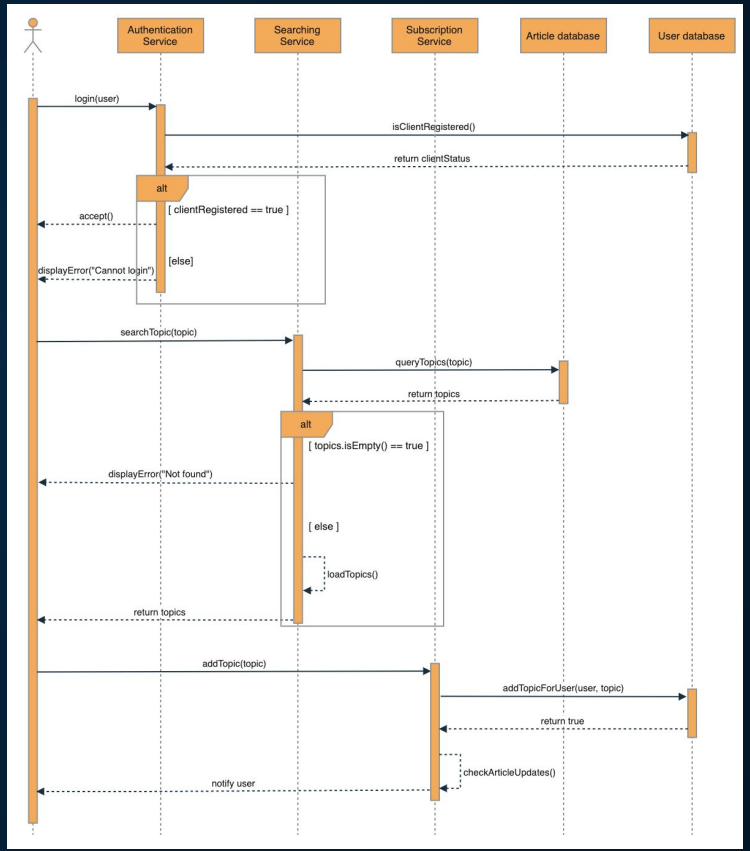
Use case 1: Client searches the article with tone filters





Use case 2: Client adds topic tag to subscription list





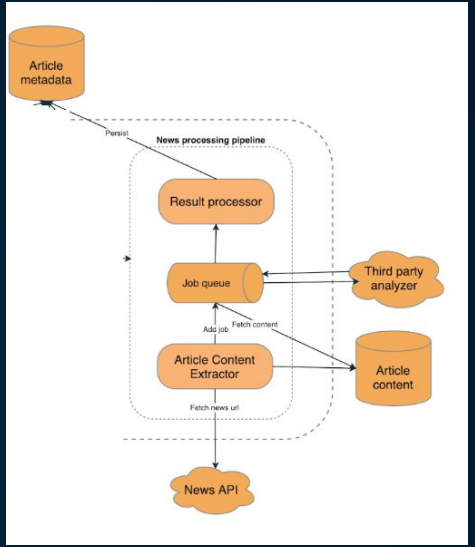
Quality Requirements and Debias?

Performance	System should limit api requests (throughput) to 100 per hour for each client querying the news database for articles
Integrity	Application should verify the user's request before allowing them to use its features.
Availability	-Installation of a new version should leave all database contents and all personal settings unchanged -System should not be unavailable more than 1 hour per 1000 hours of operation.
Portability	web app should be installable as a standalone app on the client's device
Reliability	80+ on lighthouse performance score
Privacy	Client's search history for articles and subscription topics will be stored locally on their device

Concurrency

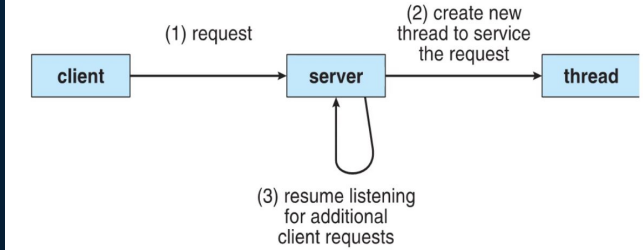
Database

- A queue system is used to ensure that articles are not being processed more than once, and are not being inserted into the database more than once
- By using a database that ensures ACID properties are being followed we can ensure that multiple users requesting data from the same table get consistent results



Handling concurrent users

In order to make sure many users can use the system concurrently we spin off threads that can retrieve data from the database so that the server can move onto the next request until the response is ready

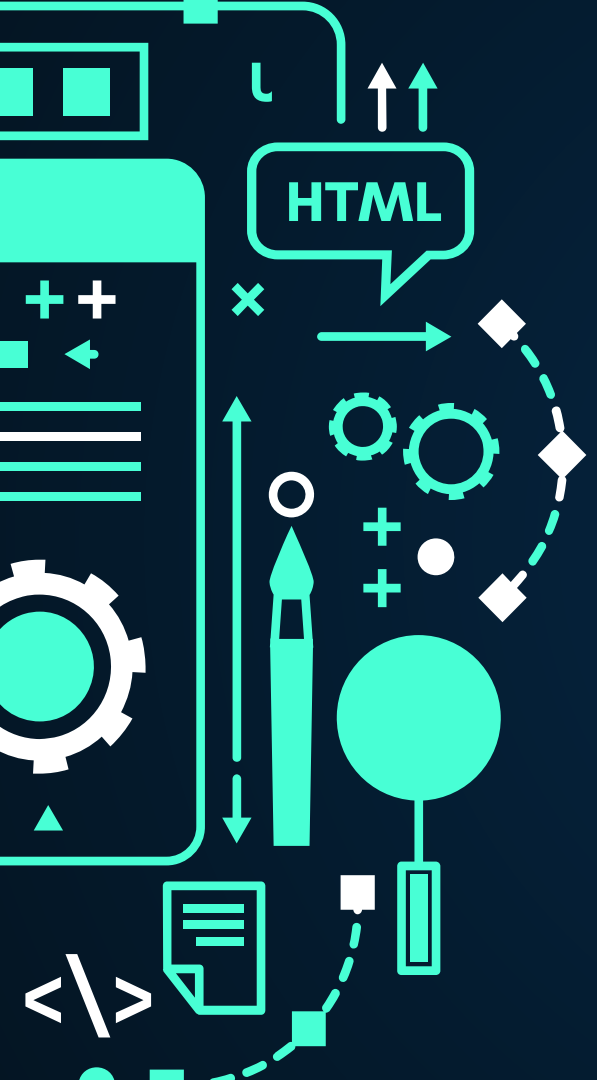


Lessons learned

- Determining the data flow is a good place to start for component planning

- Defining too many use cases beyond their basic functions in the initial stage will increase the difficulty of the system design

- While it does increase difficulty, defining more use cases could make it easier for the design to help support more complex functions



THANK YOU!

Feel free to ask any questions.

https://viloil.github.io/EECS4314-Team-Project/