

Main Functionality:

- Add license plates to a watchlist.
- Receive real-time alerts when a match is found.
- View details (image, GPS, timestamp) of detected vehicles.

Goal:

- Efficient interaction
- Asynchronous notification: instant updates when a target hit
- Scalable

Quantitative requirements:

**UI:**

First page: Log-in page

Notification pop-up:

- License plate number the police entered; reason for search
- Image of the detected license plate
- GPS + timestamp
- Yes/No choice for human verify if this is the car they wanted to find

Search page:

- License plate number + state of the license plate
- Good to have:
  - Characteristics of the car( color, type, manufacturer, etc.)
  - Which crime/reason to find this car

Choices for UI:

React + TypeScript

Pro:

- Fast, component-based structure
- Scalable
- Type safety with TypeScript

**Backend:**

- Interact with the system to handle license plate of interest watchlist
- Manage authentication login
- Process notification when a match is found

options:

FastAPI:

Pro: faster, modern python framework

Con: fewer plugins available

**Node.js + Express**

Pro:

- Fast for real-time applications
- Large ecosystem

- Socket.IO available for push notifications

Con:

- Requires additional security implementations
- Callback-heavy if not managed properly

Proposed final choice:

- React.js + Typescript for frontend
- Node.js + Express + Socket.io for backend

Justification of choice:

- Node.js uses an event-driven, asynchronous model, making it ideal for handling many concurrent connections.
  - Socket.IO provides a simple way to handle real-time communication.
  - Works well with microservices and horizontal scaling
- 

## Technology selections

Node.js with Express.js as backend framework and server

React with [Ant Design](#) Library

TypeScript both on the frontend and backend -

AWS DynamoDB for storage

JWT for token-based authentication - is this safe enough or do we need some other service?

- Register third party authorization?
- JWT is more secure than cookie based authentication

*Socket.io for dynamic updates with websocket communication*

Axios for issuing AJAX requests

Bcrypt for salting and hashing passwords

Parcel as bundler

CI/CD set up with GitHub Actions

AWS EC2 for deployment

---

## Frontend main use case

1. Authentication
  - a. User visits the login page
  - b. JWT token based authentication
  - c. Redirected to the dashboard upon success
2. Dashboard (add, remove, view plate)

- a. Users can add or remove license plates to/from the active watchlist.
  - b. Users can see all watchlist entries, timestamps, and reasons.
  - c. (optional) Search/filter
- 3. Notification center
  - a. The user receives an email alert via AWS SNS.
  - b. The user clicks the email which opens the Notification Center page.
  - c. The user reviews past alerts in the UI.
  - d. The user clicks an alert to view full details (plate, location, timestamp).