System Documentation for AID Organization

Table of Contents

---

1. Introduction

This document provides a comprehensive overview of the public-facing website for the American Islamic Diversity (AID) organization. The system is designed to facilitate the collection of data from recipients (families in need) and volunteers, as well as to manage the delivery process of aid. The system includes a public website, an admin interface, and a backend database to store and manage all relevant data.

The public website allows recipients to register and update their information, while volunteers can sign up to assist with deliveries. The admin interface provides tools for managing recipients, volunteers, and delivery logistics.

Key Features:

- **Multilingual Support: English, Arabic, Farsi, Urdu, Spanish, Myanmar, Pashto.**
- **Dynamic Map Interface: Real-time visualization of recipients for volunteers to select and deliver to, and pickup locations.**
- **SMS Integration: Automated messages for registrations, reminders, and delivery updates via Twilio.**

- **Admin Controls: Logging, volunteer/recipient management, and assigning volunteers for delivery.**
- **Scalable Database: MySQL-driven architecture with triggers, indexes, and archival tables.**

User Roles:

1. **Recipients: Families/individuals requesting aid.**
2. **Volunteers: Individuals delivering aid.**
3. **Admins: Superusers managing the platform.**

System Architecture

1 Frontend Components

- **Languages: HTML, CSS, JavaScript, PHP (for dynamic rendering).**
- **Libraries:**
  - **Leaflet.js: Interactive map rendering with clustering for dense locations.**
  - **OpenCage Geocoder: Converts addresses ↔ latitude/longitude.**
  - **MapBox: For address autocorrect.**
  - **Font Awesome: Icons for UI elements.**
- **Responsive Design: Works on mobile and desktop.**

2 Backend Components

- **Database: MySQL (tables: recipient, volunteer, delivery, logs, etc.).**
- **APIs:**
  - **Twilio: SMS/call handling, IVR menus, and message logging.**
  - **PHP Scripts: CRUD operations, session management, and data validation.**
- **Security:**
  - **Admin authentication via admin table.**
  - **Volunteer/recipient sessions for editing data.**

Database Structure

The system uses a MySQL database to store all data, all the queries will be contained in DB.sql file. Below is a breakdown of the key tables and their purposes:

1. Admin Tables

- **admin: Stores admin login credentials.**
- **admin_logs: Logs actions performed by admins (e.g., login, updates, deletions).**
- **admin_settings: Stores settings such as enabling/disabling volunteer selection for deliveries, only this one is the only thing.**

2. Volunteer Tables

- **volunteer: Stores volunteer information, including contact details, preferences, and approval status.**

- **volunteer_archive: Archives deleted volunteer records.**

3. Recipient Tables

- **recipient: Stores recipient information, including contact details, address, and approval status.**
- **recipient_details: Stores additional details about recipients, that wouldn't be needed to be fetched in some important functions, (e.g., household information, income, etc.).**
- **recipient_children: Stores information about the children of recipients.**
- **recipient_archive: Archives deleted recipient records.**

4. Delivery Tables

- **delivery: Tracks the status of deliveries, linking volunteers to recipients(contains exact number of recipients in the recipient table).**
- **delivery_archive: Archives all the delivery table rows that were used at x time.**
- **delivery_logs: Logs delivery-related actions (e.g., selection, confirmation, completion).**

5. SMS and Logging Tables

- **sms_templates: Stores SMS message templates for recipients and volunteers so admins can reuse them.**
- **sms_logs: Logs all SMS messages sent and record the latest received SMS after it.**
- **location_logs: Logs volunteer location data when they access the delivery page.**
- **collect_location: Stores pickup location details for deliveries.**

6. Indexes and Triggers

- **Indexes are created on frequently queried columns (e.g., recipient.approved, delivery.status).**
- **A trigger (after_recipient_insert) automatically creates a delivery record when a new recipient is added.**

Public Website Functionality

The public website consists of four main pages:

1. Recipient Registration Form

- **Purpose: Allows families in need to register for assistance.**
- **Fields: Name, phone, email, address, language preference, and other relevant details (the children form is only shown when num of children value is above 0).**
- **Process:**
    1. **User selects their preferred language.**
    2. **User fills out the form and submits it.**
    3. **Data is saved to the recipient, recipient_details, and recipient_children tables.**
    4. **An SMS confirmation is sent to the recipient.**

2. Volunteer Registration Form

- **Purpose: Allows individuals to sign up as volunteers.**
- **Fields: Name, phone, email, zip code, preferences (e.g., car size, availability), language preference, and notification-preference (all selected values will be saved in the same column separated with a comma).**
- **Process:**
    1. **User selects their preferred language.**

2. User fills out the form and submits it.
3. Data is saved to the volunteer table.
4. An SMS confirmation is sent to the volunteer.

3. Recipient Edit Page

- Purpose: Allows recipients to update their information.
- Process:
  1. User enters their phone number to access their record.
  2. User can edit address-related info, name, email, and phone number.
  3. Updates are saved to the recipient table.

4. Volunteer Delivery Page

- Purpose: Allows volunteers to view and manage deliveries.
- Features:
  - Map Interface: Displays the volunteer's location (red marker), pickup locations (yellow circle), and recipients (blue and green markers).
  - Showed recipient constraint: Only recipients who are approved, they replied with ”Yes”, and if the admin enabled the selection option they will see none selected recipient.
  - Selection Process:
    - Volunteers can select recipients (blue markers) if the admin has enabled the selection option.
    - Selected recipients are marked with green markers and are temporarily reserved for 10 minutes.
    - If a volunteer does not confirm a selected recipient within 10 minutes, the recipient becomes available for other volunteers.
  - Summary Panel:
    - Displays pickup location details and selected recipients.
    - Volunteers can confirm pickup and delivery actions.
    - Once all deliveries are confirmed, volunteers can view an optimized route for up to 10 recipients.

Admin Interface

The admin interface provides tools for managing recipients, volunteers, and deliveries. It consists of seven pages:

1. Login Page

- Admins log in using credentials stored in the admin table.

2. Admin Dashboard

- Sections:
  1. Manage:
     - Delivery Table: View and manage delivery statuses.
     - Pickup Locations: Add, remove, or activate pickup locations.
  2. Archives:
     - View archived records for deliveries, recipients, and volunteers.
  3. Logs:

        - **View admin logs, Admin Messages, All Messages, delivery logs, and volunteer location logs.**
        - **Inside All Messages: messages sent, received or failed to be sent to either recipient, volunteer. All the "All Messages" data are fetched from Twilio API.**

3. Volunteer and Recipient Pages

- **SMS Panel: Admins can send SMS messages to selected recipients or volunteers using templates or custom messages.**
- **Tables: Display filter options, and a selection needed columns that would be important for the SMS messages selection. Tables also contain actions for recipients or volunteers with options to edit, approve, disapprove, or delete records.**

4. Rec.xlsx and Vol.xlsx Pages

- **Excel Export: Admins can download volunteer or recipient data as Excel files.**
- **Tables: Display all columns from the data base of recipients or volunteers with options to edit, approve, disapprove, or delete records.**

4. Admin Delivery Page

- **Purpose: Allows admins to assign recipients to volunteers manually.**
- **Features:**
    - **Admins can enter a volunteer's phone number to log in as that volunteer.**
    - **Admins can view all approved and replies with "Yes" recipients on the map and assign them to volunteers.**
    - **Black markers indicate recipients already assigned to other volunteers.**

API Integrations

The system integrates with several third-party APIs to enhance functionality:

1. Twilio

- **Purpose: Handles SMS messaging and phone calls.**
- **Features:**
    - **Sends automated SMS messages to recipients and volunteers upon registration.**
    - **Logs all sent and the latest received (after sending) SMS messages in the sms_logs table.**
    - **Forwards phone calls to designated employees based on the caller's selection (e.g., language, who to forward the call to).**
    - **Provides a webhook to receive SMS replies and call data.**

2. Leaflet (OpenStreetMap)

- **Purpose: Displays interactive maps for the delivery process.**
- **Features:**
    - **Shows volunteer locations, pickup points, and recipient locations on a map.**
    - **Uses clustering to group nearby recipients into a single marker for better visibility.**
    - **Provides hoverable menus for selecting or deselecting recipients.**

3. OpenCageData

- **Purpose: Converts addresses into geographic coordinates (geocoding).**

- **Features:**
  - **Converts manually entered addresses into latitude and longitude for storage in the database.**
  - **Reverse geocodes coordinates from the "Locate Me" feature to display the nearest address.**

4. MapBox

- **Purpose: Auto-correction for address input.**
- **Features:**
  - **It helps recipients in New Recipient Page with correcting their address inputs.**

5. Font Awesome

- **Purpose: Provides icons for the user interface (e.g., language selection, logout).**

Folder Structure

The system is organized into the following folders and files:

1. Root Folder

- **Files:**
  - **deliver.css, deliver.js, deliver.php: Handle the volunteer delivery page.**
  - **general.css, general.js: Contain global styles and scripts.**
  - **header.php: Provides a consistent header across all public pages.**
  - **logo.png: Organization logo displayed in the header.**
  - **recipientform.php, volunteerform.php: Handle recipient and volunteer registration forms.**
  - **recipientpage.php, recipientpage.js, recipientpage.css: Handle the recipient edit page.**

2. php Folder

- **Files:**
  - **add_recipient.php: Adds recipient data to the database and sends an SMS confirmation.**
  - **add_volunteer.php: Adds volunteer data to the database and sends an SMS confirmation.**
  - **db.php: Manages database connections.**
  - **edit_recipient.php: Updates recipient data.**
  - **fetch_recipients.php: Retrieves recipient data for the delivery page.**
  - **volunteer_login.php: Manages volunteer login sessions.**
  - **location_logs.php: Logs volunteer location data.**
  - **logout.php: Handles user logout.**
  - **recipient_login.php: Manages recipient login sessions.**
  - **select_recipient.php: Handles recipient selection and delivery actions.**

3. extras Folder

- **Files:**
  - **DB.sql: Contains SQL queries to create the database tables.**
  - **Legacy PHP and Excel files: Used for importing old data into the new system.**

4. Admin Folder

- **Files:**
  - **login.php: Handles admin login.**
  - **styles.css: Contains global styles for admin pages.**
  - **admin_header.php: Provides a consistent header for admin pages.**
  - **admin_delivery.css, admin_delivery.js, admin_delivery.php: Handle the admin delivery page.**
  - **admin_volunteers.php, admin_recipients.php: Manage volunteer and recipient data.**
  - **admin_volunteers_excel.php, admin_recipients_excel.php: Export volunteer and recipient data as Excel files.**

5. Admin/calls Folder

- **Files:**
  - **index.php: Handles incoming phone calls and routes them based on the caller's selection.**
  - **language.php: Provides language-specific menus for callers.**
  - **route_call.php: Routes calls to designated employees.**
  - **whisper.php: Plays a whisper message to employees when receiving a call.**

6. Admin/php Folder

- **Files:**
  - **admin_fetch_recipient.php: Retrieves recipient data for the admin delivery page.**
  - **admin_volunteer_login.php: Manages volunteer login sessions for the admin delivery page.**
  - **enable_selection.php: Updates the volunteer selection setting in the database.**
  - **send_sms.php, send_sms_volunteers.php: Send SMS messages to recipients and volunteers.**
  - **receive_sms.php, receive_sms_volunteers.php: Handle incoming SMS replies.**

Call Handling

The system uses Twilio to handle incoming phone calls and route them to designated employees. Below is the call handling workflow:

1. Incoming Call Workflow

1. **Call Received:**
   - **The call is received by index.php in the Admin/calls folder.**
   - **The system checks if the call is within working hours (9 AM to 10 PM).**
   - **If outside working hours, the caller is informed that the organization is closed.**
2. **Language Selection:**
   - **The caller is prompted to select a language (English or Arabic).**
3. **Menu Selection:**
   - **Based on the caller type (recipient or volunteer), the system provides a menu of options.**
   - **For example, recipients can choose to speak with Ruba, Yousef, or Ordy.**
   - **Volunteers can choose to speak with Ruba, Omar, or Abed.**
4. **Call Routing:**
   - **The call is routed to the selected employee.**

- A whisper message is played to the employee, informing them of the caller's identity and type (e.g., "Call from AID: Recipient John Doe").

2. SMS Handling

- **Incoming SMS:**
  - Incoming SMS messages are handled by receive_sms.php and receive_sms_volunteers.php.
  - The system checks if the message is from a registered recipient or volunteer.
  - If the message is received within 12 hours of the original SMS, the system sends a response based on the selected option.
- **Outgoing SMS:**
  - Outgoing SMS messages are handled by send_sms.php and send_sms_volunteers.php.
  - Messages are logged in the sms_logs table.

Technical Details

1. Session Management

- **Volunteer Sessions:**
  - Created by volunteer_login.php when a volunteer logs in.
  - Stores the volunteer's ID, name, and pickup status.
- **Recipient Sessions:**
  - Created by recipient_login.php when a recipient logs in.
  - Stores the recipient's ID and other relevant data.
- **Admin Sessions:**
  - Created by login.php when an admin logs in.
  - Stores the admin's ID and permissions.

2. Logging

- **Admin Logs:**
  - All admin actions are logged in the admin_logs table.
- **Delivery Logs:**
  - All delivery-related actions are logged in the delivery_logs table.
- **Location Logs:**
  - Volunteer locations are logged in the location_logs table.

3. Security

- **Input Validation:**
  - All user inputs are validated to prevent SQL injection and other attacks.
- **Session Protection:**
  - Sessions are destroyed on logout to prevent unauthorized access.
- **Error Handling:**
  - Errors are logged and displayed to admins for debugging.

Editing Guideline

1. Database Tables & Archives

Key Tables:

- **recipient, volunteer, delivery, and their archive tables (*_archive).**
- **What to Watch:**
  - **Triggers:**
    - **The after_recipient_insert trigger auto-creates a delivery entry when a new recipient is added.**
    - **If you modify the recipient table schema, update this trigger if needed.**
  - **Archive Syncing:**
    - **When editing a table (e.g., adding a column), ensure its archive table is updated identically.**
    - **Example: If you add a notes column to recipient or recipient_details, add it to recipient_archive too.**
  - **Indexes:**
    - **Indexes on approved, replied, and status columns speed up queries.**
    - **Avoid removing these unless necessary.**

2. Forms & Form-Backend

Public Forms (recipientform.php, volunteerform.php):

- **Validation:**
  - **Client-side (JavaScript) and server-side (PHP) validation must match.**
  - **Example: If a phone number is validated as +1-XXX-XXX-XXXX in JS, ensure the PHP regex matches it.**
- **SMS Integration:**
  - **Editing add_recipient.php or add_volunteer.php? Test SMS templates to avoid broken placeholders (e.g., $name).**
  - **Twilio webhooks (receive_sms.php) rely on exact URL paths—don't rename files without updating Twilio console.**

Admin Forms (admin_recipients.php, admin_volunteers.php):

- **Filters:**
    - **The fixed filter row uses inline JavaScript. Editing column names? Update the JS filterTable() function.**
- **Edit Function:**
    - **Avoid allowing edits to volunteer_id or recipient_id—these break foreign keys in delivery.**

3. Admin-Volunteer & Admin-Excel Pages

Volunteer/Recipient Management:

- **Batch Actions:**
    - **The "Select All" checkbox uses JavaScript. If pagination is added, refactor to handle partial selections.**
- **Excel Exports (admin_recipients_excel.php):**
    - **Column order matches the database. Adding/removing columns? Update the Excel header array.**
    - **Large exports may timeout—increase PHP max_execution_time.**

Admin-Editor Function:

- **Approval Workflow:**
    - **When approving a recipient, ensure delivery entries are created (mimic the after_recipient_insert trigger).**
- **Session Dependencies:**
    - **Admin actions rely on admin_id stored in sessions. Test logout/login flows after edits.**

4. Admin-Filter Selection

- **Dynamic Filters:**
    - **Filters for replied, approved, etc., are generated from database ENUM values.**
    - **If you modify ENUM options (e.g., adding 'Temporary' to replied), regenerate the filter dropdowns.**
- **SQL Injection Risks:**
    - **Filters use $_GET parameters. Always sanitize with mysqli_real_escape_string() or prepared statements.**

5. Admin-Delete Function

Critical Notes:

- **Archive Before Delete:**
  - **The DELETE action in admin pages moves records to *_archive tables. Test this with foreign keys (e.g., deleting a volunteer with active delivery entries).**
- **Orphaned Data:**
  - **Deleting a recipient? Use ON DELETE CASCADE in delivery or manually clean up linked records.**
- **Logging:**
  - **Ensure deletions log to admin_logs with action_type 'DELETE'.**

6. General Pitfalls

- **Inline CSS/JS:**
  - **Try to avoid editing inline styles/scripts (e.g., in admin_recipients.php), unless you were stuck like me. Centralize changes in general.css/js.**
- **Language Files:**
  - **Text in JS files (e.g., deliver.js) is hardcoded in a translations object. Adding a new language? Update all instances.**
- **Map Clustering:**
  - **Leaflet marker clustering uses latitude/longitude. If geocoding breaks, recipients won’t appear on maps.**

7. Testing Checklist After Edits

1. **Database:**
   - **Verify triggers and archive tables sync with schema changes.**
2. **Forms:**
   - **Submit test data with invalid inputs to check validation.**
3. **Admin Actions:**
   - **Test delete, approve, and SMS send actions.**

Known Issues and Future Improvements

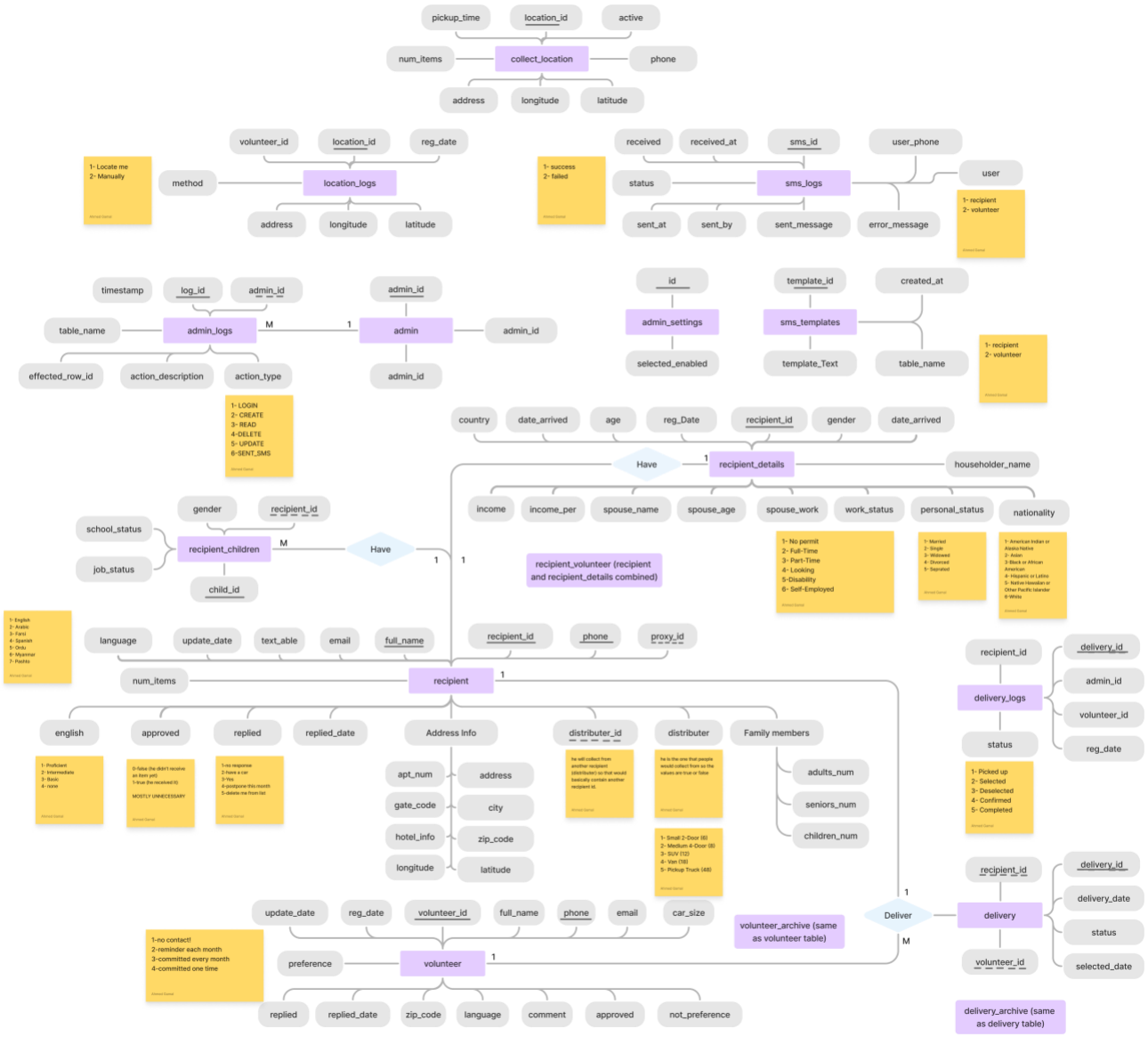
- **Issues:**
  - **Code Duplication: Multiple instances of similar functions across files.**
  - **Inline Styles: Scattered inline CSS and JavaScript.**
  - **Limited Scalability: Single-file approaches reduce modularity.**
- **Future Improvements:**
  - **Refactor code to consolidate repetitive functions.**
  - **Separate inline styles and scripts into dedicated files.**
  - **Implement robust error handling for all actions.**
  - **Optimize database queries for performance.**
- **Small Note:**
  -

Conclusion

This documentation provides a comprehensive overview of the American Islamic Diversity (AID) public website and admin interface. The system is designed to streamline the process of registering recipients and volunteers, managing deliveries, and handling communications. By integrating third-party APIs such as Twilio, Leaflet, OpenCageData, and MapBox the system offers a robust and user-friendly experience for all stakeholders.

For further assistance or customization, please refer to the provided folder structure and file descriptions, and to the Editing Guideline section.

ERD Diagram


pickup_time
location_id
active
num_items
collect_location
phone
address
longitude
latitude
volunteer_id
location_id
reg_date
1- Locate me
2- Manually
method
location_logs
address
longitude
latitude
received
received_at
sms_id
user_phone
1- success
2- failed
status
sms_logs
user
1- recipient
2- volunteer
sent_at
sent_by
sent_message
error_message
timestamp
log_id
admin_id
admin_id
table_name
admin_logs
M
1
admin
admin_id
effected_row_id
action_description
action_type
admin_id
1- LOGIN
2- CREATE
3- READ
4- DELETE
5- UPDATE
6-SENT_SMS
id
admin_settings
selected_enabled
template_id
created_at
sms_templates
template_Text
table_name
1- recipient
2- volunteer
country
date_arrived
age
reg_Date
recipient_id
gender
date_arrived
Have
1
recipient_details
householder_name
income
income_per
spouse_name
spouse_age
spouse_work
work_status
personal_status
nationality
1- No permit
2- Full-Time
3- Part-Time
4- Looking
5-Disability
6- Self-Employed
1- Married
2- Single
3- Widowed
4- Divorced
5- Separated
1- American Indian or Alaska Native
2- Asian
3-Black or African American
4- Hispanic or Latino
5- Native Hawaiian or Other Pacific Islander
6-White
gender
recipient_id
school_status
job_status
recipient_children
M
Have
1
1
child_id
recipient_volunteer (recipient and recipient_details combined)
1- English
2- Arabic
3- Farsi
4- Spanish
5- Urdu
6- Myanmar
7- Pashto
language
update_date
text_able
email
full_name
recipient_id
phone
proxy_id
num_items
recipient
1
english
approved
replied
replied_date
Address Info
distributer_id
distributer
Family members
1- Proficient
2- Intermediate
3- Basic
4- none
0-false (he didn't receive an item yet)
1-true (he received it)
MOSTLY UNNECESSARY
1-no response
2-have a car
3-Yes
4-postpone this month
5-delete me from list
apt_num
address
gate_code
city
hotel_info
zip_code
longitude
latitude
he will collect from another recipient (distributer) so that would basically contain another recipient id.
he is the one that people would collect from so the values are true or false
adults_num
seniors_num
children_num
1- Small 2-Door (6)
2- Medium 4-Door (8)
3- SUV (12)
4- Van (18)
5- Pickup Truck (48)
recipient_id
delivery_id
admin_id
delivery_logs
volunteer_id
reg_date
status
1- Picked up
2- Selected
3- Deselected
4- Confirmed
5- Completed
update_date
reg_date
volunteer_id
full_name
phone
email
car_size
volunteer_archive (same as volunteer table)
Deliver
1
M
recipient_id
delivery_id
delivery_date
delivery
status
volunteer_id
selected_date
1-no contact!
2-reminder each month
3-committed every month
4-committed one time
preference
volunteer
1
replied
replied_date
zip_code
language
comment
approved
not_preference
delivery_archive (same as delivery table)