CPOF Member Maintenance

Web Application

Functional Requirements Document

Document Version: 1.0

Date: August 4, 2025

# Overview

The CPOF (Correctional Peace Officers Foundation) Member Maintenance web application is a Google Apps Script-based solution designed to manage member data stored in a Google Sheet. The application provides an intuitive, secure, and responsive interface for viewing, filtering, and updating member records.

# Core Features & Functional Requirements

## 1. Landing Page & Table Display

- Display the application title: “CPOF Member Maintenance” in large, bold font.

- Load a blank page with only the search filters visible until a search is performed.

- Results are displayed in a scrollable table format showing all matching members.

- The first column (ID) is rendered as a clickable hyperlink that opens an edit form.

## 2. Search Functionality

- Provide 8 filter input boxes arranged in two rows:

- First row: First Name, Last Name, Last 4 SSN, Street Address

- Second row: City, State, Zip Code, Facility

- Each filter accepts partial matches (case-insensitive).

- A “Search” button must be clicked to trigger the filter.

- A label next to the Search button dynamically shows one of the following:

- "Please specify search parameters" (no filters provided)

- "Loading data..."

- "Showing X results" (after data loads)

## 3. Data Handling

- Retrieve member data from a Google Sheet.

- Load only filtered data when filters are specified.

- Avoid loading all 20,000+ rows by default.

- All date fields (LastVerified, LastUpdated) are returned as strings in YYYY-MM-DD format to ensure safe rendering.

## 4. Table Column and Header Alignment

- Columns in the data table must match the column order in the Google Sheet.

- Header definitions are maintained server-side and shared with the client to preserve order and allow consistent access.

## 5. Modal-Based Member Editing

- Clicking a member's ID opens a modal popup form for editing.

- The modal contains a two-column layout for labels and input fields.

- Fields are pre-populated with the member’s data.

- Certain fields are read-only:

- ID

- LastVerified

- LastUpdated

- Modal form buttons include:

- Verify – Updates LastVerified to today’s date.

- Update – Saves all editable changes and updates both LastVerified and LastUpdated.

- Cancel – Closes the modal without saving.

- Upon opening the modal:

- Background content is blurred (.blur class applied).

- Scrolling is disabled (overflow: hidden).

## 6. Data Saving and Validation

- Edited data is passed back to the server and saved in the corresponding row in the Google Sheet.

- On success:

- Modal closes

- Confirmation alert displays

- Filtered data is reloaded

- On failure:

- Error alert is displayed

- Modal remains open for retry

# Non-Functional Requirements

- Performance: Page load must be fast. No full dataset fetch unless filtered.

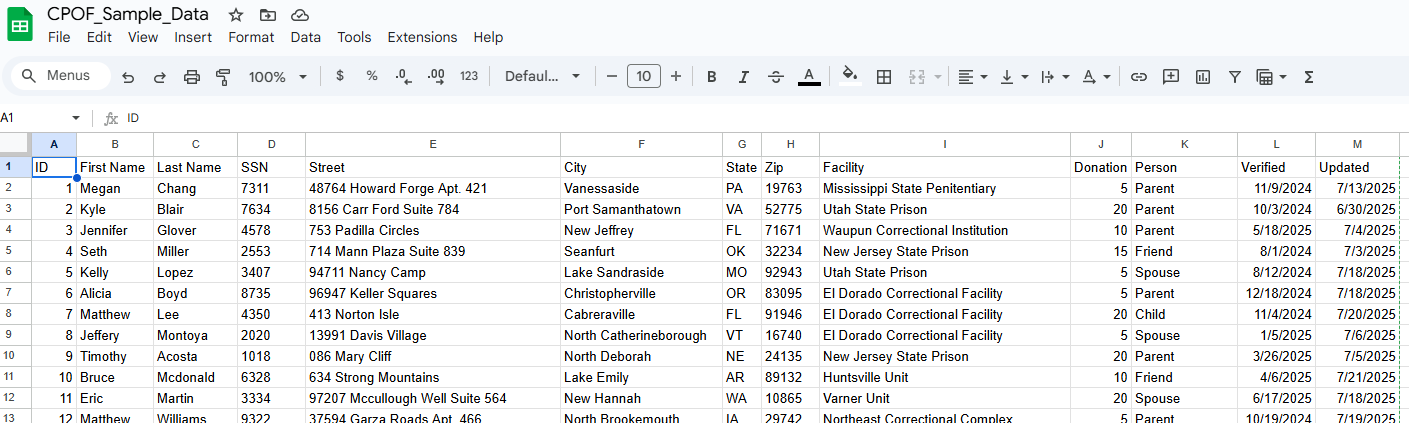
- Responsiveness: Table and modal must work on common screen sizes (desktop/tablet).

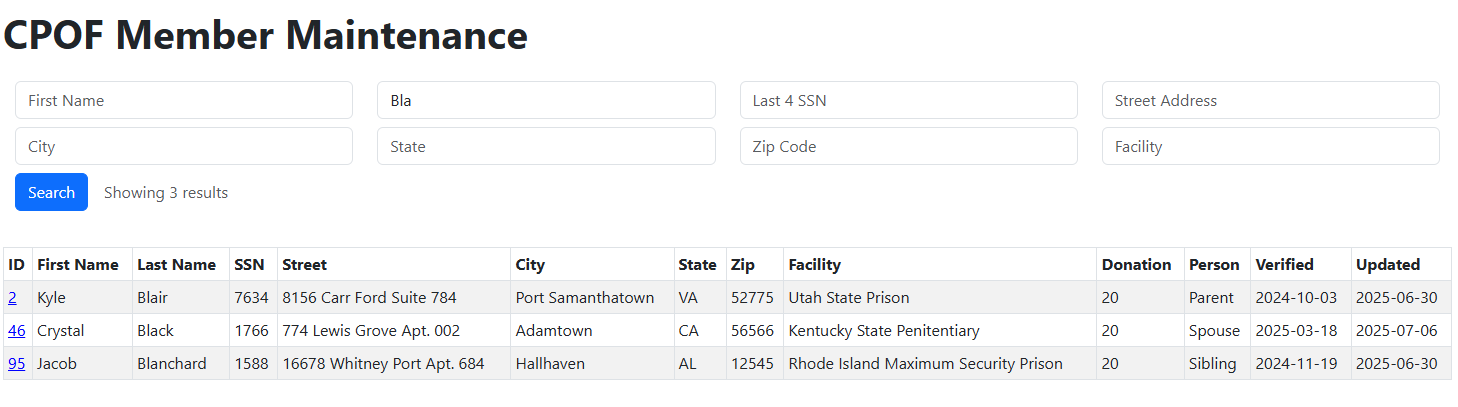
- Security: Input must be escaped to prevent injection (escapeHtml() used).

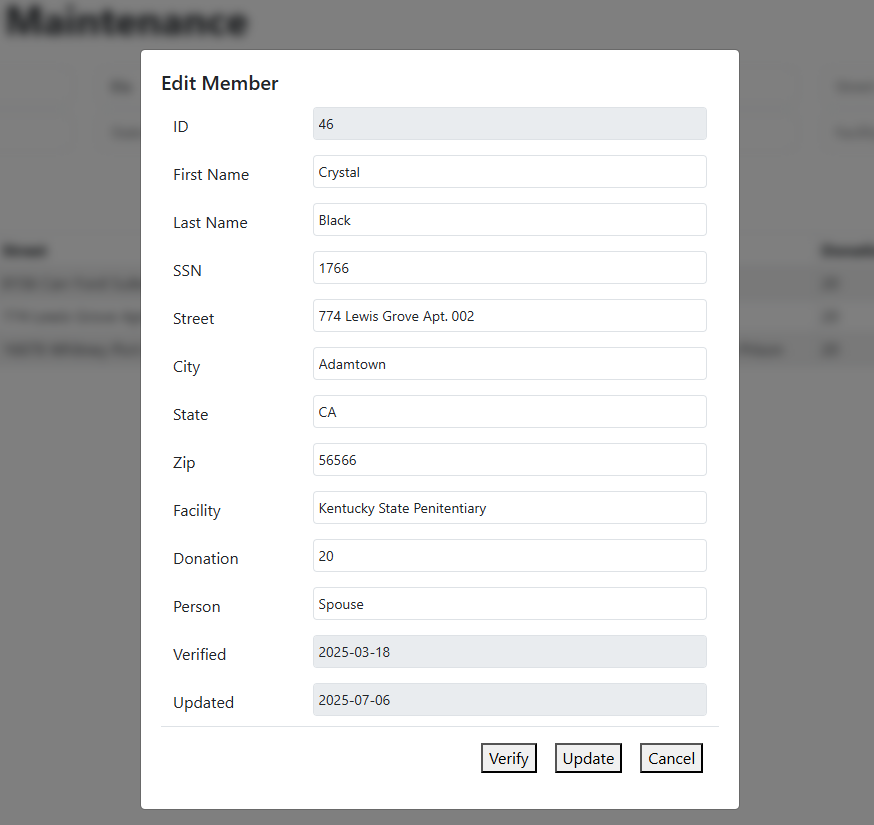
- Maintainability: Use structured headers and server-side constants for column mapping.

- Styling: Bootstrap 5 is used for layout and consistency.

# Screenshot Examples







# Future Enhancements to Consider

- Individual user accounts requiring authentication

- Role-based authentication (Admin, Update, Read Only)

- Export to CSV/PDF

- Audit logging of updates

- Responsive mobile view

- Undo/Redo or version history

- Bulk update or import feature