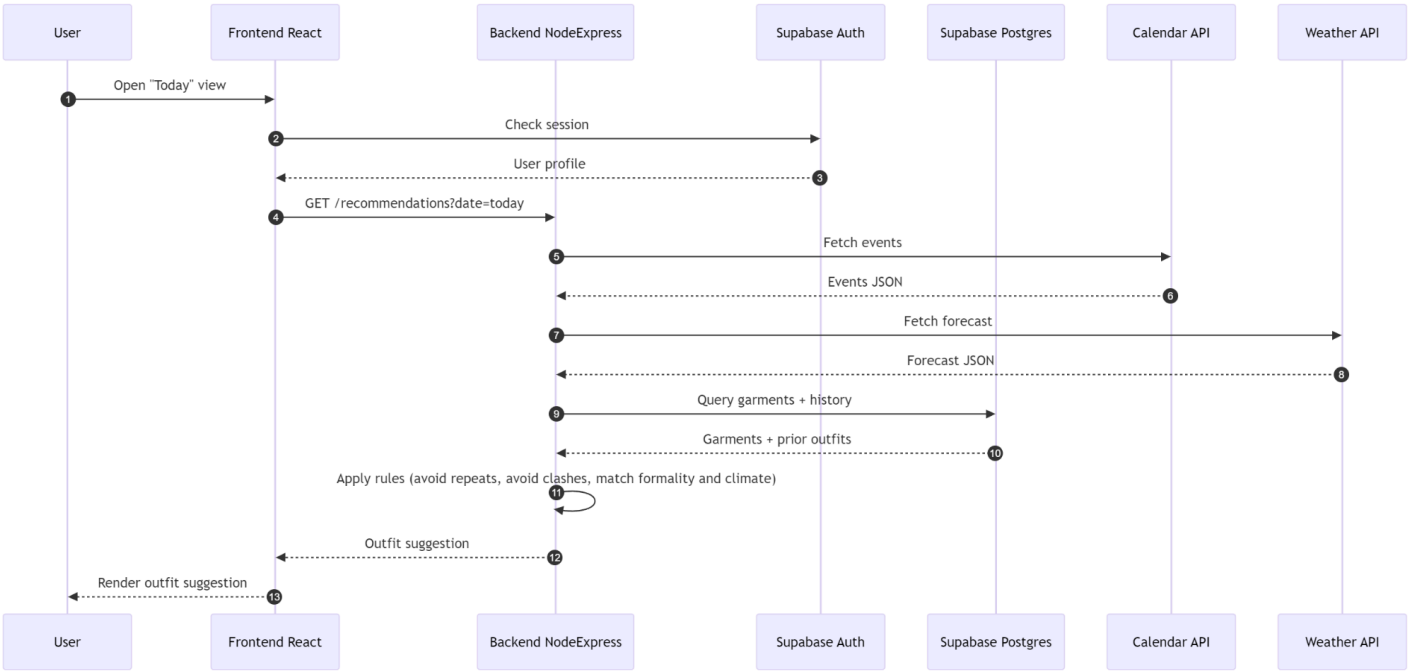
PROJECT PROPOSAL BY Yuval Shimoni

WearHouse

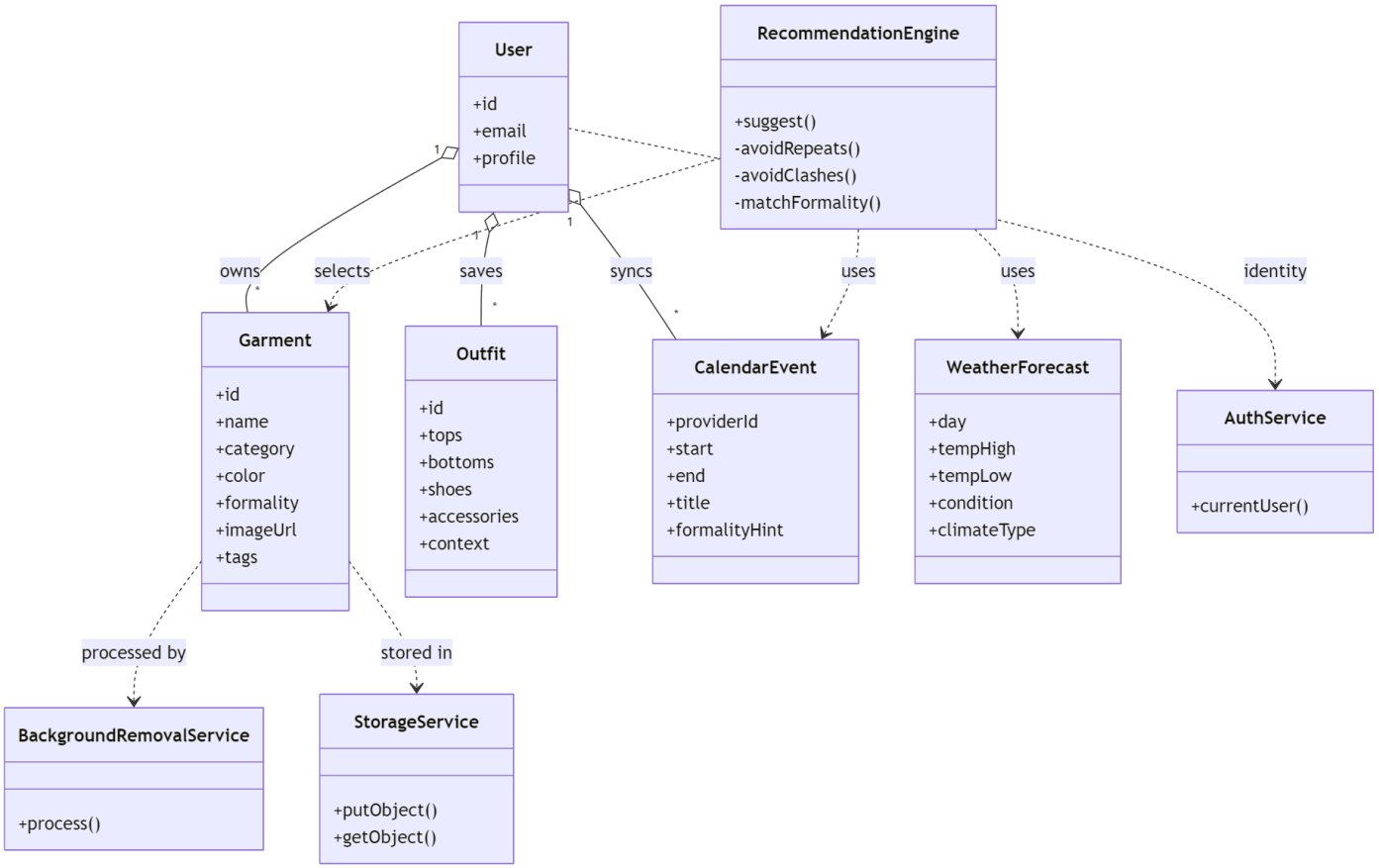
Project Abstract

In this project, I propose developing WearHouse, a smart digital closet app designed to help users manage their wardrobe and plan outfits more efficiently. Many people struggle with decision-making and fatigue when choosing their daily outfits, especially when they try to account for different weather conditions, various social/professional events, and the desire to have variety in their outfits. WearHouse addresses this by allowing users to integrate calendar events and local weather forecasts into the app, ensuring that recommendations are not only stylish but context-appropriate. Users upload images of their clothing items, which are automatically processed with a background remover to create a clean visual of their clothes. Items can be tagged by category, color, and formality. This enables the user to utilize the system to combine pieces into practical outfits. The app will include an outfit generator that uses simple rules to avoid clashes, prevent repeats, and align clothing choices with formality and climate types. The intended users include anyone who cares or needs to care about what they wear on a regular basis. Specifically, those who want to save time doing so, while being stylish, experimental, and prepared. As a stretch goal, WearHouse would like to develop an avatar-based try-on feature that would allow users to preview their outfits on a personal photo overlay or maybe even a digital mannequin.

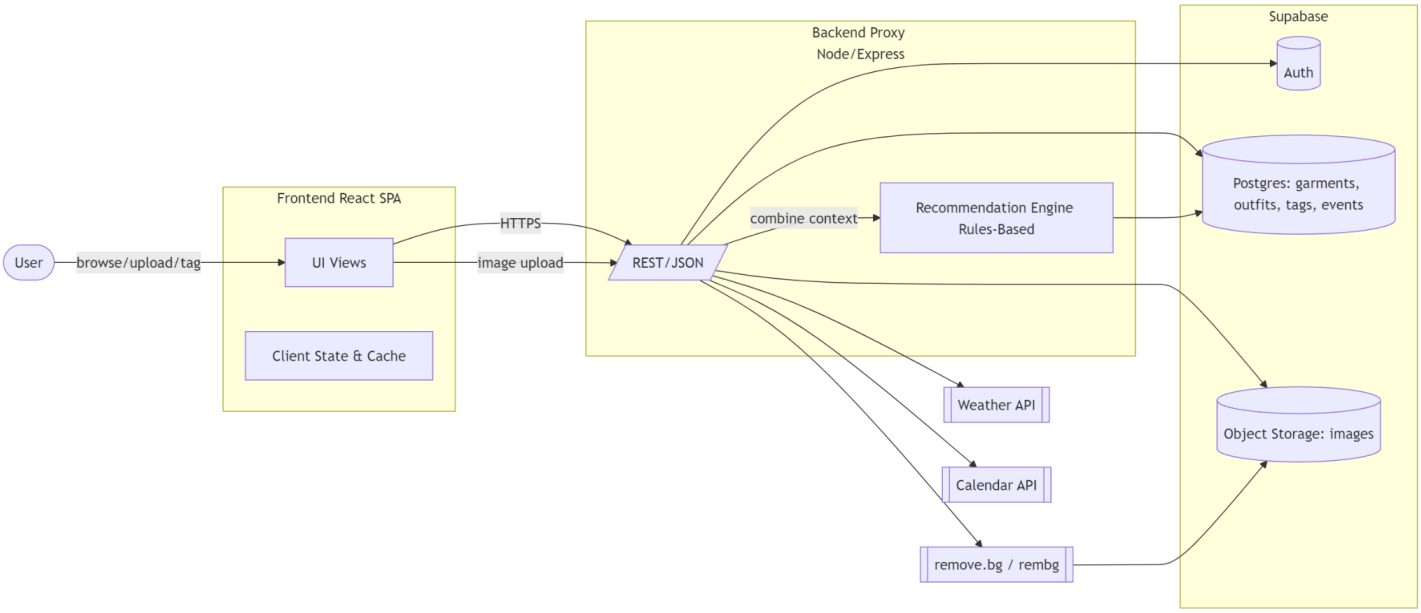
Conceptual Design



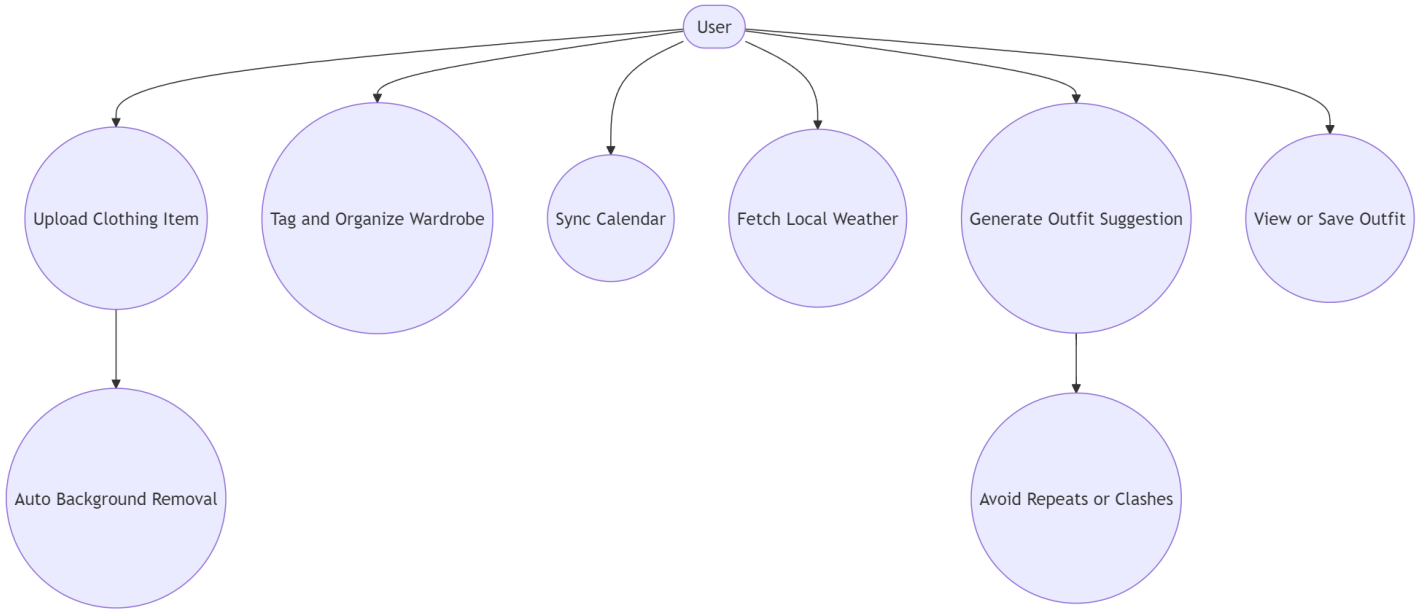
This sequence diagram traces the step-by-step process of generating an outfit recommendation for a given day. It begins when the user opens the “Today” view, checks authentication, fetches calendar events and weather, and queries stored garments. The backend applies rules (avoiding repeats/clashes, matching formality/climate) and returns an outfit suggestion to the frontend for display.



This class diagram defines the system's main objects: User, Garment, Outfit, CalendarEvent, WeatherForecast, and service classes. It shows ownership (ex: a user owns many garments and outfits) and dependencies (ex: the RecommendationEngine uses garments, calendar events, and weather). Service classes use external systems (ex: Auth, Storage, Background Removal) to keep the design modular.



This architecture diagram shows how the WearHouse system is split into a Frontend (React SPA), a Backend Proxy (Node/Express), and Supabase for auth, database, and storage. It also integrates with external APIs: remove.bg/rembg, Weather API, and Calendar API. Data flows from the user through the frontend, into the backend, which coordinates services and applies outfit recommendation rules.



This use case diagram highlights the main actions a user can perform: upload clothing, tag and organize, sync their calendar, fetch weather, and generate outfit suggestions. Each step flows logically, for example, uploading leads to background removal, and outfit generation considers weather, calendar events, and rules. It shows the system's scope from the user's perspective and ensures core features are captured.

Background

There are some apps that currently exist which have similar ideas, however, these apps (like Stylebook and Cladwell) are more focused on cataloging clothes and static outfit generation. They lack integration of external data and do not have much personalization beyond tags. WearHouse differentiates itself by using contextual recommendations (with the calendar and weather integration), automated background removal for clean visuals, and its future stretch with the avatar-based try-on.

References

Stylebook. Your Closet App for iPhone and iPad. Available: https://www.stylebookapp.com

Cladwell. Personal Styling App. Available: https://cladwell.com

Proof of Concept

https://github.com/YuvalS03/WearHouse

Required Resources

- Knowledge: web dev (React, Node.js), API integration, some image processing
- Software: Supabase Postgres + authorization + storage), Remove.bg/rembg, GitHub
- Hardware: personal laptop