

Social Media Web

OTJS

Farhan Tanvir (45)
Joty Saha (51)
Asef Sami (53)
Ahona Omi (59)

Department of Computer Science and Information Engineering
University Of Dhaka

January 26, 2025

The Social Media Web Named OTJS is a scalable platform that allows users to interact through posts, likes and real-time updates, built with React, Typescript, Tailwind CSS, and Appwrite for a seamless and responsive experience.

Design Patterns Used In Our Project

- **Singleton Pattern:** In our project, the Singleton Pattern is used for the AppwriteConfig class. This ensures that there is only one instance of the configuration, providing a global point of access to these settings throughout the application.
- **Factory Pattern:** The Factory Pattern is utilized in the EntityFactory class to centralize the creation of user and post objects. It is also used in creating query and mutation hooks for form fields and API interactions, which helps in maintaining a clean and consistent method of object creation across our application.

Design Patterns Used In Our Project

- **Repository Pattern(MVC):**This pattern is used in UserRepository and PostRepository classes to abstract database operations. It separates the data access logic from business logic, providing a clean API to perform CRUD operations on user and post data.
- **Observer Pattern:**The Observer Pattern is evident in the use of React Query's `queryClient.invalidateQueries()` method. This method is used to notify components to refetch data when mutations occur, ensuring that the UI remains in sync with the latest state and changes.

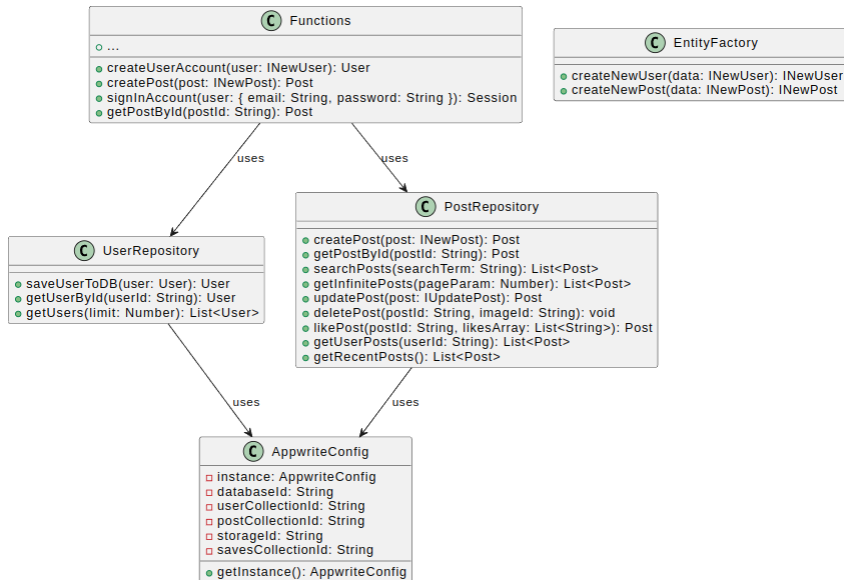
Design Patterns Used In Our Project

- **Command Pattern:** In the createMutationHook setup, the Command Pattern is implied where actions (such as signing in or creating a post) are encapsulated with their side effects, allowing these operations to be executed flexibly and consistently across the application.
- **Strategy Pattern:** The Strategy Pattern is implemented in the PostSorter class where different sorting strategies (e.g., ChronologicalSort, PopularitySort) can be applied to sort posts. This allows the sorting logic to be extended or modified without altering the clients that use it.

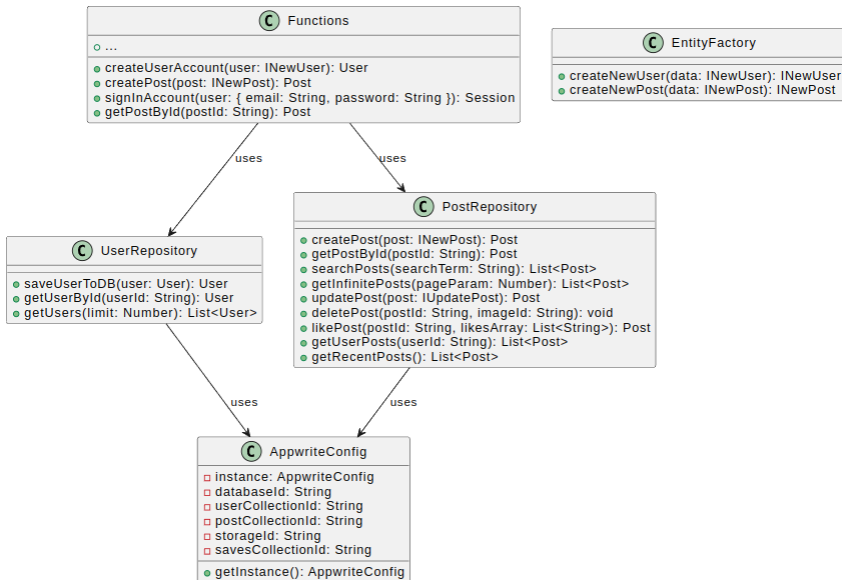
Design Patterns Used In Our Project

- **Builder Pattern:** In methods like `updatePost()` and `createPost()` in `PostRepository`. Gradually builds the object (`newPost` or `updatedPost`) by adding properties and modifying it step by step.

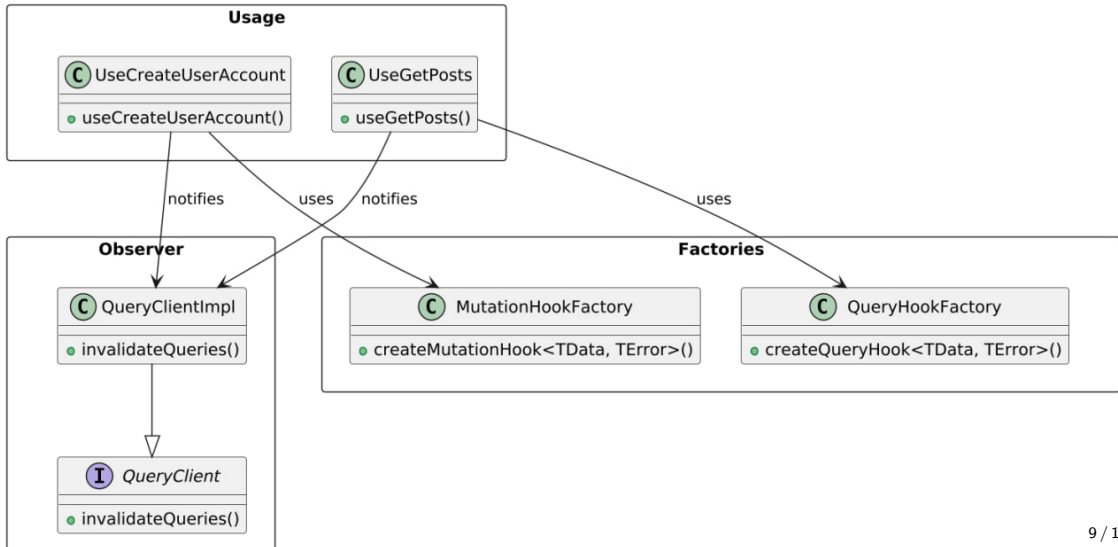
UML diagram of API.tsx



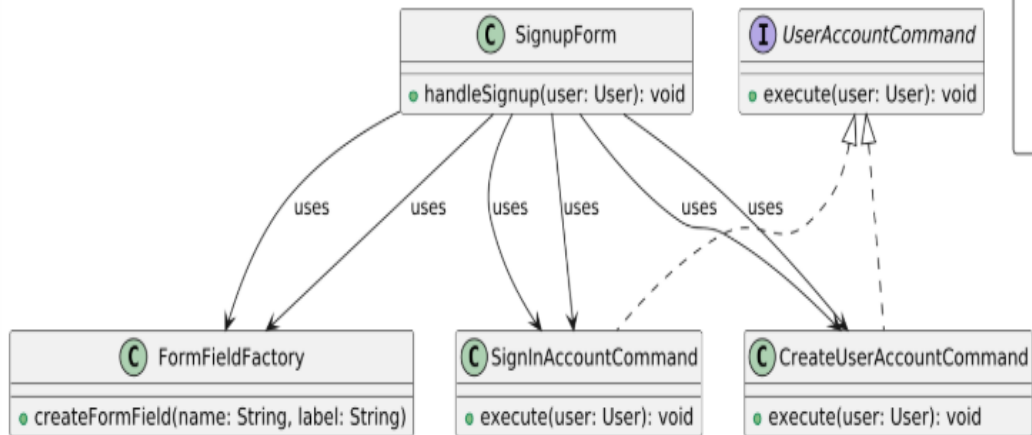
UML diagram of API.tsx



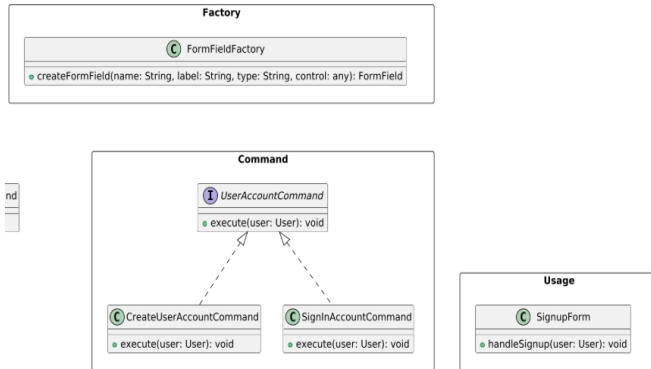
UML diagram of queries.tsx



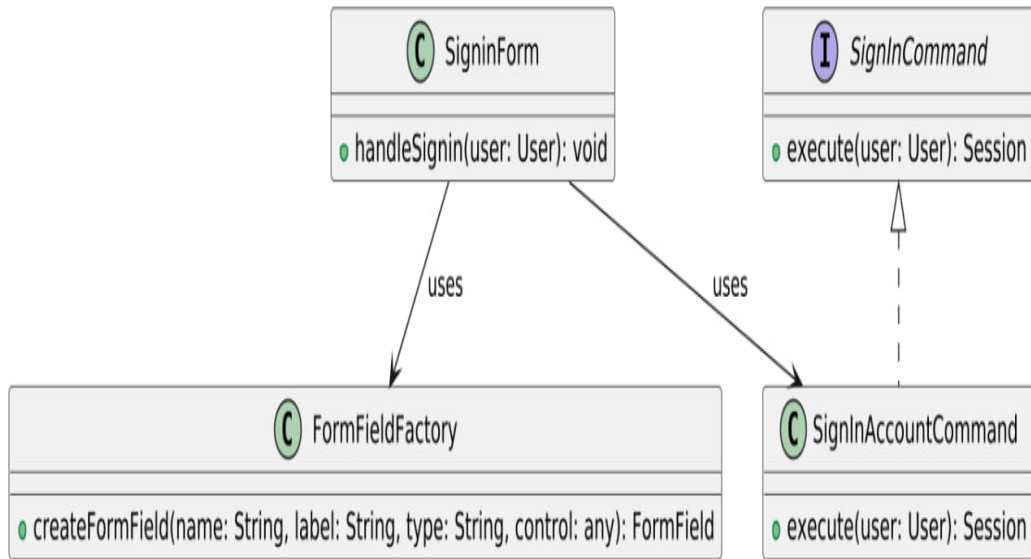
UML diagram of SignUpForm.tsx



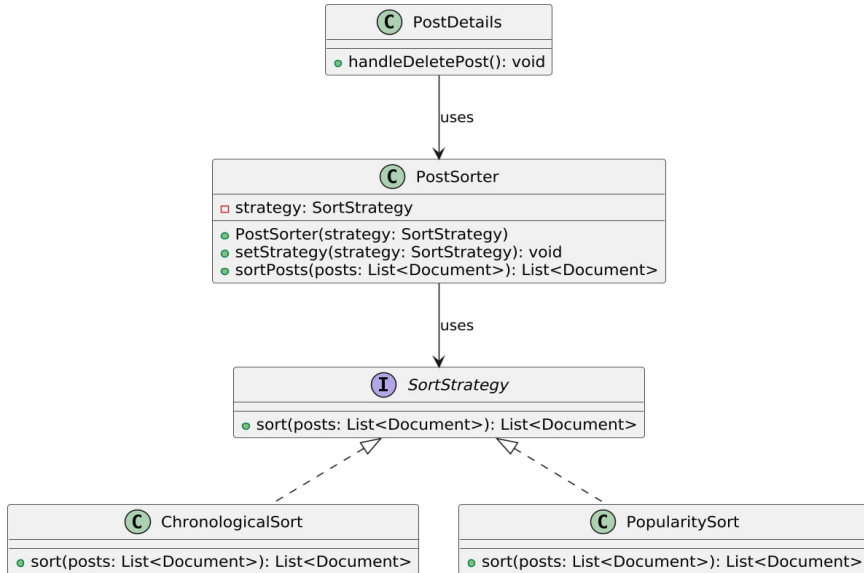
UML diagram of SignUpForm.tsx



UML diagram of SignInForm.tsx



UML diagram of PostDetails.tsx



[Go to this YouTube video link](#)

Thank You