Glossary - Crime Reporting App

Term	Definition
Account	A registered user's profile including personal information like name, email, and phone.
API	Backend endpoints that handle client requests such as login, signup, report submission, etc.
Application	The mobile app used to report crimes, view locations, and interact with police data.
AsyncStorage	Local storage mechanism in React Native used to save tokens securely on the device.
authContext	React context that stores and provides authentication state throughout the frontend.
authService	Frontend service that manages user authentication (login/signup/token validation).
AlertsScreen	Displays a list of past crime reports submitted by the logged-in user.
Boundary Class	UI-facing class or screen that interacts directly with the user.
CrimeTypeGrid	UI component displaying a grid of crime categories that users can report.
CrimeReportModal	A modal screen that allows users to confirm report submission details.
CrimeType	An entity representing the title, ID, color, and icon of a crime category.
crimeReportService	Frontend service that submits crime reports, fetches top crimes, and gets nearest station and user email.
Control Class	Coordinates business logic and interacts with entities and UI (boundaries).
DB Connection	Refers to get_db_connection() used to connect to the MySQL database.
EditProfileScreen	UI screen allowing users to change their name, phone number, or password.

Entity Class	Core data structures used across the app (e.g., Report, Location, CrimeType).
FeedbackScreen	UI screen where users submit ratings and feedback messages.
FeedbackRequest	JSON object sent to the backend containing rating and message for user feedback.
findNearestPoliceStation	Use case that returns the closest NPC based on the user’s coordinates.
GeoJSON	Geospatial file format used in the app to store and parse police station location data.
index.tsx (Report Screen)	Main screen where user selects a crime, views location, and submits report.
Location	Entity holding GPS information (latitude, longitude, and human-readable address).
locationContext	React context that fetches current device location and converts to a readable address.
LoginResponse	Object returned from successful login API call. Contains token and user info.
LoginScreen	Initial screen where user enters email and password to access the app.
MapScreen	Displays a map with the user’s location and police stations. Allows user to manually change location.
Modal	A popup overlay UI element used to get confirmation from user (e.g., for submitting reports).
Most Common Crime	A statistical result showing which crimes occur most frequently in a given area.
MySQLDatabase	Backend SQL database used to store persistent data such as users, reports, and feedback.
NearestStation	JSON object returned by backend with NPC name and division code closest to the user.
NPCGeoJSONdata	The police station dataset file used in both backend and frontend for distance calculations and map display.
NPCDataService	Frontend utility to extract name, phone, and coordinates from raw police station GeoJSON data.

ProfileScreen	Displays name, email, phone, and options to logout, give feedback, or go to edit profile.
RankingController	Backend logic that queries top crime types based on location.
Report	Main entity for a submitted report — includes type, time, coordinates, and station.
ReportScreen	Common alias for index.tsx, where the reporting process begins.
Reverse Geocoding	Converting coordinates (latitude, longitude) into a human-readable address.
SMSController	Backend service that formats and sends SMS notifications to the nearest NPC upon successful crime report.
SignupScreen	UI screen to register a new user with email, name, phone number, and password.
Submit Feedback	The process of sending user ratings and comments to the backend.
useLocation()	Custom hook to retrieve and access the locationContext.
ValidationResult	API response returned when validating a user's email and password.
View Crime History	Allows users to see a timeline or list of their past crime reports.
View Crime Offenses	Feature to see common crime types in a region based on division.

A. App Architecture Terms	
Term	**Definition**
Boundary Class	A user interface component that interacts directly with the user (e.g., screens like LoginScreen, FeedbackScreen).

Control Class	Handles business logic and coordination between UI and data (e.g., authService, crimeReportService).
Entity Class	Represents core data used across the system (e.g., CrimeType, Report, FeedbackRequest).
BCE Model	A design pattern dividing the app into Boundary, Control, and Entity layers based on use case behavior.
AsyncStorage	External local storage on the device used for persisting user tokens.
API	A backend endpoint that handles frontend requests for login, signup, crime reporting, etc.
B. Frontend Context & Services	
Term	**Definition**
authContext	React Context that manages authentication state and user token storage.
locationContext	React Context that fetches and stores the current GPS coordinates and reverse geocoded address of the user.
authService	Frontend service that interacts with user login, signup, and token validation APIs.
crimeReportService	Frontend service that handles report submission, fetches top crimes, and nearest police station.
policeDataService	Extracts name, tel, type, and coordinates from GeoJSON police station entries.
C. Frontend Screens / Components	
Term	**Definition**
LoginScreen	UI screen for user login. Interacts with authService.
SignupScreen	UI screen for account creation. Interacts with authService.
EditProfileScreen	UI screen for editing name, phone, and password.

FeedbackScreen	Screen that allows users to submit app-related feedback.
AlertsScreen	Displays a user’s past crime report history.
MapScreen	Shows user location and nearest police stations on a map.
Index	Report screen file. Handles report UI logic and integrates CrimeTypeGrid, LocationInfo, and CrimeReportModal.
CrimeTypeGrid	Grid component showing selectable crime types.
LocationInfo	UI component showing current user location and nearest police station.
CrimeReportModal	Modal popup shown after selecting a crime type to confirm report.
D. Data Structures / Entities	
Term	**Definition**
CrimeType	Defines a type of crime with id, title, icon, and color.
Report	A crime report including type, location, police station, and timestamp.
Location	Represents a user's GPS data with latitude, longitude, and name.
NearestStation	Nearest police station info fetched from backend (name, divcode).
FeedbackRequest	JSON object containing user feedback message and rating.
LoginResponse / ValidationResult / CheckUserResponse	Responses returned from authentication APIs.
E. Backend Services & Controllers	
Term	**Definition**
auth.py	Backend controller handling login, signup, user validation.
location.py	Returns nearest police station based on GPS coordinates.
ranking.py	Backend controller to fetch most reported crimes in division.

sms.py	Sends SMS to nearest NPC on report confirmation.
F. External Resources	
Term	**Definition**
NPCGeoJSONdata	GeoJSON file of police stations (SingaporePoliceForceEstablishments2018GEOJSON.geojson). Used to locate and extract NPC coordinates and metadata.
MySQLDatabase	SQL database used to store user info, crime reports, and feedback. Connection managed by get_db_connection() from db.py.
policeStationsData	Parsed police station dataset from the GeoJSON file for frontend map display.
extractPoliceStationInfo()	Parses a police station feature and extracts name, coordinates, tel, and type.
G. App Logic Concepts	
Term	**Definition**
Find Nearest Police Station	Calculates which police station is geographically closest to the user. Uses backend and policeDataService.
Display Most Common Crime Type	Shows the top reported crimes in the user's area. Uses ranking controller and crimeReportService.
Send SMS Notification	Sends SMS with location and crime details to the NPC. Uses sms controller.
User Feedback Submission	Allows users to submit textual and rating feedback to backend.
History Retrieval	Loads previous reports submitted by user. Displays in AlertsScreen.