David Goggins Society – Design Document

Group Members

- Tom Materne – a1825150
- Jack Scholten – a1826009
- Julian Lovell – a1829617
- Tom Gehling – a1838949

Contents

Project Specifications

A Vue3 website that is able to manage university clubs. This website will need to handle membership signup for students, club management and administration. The clubs should also be able to create and manage events. These events should then be on the member's home page. Their homepage should also include club updates. The members will have different roles within each club.

Members will create and authenticate their account with an email and password. This should be stored in a database. The account should be able to link social media accounts, so that they can login with those instead in the future. The members should be able to view event updates and see upcoming events in a calendar or list type view. The members should also have a page to view the club announcements/posts.

Clubs are managed by members with specific roles. A club manager can manage other users roles & other club information. Clubs can make posts to the members. Clubs can be open or invite only. Clubs can create events, and the RSVP status of the members should be visible.

A system admin is a privileged member who can manage/delete club and delete users. Can change other users to admins as well.

Application Name & Style

Plooto

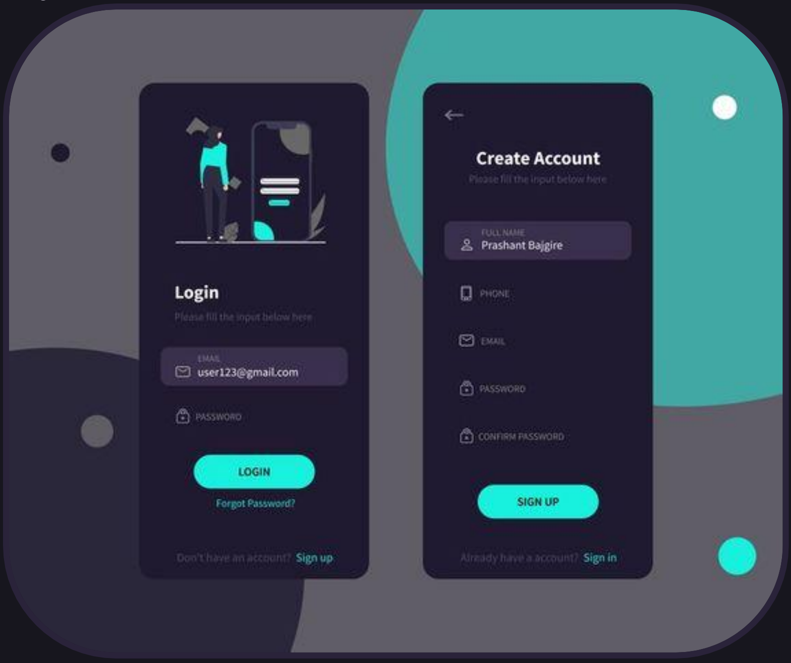
This application name was inspired by the famous company Mars, which named itself after a planet, so we've decided to follow suit and name our website after the infamous non-planet, Pluto. We expect it to inspire a sense of curiosity. When we brainstormed design ideas, we wanted to keep space as a key part of our identity, so the purple theme stems from our visions of space.

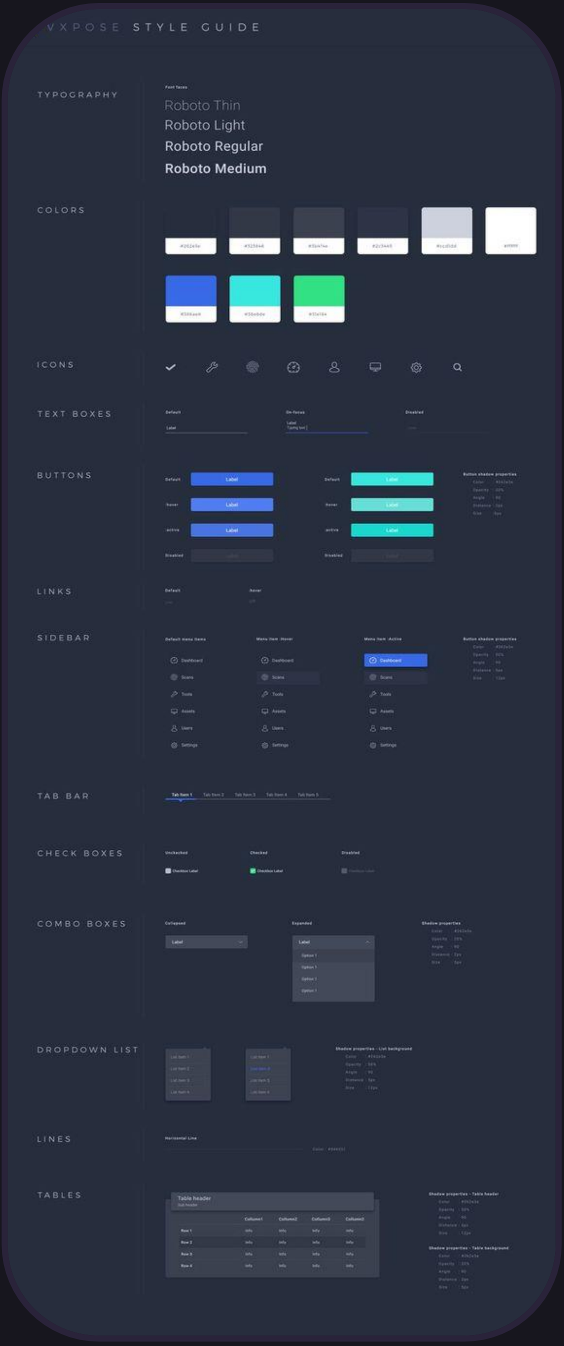
Figure 1 https://wallup.net/wp-content/uploads/2016/01/9387-space-planet-space_art-purple.jpg

Figure 2 https://wallpaper.dog/large/10701736.jpg

Research & Design

Style



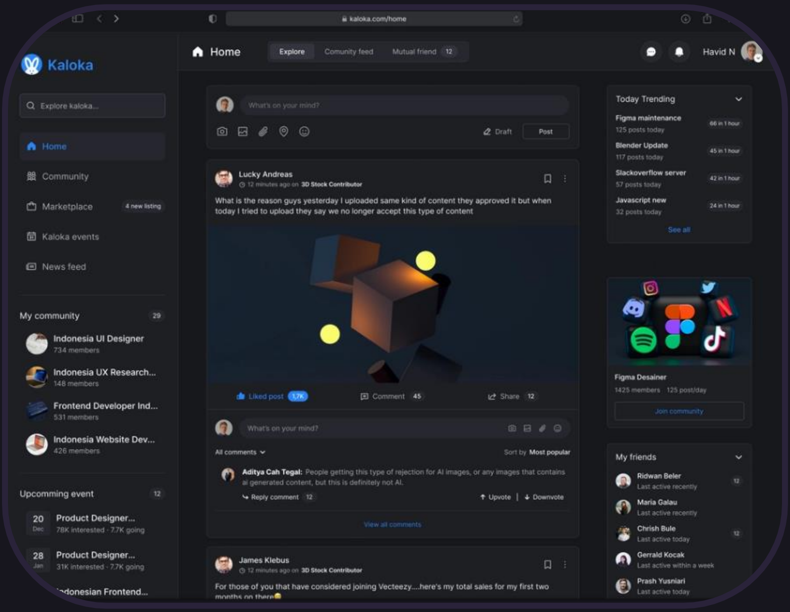


This mobile login/create account page has really nice buttons and input fields. Radius placed on all edges to maintain clean looking design.

Minimal colour scheme, fonts and overall design to make it simple and easy to use as well as being appealing to look at.

Purple as the primary colour on the website to give a unique look from other similar sites, darker shade provides a 'dark mode' style colour scheme which is the aim. Orange is a contrasting colour to purple so this will be used selectively throughout the site as an accent colour.

Structure



Lucky Andreas

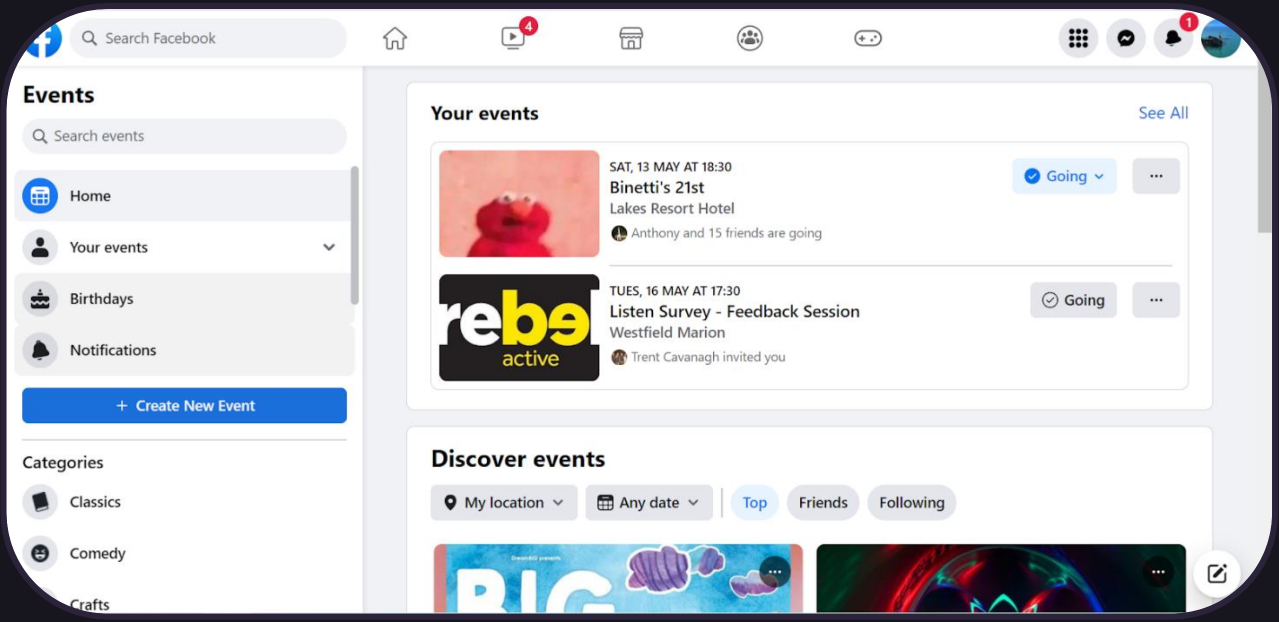
What is the reason guys yesterday I uploaded same kind of content they approved it but when today I tried to upload they say we no longer accept this type of content

Aditya Cah Tegal: People getting this type of rejection for AI images, or any images that contains ai generated content, but this is definitely not AI.

James Klebus

For those of you that have considered joining Vecteezy....here's my total sales for my first two months on there

From this pinterest image, we like the structure of the navigation, with the sidebar that has links to the different pages, and the top bar with user image and notifications & settings buttons. The layout of the timeline was also very familiar, most modern social media sites have similar structure, with endless scrolling of content in their "feed". This is where we will display upcoming events and posts.

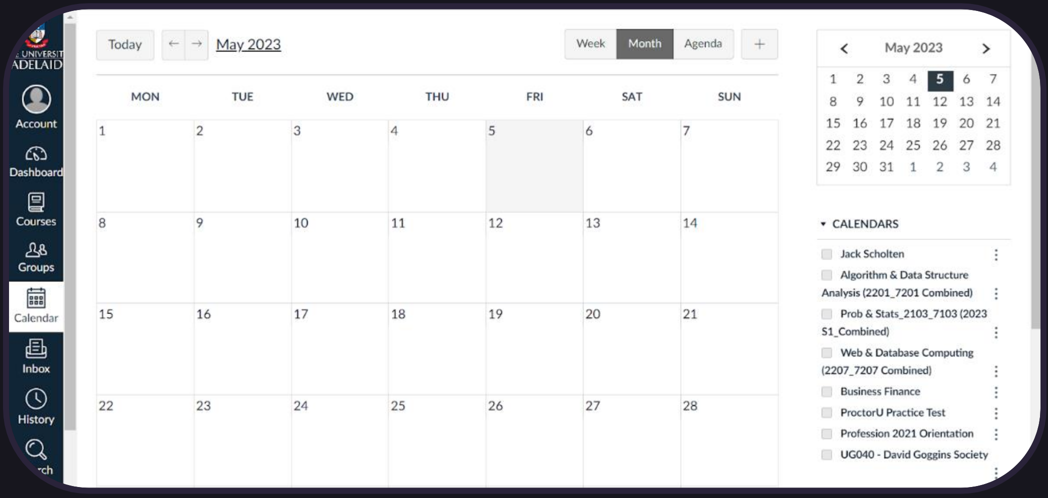


SAT, 13 MAY AT 18:30
Binetti's 21st
Lakes Resort Hotel
Anthony and 15 friends are going

TUES, 16 MAY AT 17:30
Listen Survey - Feedback Session
Westfield Marion
Trent Cavanagh invited you

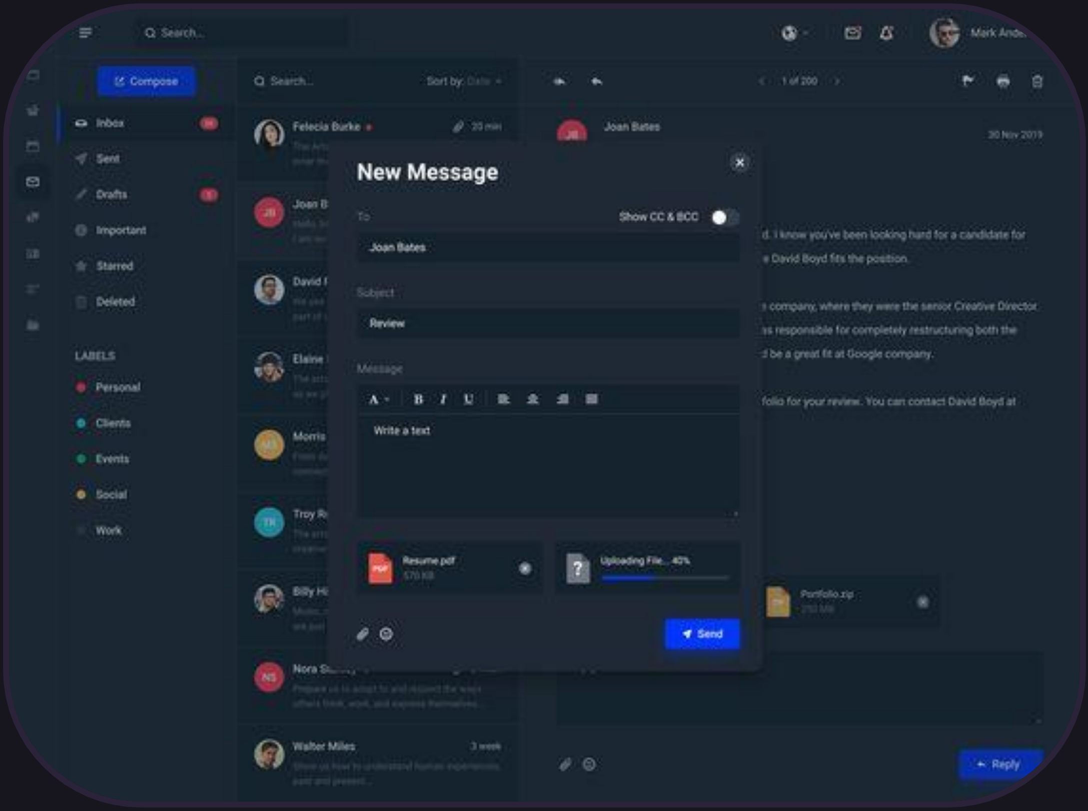
The facebook events page has a similar structure in regards to events, with the sidebar for navigation and the topbar for login information & notifications. We will adapt this layout with the "your events" and "discover events" into our own website.

Features


ADELAIDE
Account
Dashboard
Courses
Groups
Calendar
Inbox
History
Today
May 2023
Week
Month
Agenda
MON
TUE
WED
THU
FRI
SAT
SUN
May 2023
CALENDARS
Jack Scholten
Algorithm & Data Structure Analysis (2201_7201 Combined)
Prob & Stats_2103_7103 (2023 S1_Combined)
Web & Database Computing (2207_7207 Combined)
Business Finance
ProctorU Practice Test
Profession 2021 Orientation
UG040 - David Goggins Society

Website supports a calendar for scheduling and displaying. The user can access this calendar to view any of their own upcoming events.

Plooto will also implement manageable notifications per club and event, that the user can opt in and out on. As is present on the figma design, we have administration for clubs, posts and events.

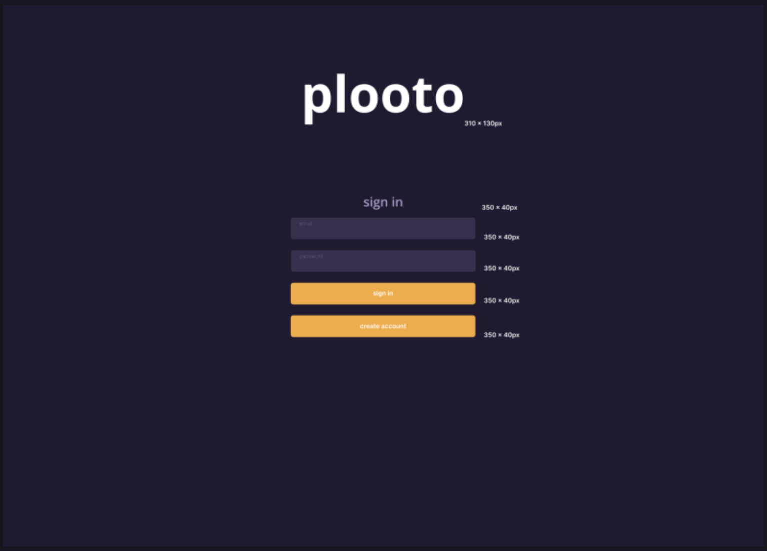


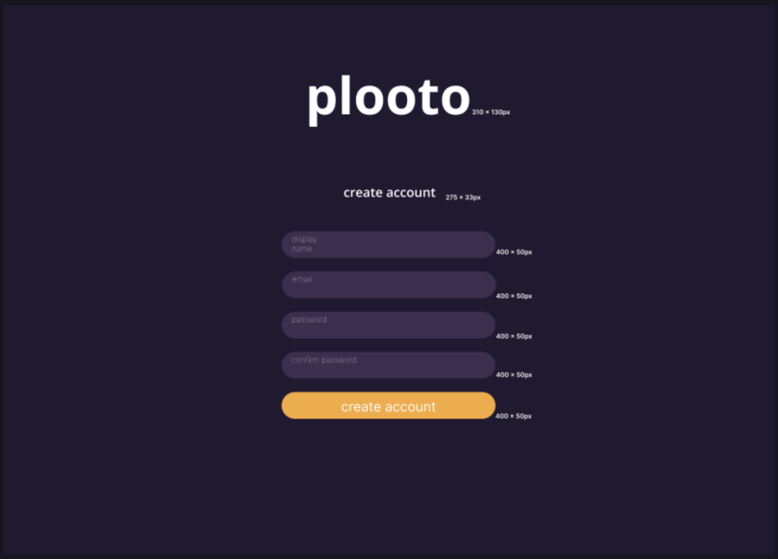
For editing controls we will implement a similar sort of modal / popup, as it’s unobtrusive and should be instant, without needing to leave the page. This will also preserve any page state and scrolling positions.

Page Mock-ups

Combing our research design with our decided application style of purple, in the theme of Plooto, we decided to make various page mock-ups for our website. These are not complete implementations; they exist as style guides and layout references for when we build our app. These designs were made using figma.

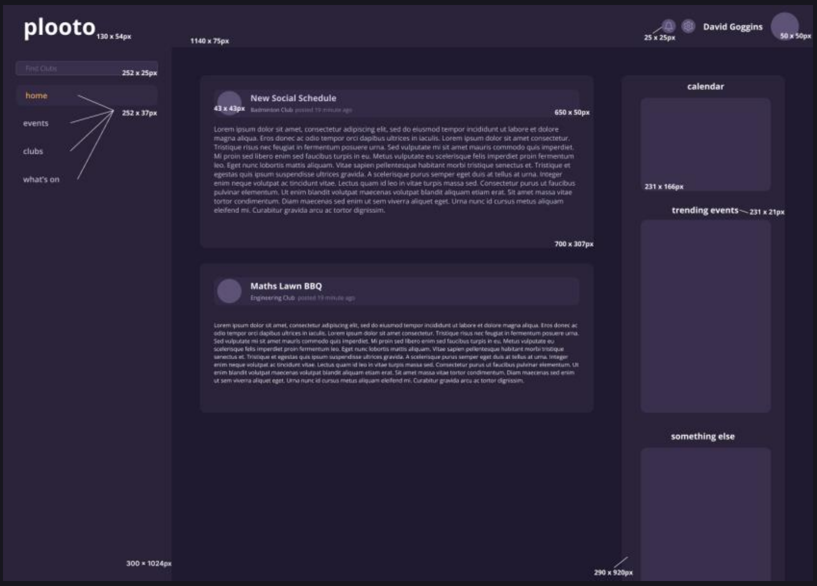
Sign Up & Create Account

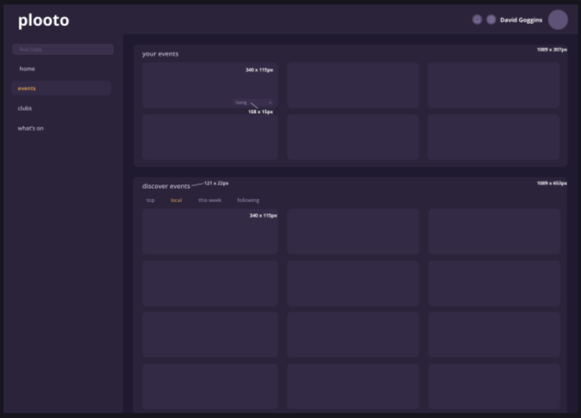


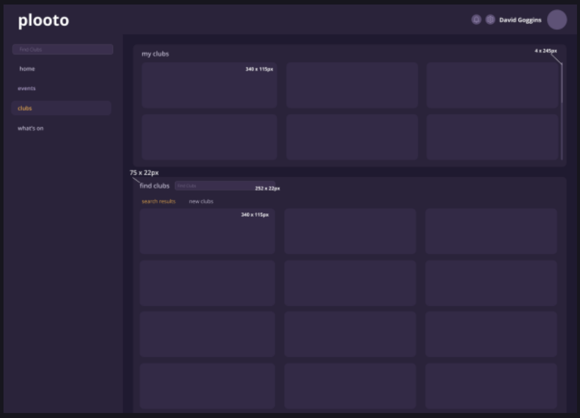


These pages were designed to be simplistic in nature, unobtrusive to the workflow and clear in what needs to be done. The interactions in this page are quite simple, the user would input their login information, and upon clicking login it would verify if their information is correct or not. If the login returns unsuccessful, then an error message would display. In the create account page, some basic validation would be necessary before creating the account, like confirming the password is the same. Potentially, this onboarding process could require a verified email, but for simplicity we will skip this step.

User Pages





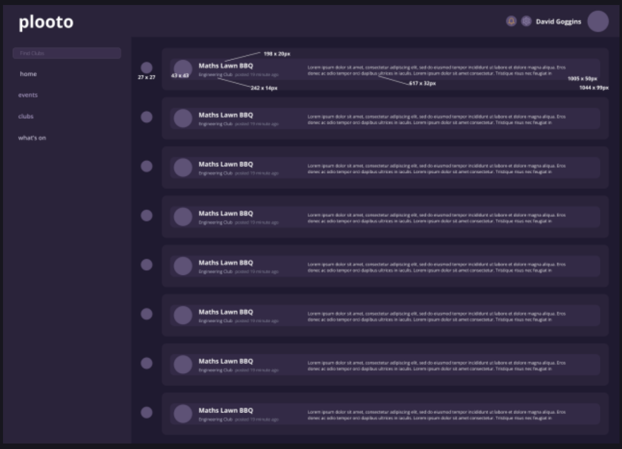


There is a consistent layout across these pages, with the navigation being present on the sidebar, and a settings, notification, and login information in the top bar on the right.

The home page is the page that the user is going to land on when they first visit the website, and is likely the page that they will spend the most time on. The homepage features the familiar timeline/feed structure, where the main content is present in the middle and is scrollable. The aside in the homepage will hold a calendar, that will display upcoming events in a calendar format, and this will be directly interactable and also could bring the user to a more complete, full-page calendar. There also exist other quick and easy information for the user like trending events, and room for more components that will be decided later on.

The events and clubs page manages the users RSVP'd events, and clubs that they are a part of. The idea of the clubs page is that in order to join a club, you would simply search for it, and on the same page you would join the club, or send a request to join the club if it's a private club. There would be multiple different ways to search for a club. A similar concept applies to events, where you would set your RSVP status to each event on this same page. Each club and event would have their own dedicated pages as well, and you could share this link to the event/club with someone else, and if they had an account they would be able to see the same content.

Notifications & Settings



Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in

Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in

Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in

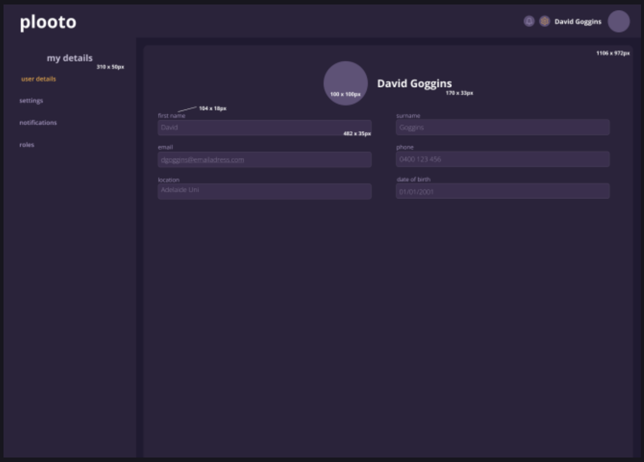
Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in

Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in

Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in

Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in

Maths Lawn BBQ
Engineering Club posted 19 minute ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eros donec ac odio tempor orci dapibus ultrices in iaculis. Lorem ipsum dolor sit amet consectetur. Tristique risus nec feugiat in



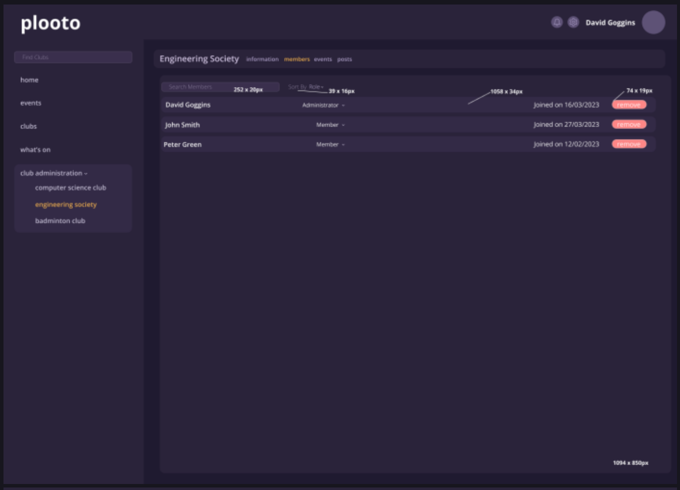
David Goggins

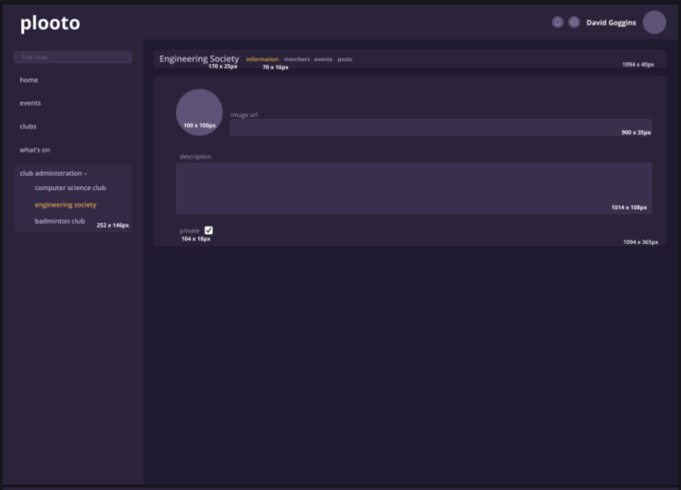
first name: David	surname: Goggins
email: dgoggins@emailadress.com	phone: 0400 123 456
location: Adelaide Uni	date of birth: 01/01/2001

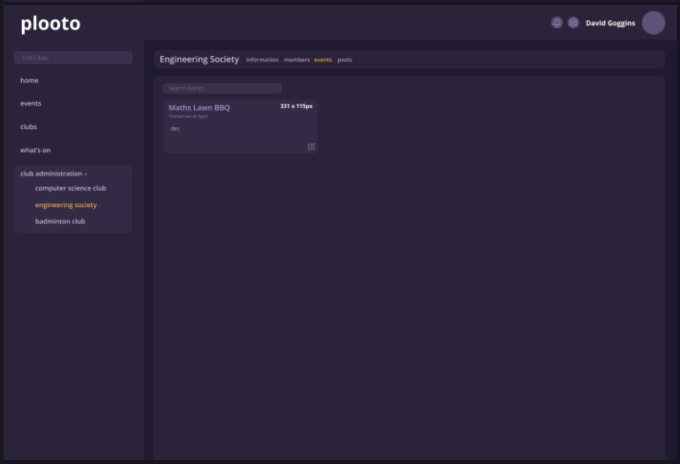
These two pages relate to each other because they are both the pages that exist in the top bar. In terms of the notifications page, this page would be present through a “show all” button in the notifications popup, which is what you would first see when you clicked the notifications button. This would show read/unread status of each notification.

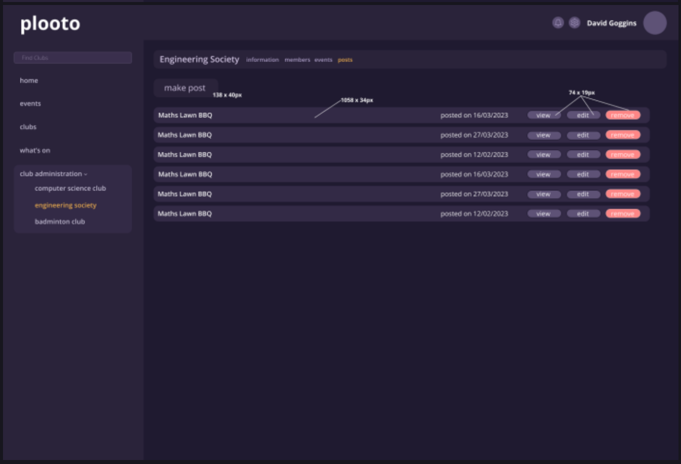
The settings page has multiple sub-sections and they’re all pretty basic forms for user input. It is inside here that the user would set their email notification settings for each event, club, etc. In addition, if the user was designated as a system administrator, then the “admin” subsection is where they would perform these administrator tasks.

Club Management

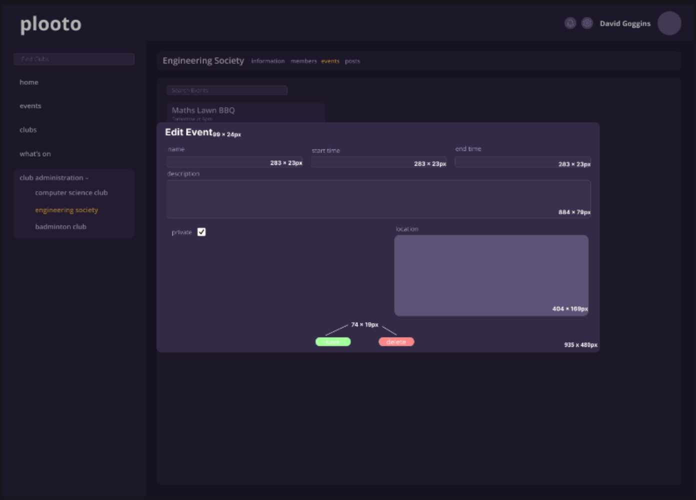


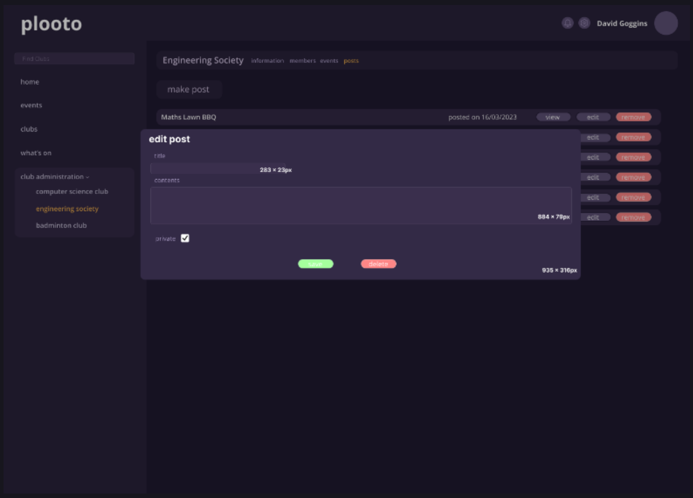






These pages offer management capabilities for the clubs. Each subsection has a different purpose. A club is able to create, edit, and delete events and posts. The club is also to have a look at who is a member of the club, change roles, and remove them. The editor controls for posts and events are popup modals, as demonstrated below.





If these pages are too difficult to view in this document, they are available under /planning/pages/ in the github repository.

Peer Feedback

Group 1

- Dark mode isn't the common choice among most people, need choice of light mode in settings
- Enjoyed the minimal design, made the page easy to scan and understand all the information
- Well made layout made it very user friendly to understand how to operate the website and how to navigate it

Group 2

- How to show if notifications have been read or not (Disappear or not), more memory if it doesn't disappear
- Colour pallet easy on the eyes
- minimalistic design is good
- make sure sign in pages have a ‘forgot password’ button
- Checking who can see what information (such as location and date of birth) about users
- Could add a page that shows where members are located to help plan events

Group 3

- Use correct capitalisation
- Need a dedicated page for events/posts

Review

From the group feedback we’ve identified some clear goals to improve our design:

- Correct capitalisation
- “Forgot Password” button
- Clearly show notifications being “read” or “unread”
- Light mode support
  - Light/dark mode toggle

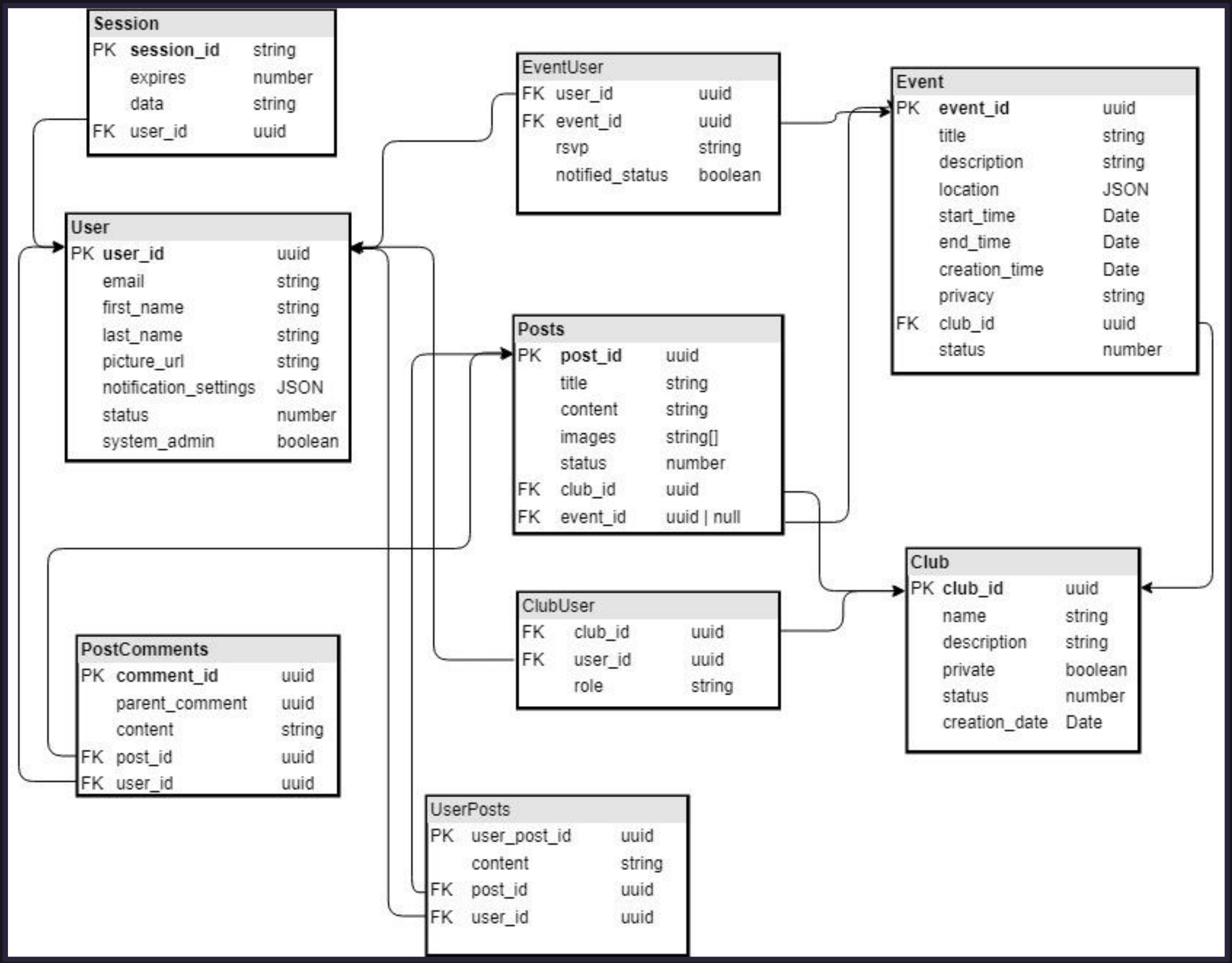
When comparing our website to design to the usability heuristics, we’ve identified these areas for improvement:

- Need more statistics/information readily available to users/clubs, in accordance with “Visibility of system status”
- Perhaps change “administration” to “management”, as some users may not understand that administration includes the functions for managing aspects of clubs (“Match between system and the real world”).
- Ensure that in the final design, there is “Undo” buttons for the actions, and that in each modal there is a clear “Close” or “Cancel” button.
- Add in shortcut buttons or shortcut keybinds to allow for quick and easy navigation.

And when assessing our design for kinematic/cognitive load, we’ve identified these as areas for more potential:

- Make sure there is proper feedback on user actions, so when creating or updating posts/events it is clear to the user that the update has been successful.
- Keep the navigation consistent across the different pages, such that all navigation should only be done in the sidebar, and not in the top bar of a subpage.

Schema Design

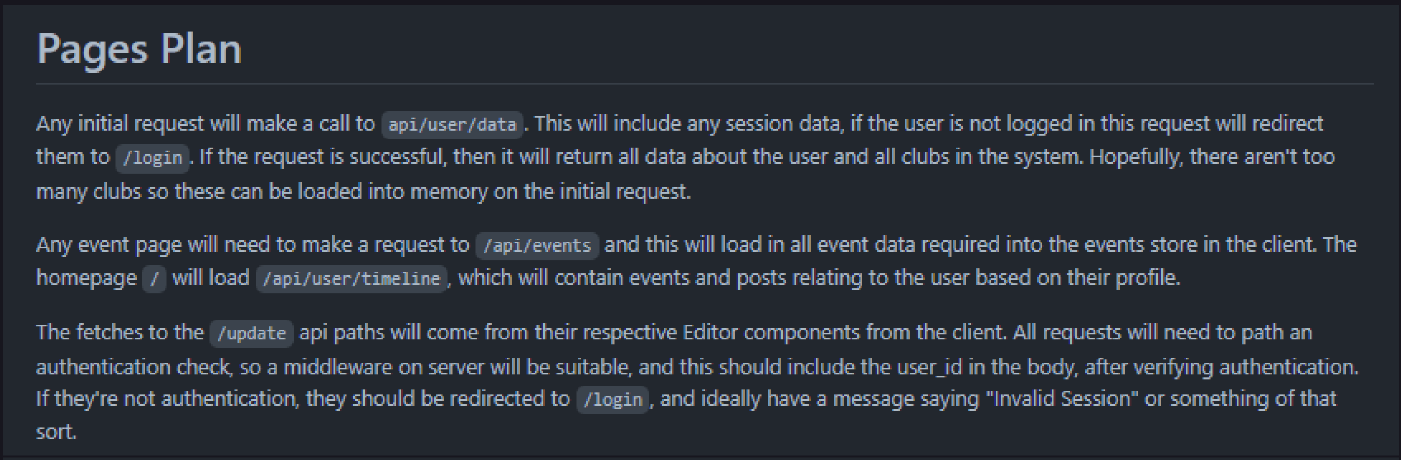


This schema includes post comments, which was not included in the original design. These comments will be a feature that will only be implemented if we have enough time at the end, as they are not critical to our app’s main functions.

Data Plan

The data plan exists in the repository under /planning/data_plan.md

As it is an interactive markdown document, it is best viewed inside a markdown renderer, however we've included screenshots from the document as rendered on github.

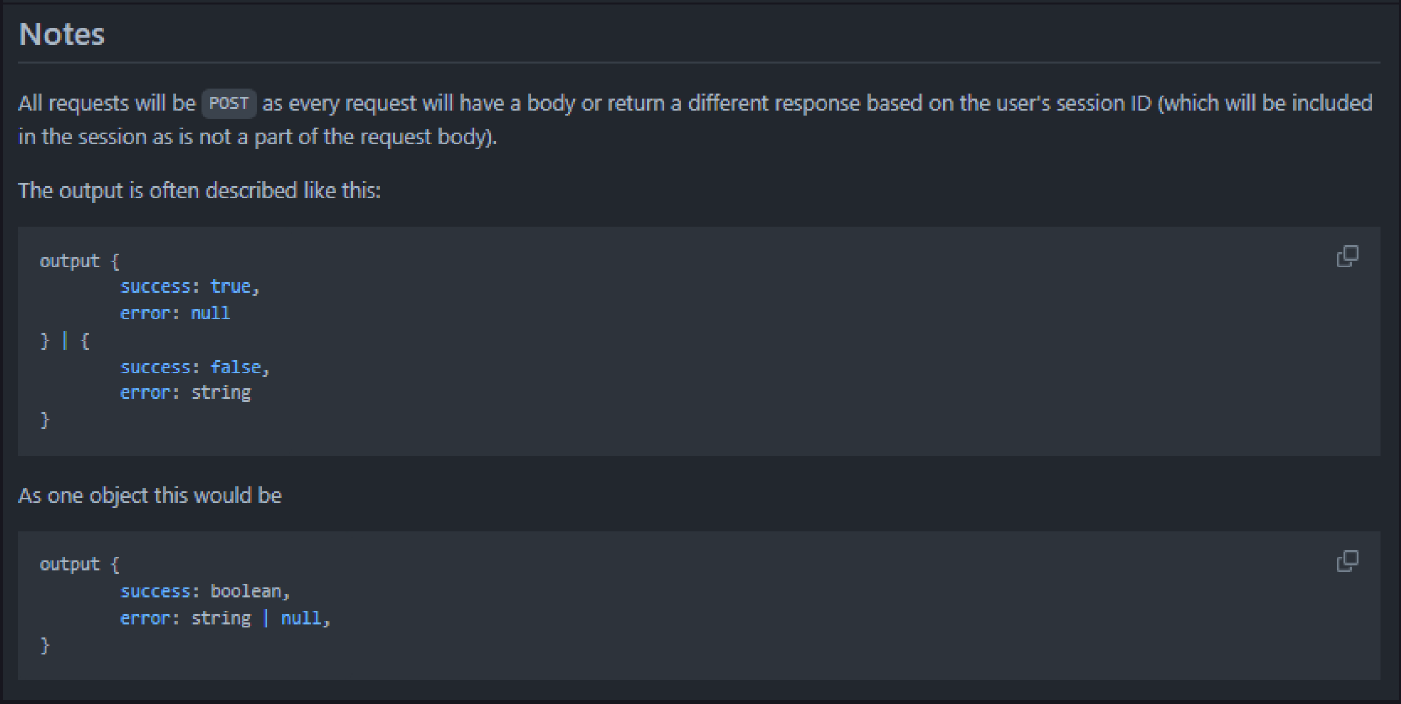


Pages Plan

Any initial request will make a call to `api/user/data`. This will include any session data, if the user is not logged in this request will redirect them to `/login`. If the request is successful, then it will return all data about the user and all clubs in the system. Hopefully, there aren't too many clubs so these can be loaded into memory on the initial request.

Any event page will need to make a request to `/api/events` and this will load in all event data required into the events store in the client. The homepage `/` will load `/api/user/timeline`, which will contain events and posts relating to the user based on their profile.

The fetches to the `/update` api paths will come from their respective Editor components from the client. All requests will need to path an authentication check, so a middleware on server will be suitable, and this should include the user_id in the body, after verifying authentication. If they're not authentication, they should be redirected to `/login`, and ideally have a message saying "Invalid Session" or something of that sort.



Notes

All requests will be `POST` as every request will have a body or return a different response based on the user's session ID (which will be included in the session as is not a part of the request body).

The output is often described like this:

```
output {
	success: true,
	error: null
} | {
	success: false,
	error: string
}
```

As one object this would be

```
output {
	success: boolean,
	error: string | null,
}
```

User

▼ `POST` `` `/api/user/create` ``
Attempts to create a user with the given input body
Sources:

- `/views/Signup.vue`

```
body {
        email: string,
        first_name: string,
        last_name: string,
        display_picture: string
}
```

`display_picture` will be a URL of where the image is hosted. Will perform an `INSERT` query and return the ID, success will be false if the DB is down or some other IO operation cancels it.

```
response {
        success: true,
        ID: string
} | {
        success: false,
        error: string
}
```

▼ `POST` `` `/api/user/update` ``
Attempts to update a user update query given the user ID, and updates with the other data specified. This will also handle user deletions, in which case the `status` will be set to 0.
Sources:

- `/views/settings/Account.vue`

```
body {
        id: string,
        email: string,
        first_name: string,
        last_name: string,
        display_picture: string,
        notification_settings: any,
        status: 0 | 1
}
```

`display_picture` will be a URL of where the image is hosted.

```
response {
        success: true,
        error: null
} | {
        success: false,
        error: string
}
```

▼ `POST` ``` `/api/user/timeline` ```
Loads relevant posts and events for a given user.
Sources:

- `/views/Home.vue`

```
body {
	user_id: string // this may also be gotten from session
}
```

```
response {
	events: Event[],
	posts: Post[],
} | {
	success: false,
	error: string
}
```

▼ `POST` ``` `/api/user/data` ```
Loads relevant user information for a given user and all clubs for storage within local memory.
Sources:

- `/views/Home.vue`

```
body {
	user_id: string // this may also be gotten from session
}
```

UserInformation is the data form the `Users` table.

```
response {
	clubs: Club[],
	user: UserInformation
} | {
	success: false,
	error: string
}
```

Club

▼ `POST` `` `/api/club/create` ``
Sources:

- club creation component

```
input {
        name: string,
        description: string,
        private: boolean,
}
```

Will perform an INSERT query and return the ID, success will be false if the DB is down or some other IO operation cancels it.

```
output {
        success: true,
        ID: string
} | {
        success: false,
        error: string
}
```

▼ `POST` `` `/api/club/update` ``
Sources:

- `/views/club-admin/Information.vue`

```
input {
        id: string,
        name: string,
        description: string,
        private: boolean,
        status: 0 | 1
}
```

```
output {
        success: true,
        error: null
} | {
        success: false,
        error: string
}
```

Events

▼ `POST` `` `/api/events` ``
Sources:

- event creation component

```
input {
        user_id: string
}
```

```
output {
        success: true,
        events: Event[]
} | {
        success: false,
        error: string
}
```

▼ `POST` `` `/api/event/create` ``
Sources:

- event creation component

```
input {
        title: string,
        description: string,
        location: {
                lat: number,
                lon: number
        },
        start_time: Date,
        end_time: Date,
        club_id: string,
        privacy: "MEMBER" | "OPEN" | "INVITE ONLY"
}
```

Will perform an INSERT query and return the ID, success will be false if the DB is down or some other IO operation cancels it.

```
output {
        success: true,
        ID: string
} | {
        success: false,
        error: string
}
```

▼ `POST` `` `/api/event/update` ``
Sources:

- `components/EventEditor.vue`

```
input {
        id: string,
        title: string,
        description: string,
        location: {
                lat: number,
                lon: number
        },
        start_time: Date,
        end_time: Date,
        club_id: string,
        privacy: "MEMBER" | "OPEN" | "INVITE ONLY",
        status: 0 | 1
}
```

Will perform an INSERT query and return the ID, success will be false if the DB is down or some other IO operation cancels it.

```
output {
        success: true,
        error: null,
} | {
        success: false,
        error: string
}
```

Posts

▼ POST `/api/post/create`

Sources:

- post creation component

```
input {
        title: string,
        content: string,
        images: string[],
        club_id: string,
        event_id: string | null
}
```

Will perform an INSERT query and return the ID, success will be false if the DB is down or some other IO operation cancels it.

```
output {
        success: true,
        ID: string
} | {
        success: false,
        error: string
}
```

▼ POST `/api/post/update`

Sources:

- post editor component

```
input {
        id: string,
        title: string,
        content: string,
        images: string[],
        club_id: string,
        event_id: string | null,
        status: 0 | 1
}
```

```
output {
        success: true,
        error: null
} | {
        success: false,
        error: string
}
```