

Implementation Plan - Visit Geo-fencing and Cancel Flow Enhancement

Overview

- Update the officer visit detail page to implement strict geo-fencing (25m radius), visual GPS status indicator, and a structured cancellation flow with mandatory documentation for specific cancellation reasons.

User Review Required

- **IMPORTANT**
- Geo-fencing Distance: Changing from 30m (backend constant) to 25m as specified. This requires updating both frontend validation and backend `GEO_FENCE_THRESHOLD_METERS` constant.
- **WARNING**
- Selfie Upload Requirement: For cancellation reasons (Shifted, Passed Away, Not Present), a selfie upload will be mandatory. Officers cannot proceed without uploading a photo.

2.

Proposed Changes

Backend Updates

- [MODIFY] `visitController.ts`
- Line 15: Update geo-fence threshold from 30m to 25m

- `const GEO_FENCE_THRESHOLD_METERS = 25; //`
`Changed from 30`
- Lines 60-82: Update error message to reflect 25m requirement

- `throw new AppError('Officer must be within 25 meters of the citizen location to start/complete the visit', 400);`

Frontend Updates

- [MODIFY]
- page.tsx
- New State Variables (after line 68):
 -
 - `cancelDialogOpen: boolean` - Controls cancel dialog visibility
 - `cancelReason: string` - Selected cancellation reason
 - `cancelRemarks: string` - Remarks for cancellation
 - `cancelSelfie: File | null` - Uploaded selfie file
 - `distanceFromCitizen: number | null` - Calculated distance in meters
 - `isWithinRange: boolean` - Whether officer is within 25m
- New Helper Function (after line 143):
 -
 - `const calculateDistance = (lat1: number, lng1: number, lat2: number, lng2: number): number => {`
 - `const R = 6371000; // Earth's radius in meters`
 - `const dLat = (lat2 - lat1) * Math.PI / 180;`

- const dLng = (lng2 - lng1) * Math.PI / 180;
- const a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(lat1 * Math.PI / 180) * Math.cos(lat2 *
Math.PI / 180) *
- Math.sin(dLng/2) * Math.sin(dLng/2);
- const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
- return R * c; // Distance in meters
- };
- Update
- refreshLocation
- function (lines 123-143):
-
- Calculate distance from citizen's location
- Set isWithinRange based on 25m threshold
- Update distanceFromCitizen state
- Update GPS Button UI (lines 330-333):
-
- Show "GPS OK" in green when isWithinRange === true
- Show distance in meters when not in range
- Disable "Start Visit" button when not in range
- New Cancel Dialog Component (replace lines 344-357):
-
- Dialog with 4 radio options: Shifted, Passed Away, Not
Present, Reschedule
- Conditional rendering of remarks textarea (for first 3
options)
- Conditional rendering of selfie upload (mandatory for
first 3 options)
- Validation to ensure selfie is uploaded before
submission

- Update
- handleStartVisit
- (lines 195-219):
 - Add geo-fence validation before allowing start
 - Show error toast if not within 25m range
 - Redirect to assessment form on successful start
- New handleCancelVisit function:
 - Upload selfie to backend (if provided)
 - Call cancel API with reason, remarks, and selfie URL
 - Handle different cancellation reasons appropriately

Verification Plan

Automated Tests

1. Test distance calculation function with known coordinates
2. Verify GPS status updates correctly based on location
3. Test cancel dialog validation (selfie required for specific reasons)

Manual Verification

- **Geo-fencing Test:**
 - Navigate to visit detail page
 - Verify "GPS OK" shows when within 25m (or in development mode)
 - Verify "Start Visit" button is disabled when out of range

● **Start Visit Flow:**

- Click "Start Visit" when GPS OK
- Verify redirect to assessment form (same page, status changes to IN_PROGRESS)

● **Cancel Visit Flow:**

- Click "Cancel" button
- Select "Shifted" → Verify remarks field and selfie upload appear
- Try to submit without selfie → Verify error message
- Upload selfie and submit → Verify success

● **Reschedule Flow:**

- Select "Reschedule" option
- Verify no selfie requirement
- Verify date/time picker appears