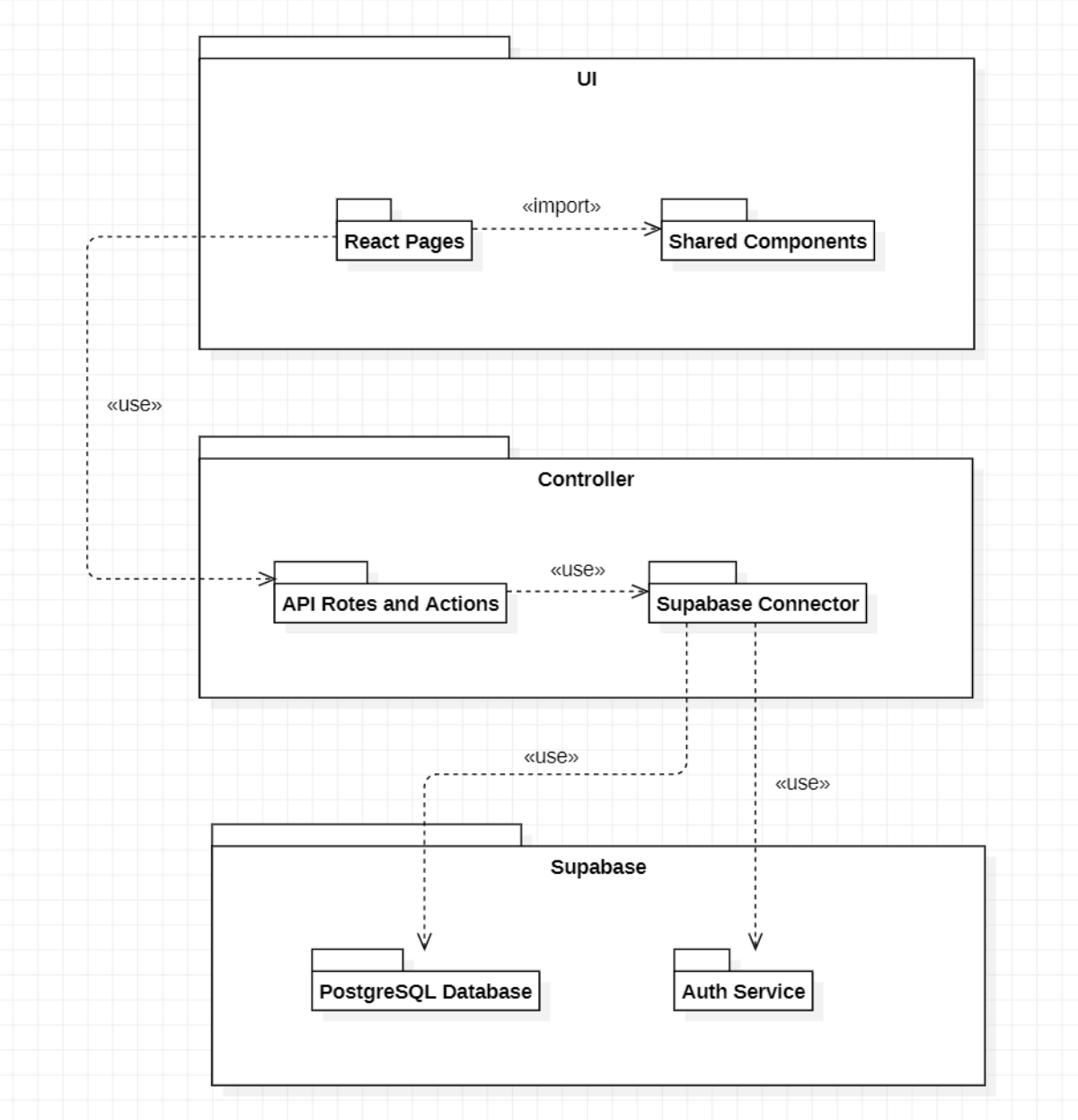
Packages diagram

Victoria Vavulina

Riku Toivanen

Monami Kirjavainen

The package diagram:



1. **UI Package:** Contains the React Pages that form the user-facing interface of the application. This is the presentation layer where users interact with the system.

2. **Shared Components Package:** Houses reusable React components (buttons, forms, navigation elements, etc.) that are imported by React Pages.

3. **Controller Package:** Serves as the intermediary layer between the UI and backend services. It contains the API Routes and Actions that handle business logic and uses the Supabase Connector to communicate with the database layer.

4. **API Routes and Actions Package:** Defines the API endpoints and server actions that the UI can call. This package handles HTTP requests, input validation, and delegates data operations to the Supabase Connector.

5. **Supabase Connector Package:** Provides an abstraction layer for database operations, connecting the Controller to Supabase services.

6. **Supabase Package:** The backend-as-a-service platform that hosts and manages the PostgreSQL Database and Auth Service.

7. **PostgreSQL Database Package:** The relational database that stores all application data. It's accessed through Supabase.

8. **Auth Service Package:** Manages user authentication and authorization within Supabase. It handles user registration, login, session management, and access control, ensuring secure access to application resources.

React Pages <<import>> Shared Components: Pages in the app (like page.tsx) bring in and reuse shared UI pieces. Shared components are buttons, cards, layouts, forms, etc.

React Page <<use>> API Routes and Actions: Pages call server functions to get or send data. In Next.js this is usually inside app/api/... or inside “Server Actions”.

API Routes and Actions <<use>> Supabase Connector: API routes do not talk to the database directly. They call a helper module (the “Supabase Connector”) that wraps Supabase calls.

Supabase Connector <<use>> PostgreSQL Database: The connector uses the Supabase client library to read and write data in the PostgreSQL database.

Supabase Connector <<use>> Auth Service: The connector also handles user login, logout, tokens, and session validation through Supabase Auth.