ChatGPT Initial Specification

Below is a basic Requirements Specification document based on your app concept:

Street Cleaning Alert & Car Tracker App – Requirements Specification

1. Introduction

Purpose:

The purpose of this app is to help users in San Francisco avoid parking tickets during street cleaning. The app will track the user’s car location, integrate with the city’s street cleaning schedule, and send timely alerts so the user can move their car before a scheduled cleaning.

Scope:

The app will be built for iOS devices. It will incorporate map integration, data from the San Francisco government’s street cleaning API, notification systems (push and text alerts), family sharing capabilities, and integration with an AirTag already associated with the user’s Apple Find My account.

---

2. Functional Requirements

2.1 User Interface & Experience

- **Map View:**
- Display the current location of the user’s car.
- Allow users to set a car location by pinning a point on the map or automatically via the AirTag.
- **Countdown Timer:**
- Display a countdown or timer for upcoming street cleaning events based on the car’s location.
- **Notification System:**
- Send push notifications (and optionally SMS/text alerts) to the user.

- Alert designated family members when the car is in a location scheduled for street cleaning.

- **Family Sharing:**

- Allow the user to add family members (via their iPhone accounts) to receive alerts and share location data.

2.2 Data Integration & Processing

- **Street Cleaning Data:**

- Connect to and regularly fetch data from the San Francisco government API regarding street cleaning schedules.

- Parse and map the received data to specific geographic locations.

- **AirTag Integration:**

- Integrate with Apple's Find My service to read data from an AirTag associated with the car.

- Use the AirTag data to verify or update the car's location automatically.

2.3 User Account & Data Management

- **User Authentication:**

- Enable user registration and login (possibly using Apple Sign-In or Firebase Authentication).

- **User Preferences:**

- Allow users to save and update their car's location, notification preferences, and family contacts.

- **Data Storage:**

- Store user settings, car locations, and alert histories securely in a backend database.

2.4 Backend Services & API Integration

- **Backend Infrastructure:**

- Use a service like Firebase or AWS to handle user data, authentication, and real-time notifications.

- **API Calls:**
- Implement backend functions to periodically query the San Francisco street cleaning API.
- Process and store the retrieved data for timely notifications.

---

3. Non-Functional Requirements

3.1 Performance

- The app should update car location and street cleaning data in near real-time.
- Notifications should be sent with minimal delay.

3.2 Scalability

- The backend must support multiple users and family sharing without performance degradation.

3.3 Security

- Ensure secure storage and transmission of user data (use encryption where applicable).
- Implement proper authentication and authorization for accessing user information and the AirTag data.

3.4 Usability

- The app must have an intuitive, easy-to-navigate interface.
- Clear, accessible alerts and timers that are easy for all age groups.

3.5 Compatibility

- Target iOS 14 and above.
- Ensure compatibility with Apple’s Find My service for AirTag integration.

3.6 Reliability

- The app must be resilient with minimal downtime.
- Provide mechanisms for error logging and recovery, especially for backend data fetching.

---

4. System Architecture Overview

4.1 Frontend

- **Development Language:** Swift for iOS.
- **Key Components:**
- Map integration (using Apple Maps or a similar framework).
- User interface elements for notifications, timers, and settings.

4.2 Backend

- **Platform:** Firebase or AWS.
- **Services:**
- Real-time database for storing user data and schedules.
- Cloud Functions (e.g., using Node.js/JavaScript) for API integration and sending notifications.
- Authentication service to manage user accounts.

4.3 Data Flow

- **Street Cleaning Data Flow:**
- The backend periodically calls the San Francisco street cleaning API, processes the data, and updates the database.
- **Location & Notification Flow:**
- The app reads the car’s location (either manually set or via AirTag data) and cross-references with the street cleaning schedule.
- When the car’s location matches a street cleaning schedule, the backend triggers notifications to the user and family members.

---

5. Assumptions & Constraints

- **API Limitations:**

- The San Francisco government API’s data frequency, rate limits, and geographic precision will affect notification timing.

- **AirTag Data Access:**

- Access to AirTag location data is subject to Apple’s Find My privacy guidelines and technical limitations.

- **User Device Requirements:**

- The app requires iOS devices with location services enabled and possibly specific hardware support for AirTag integration.

This requirements specification provides a foundational roadmap. You can refine and expand upon these requirements as you move forward with design and development. Let me know if you need more details on any specific section!