NYU

DailyCommuter

Group #5

Conor Zhang, Nissim Ram, Emmanuel Owusu-Ampaw, Kamil Paczkowski, Gideok Min, Boruch Khazanovich

04.03.2025

Overview

Why would you need an app for your commute?

- Stay informed with real-time alerts that may impact your commute.
- Get push notifications when delays may affect your usual departure time.
- Discover alternate routes to help you save time during disruptio
- Know exactly when your next bus or train is arriving — no guess

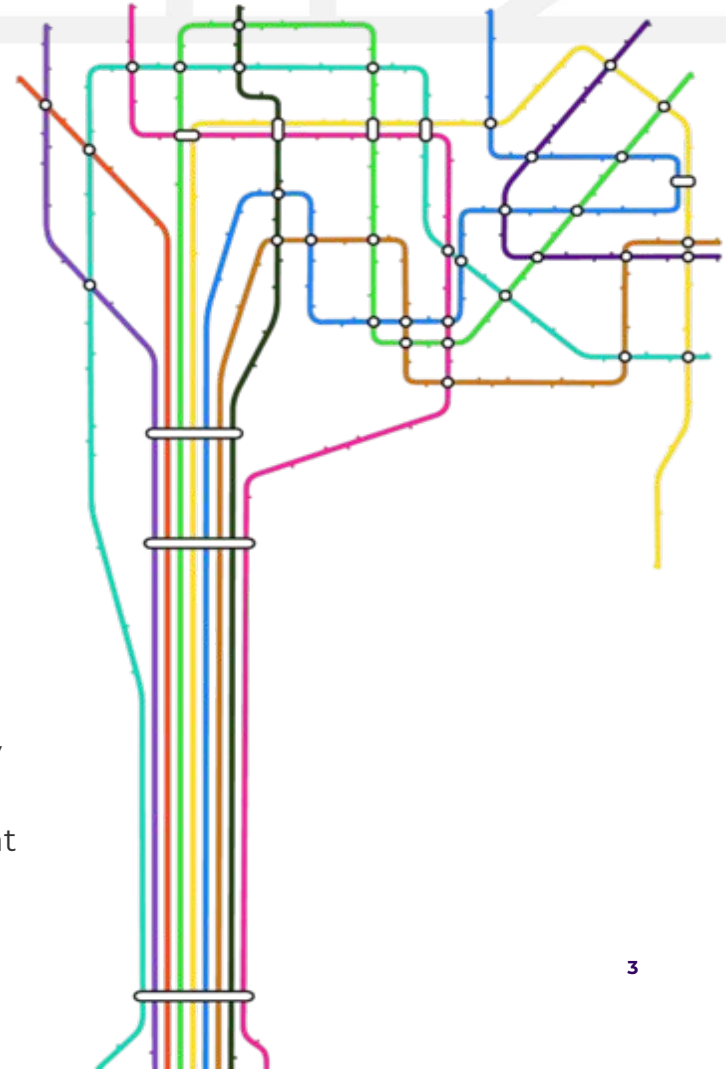
NYU

Features to be Implemented

General Implementations

- Search feature with filters that the user can toggle on / off:
  - The ability to Search/filter by individual stop
  - Search/filter by train, showing all upcoming times for that line

- Interactive Map:
  - Displays all individual stops
  - Allows users to filter the view by transit type. (do they only want to see bus or train routes)
  - Be able to connect to live transit data sources and use that information to highlight service delays

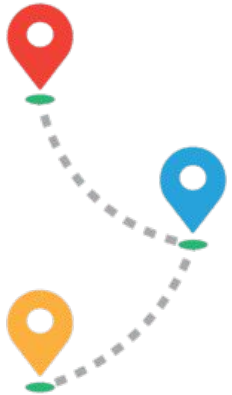


Features to be Implemented

App Specific Implementations

- Route Tracking
  - Allowing the user to input specific routes that they commonly take.
  - Provide alerts for delays, service interruptions, and arrival times

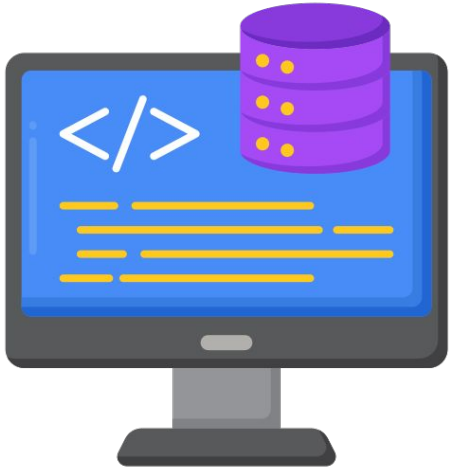
- Smart Notifications:
  - Allowing user to enter desired arrival times (e.g. School at 9AM)
  - App will then send reminders on when to leave to stay on schedule.



Frontend Implementation

- Responsive design
  - for mobile and desktop
- Visual Inspiration
  - Design will draw from Transit app's clean visual style for intuitive user experience
- Core technologies
  - React
  - OpenStreetMap for mapping
  - Chakra UI component library for non - map components like Forms, Dropdowns, Buttons etc.

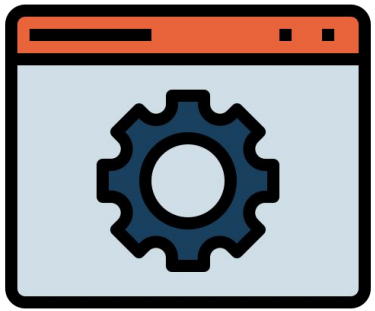
Backend Implementation



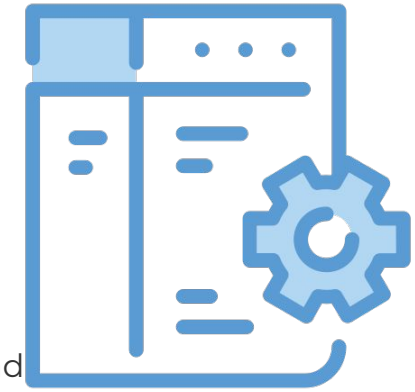
- Backend is implemented using Flask (Python) with Sqlite3.

- As of today, the database is retrieving updates for:
  - All general train updates
  - Service alerts for trains and busses

- The above data is updated on loading the homepage of the app

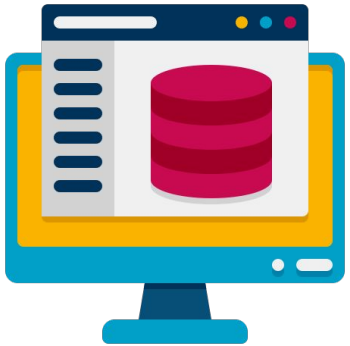


Backend Implementation



- Still to be implemented:
  - Retrieve updates for busses
  - Retrieve alerts for elevator and escalator outages
  - Create filter functions for all searching and filtering needed for the frontend
  - Store user's saved routes and stations

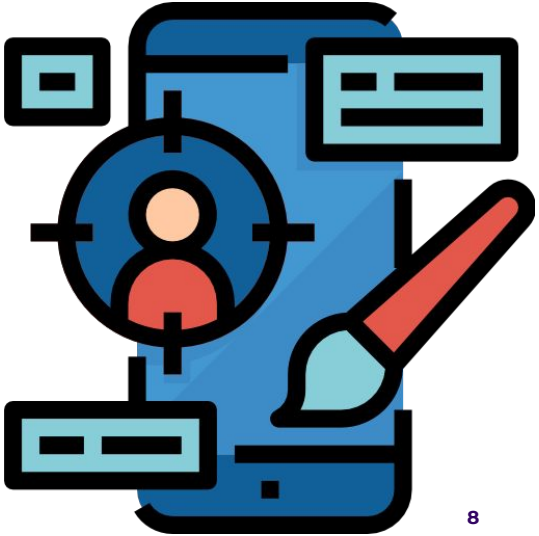
- While the login and authentication has already been implemented using the basic Flask functionality, we still need to set up the database to store the username and passwords



UI Mockups

- UI inspired by the Transit app design:
  - Clean interface with colored route cards
  - Real-time arrival countdowns

- Mockup components (in progress):
  - Homepage with search bar and pinned routes
  - Route page showing line info and Estimated Time of Arrival
  - Map view with stop markers



- Tools being used:
  - Figma for design
  - Chakra UI for implementation

NYU

Future Improvements

Scaling & Performance Enhancements

- Database Upgrade
  - Switch to a more scalable database like PostgreSQL to handle concurrent user request efficiently.
- Caching
  - Using catching for frequently accessed transit to improve load times and less API calls

https://api.mta.info/#/subwayRealTimeFeeds

https://github.com/DailyCommuter/DailyCommuter

NYU