



Disc :: Song Sharing Platform

Symmetric Social Network Service

Team: Andy Jeong, Ariana Freitag, Leart Krasniqi, Dhvanil Shah

Date: February 27, 2019



The Crew

Core Requirements



Accessible



Interactive



Connected



Customizable

Stretch Goals



Custom
Privacy



Security



Messaging

OVERVIEW

1



2



3



4

Tech Stack

React, Nginx,
Apollo-Spotify,
MongoDB

Data / UI

Schema, Field, Type
of data entities

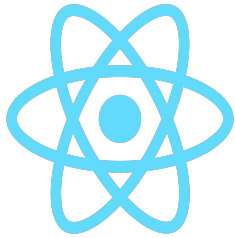
Black Box

Client, Server,
Database Interaction

High-Level Overview

Route-Handling,
Business-Logic,
Database-Level

1 TECH STACK



ReactJS



Apollo-Spotify



Nginx



MongoDB

MongoDB

1 TECH STACK

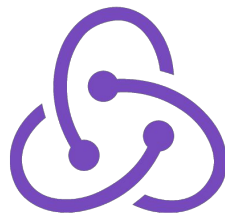
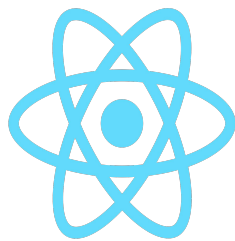
Why React?

Reusable Components

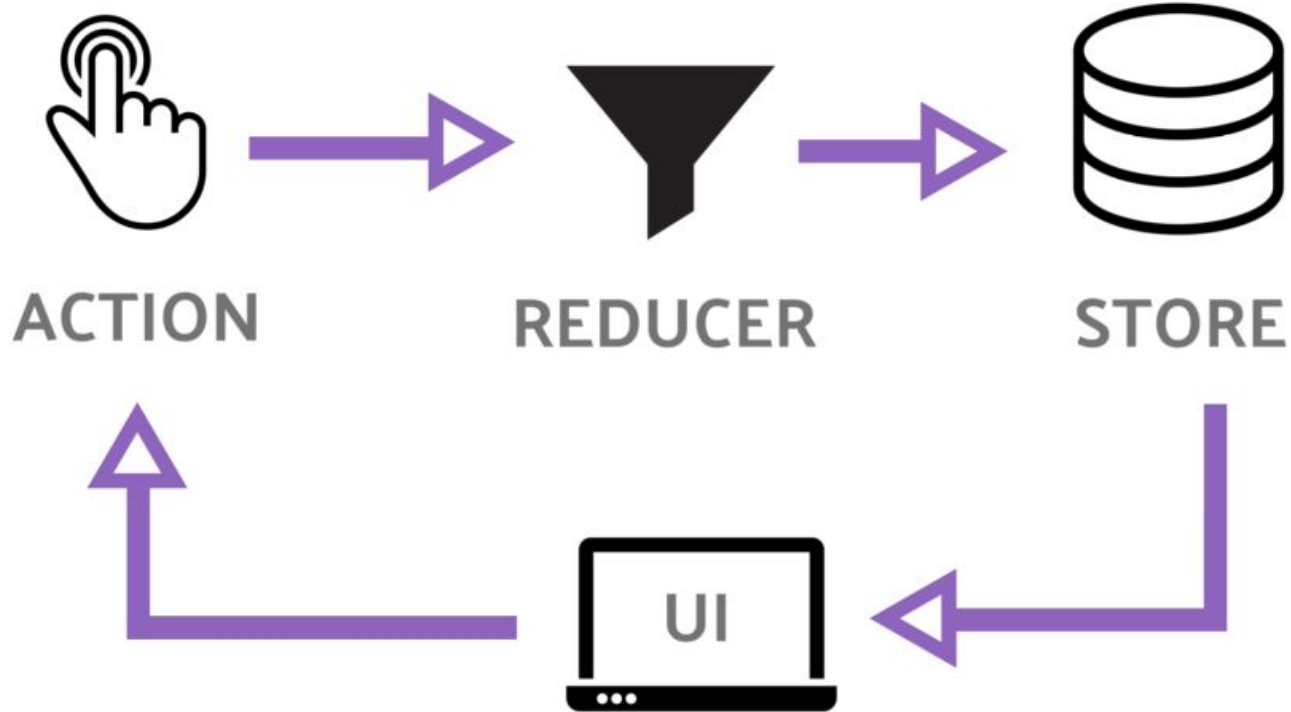
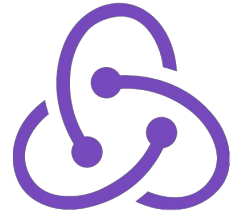
Fast Render with Virtual DOM

Redux for State Management

Axios for a Lightweight HTTP Client



1 TECH STACK



1 TECH STACK

Axios

Simple Setup

```
var instance = axios.create({  
  
  baseURL:  
  'https://some-domain.com/api/',  
  
  timeout: 1000,  
  
  headers: {'X-Custom-Header':  
  'foobar'} //possibly a JWT  
  
});
```

Sample HTTP Request

```
axios.get('/user?ID=sek')  
  
  .then(function (response) {  
  
    console.log(response);  
  
  })  
  
  .catch(function (error) {  
  
    console.log(error);  
  
  });
```

1 TECH STACK

Why NGINX + Apollo?



×



Simply Defined Request Routing

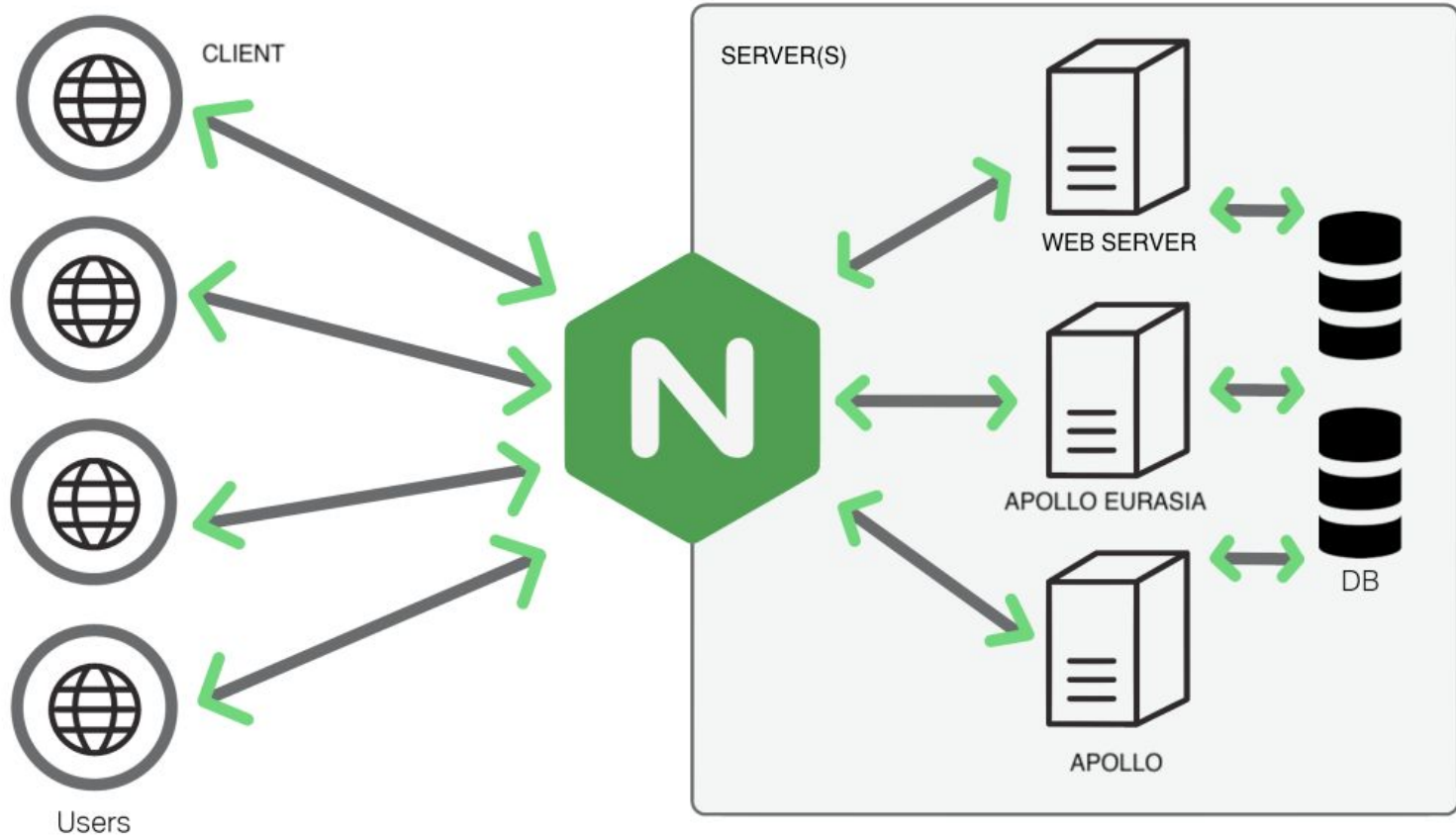
Easy Request/Reply Handlers

Built-in Testing Service

Well Documented Web Serving

Easy Content Caching and Load Balancing

1 TECH STACK



1

TECH STACK

Why MongoDB?



MongoDB
(Non-Relational DB)

MySQL

- Maturity
- Compatibility
- Cost-effective
- Replicable
- Sharding

MongoDB

- Dynamic schema
- Scalability
- Manageability
- Speed
- Flexibility

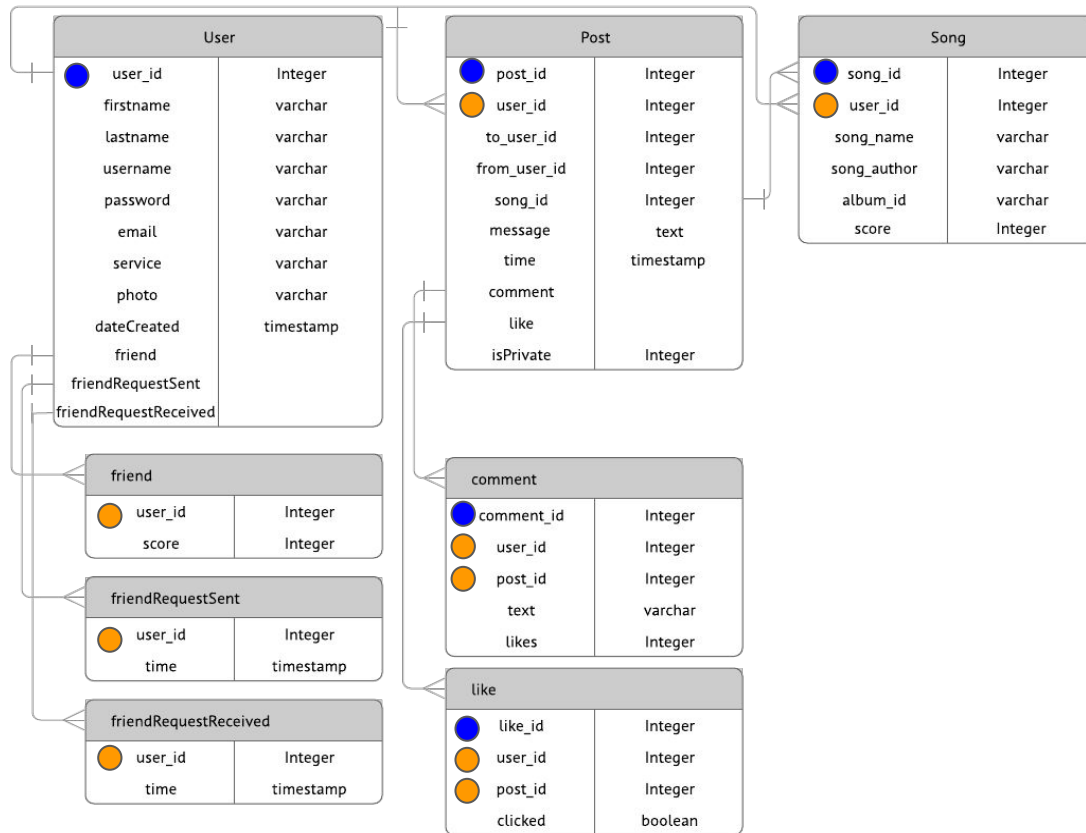
2 DATA

Entity-Representation Diagram

Primary Key ●

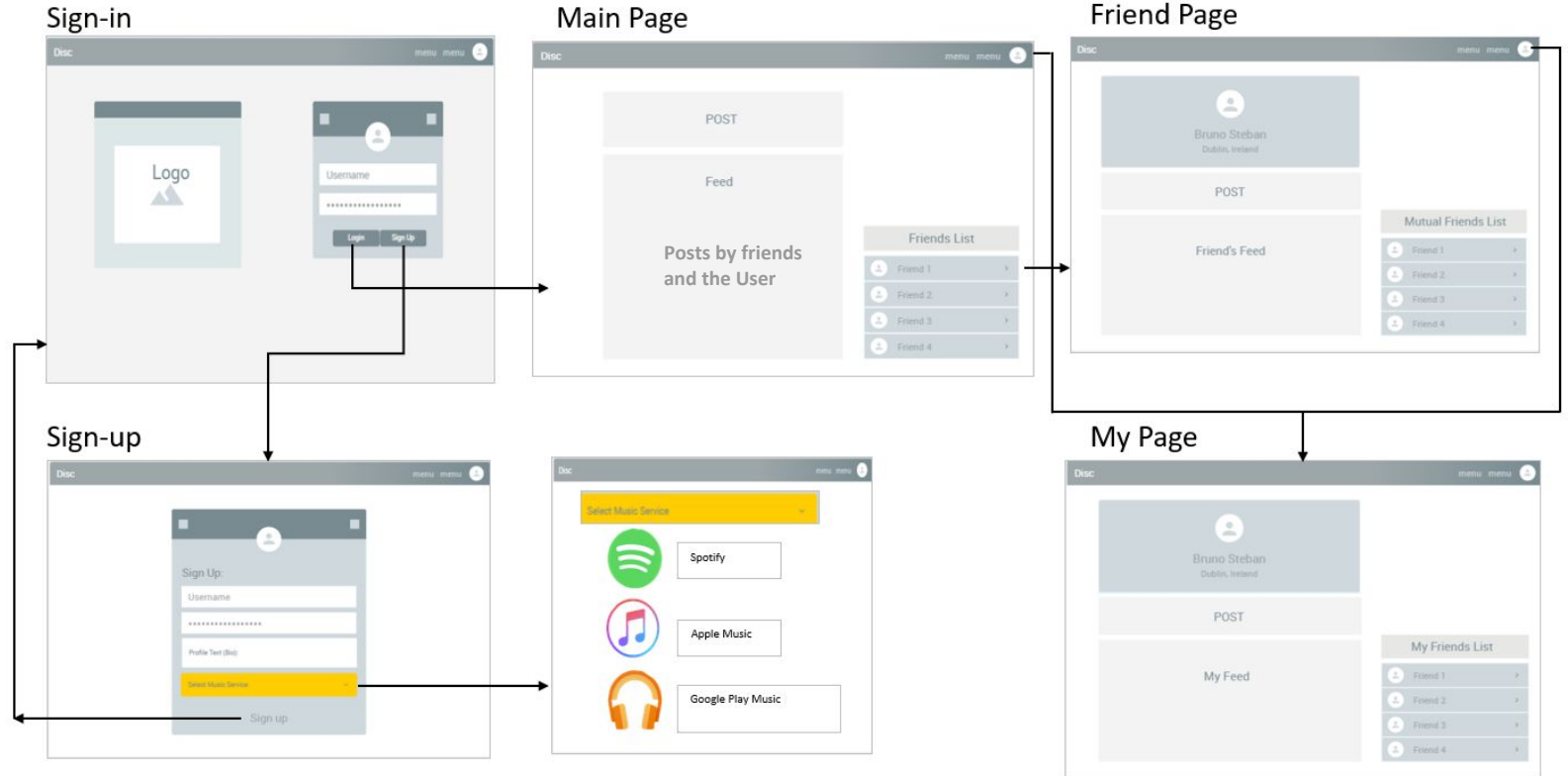
Foreign Key ●

* All required fields



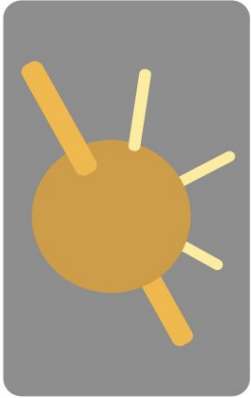
2 DATA

Process Flow Diagram

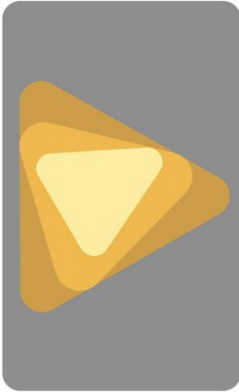


2 UI Design (In Progress)

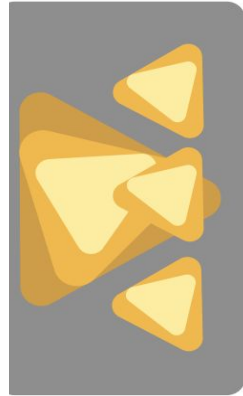
Icons



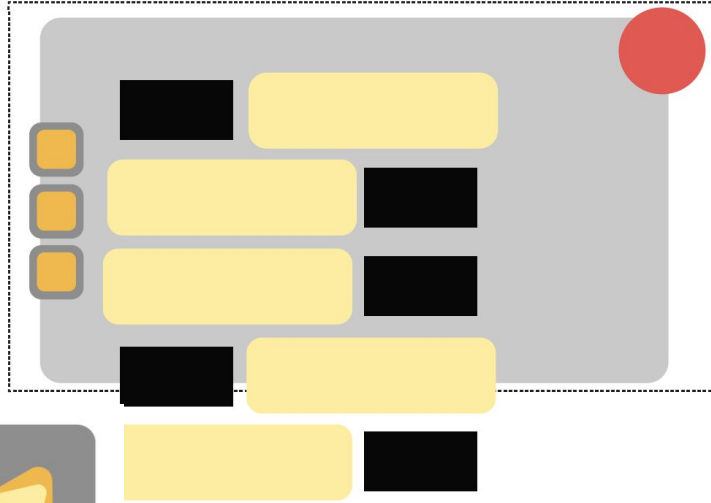
settings



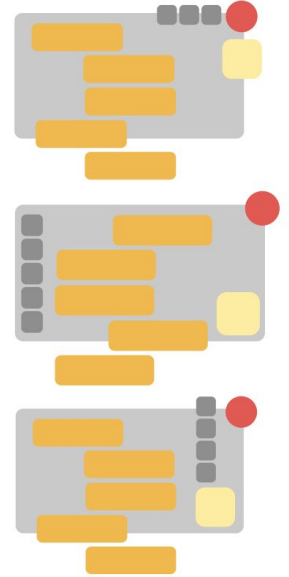
profile



friends



Website

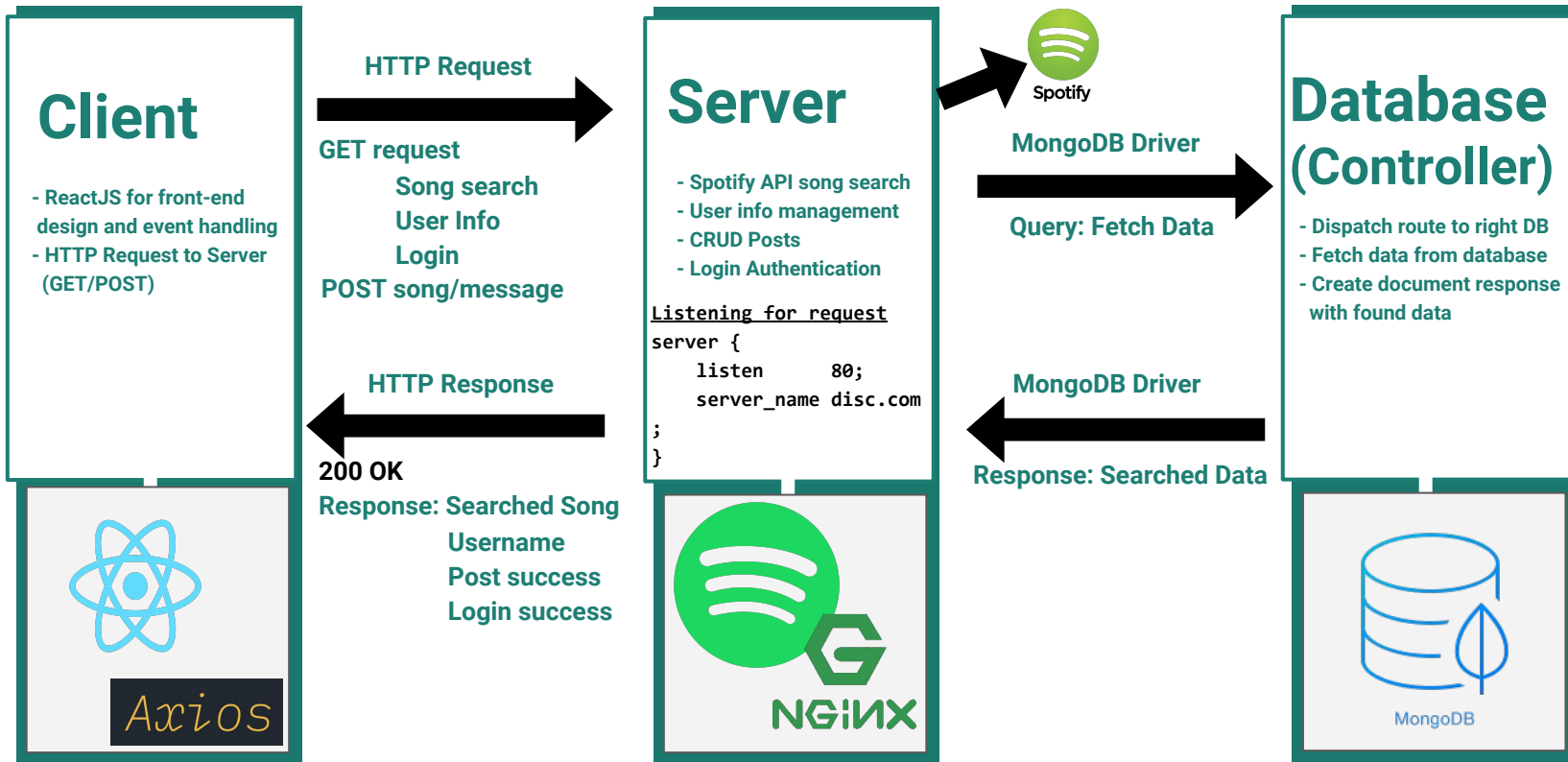


Designed by Leo Serriere A'20

3

BLACK BOX

Client-Server-DB Interaction Overview

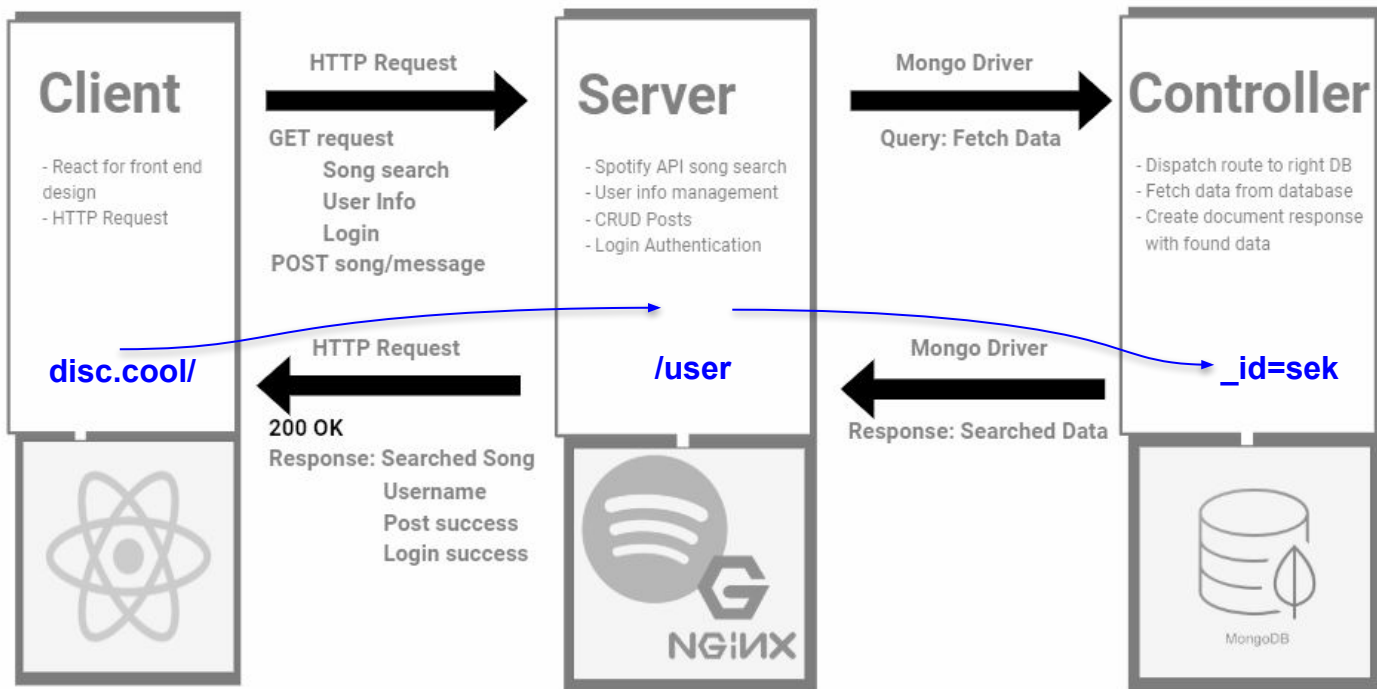
Asynchronous Handling

```
import com.mongodb.ConnectionString;
import com.mongodb.ServerAddress;
import com.mongodb.async.client.*;
import com.mongodb.connection.ClusterSettings;
import com.mongodb.connection.netty.NettyStreamFactoryFactory;
import java.util.Arrays;
import static java.util.Arrays.asList;
```


4 HIGH-LEVEL OVERVIEW

Route-handling

GET `www.disc.cool/user?ID=sek`



4

HIGH-LEVEL OVERVIEW

Route-handling

JavaScript (Front-End)

```
axios.get('/user?ID=sek')  
  
  .then(function (response) {  
  
    console.log(response);  
  
  })  
  
  .catch(function (error) {  
  
    console.log(error);  
  
  });
```

Java (Back-End)

```
public static void init(Environment environment)  
{  
    environment.routingEngine()  
        .registerAutoRoute(Route.sync("GET",  
            "/user?ID=sek", handlerFunction()));  
}
```

4 HIGH-LEVEL OVERVIEW

Business-Logic

- 1 Support song search through Spotify API (Goal: music services++)
- 2 Friends list module filtered by user's relationship to the friend on current page
- 3 Post privacy setting adjustable only by the owner of the post
- 4 Profile page info is viewable to all users but only editable by the specified user