CCSA_Tech_Spec_v1.24

Title: Crosswalk Church Safety Alerts (CCSA) - Technical Specification (Version 1.24)
Date: 08-29-2025
Time: 11:45 AM MST
Approved: 08-29-2025
Author: Tim Caldwell with G (Grok, xAI)

1. Introduction

1.1 Background

The *Crosswalk Church Safety Alerts (CCSA)* is a Proof of Concept (POC) to monitor safety incidents near **Crosswalk Lutheran Ministries**, which holds its services on Sundays at the Caesar Chavez High School Auditorium at 3921 W. Baseline Rd, Laveen, AZ 85339. The mailing address is PO Box 13, Laveen, AZ 85339. The system is configurable for additional sites, using free Google Suite and Google App Server (GAS) tools. Built on a previous POC, *CWLK_Safety_Monitor_v03.03.14* by **Tim Caldwell**, this POC leverages proven capabilities of pulling data from multiple free sources, aiming for a cost-free initial deployment.

1.2 Objectives

1.2.1 Visualize local and national incidents with keyword-based alerts for the Safety Team.
1.2.2 Integrate free local data sources via APIs or site scraping.
1.2.3 Provide an interactive dashboard for up to 25 users, with potential for scaling.
1.2.4 Develop and deploy a no-cost POC.

1.3 Scope

1.3.1 Conduct a weekly 'Friday Night Review' containing all of the weeks events of note to Safety Team Leader and Operations Manager.
1.3.2 Proactively prevent security/safety issues.
1.3.3 Enable real-time alerts on Sundays from 6 AM to 1 PM.
1.3.4 Use 2025 sample data for testing.
1.3.5 Exclude paid or offline components.

2. Requirements

2.1 Functional Requirements

2.1.1 Incident Visualization

2.1.1.1 Display a color-coded map (red = high risk, green = low) with an admin-configurable radius (default 1 mile).
2.1.1.2 Show incident details in pop-ups: date (MM-DD-YYYY), keyword, risk level (0-10), sentiment (positive/negative/neutral).
2.1.1.3 Use sample data from *CWLK_Safety_Monitor_v03.03.14*.

2.1.2 Data Integration

2.1.2.1 Aggregate data from *incident_data*, *alert_log*, and news sources (e.g., AZFamily, X).
2.1.2.2 Match keywords (e.g., “shooting”), geolocate using latitude/longitude, and assign a unique ID using a hash of timestamp (publishing time from *RawPubDate*) and location (e.g., `Utilities.computeDigest(Utilities.DigestAlgorithm.MD5,` ${timestamp}_${location}`).toString()`). If no location is provided, use the geofence center (33.375898, -112.145216) with a “NoLoc” flag for manual review.
2.1.2.3 For multiple sources reporting the same event with different locations (content similarity >80% via text comparison), average latitudes/longitudes; otherwise, flag as separate incidents.
2.1.2.4 Control the unique Arizona Timezone (UTC-7) using a server-side .json configuration file.
2.1.2.5 Standardize timestamp handling to MM-DD-YYYY hh:mm a z, retaining *RawPubDate* for unprocessed sources.

2.1.3 User Interaction

2.1.3.1 Enable filtering by date range, keyword, or location via GAS HTML dropdowns.
2.1.3.2 Support zoom (0.1-10 miles) and pan on the map using Leaflet.js events.
2.1.3.3 Provide a Kanban/dashboard UI, configurable for DEV/QA via GAS parameters.
2.1.3.4 The dashboard auto-updates every 60 seconds using GAS timers, with an optional manual refresh button.

2.2 Non-Functional Requirements

2.2.1 Performance

2.2.1.1 Ensure the map loads in under 2 seconds for 25 concurrent users on standard broadband.

2.2.2 Scalability

2.2.2.1 Handle up to 5,000 daily updates; maximum estimated at 10,000,000 cells based on Google Sheets limits.

2.2.3 Security

2.2.3.1 Restrict access to authorized users with JSON Web Token (JWT) authentication.
2.2.3.2 Use HTTPS via GAS with TLS 1.3 encryption.
2.2.3.3 Implement role-based access control (RBAC) in Google Sheets.
2.2.3.4 Audit logs for unauthorized access attempts, triggering email alerts if more than 3 attempts per hour.
2.2.3.5 Sanitize all user inputs with GAS to prevent SQL injection.

3. System Architecture

3.1 Frontend

- **Frontend**: GAS HTML with Leaflet.js, palette (#80AC9E, #CB935B, #D14A42, #2B565A, #DCDCE2) matching the selected logo.

3.2 Backend

- **Backend**: GAS, Google Sheets.

3.3 Database

All sheets support DEV (development), TEST (testing), and PROD (production) environments via script properties (e.g., `PropertiesService.getScriptProperties().getProperty('env') = 'DEV'` for table switches).

3.3.1 Tables per Environment

- DEV: dev_incident_data, dev_safety_zones, dev_alert_log, dev_keywords, dev_audit_log, dev_error_log, dev_backup_log.
- TEST: test_incident_data.
- PROD: incident_data, safety_zones, alert_log, keywords, audit_log, error_log, backup_log. Notes: DEV/TEST tables have frozen/hidden id/hash_id; PROD no freezing/hiding. Output diagnostics and counter bug fixed for clarity.

3.4 Geospatial Services

- Use OpenStreetMap with Leaflet.js for free tile layers, cached in GAS memory.
- Implement GAS geocoding with a custom function using Google Maps Geocoding API.

4. Data Management

4.1 Data Sources

- News: AZFamily, FOX 10 Phoenix.
- Sample: Chavez Park Lake (May 25, 2025).

4.2 Table Structures

4.2.1 dev_incident_data

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	-64,-109,9,-2,-53,-128,-71,122,118,-93,-82,-6,73,33,62,25	MD5 hash for data integrity
C	timestamp	DATETIME	NOT NULL	08-23-2025 10:06 PM MST	Incident time in MST format
D	location	VARCHAR(100)		35th Ave & Baseline	Location
E	latitude	DECIMAL(9,6)		33.375898	Latitude
F	longitude	DECIMAL(9,6)		-112.145216	Longitude
G	type	VARCHAR(50)		Mock Type	Incident type
H	description	VARCHAR(255)		Mock Desc	Description
I	source	VARCHAR(50)		Mock Source	Data source
J	keyword	VARCHAR(50)		mock	Matched keyword
K	risk_level	INTEGER		5	Risk score (0-10)

Column	Field	Type	Constraints	Example	Description
L	sentiment	VARCHAR(20)		neutral	Sentiment analysis (positive/negative/neutral) from X posts or sources
M	RawPubDate	VARCHAR(255)		08-23-2025 10:06 PM MST	Original publish date
N	noLocErrorProcessed	BOOLEAN	NOT NULL	TRUE	Flags if NoLoc error processed

Notes: Stores mock incident data for dev; id/hash_id frozen/hidden; redundant "campus" column removed.

4.2.2 dev_safety_zones

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	Utilities.computeDigest(...)	MD5 hash for data integrity
C	zone_name	VARCHAR(50)		Laveen Mock	Geofence area name
D	center_lat	DECIMAL(9,6)	NOT NULL	33.375898	Center latitude
E	center_lon	DECIMAL(9,6)	NOT NULL	-112.145216	Center longitude
F	radius	INTEGER	NOT NULL	1	Mile radius

Column	Field	Type	Constraints	Example	Description
G	active	BOOLEAN	NOT NULL	TRUE	Zone status

Notes: Defines safety zones for dev; id/hash_id frozen/hidden; no redundant "Campus" column.

4.2.3 dev_alert_log

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	Utilities.computeDigest(...)	MD5 hash for data integrity
C	timestamp	DATETIME	NOT NULL	08-23-2025 10:06 PM MST	Alert time in MST format
D	incident_id	INTEGER	FOREIGN KEY to dev_incident_data.id	1	Linked incident
E	alert_type	VARCHAR(20)		High Risk	Alert category
F	description	VARCHAR(255)		Shooting detected	Alert description
G	action	VARCHAR(50)		Flag	Action taken

Notes: Logs alerts for dev; id/hash_id frozen/hidden.

4.2.4 dev_keywords

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	Utilities.computeDigest(...)	MD5 hash for data integrity
C	keyword	VARCHAR(50)	NOT NULL	shooting	Matched keyword
D	weight	INTEGER	NOT NULL	2	Priority weight
E	include_exclude	BOOLEAN	NOT NULL	TRUE	Include or exclude flag
F	status	VARCHAR(20)		active	Keyword status

Notes: Manages keywords for dev; id/hash_id frozen/hidden.

4.2.5 dev_audit_log

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	Utilities.computeDigest(...)	MD5 hash for data integrity

Column	Field	Type	Constraints	Example	Description
C	timestamp	DATETIME	NOT NULL	08-23-2025 10:06 PM MST	Audit time in MST format
D	action	VARCHAR(50)	NOT NULL	Data Update	Action performed
E	details	VARCHAR(255)		Updated data	Action details
F	user	VARCHAR(50)		Admin	User performing action

Notes: Audits actions for dev; id/hash_id frozen/hidden.

4.2.6 dev_error_log

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	timestamp	DATETIME	NOT NULL	08-23-2025 10:06 PM MST	Error time in MST format
C	error	VARCHAR(100)	NOT NULL	Update Fail	Error type
D	incident_id	INTEGER		1	Linked incident ID
E	details	VARCHAR(255)		Data mismatch	Error details
F	error_details	VARCHAR(255)		Detailed error description	Description of error

Notes: Logs errors for dev; id frozen/hidden (no hash_id).

4.2.7 dev_backup_log

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	Utilities.computeDigest(...)	MD5 hash for data integrity
C	timestamp	DATETIME	NOT NULL	08-23-2025 10:06 PM MST	Backup time in MST format
D	url	VARCHAR(255)		https://drive.google.com/...	Backup URL
E	status	VARCHAR(20)		Success	Backup status
F	campus	VARCHAR(50)		all	Associated campus
G	backup_status	VARCHAR(50)		Completed	Backup completion status

Notes: Tracks backups for dev; id/hash_id frozen/hidden; 5 mock rows.

4.2.8 test_incident_data

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	Utilities.computeDigest(...)	MD5 hash for data integrity
C	timestamp	DATETIME	NOT NULL	08-23-2025 10:06 PM MST	Incident time in MST format
D	action	VARCHAR(50)		Test Action	Action performed
E	details	VARCHAR(255)		Test Details	Details

Notes: Stores test incident data; id/hash_id frozen/hidden.

4.2.9 backup_log

Column	Field	Type	Constraints	Example	Description
A	id	INTEGER	PRIMARY KEY, AUTO_INCREMENT	1	Unique identifier
B	hash_id	VARCHAR(255)	NOT NULL	Utilities.computeDigest(...)	MD5 hash for data integrity
C	timestamp	DATETIME	NOT NULL	08-23-2025 10:06 PM MST	Backup time in MST format

Column	Field	Type	Constraints	Example	Description
D	action	VARCHAR(50)	NOT NULL	Data Backup	Action performed
E	details	VARCHAR(255)		Backup completed	Details

Notes: Non-dev backup log; no freezing/hiding specified.

4.3 Interdependencies

- dev_incident_data links to dev_alert_log via incident_id (FOREIGN KEY).
- Keywords in dev_keywords used for matching in dev_incident_data.
- Safety zones in dev_safety_zones used for geofencing latitude/longitude checks.

4.4 Triggers

- onEdit in GAS for real-time audit to dev_audit_log.
- Time-based for backups (every week), monitoring (every 60 mins).

4.5 Audit Features

- dev_audit_log tracks all changes (onEdit trigger).
- Sheet_Audit (4.5.4) with real-time logging.

4.6 Automated Backups

- Weekly CSV to Drive, triggered by GAS (backup_script with environment selector).
- UI for manual "Backup Now".
- Logs to dev_backup_log with campus.

4.7 Data Retention

- 90 days in sheets, 180 in backups—GAS archive script.

4.8 Multi-Property Management

- Configurable for Laveen/Phoenix sites, with unique branding/logos in CCSA_Logos folder.

5. Deployment

5.1 Infrastructure

- Google Sheets/GAS, no-cost.

5.2 Deployment Steps

- Create sheets/triggers via setupSheets/setupTriggers.
- Deploy GAS scripts, test with sample data.

6. Use Cases

- Emergency alerts, mapping, trends, responses (detailed in spec).

7. Incident Prevention Strategy

- Monitor keywords/X trends, train team, ethical data handling.

8. System Operations

- Scripts: Audit, Monitor, Alerts.
- UI: Dashboard, admin tools.

9. Development

- ES5 JS, JSDoc headers, GAS setup.

9.1 Overview

9.1.1 A GAS/JavaScript POC for safety monitoring.

9.2 Setup

9.2.1 Create Google Sheets with tabs using setupSheets.gs.
9.2.2 Deploy GAS scripts, set triggers.
9.2.3 Multiple Scripts and Log: Includes setupSheets.gs, setupTriggers.gs, setup_log.
9.2.4 Configurable Prioritization: Stores settings in script properties.
9.2.5 Version Control: Version_Log tracks deployments.

9.3 Coding Guidelines

9.3.1 ES5 JavaScript: Uses ES5 for GAS compatibility.
9.3.2 JSDoc: Enhances code with /** ... */ comments.

9.4 Scripts

9.4.1 Deduplication:

```
/**
 * Project: Crosswalk Church Safety Alerts (CCSA)
 * Version: v1.1
 * Date: 07-12-2025
 * Last Updated: 02:53 AM MST, 07-21-2025
 * Author: Tim Caldwell with Grok (xAI)
 * Description: Manages sheet audits, documentation, and alert
deduplication.
 * Usage: deduplicateAlerts() removes duplicates.
 * Notes: Builds on CWLK_Safety_Monitor logic.
 * Dependencies: SpreadsheetApp, Utilities
 * @param {Object} alertData - Contains timestamp and location.
 * @returns {boolean} - True if added, false if duplicate.
 */
function deduplicateAlerts(alertData) {
  const sheet =
SpreadsheetApp.getActiveSpreadsheet().getSheetByName("alert_log");
  const ids = sheet.getRange("A2:A").getValues().flat();
  const id = Utilities.computeDigest(Utilities.DigestAlgorithm.MD5,
`${alertData.timestamp}_${alertData.location ||
'33.375898,-112.145216'}`).toString();
  if (ids.includes(id)) return false;
  sheet.appendRow([id, alertData.type, new Date()]);
  return true;
}
```

9.5 Testing

9.5.1 Run with 10 sample incident_data records, validate map rendering, simulate 25-user load.

10. Product Management

10.1 Product Vision

10.1.1 A free POC by Tim Caldwell to enhance church safety.

10.2 Roadmap

10.2.1 Setup (Week 1).
10.2.2 Build (Weeks 2-3).
10.2.3 Test (Weeks 4-6).
10.2.4 Beta by Aug-Sep 2025, v1.1 by Sep 2025.

10.3 Requirements Prioritization

10.3.1 High: Alerts, config.
10.3.2 Medium: UI.

10.4 Metrics

10.4.1 Target 90% alert accuracy, validated by 100 incidents.
10.4.2 Load Testing for 25 Users: Simulate 25 tabs, monitor with Logger.log().

11. Appendices

11.1 Incident Data (2025 Samples)

Timestamp	**Keyword**	**Incident**	**Location**	**Source**
06-24-2025 12:11 AM MST	Traffic Accident	Motorcycle crash	16th Street and Camelback Road, Phoenix, AZ	AZFamily
05-26-2025 03:45 PM MST	cesar chavez	Body found	Alvord Lake at Cesar Chavez Park, Laveen, AZ	AZFamily
07-08-2025 09:00 AM MST	shooting	2 shot	Laveen Village, Phoenix, AZ	AZFamily

11.2 Potential Data Sources

11.2.1 News: AZFamily, FOX 10 Phoenix.
11.2.2 RSS: AZFamily RSS, FOX 10 Phoenix RSS.
11.2.3 Social Media: X.
11.2.4 CommunityCrimeMap.

11.3 Branding Assets

11.3.1 Official logos received from Senior Pastor Dan on 07-17-2025, approved for use. Store in Google Drive’s “CCSA_Logos” folder, linked by property ID in *safety_zones*, with a sync script (*syncLogos.gs*) updating hourly.
11.3.2 Note: Images (PNG/JPG, max 500KB) are for official branding; formal set pending further approval if needed.

11.4 Specification Change Log

Date	Version	Change	Rationale
08-15-2025	v1.0	Initial draft with basic structure, objectives, requirements, architecture, and appendices.	Established baseline framework for CCSA POC, initiated by Tim Caldwell during planning session; focused on free tools and safety monitoring.
08-18-2025	v1.1	Added phx_fix column to dev_incident_data for Phoenix grid tolerance; updated geospatial services with geocodeAddress.	Enhanced handling of location data in Phoenix area to improve accuracy; based on user feedback for multi-campus support.
08-20-2025	v1.2	Standardized timestamps to MM-DD-YYYY hh:mm a MST across all logs and tables; retained RawPubDate.	Ensured consistent time formatting for Arizona area UTC-7, improving data integration and readability; tested with sample data conversion.
08-22-2025	v1.3	Added campus_attempted and attempted campus ID to dev_error_log for multi-campus tracing; updated error logging behavior.	Supported multi-property management by tracking campus context in errors; aligned with user request for better debugging in multi-site setups.
08-24-2025	v1.4	Introduced DEV/TEST/PROD table separation in 3.3.1 with script properties for switches; removed redundant columns.	Enhanced environment clarity and scalability; reflected user feedback to avoid data contamination across dev/test/prod.
08-25-2025	v1.5	Added threshold JSON config for risk_levels; integrated with admin UI.	Allowed configurable alert thresholds via server-side JSON; improved flexibility for safety team adjustments.
08-25-2025	v1.6	Added NoLoc/phx_fix validations in QA scripts; expanded testing with erring conditions.	Streamlined location error handling and QA automation; per user input to elevate POC without breaking free-tier limits.

Date	Version	Change	Rationale
08-25-2025	v1.7	Aligned dev_error_log to 7 columns (id, timestamp, error, incident_id, details, error_details, campus_attempted); fixed data shift.	Prevented column misalignment in logs; based on user feedback for integrity in error tracking.
08-25-2025	v1.8	Reinstated phx_fix in dev_incident_data; detailed Phoenix grid tolerance.	Re-added for consistent location processing in Phoenix area; addressed user request after testing.
08-25-2025	v1.9	Added Latest_timestamp column H in Overwatch sheet for tracking updates.	Enabled better monitoring of data freshness; improved trend analysis capabilities.
08-25-2025	v1.10	Introduced config.json for timestamp settings; server-side timezone control.	Centralized configuration for Arizona UTC-7; simplified maintenance and standardization.
08-25-2025	v1.11	Fixed 'id' in safety_zones for unique integrity; AUTO_INCREMENT ensured.	Ensured no duplicate zones in multi-campus; per user feedback for data consistency.
08-25-2025	v1.12	Removed manual month fix in timestamp parsing; automated handling.	Streamlined code for efficiency; reduced error-prone manual adjustments.
08-25-2025	v1.13	Added fallback to America/Arizona timezone in config.	Guaranteed UTC-7 consistency across all operations; fallback for edge cases.
08-26-2025	v1.14	Fixed 'dd' format for day of month in timestamps.	Ensured accurate date formatting; addressed parsing bugs from threads.
08-26-2025	v1.15	Cleaned timestamp parsing logic; removed redundancies.	Improved code efficiency and error-free conversion; based on dev debugging.
08-26-2025	v1.16	Enhanced timestamp debug logging in scripts.	Added troubleshooting tools for QA; facilitated faster issue resolution.

Date	Version	Change	Rationale
08-26-2025	v1.17	Regex fix for case-insensitivity in keyword matching.	Allowed flexible matching (e.g., 'Shooting' = 'shooting'); user-requested for robustness.
08-26-2025	v1.18	Strict timestamp validation rules added.	Prevented invalid data entry; enhanced data integrity.
08-26-2025	v1.19	Sync'd with updated GAS script properties for env switches.	Aligned DEV/TEST/PROD; improved multi-env seamless operation.
08-26-2025	v1.20	Directed logs to dev_audit_log for all changes.	Centralized auditing; integrated with onEdit trigger for real-time.
08-26-2025	v1.21	Added debug for timestamps/config in QA scripts.	Provided tools for timestamp troubleshooting; per dev threads.
08-26-2025	v1.22	Merged integrity checks into single script.	Consolidated QA for efficiency; reduced code duplication.
08-26-2025	v1.23	Consolidated header/row validations in tables.	Ensured data integrity fixes across all sheets; addressed shift issues.
08-29-2025	v1.24	Expanded Change Log details; gSite automation milestone (uploads/render success); widened columns for landscapes in gDocs.	Granular history for handoff; aligned to dev threads and user request for better readability in landscape mode.

Note: All changes expanded with detailed change/rationale from threads; v1.5-1.24 captured. `error_log` corrected; script headers updated. Timestamps standardized; change log widened for landscapes in gDocs.