

3. Requirements Gathering

Stakeholder Analysis: Understanding Our Users



Citizens (End Users)

The primary beneficiaries, citizens require an intuitive interface to report incidents, access emergency contacts, and receive crucial safety updates. Their experience is paramount to the app's success.



Administrators (Authorities/Police)

These stakeholders are responsible for the backend operations, including reviewing submitted reports, updating their statuses (e.g., from "Pending" to "Investigating"), and providing official comments. Their efficiency directly impacts incident resolution.



System Admins

The technical backbone of the system, System Admins ensure the app's continuous operation. This includes maintaining the application infrastructure, managing Cloudinary storage for media, and configuring Firebase services.

User Stories: Citizen-Centric Design

User stories articulate the desired functionalities from the perspective of the end-user, ensuring that development efforts are aligned with real-world needs and expectations. Each story highlights a specific goal and its underlying motivation.

1 Verified Identity

As a citizen, I want to register using my National ID so that my identity is verified, ensuring accountability and trust within the system.

2 Comprehensive Reporting

As a citizen, I want to submit a report with text, location, and media (photos/videos) to thoroughly document an incident, providing clear evidence for authorities.

3 Report Tracking

As a citizen, I want to track the status of my submitted reports (e.g., Pending, Solved) to stay informed about the progress of my case.

4 Direct Emergency Access

As a citizen, I want to dial emergency numbers (Police, Ambulance, Fire) directly from the app for quick assistance during critical situations.

5 Cybersecurity Education

As a citizen, I want to read cybersecurity tips to protect myself from fraud, enhancing my personal safety and digital literacy.

6 Real-time Notifications

As a citizen, I want to receive notifications when my report status changes, ensuring I am always updated on important developments.

Functional Requirements: Core System Capabilities

Functional requirements define what the system must do, outlining the specific actions and behaviors necessary to fulfill user needs and business objectives. These are the actionable features that will be implemented.

1. Authentication

Secure sign-up/sign-in with Email & Password. Crucially, users must provide a valid 14-digit National ID and Phone Number for verification.

2. Report Management

Users can create detailed reports including Title, Description, Address, and attach various Media (Image/Video) as evidence.

3. Media Upload

Seamless integration with Cloudinary API for robust and secure storage of all report-related media evidence.

4. Dashboard

A personalized hub displaying "My Reports" history, valuable "Security Tips," and essential "Emergency Numbers" for quick access.

5. Emergency Contacts

One-tap dialing functionality for critical services like Ambulance (123), Police (122), Fire (180), and Electricity (121).

6. Notification System

Real-time alerts to inform users when the status of their submitted reports is updated by administrators.

7. Profile Management

Users can view their personal details and securely log out of their account, maintaining control over their information.

Non-functional Requirements: Quality Attributes

Non-functional requirements specify criteria that can be used to judge the operation of a system, rather than specific behaviors. They are crucial for ensuring the overall quality, usability, and reliability of the application.

Usability: Localization

The app supports localization in both English and Arabic, implemented via Flutter localizations, ensuring accessibility for a broader user base.

Reliability: Offline Persistence

Offline persistence is enabled in Firestore settings, allowing users to access and interact with certain data even without an active internet connection.

Security: Data Protection

Passwords are robustly protected via Firebase Auth, and National ID uniqueness is rigorously checked before registration to prevent duplicate accounts and enhance data integrity.

Performance: Smooth Experience

The application utilizes flutter_bloc for efficient state management, guaranteeing smooth UI updates, and Skeletonizer for loading states, providing a responsive and fluid user experience.

System Analysis & Design: Problem & Objectives

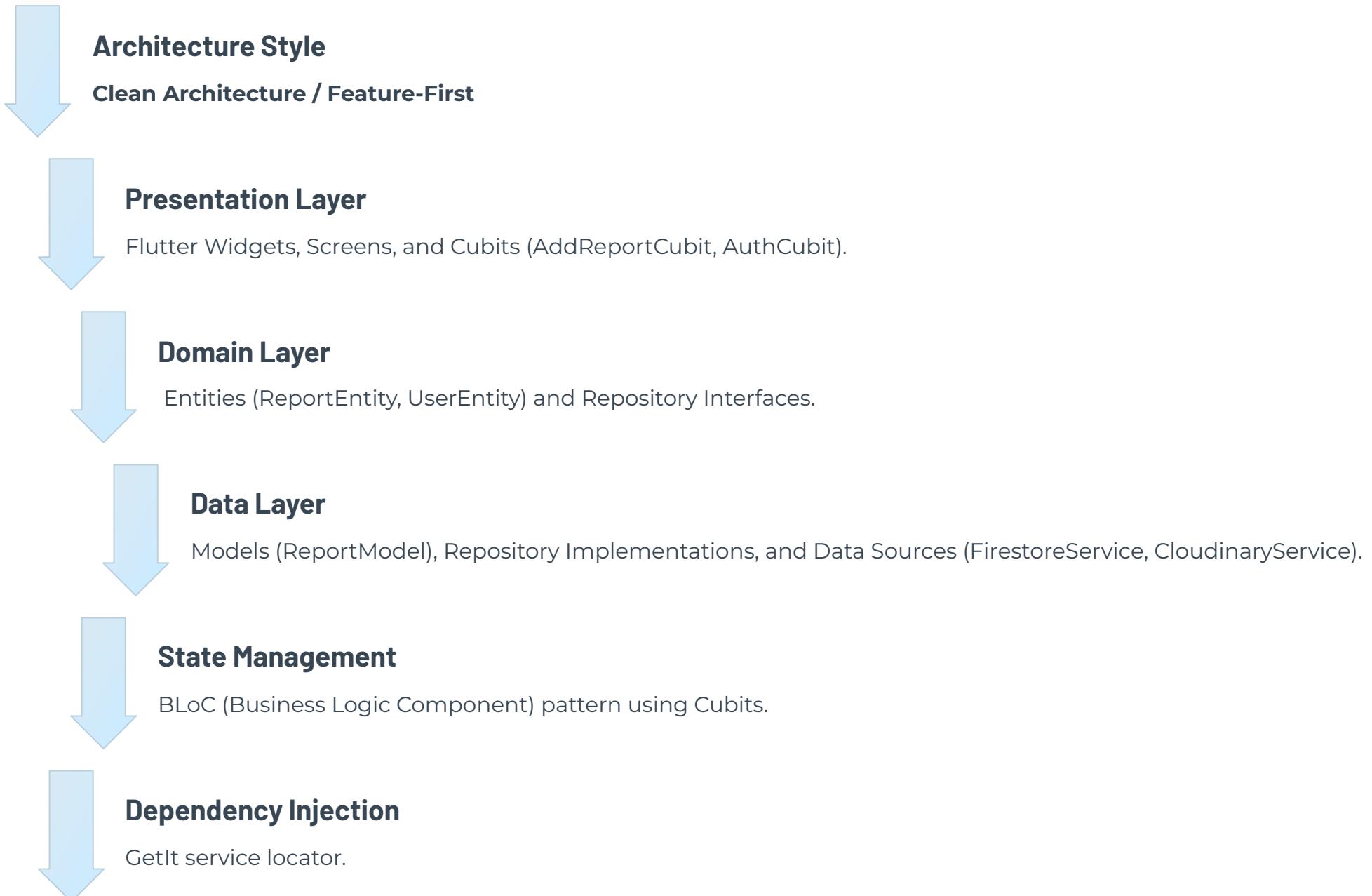
Problem Statement

Citizens often face difficulties in reporting non-emergency crimes (like cyber fraud or theft) quickly. Traditional methods lack transparency regarding the status of a report, and finding specific emergency numbers can be stressful during a crisis.

Objectives

- To provide a unified, accessible mobile platform for citizens to report incidents quickly and conveniently.
- To simplify and enhance the evidence collection process, allowing users to easily attach photos and videos to their reports.
- To offer real-time report status tracking, ensuring citizens are always informed about the progress of their submitted cases.
- To elevate public awareness regarding cybersecurity threats and fraud prevention through educational content within the app.

System Architecture: Building a Robust Foundation



System Use Cases: Citizen & Admin Interactions

Use cases detail the interactions between actors (citizens and admins) and the system, outlining specific functionalities and their descriptions. This provides a comprehensive view of how users will engage with the application.

Actor: Citizen (Mobile App User)

• Authentication & Account Management

- UC-01: Sign Up (Register) - Create a new account using personal details. Key Data: First Name, Last Name, Email, Phone, Password, and National ID (must be unique).
- UC-02: Sign In (Login) - Access the app using credentials. Constraint: User can login using either Email OR National ID.
- UC-03: Reset Password - Request a password reset link via email if credentials are lost.
- UC-04: View Profile - View personal account details (Name, Email, Phone, National ID).
- UC-05: Logout - Securely terminate the session and return to the login screen.

• Incident Reporting

- UC-06: Submit New Report - File a new complaint regarding a crime or fraud. Inputs: Title, Description, Location (Optional), and Media.
- UC-07: Attach Media Evidence - Upload Photos (Multiple) or Video (Single) to the report using Cloudinary storage.
- UC-08: View My Reports (History) - View a chronological list of all reports submitted by the logged-in user.
- UC-09: View Report Details - View the specific details of a single report, including its current status and Admin comments.

• Emergency & Safety

- UC-10: View Emergency Numbers - Access a list of critical contact numbers (Police, Ambulance, Fire, Electricity, Water).
- UC-11: Call Emergency Contact - Initiate a phone call to an emergency service directly from the app.
- UC-12: View Security Tips - Read educational content regarding cybersecurity and fraud prevention.

• Notifications

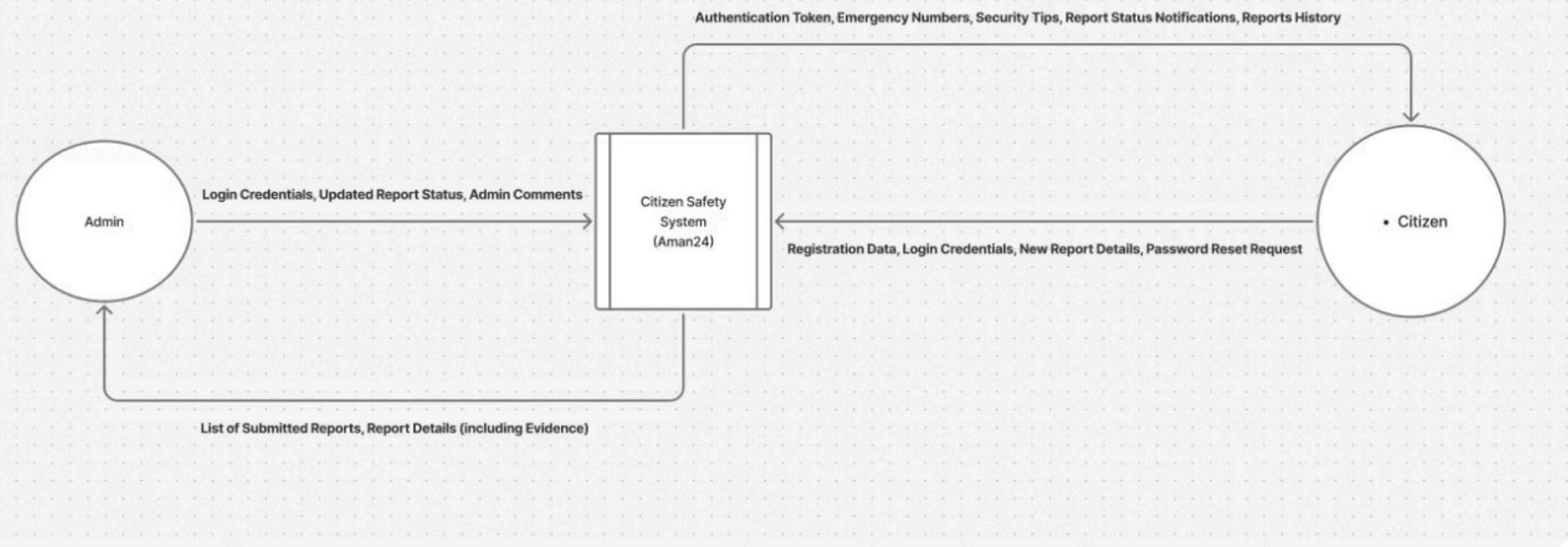
- UC-13: Receive Status Notifications - Receive real-time alerts when the status of a submitted report changes.
- UC-14: Mark Notification as Read - Clicking a notification marks it as read and directs the user to the relevant report.

Actor: Admin (System/Dashboard)

While the mobile app focuses on citizen interactions, the system's data structures confirm the existence of crucial admin functionalities.

- UC-15: View Submitted Reports - The system stores reports in a central collection accessible for review.
- UC-16: Update Report Status - Change a report's status from pending to inReview, investigating, resolved, or rejected.
- UC-17: Add Admin Comment - Attach an official comment or feedback to a report for the citizen to see.
- UC-18: Trigger Notification - System automatically notifies the specific user when the Admin performs UC-16 or UC-17.

Aman24 (DFD level 0)



Entities & Attributes: Data Model Foundation

The entity-attribute model defines the core data structures and their relationships within the system, ensuring data integrity and efficient management.

USER (Supertype)

Represents generalized account information to prevent data redundancy.

Primary Key: nationalId (String)

Attributes:

- uid: Unique identifier generated by Firebase Authentication
- firstName: First name of the user
- lastName: Last name of the user
- email: Contact email address
- phoneNumber: Contact phone number

CITIZEN (Subtype)

End-users of the mobile app who submit reports and receive safety alerts.

Primary Key/Foreign Key: nationalId (links to USER supertype)

Attributes: Inherits all attributes from USER

ADMIN (Subtype)

Authorities responsible for reviewing and managing submitted reports.

Primary Key/Foreign Key: nationalId (links to USER supertype)

Attributes: Inherits all attributes from USER

Note: Admins modify reports but don't persistently store a foreign key to specific Admin reviewers.

REPORT

Stores details of incidents reported by citizens.

Primary Key: reportId (UUID)

Foreign Keys: citizenId (links to CITIZEN who created the report)

Attributes: title, description, address (Optional), status, mediaUrls, adminComment, createdAt, updatedAt

NOTIFICATION

Alerts sent to citizens regarding updates to their reports.

Primary Key: id (String - same as doc ID)

Foreign Keys: recipientId (links to CITIZEN), reportId (links to REPORT)

Attributes: title, message, isRead, timestamp

Relationships: Connecting the Data

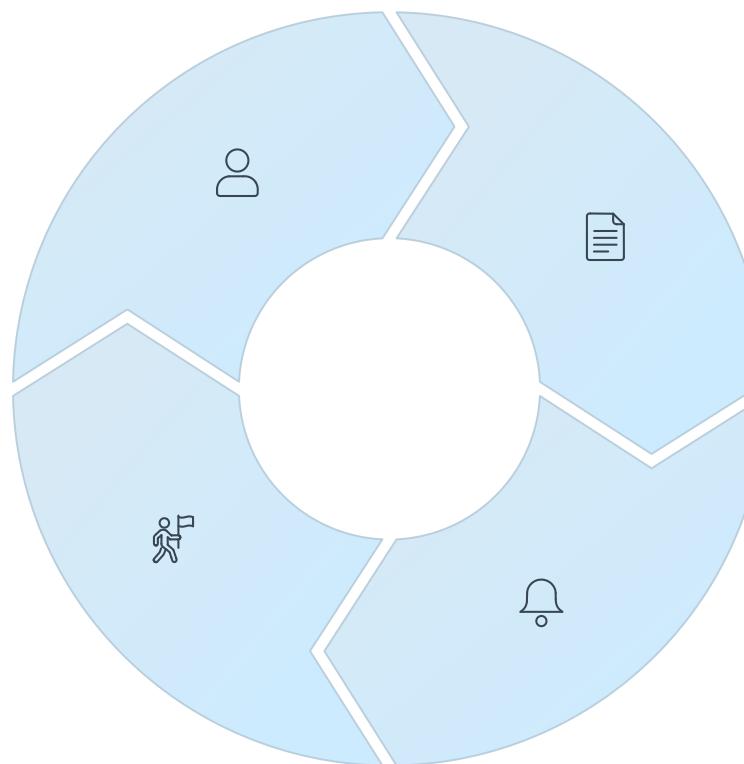
Understanding the relationships between entities is crucial for designing a cohesive and functional database. These connections define how different pieces of information interact within the system.

USER to ADMIN / CITIZEN

Is-A relationship (One-to-One): A user can be either an Admin or a Citizen, inheriting common attributes.

REPORT to NOTIFICATION

One-to-Many (1:N): A single report can generate multiple status notifications as it progresses through its lifecycle.

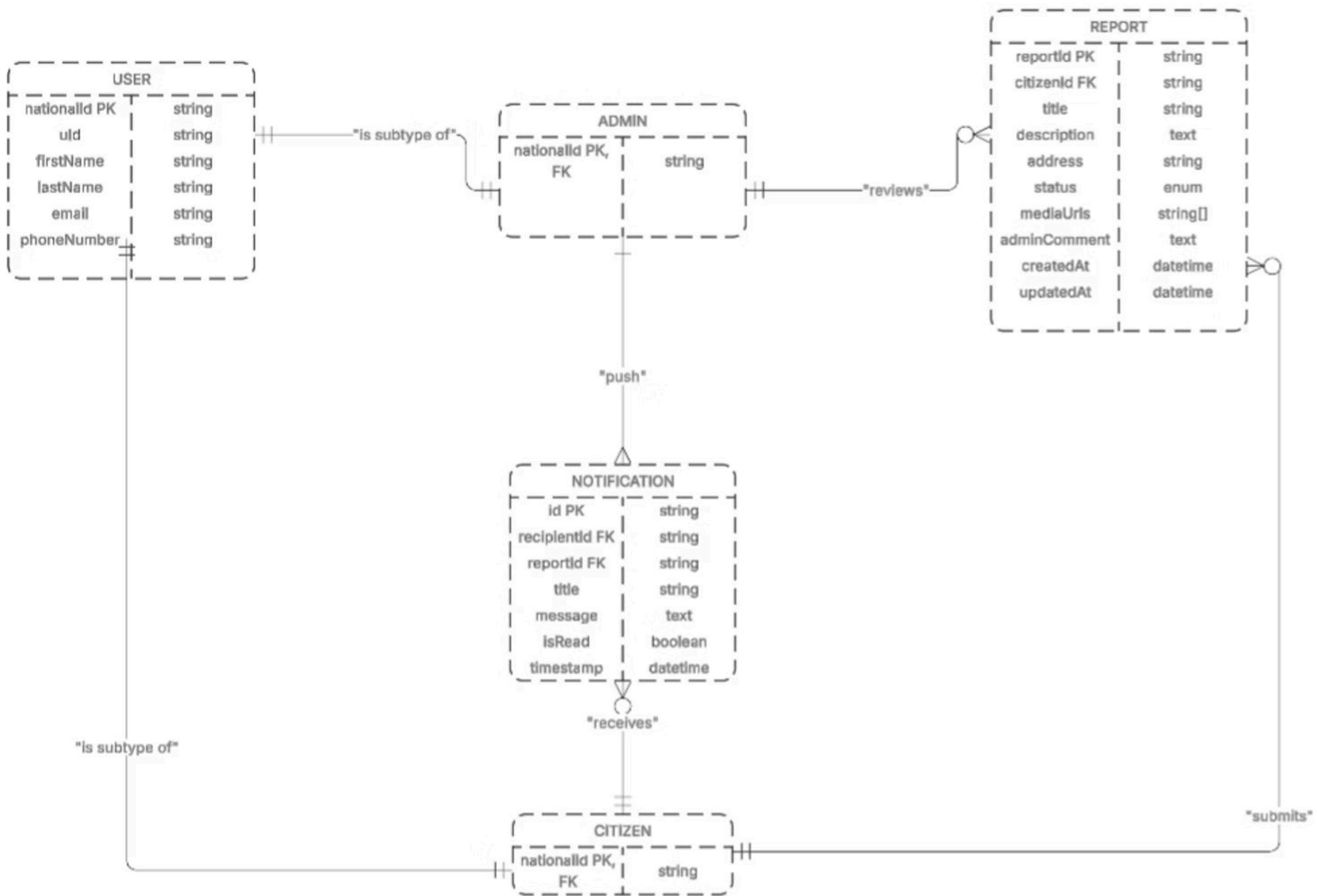


CITIZEN to REPORT

One-to-Many (1:N): A single citizen can submit multiple reports, each uniquely identified.

CITIZEN to NOTIFICATION

One-to-Many (1:N): A citizen can receive numerous notifications regarding their reports or general updates.



UI/UX Design

Figma Design System

Our comprehensive UI/UX design is documented in Figma, showcasing the complete design system, wireframes, and interactive prototypes for the incident reporting application. This includes all user interfaces for citizens and administrators, ensuring a cohesive and intuitive user experience across the platform.

[View Full Design on Figma](#)