CAP - Product Requirements Document

1. Overview

App Name: CAP (Community Alert and Protection)

CAP is a critical safety Progressive Web App designed specifically for women in public office who face unique security challenges during public engagements, rallies, town halls, and other official events. The application provides a discreet, real-time alert system that connects directly with security teams and law enforcement when users feel unsafe or threatened. The app integrates with wearable devices and includes advanced privacy protection features to ensure user safety even if the device is compromised.

Target Audience: Women political leaders, elected officials, public servants, and their security teams who require immediate emergency response capabilities during public appearances and official duties.

Key Problems Solved:

- Delayed emergency response during public events
- Inability to discreetly signal for help when under threat
- Lack of real-time location and context sharing with security teams
- Vulnerability when forced to disable safety applications under duress

2. Essential Core Features

Emergency Alert System

- One-tap panic button with customizable alert levels (Low, Medium, High, Critical)
- Automatic GPS location sharing with real-time tracking
- Background audio recording and streaming to security teams
- Pre-configured emergency messages with context about the situation
- Silent alert mode for discreet emergency communication

Wearable Integration

- Smartwatch compatibility for Apple Watch and Wear OS devices
- Discrete wearable tag support with one-button activation
- Heart rate monitoring integration for automatic stress detection
- Gesture-based activation (double-tap, shake, specific hand movements)

Decoy Security Features

- Fake home screen that appears when using decoy unlock pattern
- Hidden app functionality that appears as a normal utility app
- Duress password that triggers silent alerts while appearing to disable the app
- Automatic activation if app isn't checked-in within specified time intervals

Security Team Dashboard

- Real-time alert monitoring interface
- Live location tracking with map visualization
- Audio stream access for situational awareness
- Multi-channel communication (push notifications, SMS, email, phone calls)
- Response team coordination and dispatch features

Communication Features

- Direct messaging with security team members
- Video calling capabilities for real-time assessment
- Group chat functionality for event coordination
- Push-to-talk radio-style communication

Event Management

- Pre-event setup with location details and security protocols
- Schedule integration with automatic app activation during events
- Post-event reporting and incident logging
- Risk assessment tools and threat level indicators

3. Tech Stack

Front-End

- **React 18+** with TypeScript for robust PWA development
- **Tailwind CSS** for responsive, mobile-first design
- **Vite** for optimized build process and development experience
- **PWA Service Workers** for offline functionality and push notifications

Back-End

- **Supabase** for comprehensive backend services:
  - PostgreSQL database for user data, alerts, and incident logs
  - Real-time subscriptions for live location tracking and messaging
  - Authentication with role-based access control
  - Row Level Security for data protection
  - File storage for audio recordings and incident documentation

APIs & Integrations

- **Geolocation API** for precise location tracking
- **MediaDevices API** for audio recording capabilities
- **Web Push API** for emergency notifications
- **WebRTC** for real-time video communication
- **Google Maps API** for advanced mapping and location services
- **Twilio API** for SMS and voice call functionality
- **WebSocket connections** via Supabase for real-time data sync
- **Device sensors API** for accelerometer and gyroscope data
- **Web Bluetooth API** for wearable device connectivity

Technical Requirements

- PWA manifest for app-like experience on mobile devices
- Offline functionality for core emergency features
- End-to-end encryption for sensitive communications
- Battery optimization for background location tracking
- Cross-platform compatibility (iOS Safari, Android Chrome)

4. Design Preferences

Interface

The design should prioritize accessibility and speed with a clean, professional interface that maintains discretion while providing immediate access to emergency functions. Focus on large, easily-tappable buttons and high contrast elements for use in stressful situations.

Color Palette

- **Primary:** Deep Blue (#1e40af) - conveys trust and professionalism
- **Secondary:** Slate Gray (#64748b) - for secondary actions and text
- **Accent:** Emergency Red (#dc2626) - for critical alerts and danger states
- **Success:** Forest Green (#059669) - for safe status and confirmations
- **Background:** Light Gray (#f8fafc) and Pure White (#ffffff)

Typography

- **Headings:** Inter Bold - modern, highly legible sans-serif
- **Body Text:** Inter Regular - consistent font family for optimal readability
- **Interface Elements:** Inter Medium - for buttons and navigation

Additional Design Considerations

- High contrast mode support for visibility in various lighting conditions
- Large touch targets (minimum 44px) for stress situations
- Minimalist iconography with universally understood symbols
- Dark mode support for discrete usage
- Accessibility compliance (WCAG 2.1 AA)

5. All Screens/Pages

Authentication Screens

/login - Login Screen

- **UI Elements:** Email input field, password input field, "Remember Me" checkbox, Login button, "Forgot Password" link, biometric login option (fingerprint/face recognition)
- **Entry:** App launch, logout, session expiration
- **Exit:** Successful login → Dashboard, "Forgot Password" → Password Reset

/register - Registration Screen

- **UI Elements:** Personal information form (name, email, phone, organization), password creation fields, role selection (Official, Security Team), terms acceptance checkbox, Create Account button
- **Entry:** Login screen "Sign Up" link, invitation links

- **Exit:** Successful registration → Profile Setup, "Back to Login" → Login Screen

/forgot-password - Password Reset Screen

- **UI Elements:** Email input field, Reset Password button, verification code input (conditional), "Back to Login" link
- **Entry:** Login screen "Forgot Password" link
- **Exit:** Reset email sent → Login Screen, successful reset → Login Screen

Main Application Screens

/dashboard - Main Dashboard

- **UI Elements:** Current status indicator (Safe/At Risk/Emergency), Quick panic button (prominent), Active event card, Recent alerts list, Security team status, Weather/location widget, Quick actions menu
- **Entry:** Successful login, bottom navigation, home button
- **Exit:** Navigation menu → other screens, Panic Button → Emergency Alert

/emergency - Emergency Alert Screen

- **UI Elements:** Large panic button with alert level selector, Current location display with map, Active recording indicator, Security team response status, Cancel alert button (with confirmation), Additional information input field, Quick message templates
- **Entry:** Panic button activation, wearable trigger, automatic stress detection
- **Exit:** Alert resolution → Dashboard, False alarm → Dashboard with incident log

/profile - User Profile

- **UI Elements:** Profile photo and edit option, Personal information fields, Emergency contact management, Security preferences, Wearable device management, Account security settings, Privacy controls
- **Entry:** Navigation menu, settings
- **Exit:** Save changes → Dashboard, Navigation → other screens

/events - Event Management

- **UI Elements:** Upcoming events calendar, Create new event button, Event details cards (date, location, security level), Active event indicator, Past events list with incident reports, Event templates
- **Entry:** Navigation menu, dashboard event card

- **Exit:** Event details → Event Detail Screen, Navigation → other screens

/event-detail/:id - Event Detail Screen

- **UI Elements:** Event information display, Location with map integration, Security team assignments, Risk assessment indicators, Pre-event checklist, Activate event button, Edit event details, Emergency protocols specific to event
- **Entry:** Events list, dashboard active event
- **Exit:** Activate event → Dashboard with active status, Back → Events Screen

/security-dashboard - Security Team Dashboard

- **UI Elements:** Real-time alerts feed, Active officials monitoring grid, Map view with all tracked users, Communication panel, Response team status, Alert history and analytics, Dispatch controls
- **Entry:** Navigation (security team members only), emergency alerts
- **Exit:** Navigation → other screens, Alert response → specific alert details

/communications - Communications Hub

- **UI Elements:** Message threads list, Group chat access, Video call interface, Push-to-talk controls, Security team contacts, Emergency broadcast messages, Message search and filters
- **Entry:** Navigation menu, alert notifications, direct messages
- **Exit:** Navigation → other screens, call initiation → video interface

/settings - Settings Screen

- **UI Elements:** Alert preferences, Privacy and security options, Wearable device configuration, Notification settings, Decoy mode setup, Emergency contacts management, App permissions, Data backup options
- **Entry:** Navigation menu, profile settings
- **Exit:** Apply settings → Dashboard, Navigation → other screens

/wearables - Wearable Device Management

- **UI Elements:** Connected devices list, Device pairing interface, Battery status indicators, Gesture configuration, Test alert functions, Device-specific settings, Troubleshooting guide
- **Entry:** Settings menu, profile screen, setup wizard
- **Exit:** Save configuration → Settings, Navigation → other screens

/decoy-setup - Decoy Mode Configuration

- **UI Elements:** Fake interface customization, Duress password setup, Hidden activation methods, Testing interface, Security explanation, Recovery options setup
- **Entry:** Settings menu, initial setup wizard
- **Exit:** Save decoy settings → Settings, Test mode → Decoy Interface

/decoy-home - Decoy Interface (Hidden Mode)

- **UI Elements:** Fake app interface (calculator, notes, or weather app), Hidden emergency activation areas, Subtle status indicators, Background functionality controls
- **Entry:** Duress activation, forced app access
- **Exit:** Hidden deactivation → Real Dashboard, Time-based → Real Dashboard

/reports - Incident Reports

- **UI Elements:** Incident history timeline, Detailed report cards, Export/download options, Statistical analysis, Filter and search capabilities, Follow-up actions tracker
- **Entry:** Navigation menu, post-incident workflows
- **Exit:** Report details → specific incident, Navigation → other screens

/help - Help and Support

- **UI Elements:** Emergency procedures guide, FAQ sections, Video tutorials, Contact support options, Device troubleshooting, Quick start guide, Safety tips
- **Entry:** Navigation menu, error states, first-time setup
- **Exit:** Navigation → other screens, Support contact → external communication

6. App Menu and Navigation Structure

Primary Navigation System

Bottom Tab Navigation - Fixed bottom navigation bar for core functionality:

- **Home** (Dashboard icon) - Main dashboard and status
- **Events** (Calendar icon) - Event management and scheduling
- **Communications** (Message icon) - Chat and communication hub
- **Emergency** (Shield icon) - Quick access to emergency features
- **Profile** (User icon) - Settings and profile management

Secondary Navigation

Hamburger Menu - Accessible from top-left corner for additional features:

- Settings
- Wearable Devices
- Incident Reports
- Help & Support
- Security Dashboard (Security team only)
- Decoy Mode Setup
- Logout

Navigation Hierarchy

1. **Level 1:** Bottom tab navigation for primary functions
2. **Level 2:** Screen-specific actions and sub-features
3. **Level 3:** Detail screens and configuration panels
4. **Emergency Override:** Panic button accessible from all screens via floating action button

User Movement Flow

- **Primary Flow:** Users navigate between main sections via bottom tabs
- **Contextual Flow:** Related features accessible through in-screen navigation
- **Emergency Flow:** Emergency features always accessible via persistent panic button
- **Settings Flow:** Configuration screens accessed through hamburger menu
- **Back Navigation:** Browser back button and in-app back arrows for screen hierarchy

7. User Flow

Initial Setup and Registration Flow

1. **App Launch:** User opens SYNCA PWA on their mobile device
2. **Welcome Screen:** Brief introduction to app purpose and security features
3. **Registration:** User creates account with role selection (Official or Security Team)
4. **Profile Setup:** Complete personal information, emergency contacts, and organization details

5. **Wearable Integration:** Connect smartwatch or wearable device through Bluetooth pairing
6. **Security Configuration:** Set up decoy mode, duress passwords, and privacy preferences
7. **Permission Requests:** Grant location access, microphone, camera, and notification permissions
8. **Tutorial:** Interactive walkthrough of emergency features and panic button usage

Daily Usage Flow - Event Preparation

1. **Dashboard Access:** User logs in and views current safety status
2. **Event Creation:** Navigate to Events screen and create new public engagement
3. **Event Details:** Input location, duration, expected attendance, and security level
4. **Security Team Assignment:** Select and notify relevant security personnel
5. **Pre-Event Check:** Review emergency protocols and test wearable device connectivity
6. **Event Activation:** Enable active monitoring 30 minutes before event start

Emergency Alert Flow - Critical Path

1. **Threat Detection:** User feels unsafe and needs immediate assistance
2. **Alert Initiation:** Activate panic button via app, wearable device, or gesture
3. **Alert Level Selection:** Choose urgency level (Low/Medium/High/Critical) - defaults to High
4. **Automatic Data Collection:** App immediately captures GPS location, starts audio recording, and prepares emergency message
5. **Security Team Notification:** Instant alerts sent via push notification, SMS, and email to assigned security team
6. **Real-Time Monitoring:** Security team receives live location tracking and audio stream
7. **Response Coordination:** Security team dispatches appropriate response and maintains communication
8. **Situation Resolution:** User or security team marks incident as resolved
9. **Post-Incident Report:** Automatic generation of incident documentation for review

Decoy Mode Flow - Under Duress

1. **Forced App Access:** User is compelled to show or disable the app under threat
2. **Duress Activation:** User enters special unlock pattern or duress password

3. **Fake Interface Display:** App shows convincing decoy screen (calculator, weather, etc.)
4. **Silent Alert Trigger:** Background emergency alert sent to security team without visible indication
5. **Continued Monitoring:** App maintains location tracking and audio recording in background
6. **Security Response:** Security team receives silent alert and initiates discrete response
7. **Situation Assessment:** Security team evaluates threat level and coordinates appropriate intervention

Communication Flow - Ongoing Incident

1. **Alert Active State:** Emergency alert is active with security team responding
2. **Communication Access:** User can access messaging interface during incident
3. **Status Updates:** Security team provides real-time updates on response progress
4. **Additional Information:** User can send text updates, photos, or additional context
5. **Video Call Option:** Security team can initiate video call for visual situation assessment
6. **Coordination Updates:** Multiple security team members coordinate through group chat
7. **All-Clear Confirmation:** User confirms safety and situation resolution

Post-Incident Flow - Documentation and Follow-up

1. **Incident Conclusion:** Emergency situation resolved and users marked safe
2. **Immediate Debrief:** Quick assessment of response effectiveness
3. **Report Generation:** Automatic creation of detailed incident report with timeline
4. **Data Review:** Security team reviews audio recordings, location data, and communications
5. **Follow-up Actions:** Identify any necessary security improvements or protocol changes
6. **User Check-in:** Wellness check and support resource provision
7. **System Updates:** Update security protocols and user preferences based on learnings
8. **Archive Documentation:** Secure storage of incident data for future reference and analysis

📄 Preparing the CAP App for Submission – Summary

To ensure the CAP App is **submission-ready and professional**, the following steps should be completed:

1. Git Repository & Version Control

- Organize the repository into clear modules:
  - /frontend – React + TypeScript PWA
  - /backend – Supabase schema, SQL policies, API routes, server functions
  - /security-dashboard – Admin/security interface
  - /wearables – Wearable device integration scripts
  - /docs – PRD, architecture, diagrams
  - /tests – Unit, integration, and security tests
- Adopt a clean branching strategy:
  - main (stable), dev (active), feature/* (per module)
- Use descriptive commit messages documenting features (e.g., *"feat: add silent duress alert trigger"*).
- Tag the final release as **v1.0 – CAP Public Safety PWA**.

---

2. README File (Submission-Ready)

Your README should include:

- **Project Title**: CAP – Community Alert and Protection
- **Project Description**: Summary of the safety PWA for political leaders
- **Problem Statement & Solution**
- **Key Features**:
  - Panic button, silent duress mode, wearable triggers
  - Real-time tracking, audio streaming

  - Event management, security team dashboard, decoy mode
- **Tech Stack** (React, Tailwind, Supabase, WebRTC, Web Bluetooth, etc.)
- **Setup Instructions** (environment variables, Supabase setup, running PWA)
- **Usage Guide** (screenshots: dashboard, event setup, panic flow)
- **Team Roles** (devs, UI/UX, backend, QA)
- **Future Improvements** (geo-fencing, AI threat analysis, SOS drone integration)
- Add **screenshots/GIFs** of:
  - Emergency alert flow
  - Decoy mode
  - Wearable activation
  - Security dashboard

---

3. Demo / Video Walkthrough

Record a **5–10 minute professional demo** covering:

- User registration & onboarding
- Setting up decoy mode and wearables
- Creating and activating an event
- Triggering emergency alerts:
  - Panic button
  - Silent duress mode
  - Wearable activation
- Real-time security dashboard view
- Incident resolution & report generation

Alternatively, deploy a **live staging demo** for reviewers.

---

4. Project Pitch / Presentation

Prepare a polished presentation including (**Strictly follow the PLP standard pitch deck template as instructed**:

- Problem statement (women in public office face unique safety risks)
- CAP solution overview
- Key differentiator features (duress mode, wearable integration, decoy interface, live streaming)
- Architecture & tech stack
- Demo screenshots
- Security & privacy strategy (RLS, E2E encryption)
- Future roadmap

---

5. Documentation

Provide comprehensive documentation:

- **API Endpoints** (Swagger/OpenAPI)
- **Supabase ERD** (users, events, alerts, incident reports, device sessions)
- **System Architecture Diagram** (PWA → Supabase → Security Dashboard)
- **User Flow Diagrams** (Emergency flow, Decoy flow, Wearable flow)
- **Testing Reports**:
  - Unit tests for UI components
  - Backend logic tests
  - Security tests (RLS, permissions, duress triggers)

---

6. Final Quality Checks

Before submission, verify:

- Clean, optimized, accessible UI
- Responsive design across mobile/tablet
- Reliable background location tracking

- Stable audio streaming under low connectivity
- Battery-optimized service workers
- No console errors or API failures
- Seamless decoy mode transitions
- Wearable triggers functioning correctly
- Versioning and CI checks passing

---

7. Additional Professional Touches

To elevate the submission:

- Add **official CAP logo & icon set**
- Include a **CHANGELOG.md** tracking all releases
- Add **CONTRIBUTORS.md** with team acknowledgments
- Provide **LICENSE.md** (MIT, Apache 2.0, or custom security license)
- Create a **Security Disclosure Policy** for responsible reporting

---

✅ Submission Package Checklist

Required:

- Clean, well-structured Git repository
- Full README + documentation
- Demo video or staging environment
- Pitch deck (PDF or slides)
- Fully functional PWA with backend
- Unit/integration tests + coverage reports
- Changelog + contributors file