Go-Split

Split costs, not friendships

Harry, Jenna, Ana, Scott and Naeem

Agenda:

1. Project Overview
2. Ways of Working
3. UML Diagrams
4. Entity Relationship
5. Wireframe
6. Demo - Functionalities
7. Challenges & Bugs
8. Future Recommendations
9. Q&A

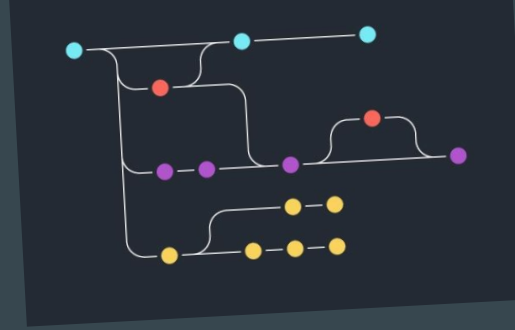
Project Overview

The Task: Design and build a full-stack application providing a service/solve a problem. Full Stack incorporates both front and back ends.

The API: A report generation for trips outlining the costs due by the organiser. Linking endpoints to the application to add, edit and delete data

Our Project: A mobile application allowing users to add friends, add trips and split costs between friends who attend the trip.

Ways of Working



Trello - Planning tasks

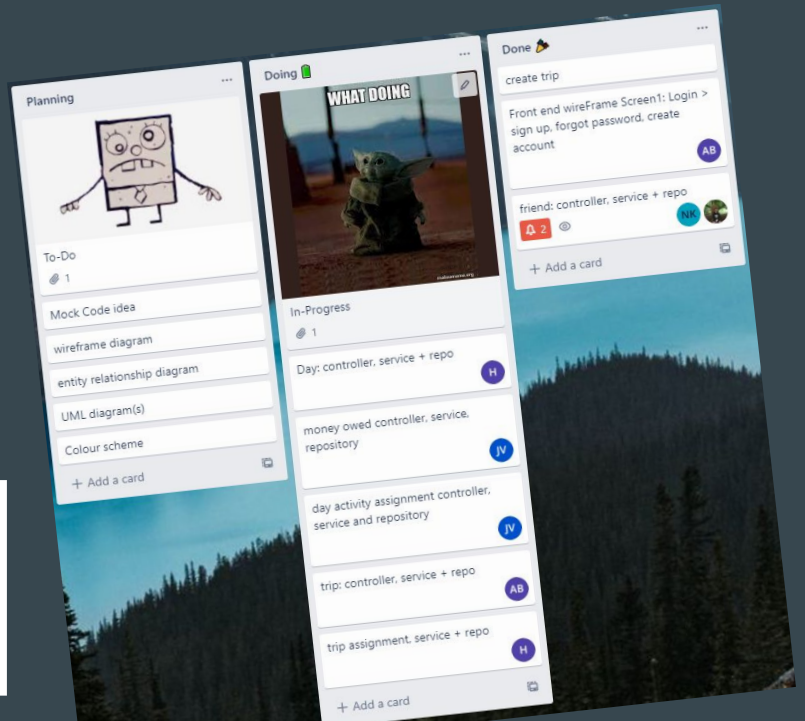
Lucid - Relationship & UML Diagrams

Figma - Wireframing

VS Code

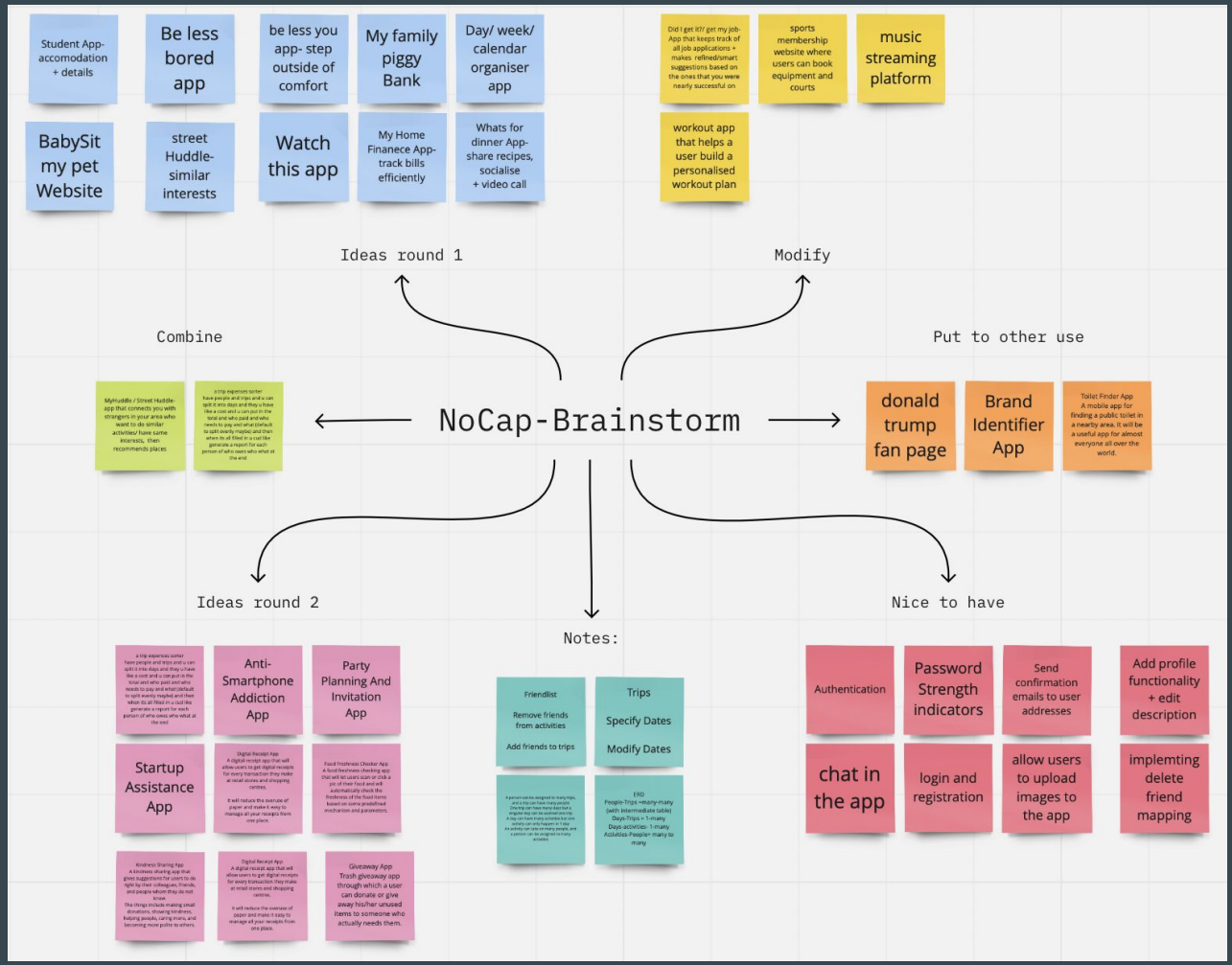
IntelliJ, Driver/Navigator

Git-Branching in VS Code & IntelliJ

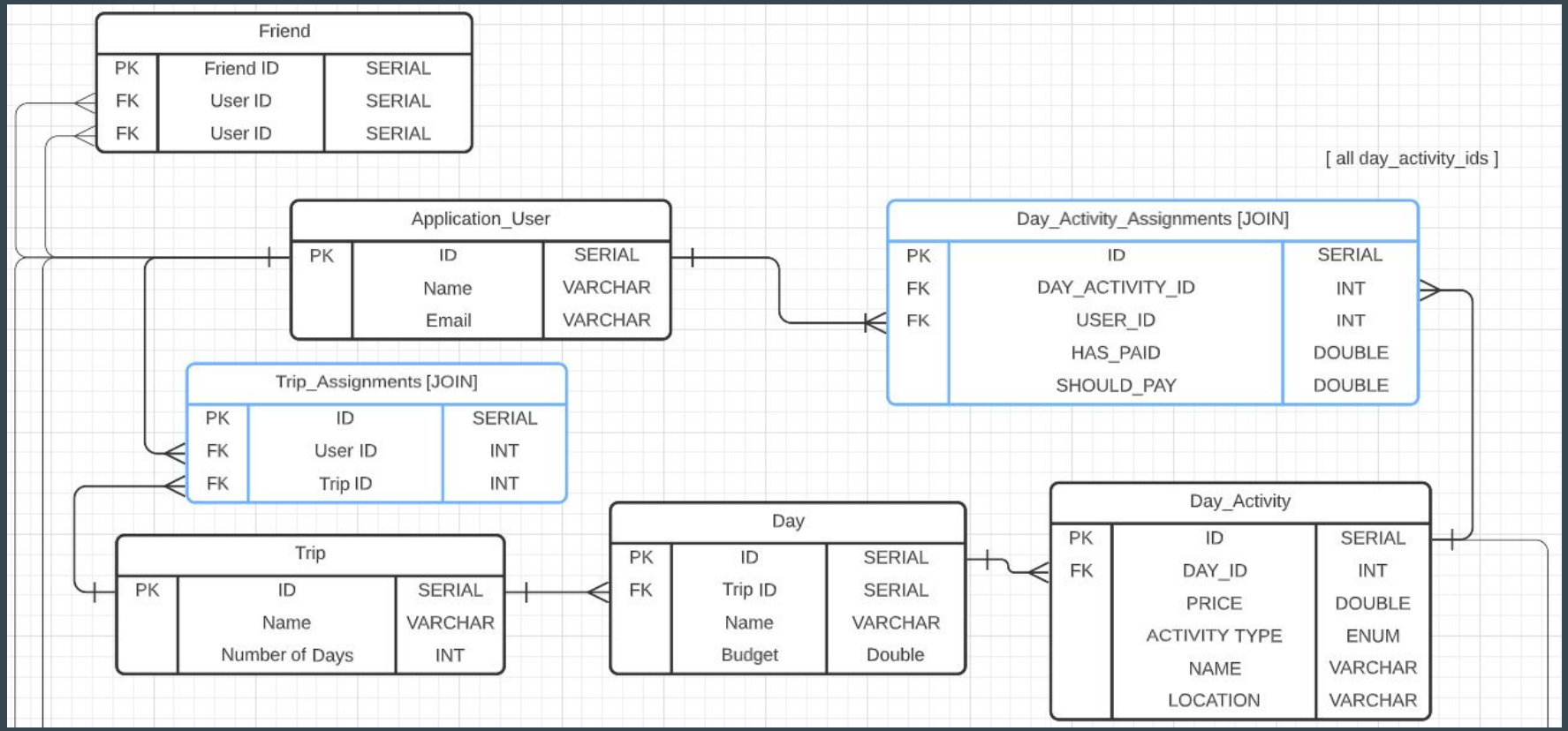


Brainstorm

- Generate
- Combine
- Modify

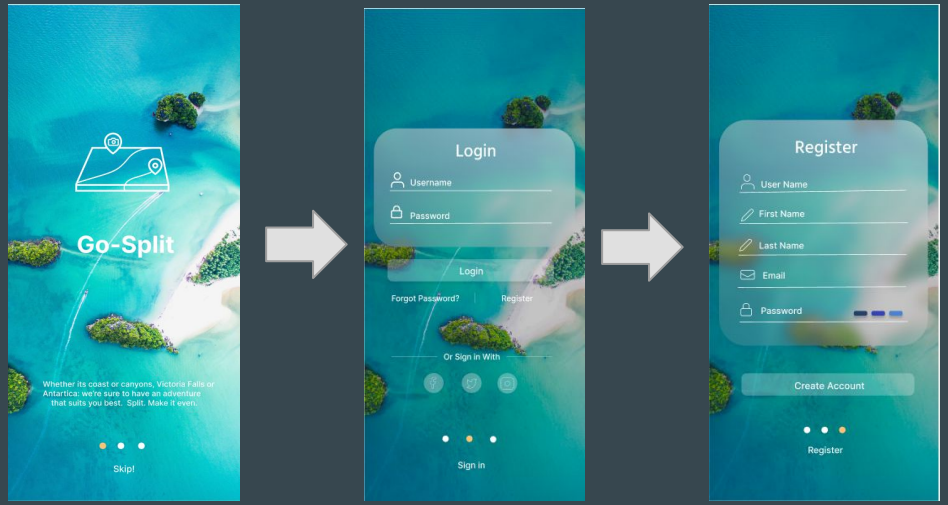


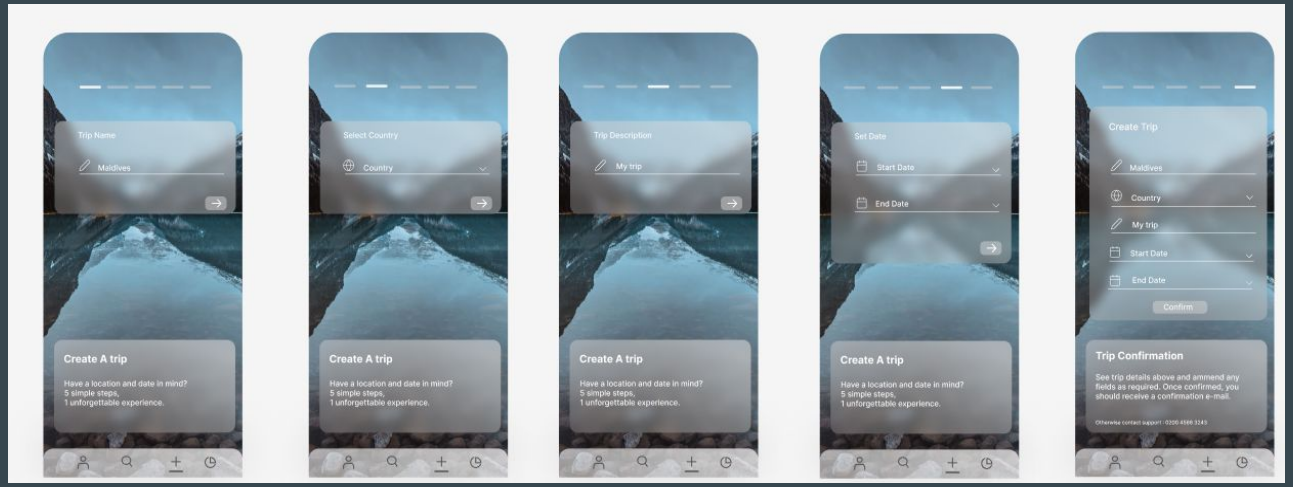
Entity Relationship Diagram- Scott

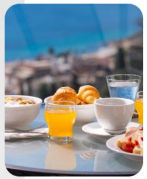


Wireframes

- Sign in
- Register
- Add trip
- View Trip
- View activities







Mouton Noir Brunch £235

Food +3

12:30 pm 13:40 pm Cancel

Atlas Mountains Hot Air Balloon £648

Sports +1

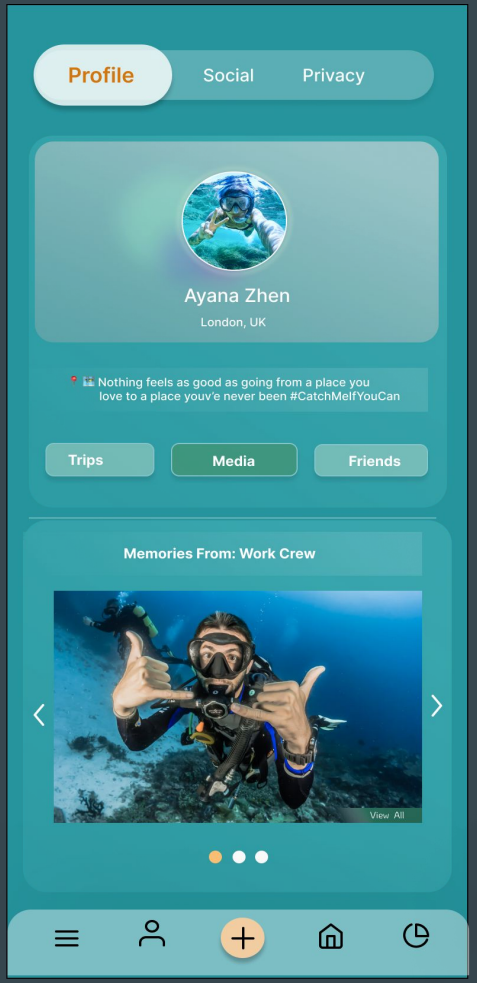
18:30 pm 19:45 pm Cancel

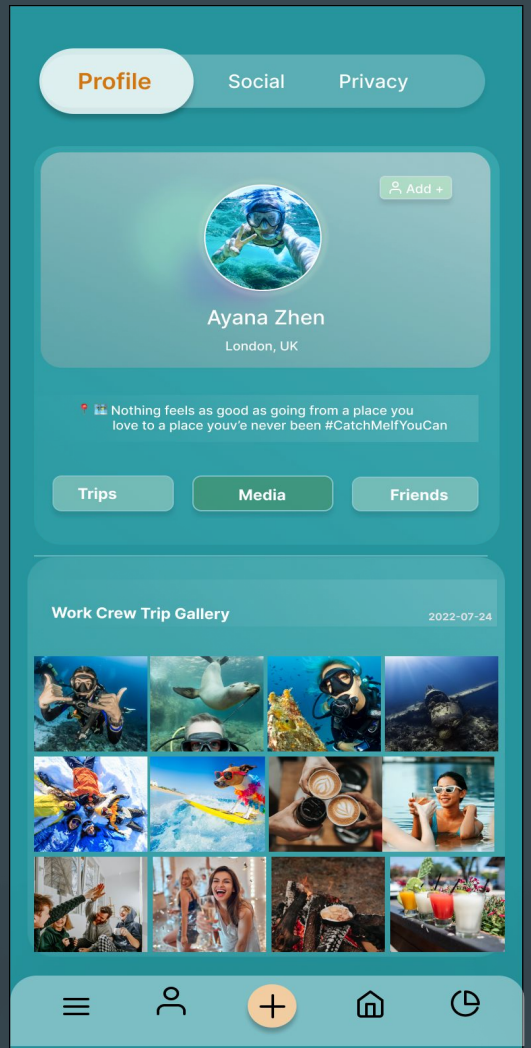
£350

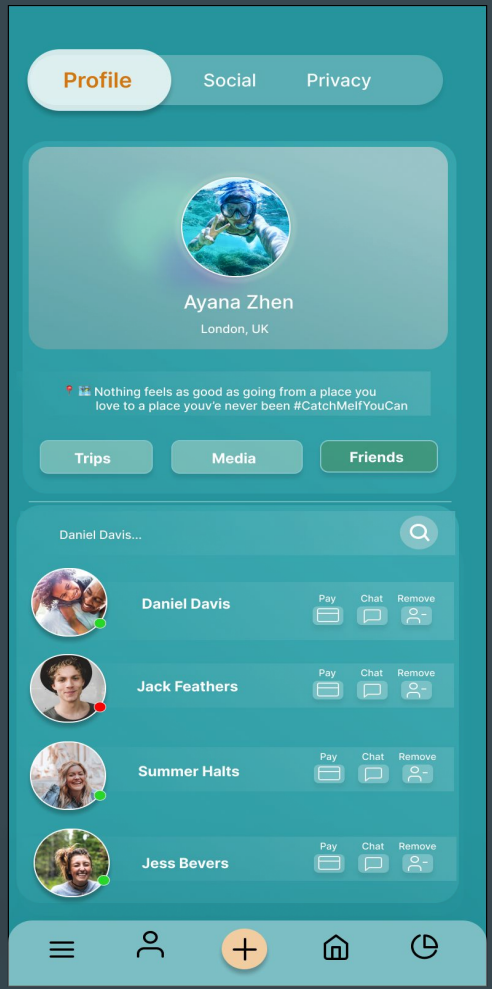
Agadir Day Trip To Legzira & +2

Profile Screens

- View Trips
- Upload Media
- View Gallery
- View friends

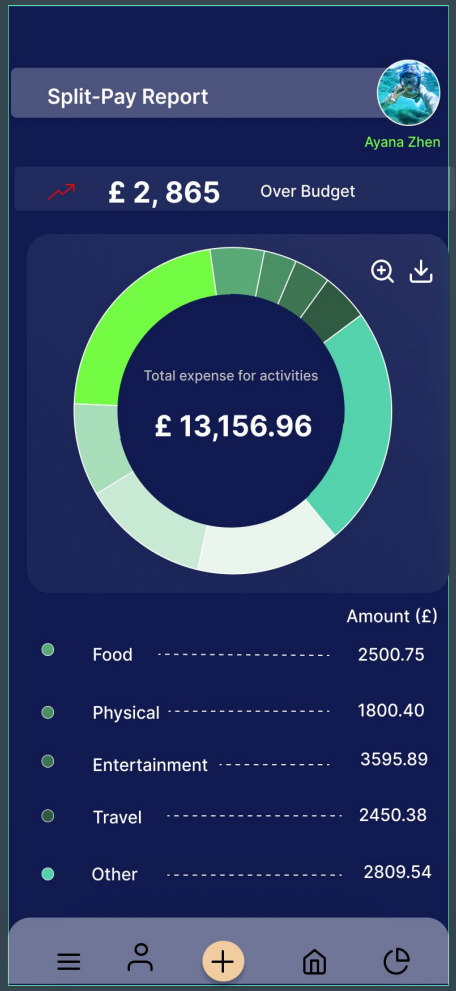


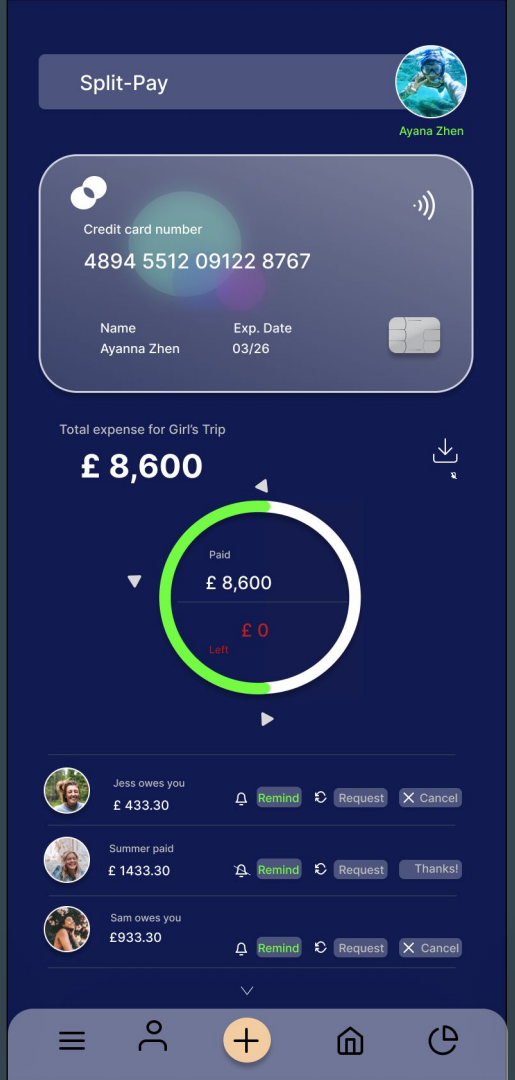


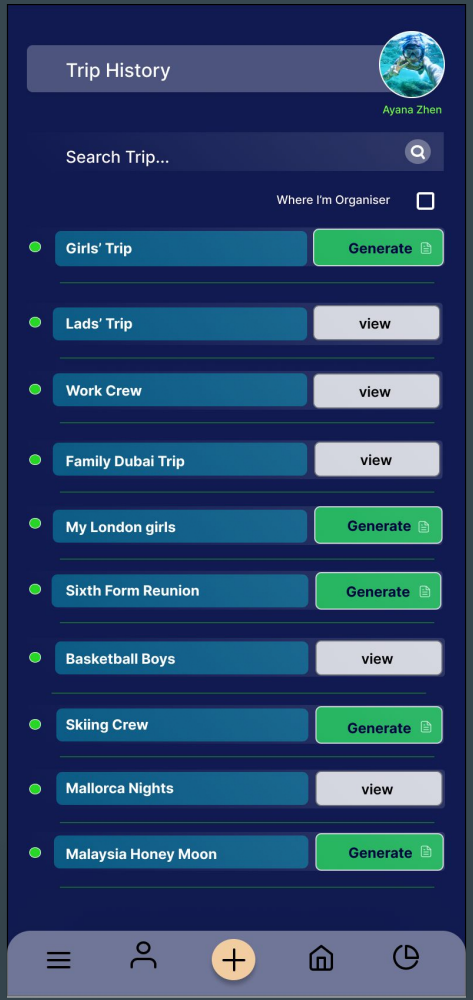


Finance Screens

- Generate Pay Reports
- Request Payment
- Remind Friends
- Improve your finance



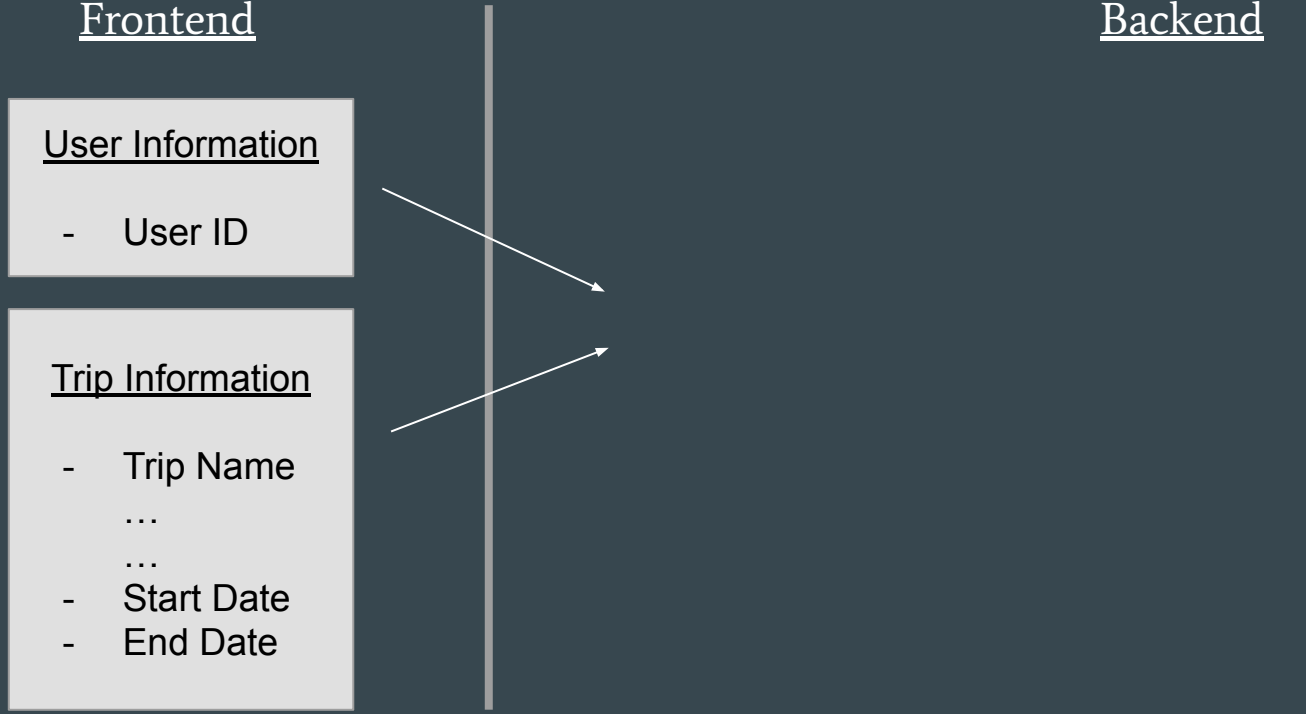




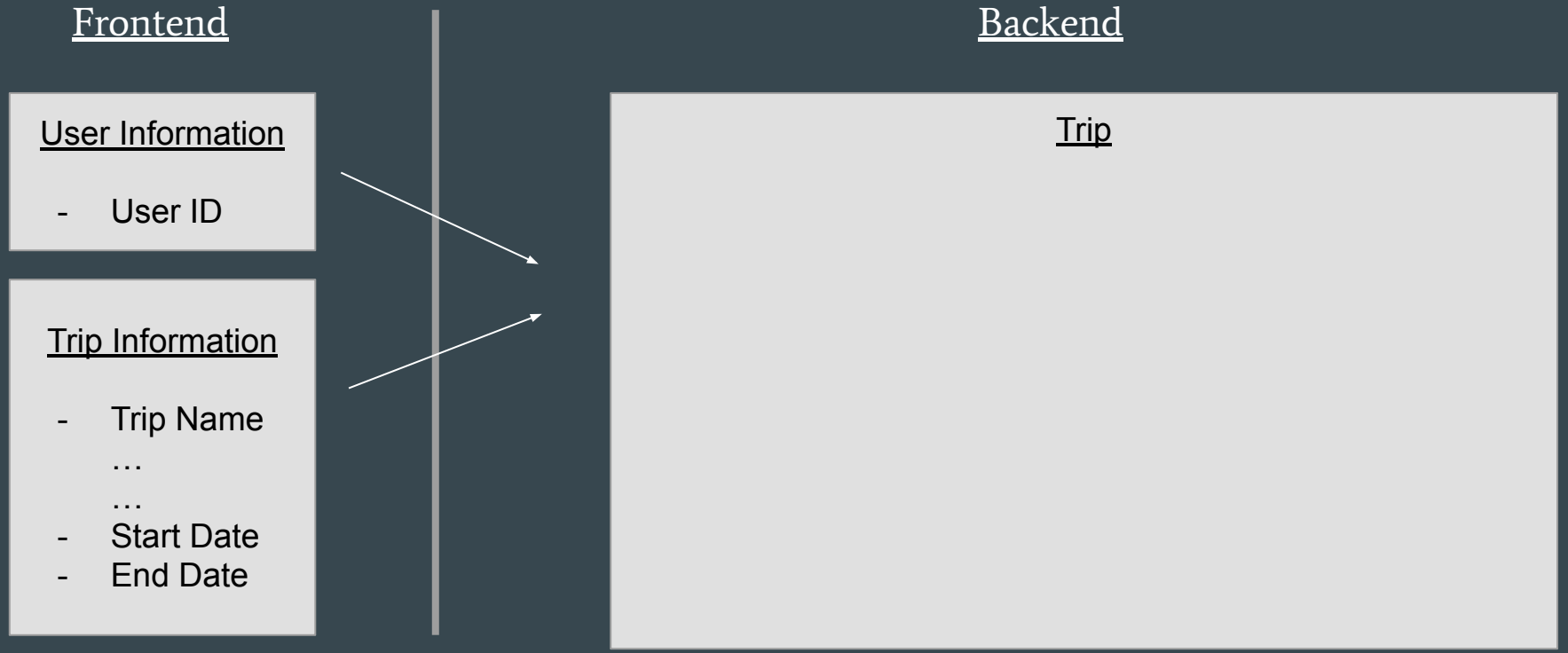
Creating a Trip

Frontend

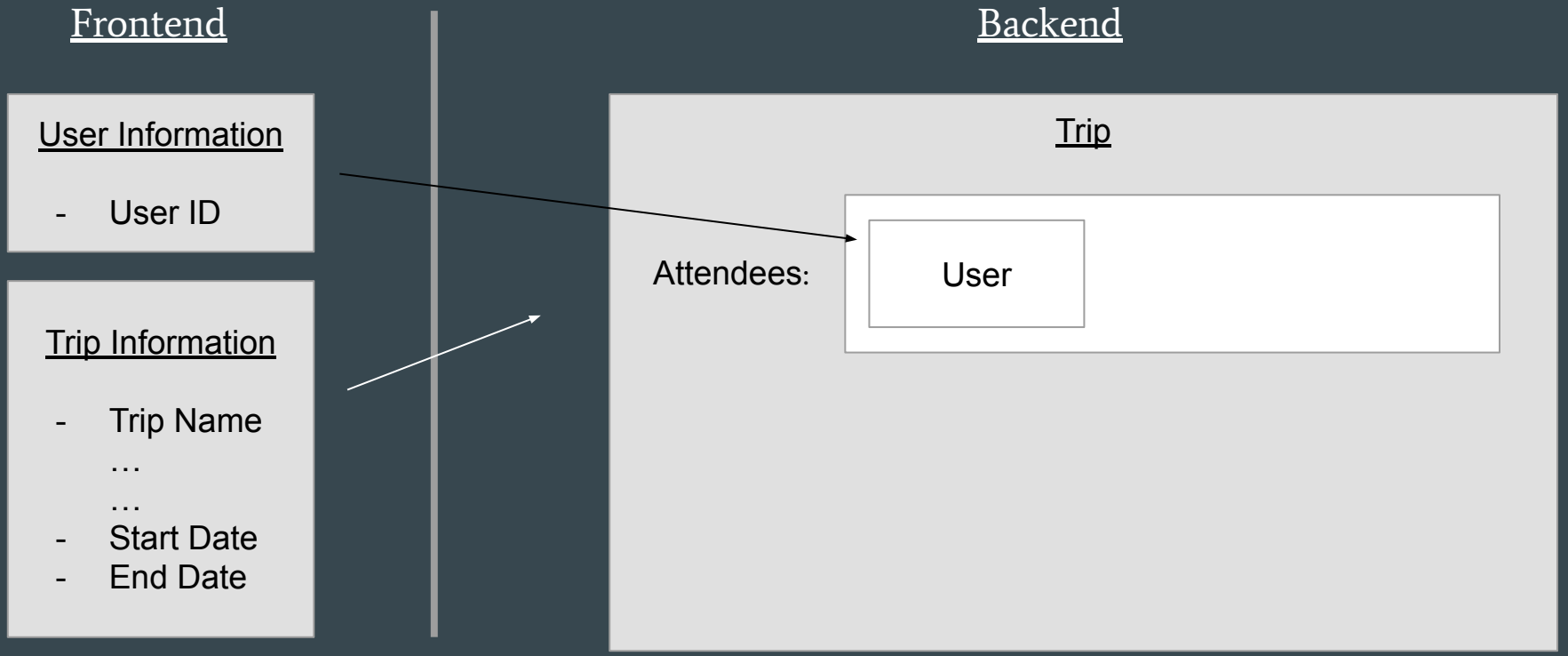
Backend



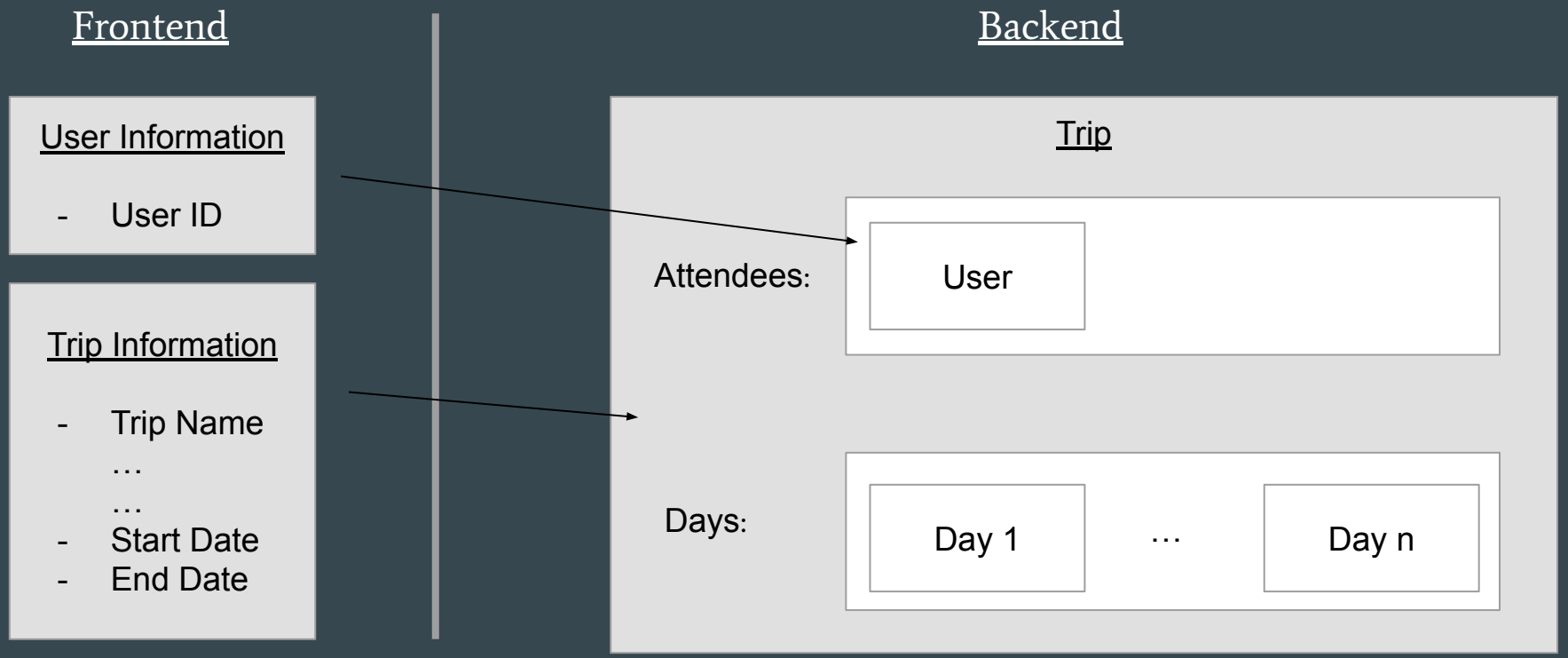
Creating a Trip



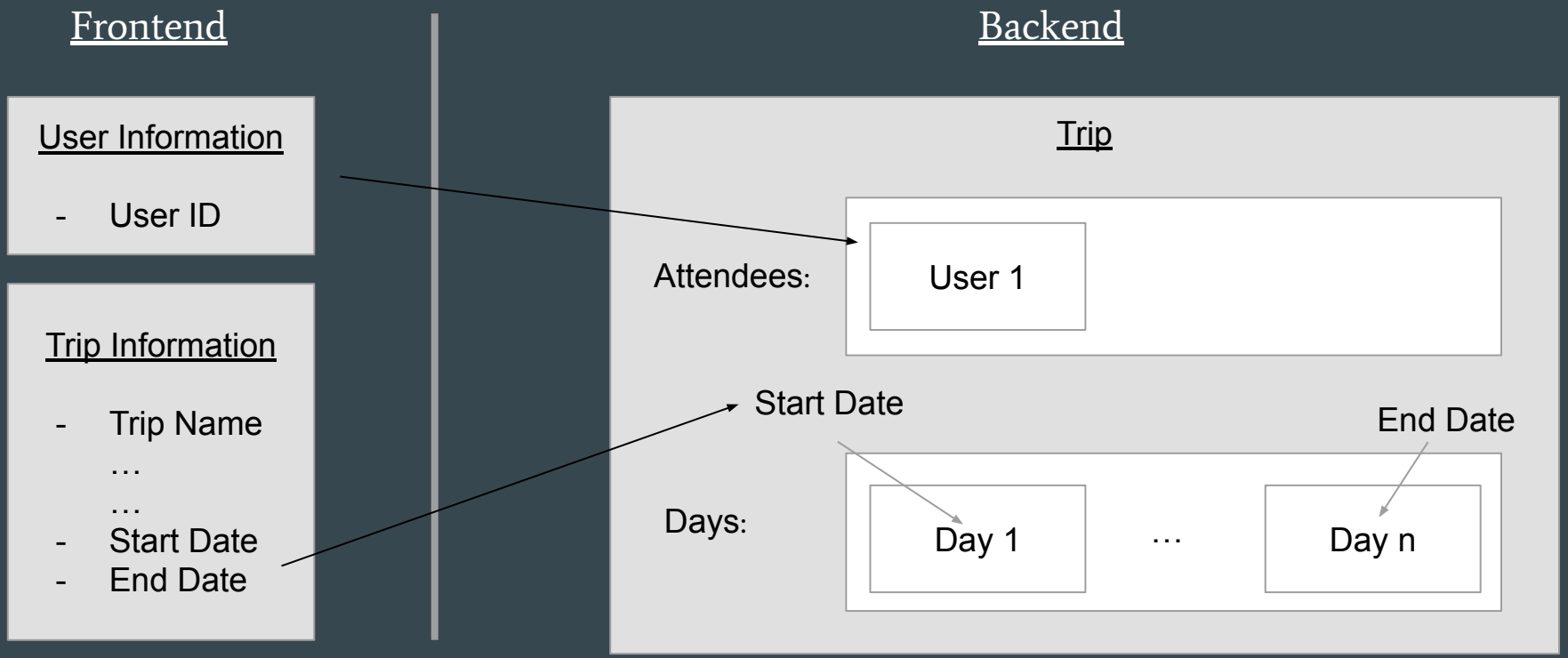
Creating a Trip

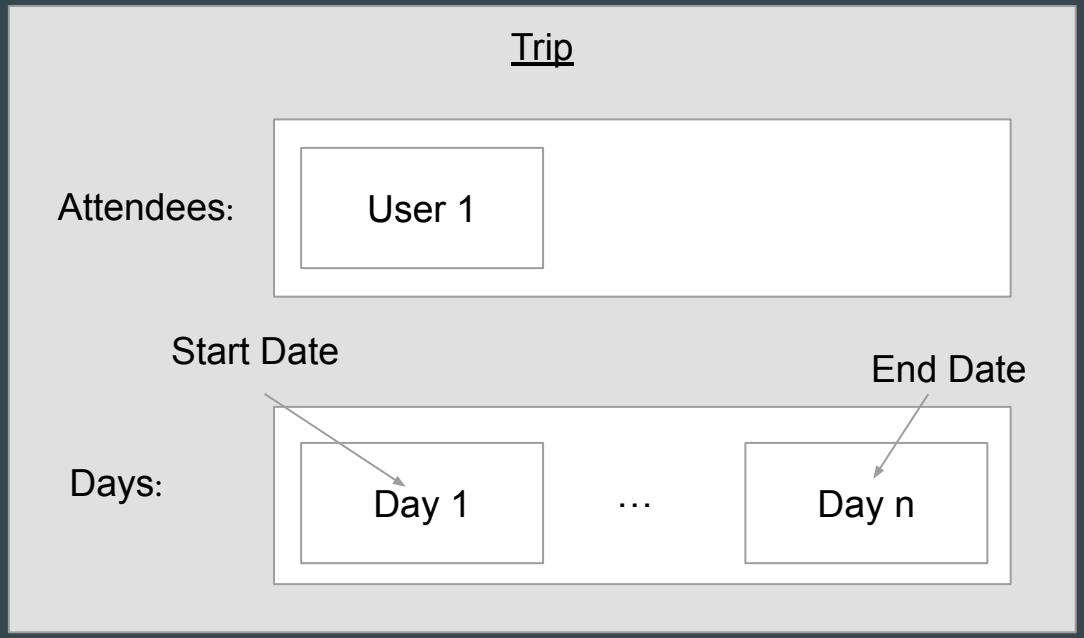


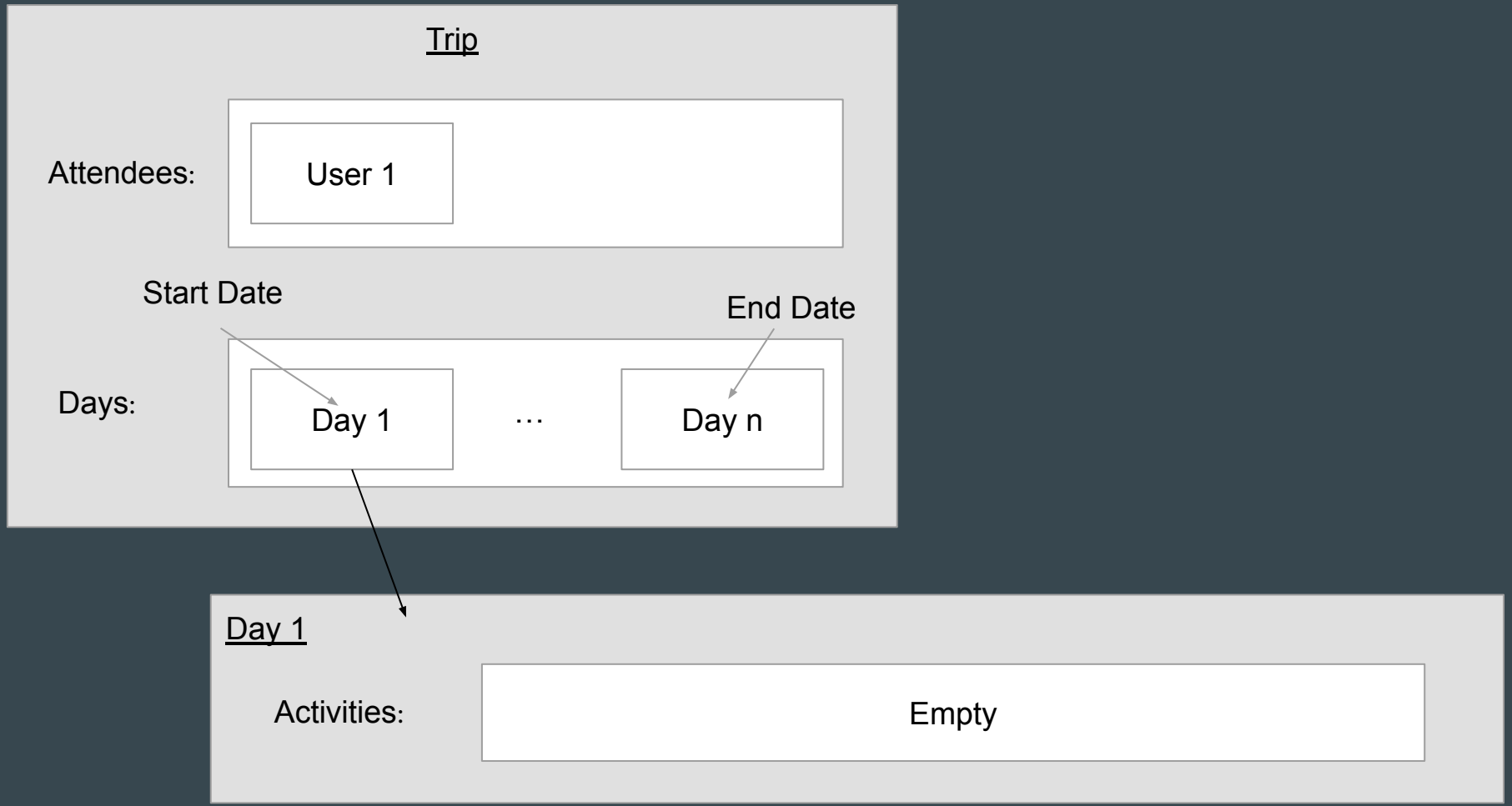
Creating a Trip



Creating a Trip



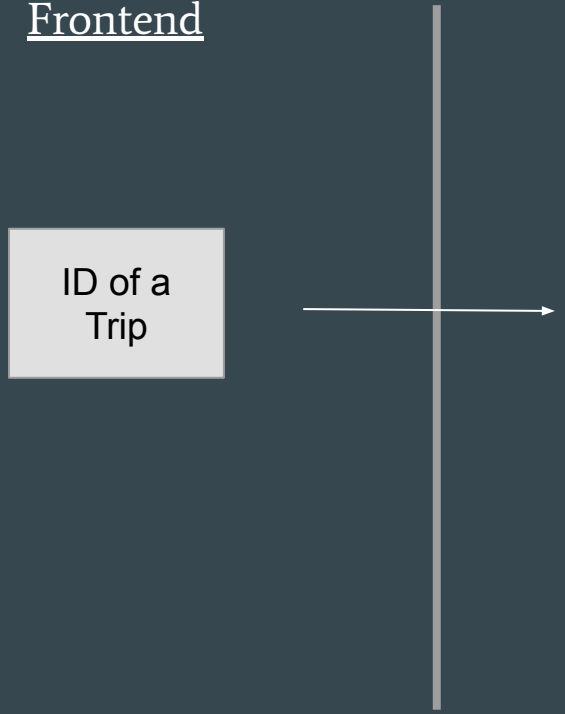

Trip
Attendees:
User 1
Start Date
End Date
Days:
Day 1
…
Day n


Trip
Attendees:
User 1
Start Date
End Date
Days:
Day 1
…
Day n
Day 1
Activities:
Empty

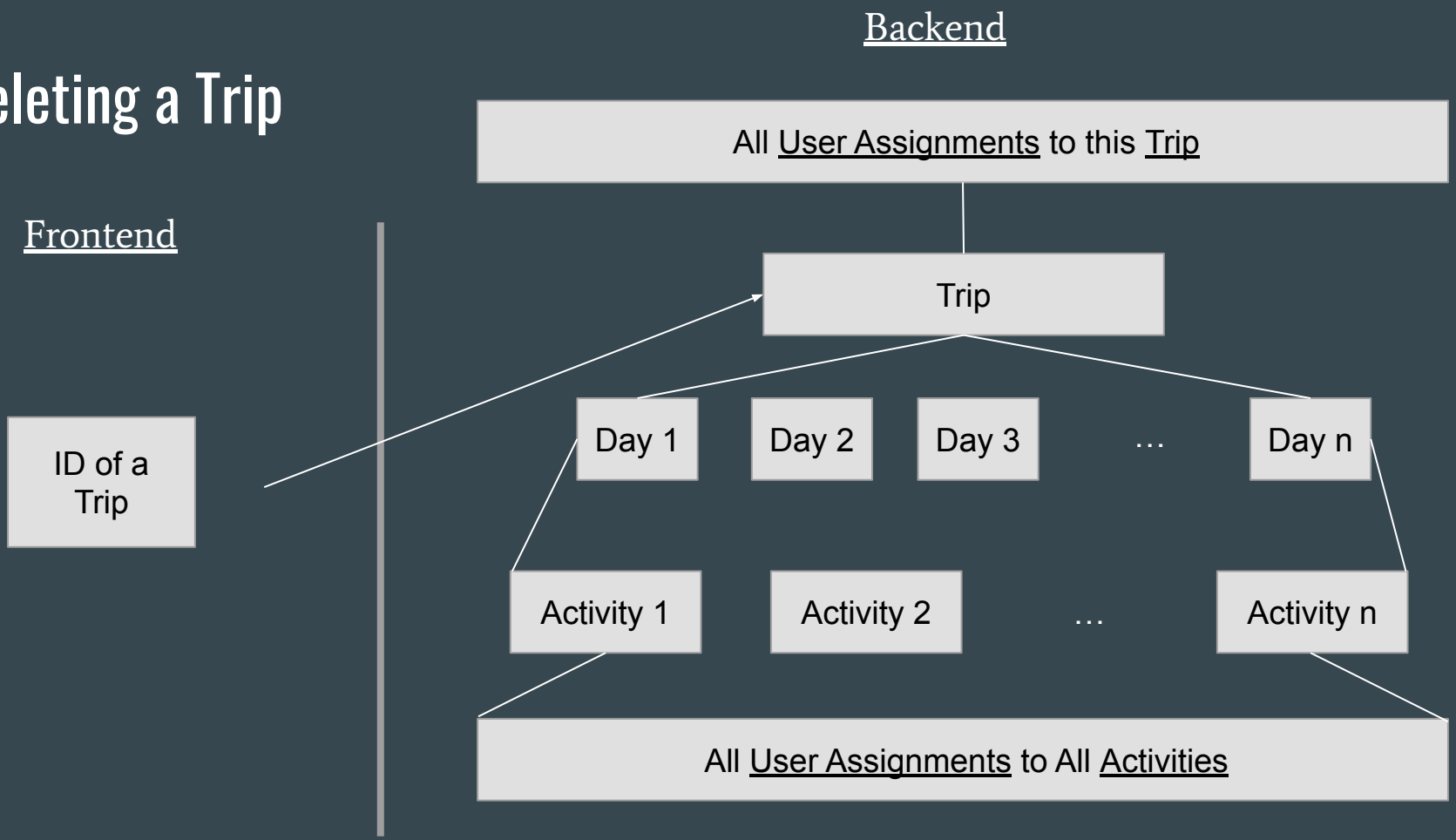
Backend

Deleting a Trip

Frontend



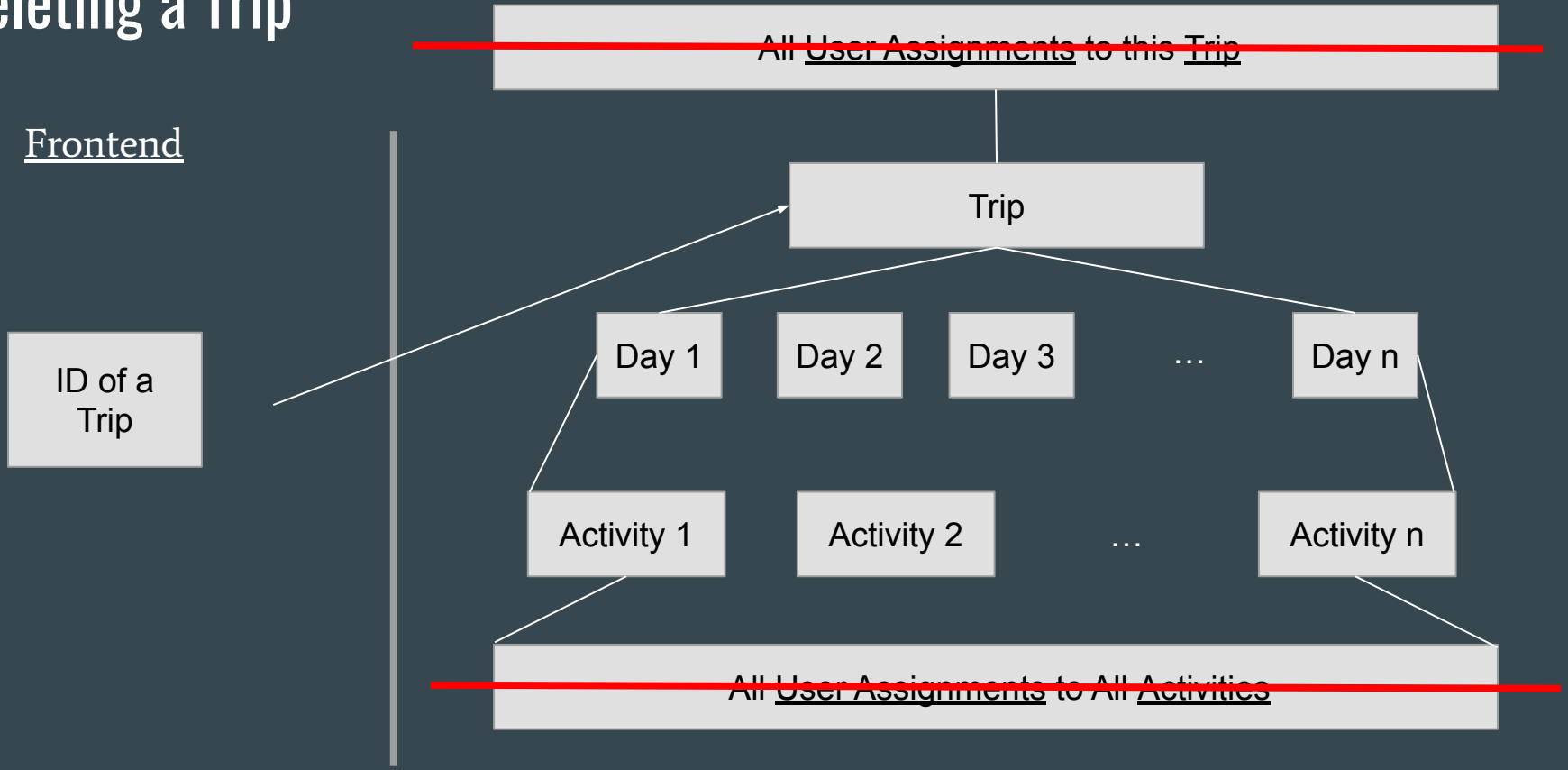
Deleting a Trip



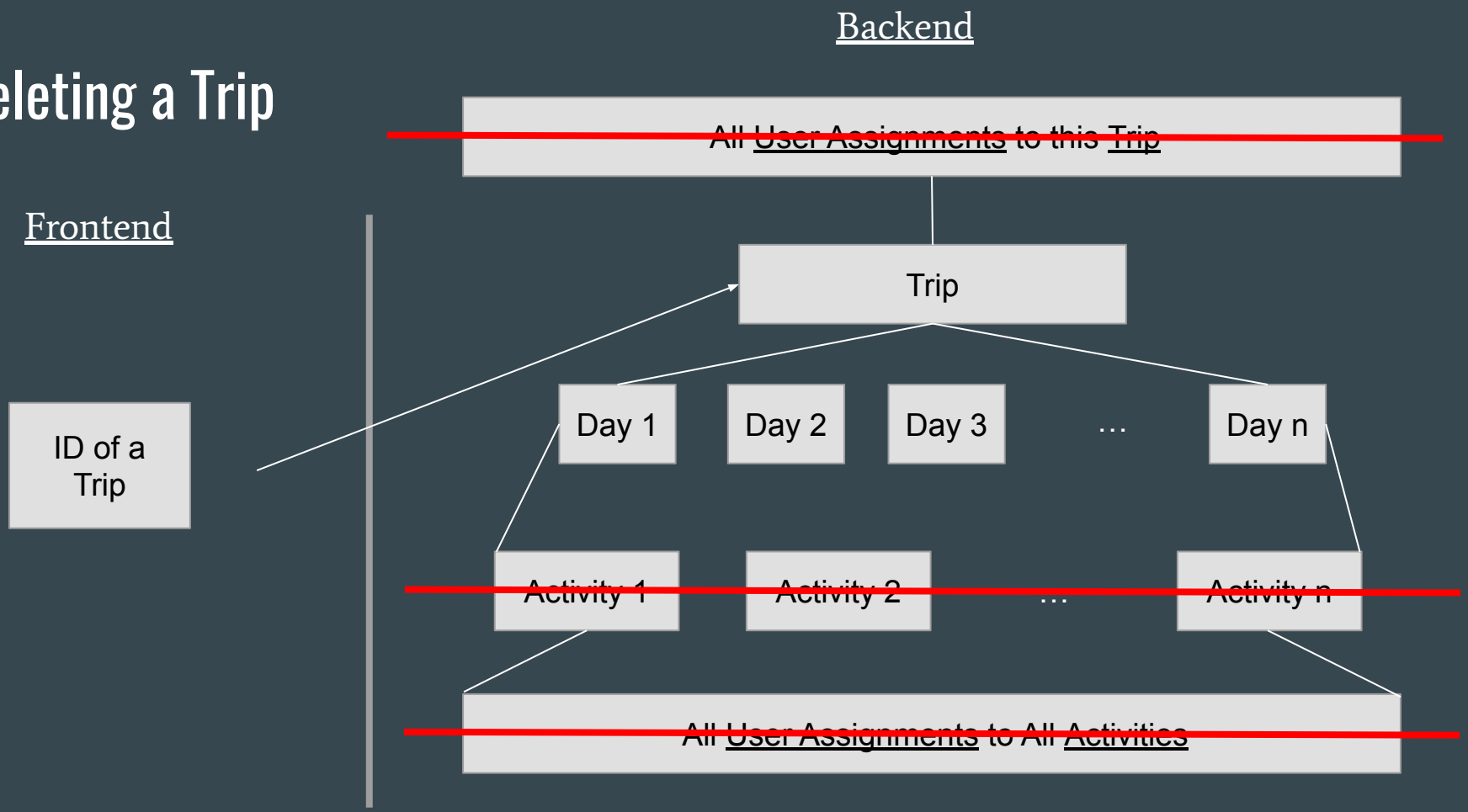
Deleting a Trip

Backend

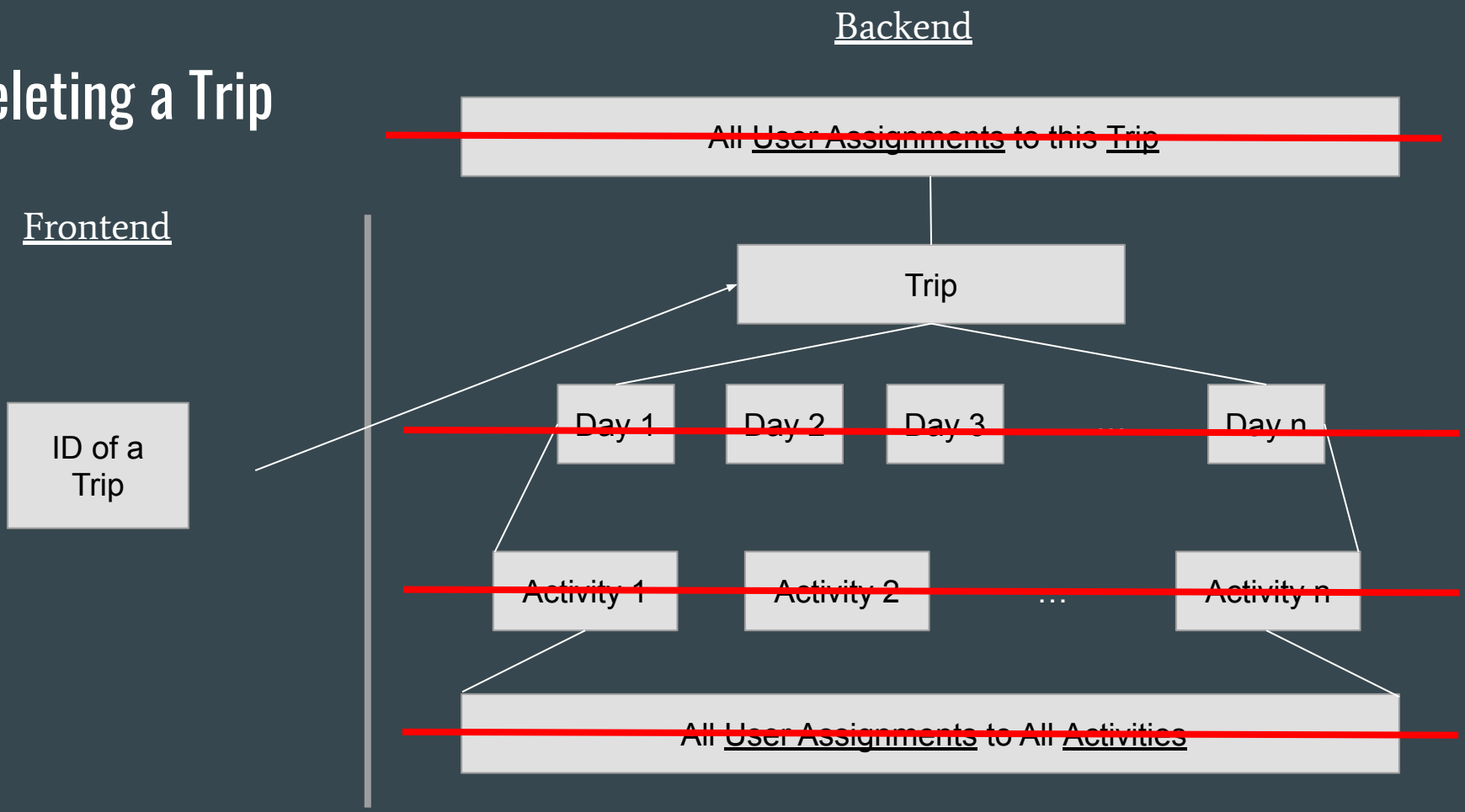
Frontend



Deleting a Trip



Deleting a Trip



Day Code- Ana

Day Class

```
public Day(String name,
           Double budget,
           LocalDateTime date,
           Trip trip) {
    this.name = name;
    this.budget = budget;
    this.date = date;
    this.trip = trip;
    this.dayActivities = Sets.newHashSet();
}
```

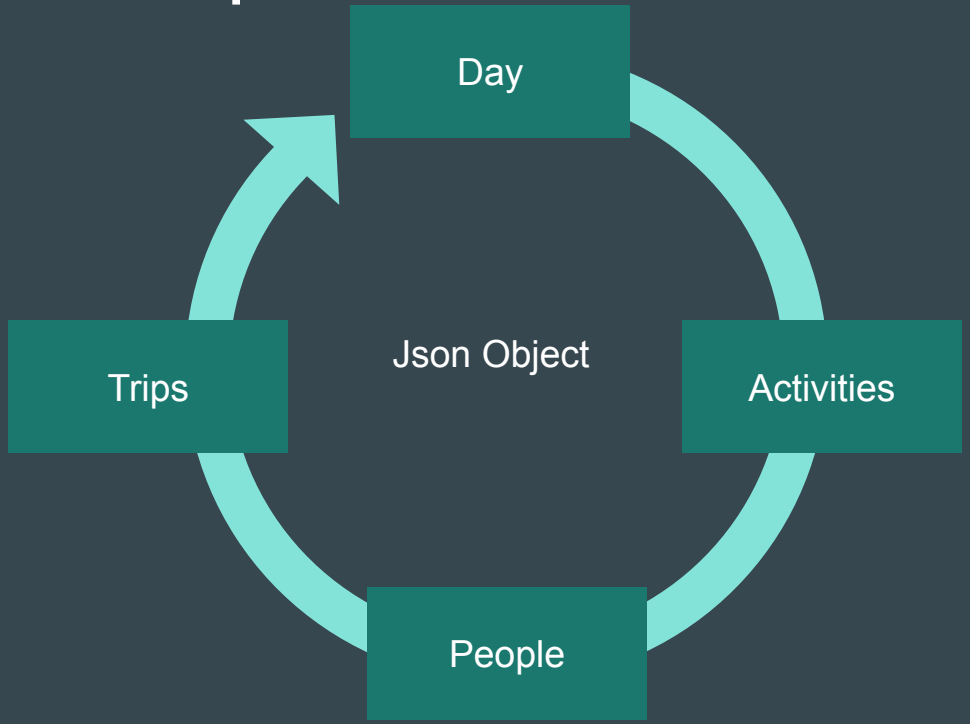
Properties: Trip and day activities

Day DTO

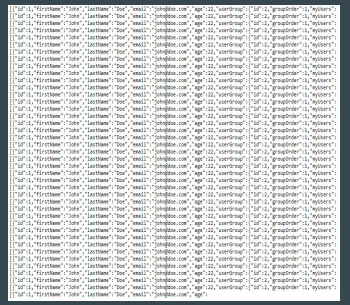
```
private Long id;
    private String name;
    private Double budget;
    private LocalDateTime date;

    public CustomDayDto(Long id,
                        String name,
                        Double budget,
                        LocalDateTime date) {
        this.id = id;
        this.name = name;
        this.budget = budget;
        this.date = date;
    }
```

No trip or Activities properties

Day code - Infinite Loop



Json Object: Infinite Loop



Json Object

Day Activity Code- Jenna

```
public List<PayeeAndPayer> generateActivityCostByUser(Long userID, Long dayActivityID) {
    List<DayActivityAssignment> dayActivityAssignments = dayActivityAssignmentRepository
            .returnActivityAssignmentsByActivityID(dayActivityID);
```

```
@GetMapping("/generateOwingFromTrip")
public List<PayeeAndPayer> generateTripCosts(@RequestParam Long tripID) {
```


-Goes through an activity and finds out who owes what to each person by activity

-Goes through all activities of a trip and adds up what each person owes each other person

-Then cancels out any owing that goes both ways!

Friends Code - Scott FrontEnd

```
const addFriend = async (fA) => {
  const currentUserUsername = "thelifeandtimesofnaeem"
  try {
    const friends = await axios
        .post(`http://localhost:8080/friend/addFriend/${currentUserUsername}/${fA.username}`)
        const data = friends.data;
        setFriends(data)
        document.getElementById(`${fA.id}addFriendBtn`).innerHTML = "Added";
  } catch (err) {
    setFriends([])
    document.getElementById(`${fA.id}friendBTNclick`).innerHTML = `${err.response.data.message}`;
  }
}
```

Add friend

- Made asynchronous so addFriend will only load once the searchUser method declared earlier loads the list of users
- addFriend called onClick, linked to each userId, which will prompt either an error message to load below the profile if unsuccessful or the button will be altered to say “added” if successful

```
const searchUsers = async (e) => {
  const username = e.target.value;
  try {
    const users = await axios.get(`http://l
    const data = users.data;
    setUsers(data)
```

```
<div id="users-list">
    {
      users.map((user, index) => {
        return (
          <div key={index} id="one-user">
            <div id="user-p-container">
            <div className="search-user-image-container">
              <img src={user.imgURL} alt="" className="user-image-style"/>
            </div>
            <h1 id="headingUser">{user.firstname} {user.lastname}</h1>
              <p id="pUsername">{user.username}</p>
            <button id={`${user.id}addFriendBtn`} className="addFriendBtn" onClick={() => addFriend(user)}>Add Friend</button>
            </div>
            <div className="err-return">
            <p id={`${user.id}friendBTNclick`} className="friendBtnclick"></p>
            </div>
          </div>
        )
      })
```

In the user list div, each item “one-user” is mapped by the data stored in users with a unique key (index) corresponding to each user

Friends Code - BackEnd

```java
@PostMapping("/addFriend/{currentUserUsername}/{friendToAddUsername}")
public String addFriend(@PathVariable("currentUserUsername") String currentUserUsername,
                        @PathVariable("friendToAddUsername") String friendToAddUsername

                        ) throws Exception{
    return friendService.addFriend(currentUserUsername, friendToAddUsername);
}
```


```java
public String addFriend(String currentUserUsername, String friendToAddUsername) {

    ApplicationUser userC = userRepository.getUserByUserName(currentUserUsername);
    ApplicationUser userF = userRepository.getUserByUserName(friendToAddUsername);

    Friend friendTA = new Friend(userC, userF, currentUserUsername, friendToAddUsername);

    Friend existingUsernameCombo1 = friendRepository.findFriendPairUsername(currentUserUsername, friendToAddUsername);
    Friend existingUsernameCombo2 = friendRepository.findFriendPairUsername(friendToAddUsername, currentUserUsername);

    if (userF == null) {
        throw new InvalidDataAccessApiUsageException("User does not exist! Check the details are right and try again, " + userC.getFirstname() + " " + userC.getLastname());
    } else if (userC == userF) {
        throw new InvalidDataAccessApiUsageException("Oi, big boy! You can't be friends with yourself! " + userC.getFirstname() + " " + userC.getLastname());
    } else if (existingUsernameCombo1 == null && existingUsernameCombo2 == null) {
        friendRepository.save(friendTA);
    } else {
        throw new InvalidDataAccessApiUsageException("You're already friends with " + userF.getFirstname() + "!");
    }

    return "You have added " + userF.getFirstname() + " " + userF.getLastname() + " to your friend list, " + userC.getFirstname() + " " + userC.getLastname();
}
```

Add Friend:

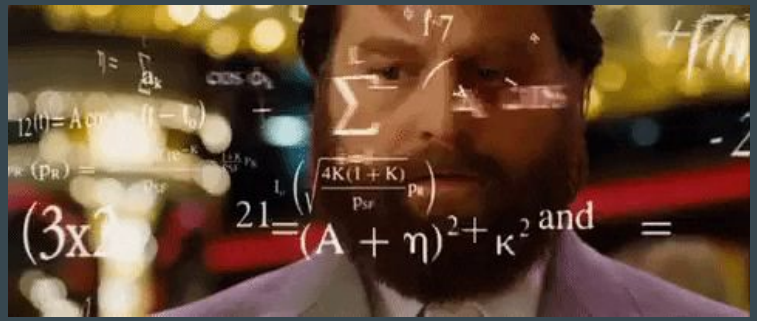
- Define PathVariable to connect service
- Call addFriend method
- User data in order of person using it and person on who they want to add
- Friend repository calls a method to check if friend pair exists.
- Added extra functionality to prevent user from adding themselves
- Prevented duplicate friends being added.

- LIVE DEMO- Jenna

Challenges & Bugs

CSS Styling...

i.e. THE BANE OF OUR EXISTENCE



The infinite JSON loop...

Data Loader

Keeping the project in scope

Future Recommendations

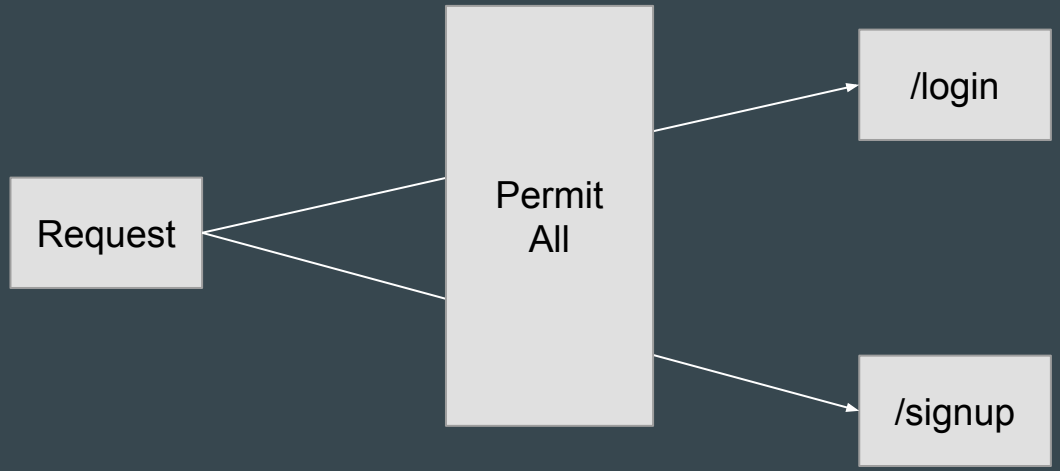
Delete Mappings

- Enable removing friends in your user profile
- Removal of activities from any day

Authentication

- We would have made more of our code dynamic, e.g.:
  - Removing the hard-coded “logged-in” user
  - Integrated the log-in feature with the app so that, upon submitting the log-in form, it would load the corresponding user profile

User Authentication Code




```
http
        .authorizeRequests() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
        .antMatchers(HttpMethod.POST, ...antPatterns: "/authenticate") ExpressionUrlAuthorizationConfigurer<...>.AuthorizedUrl
        .permitAll() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
        .antMatchers(HttpMethod.POST, ...antPatterns: "/sign_up") ExpressionUrlAuthorizationConfigurer<...>.AuthorizedUrl
        .permitAll() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
        .antMatchers(HttpMethod.GET, ...antPatterns: "/test") ExpressionUrlAuthorizationConfigurer<...>.AuthorizedUrl
        .permitAll();
```

/login

```
try {
    authenticationManager.authenticate(new UsernamePasswordAuthenticationToken(
            username,
            password
    ));
    return authenticationUserDetailsService.loadUserByUsername(username);
}
```

/login

```
try {
    authenticationManager.authenticate(new UsernamePasswordAuthenticationToken(
            username,
            password
    ));
    return authenticationUserDetailsService.loadUserByUsername(username);
}
```

```
Set<SimpleGrantedAuthority> authorities =
        user.getOwner() ? APP_OWNER.getGrantedAuthorities() :
        user.getAdmin() ? APP_ADMIN.getGrantedAuthorities() :
                          APP_USER.getGrantedAuthorities();

return new AuthenticationUserImplUserDetails(
    user.getUsername(),
    user.getPassword(),
    authorities,
        isAccountNonExpired: true,
        isAccountNonLocked: true,
        isCredentialNonExpired: true,
        isEnabled: true
);
```

/login

```
try {
    authenticationManager.authenticate(new UsernamePasswordAuthenticationToken(
            username,
            password
    ));
    return authenticationUserDetailsService.loadUserByUsername(username);
}
```

```
Set<SimpleGrantedAuthority> authorities =
        user.getOwner() ? APP_OWNER.getGrantedAuthorities() :
        user.getAdmin() ? APP_ADMIN.getGrantedAuthorities() :
                          APP_USER.getGrantedAuthorities();

return new AuthenticationUserImplUserDetails(
    user.getUsername(),
    user.getPassword(),
    authorities,
        isAccountNonExpired: true,
        isAccountNonLocked: true,
        isCredentialNonExpired: true,
        isEnabled: true
);
```

```
APP_ADMIN(Sets.newHashSet(
            APP_ADMIN_READ_SELF,
            APP_ADMIN_WRITE_SELF,
            APP_USER_READ_ALL,
            APP_USER_WRITE_ALL
)),
```

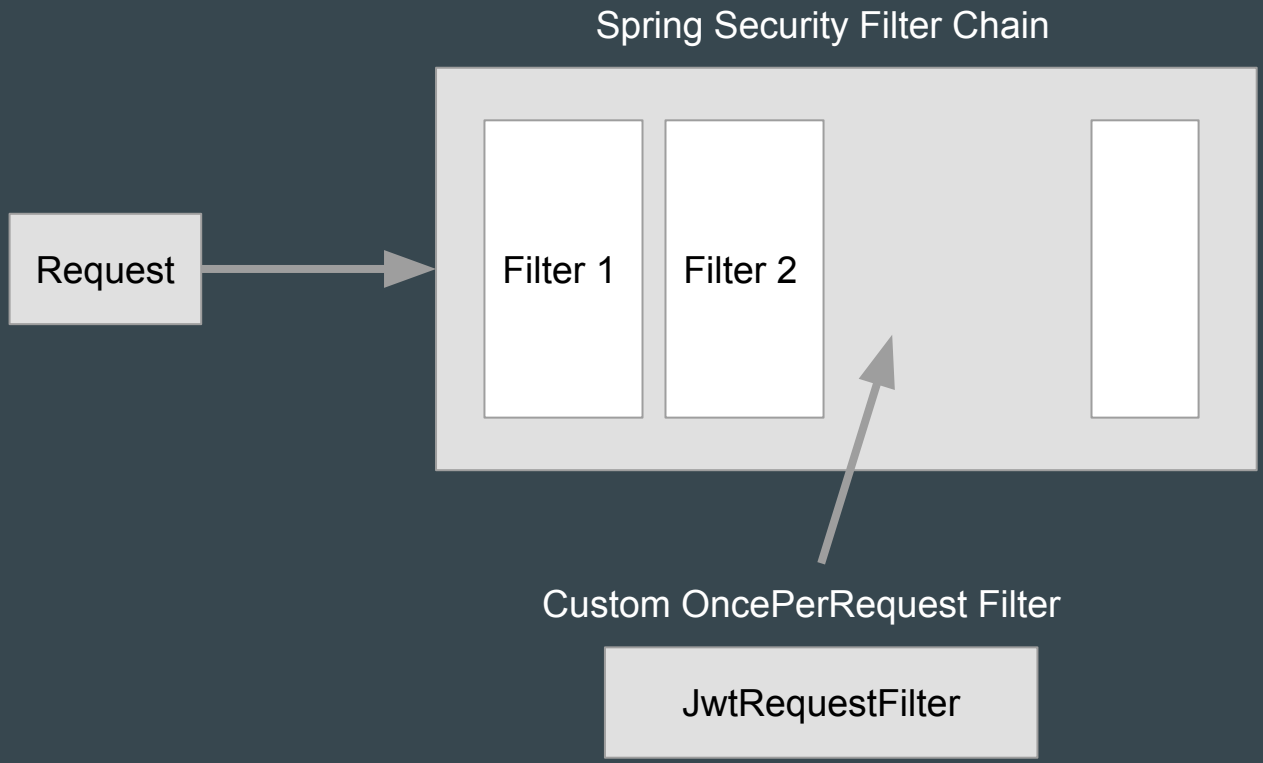
```
return Jwts
        .builder()
        .setHeaderParam( s: "typ",  o: "JWT")
        .setSubject(userDetails.getUsername())
        .claim( s: "authorities", userDetails.getAuthorities())
        .setIssuedAt(new Date())
        .setExpiration(java.sql.Date.valueOf(LocalDate.now().plusDays(jwtConfig.getTokenExpirationAfterDays())))
        .signWith(Keys.hmacShaKeyFor(jwtConfig.getSecretKey().getBytes()))
        .compact();
```


```
{
    "success": true,
    "message": "Token generated",
    "payload": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.
        eyJzdWIiOiJhZG1pbjEiLCJhdXRob3JpdGllcyI6W3siYXV0aG9yaXR5Ij
        oiYXBwX3VzZXI6cmVhZF9hbGwifSx7ImF1dGhvcml0eSI6ImFwcF91c2Vy
        OndyaXRlX2FsbCJ9LHsiYXV0aG9yaXR5IjoiYXBwX2FkbWluOnJlYWRfc2
        VsZiJ9LHsiYXV0aG9yaXR5IjoiYXBwX2FkbWluOndyaXRlX3NlbGYifV0s
        ImlhdCI6MTY1NzIyNjE1MywiZXhwIjoxNjU3MjM4NDAwfQ.
        Sqodc2iFv1FyHKX0k6ErdXhWZlkfUYnJX0DI7NdTj0q6G_0L6TmcNHWWGM
        iEkH88nfuCdnGaMO7s-OxzAmoCcQ"
}
```

```
{
  "success": true,
  "message": "Token generated",
  "payload": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.
      eyJzdWIiOiJhZG1pbjEiLCJhdXRob3JpdGllcyI6W3siYXV0aG9yaXR5Ij
      oiYXBwX3VzZXI6cmVhZF9hbGwifSx7ImF1dGhvcml0eSI6ImFwcF91c2Vy
      OndyaXRlX2FsbCJ9LHsiYXV0aG9yaXR5IjoiYXBwX2FkbWluOnJlYWRfc2
      VsZiJ9LHsiYXV0aG9yaXR5IjoiYXBwX2FkbWluOndyaXRlX3NlbGYifV0s
      ImlhdCI6MTY1NzIyNjE1MywiZXhwIjoxNjU3MjM4NDAwfQ.
      Sqodc2iFv1FyHKX0k6ErdXhWZlkfUYnJX0DI7NdTj0q6G_0L6TmcNHWWGM
      iEkH88nfuCdnGaMO7s-OxzAmoCcQ"
}
```



```
{
  "sub": "admin1",
  "authorities": [
    {
      "authority": "app_user:read_all"
    },
    {
      "authority": "app_user:write_all"
    },
    {
      "authority": "app_admin:read_self"
    },
    {
      "authority": "app_admin:write_self"
    }
  ],
  "iat": 1657226153,
  "exp": 1657238400
}
```


Spring Security Filter Chain
Request
Filter 1
Filter 2
Custom OncePerRequest Filter
JwtRequestFilter

```
Jws<Claims> claimsJws = Jwts.parser()
                        .setSigningKey(Keys.hmacShaKeyFor(jwtConfig.getSecretKey().getBytes()))
                        .parseClaimsJws(token);
```

```
if (expiration.before(new Date())) {
    filterChain.doFilter(request, response);
    return;
}
```

Set of authorities

```
SecurityContextHolder.getContext().setAuthentication(authentication);
```

```
} catch (JwtException e) {
    throw new IllegalStateException(String.format("Token %s is invalid.", token));
}
```

```
@GetMapping("/get_all")
@PreAuthorize("hasAuthority('app_user:read_all')")
```

```
import React, { useState } from 'react'


const [authenticatedUser, setAuthenticatedUser] = useState(null)
```

Thanks for listening