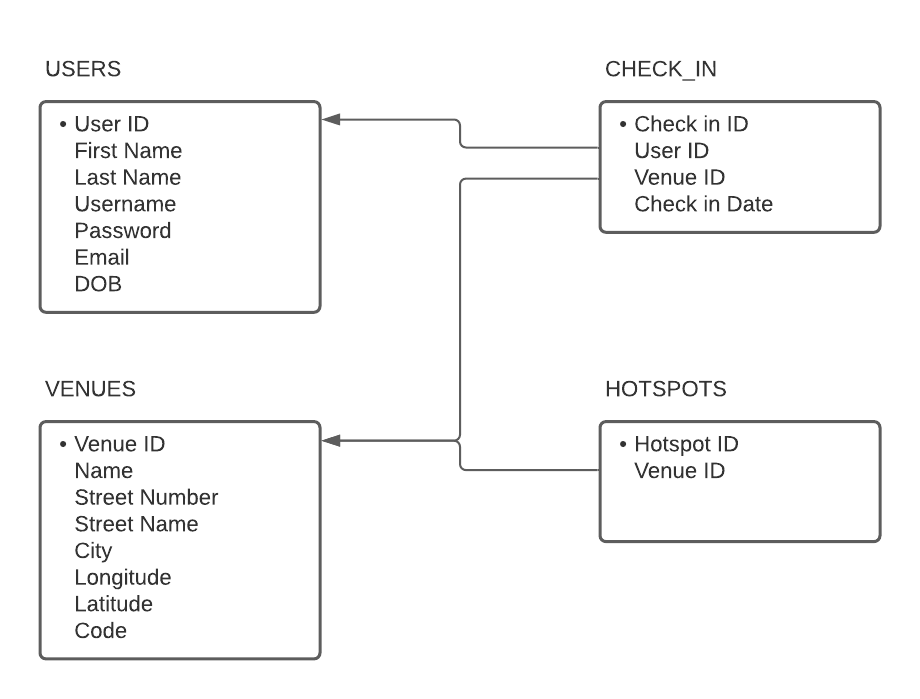
Data

1. Data Stored in Database
   * User object
     + User ID (unique integer)
     + First name (string)
     + Last name (string)
     + Username (unique string)
     + Password (string)
     + Email (unique string)
     + Date of birth (date)
   * Venue object
     + Venue ID (unique integer)
     + Name (string)
     + Street number (string)
     + Street name (string)
     + City (string)
     + Longitude (double)
     + Latitude (double)
     + Code (string)
   * Check in object
     + Check in ID (unique integer)
     + User ID (integer)
     + Venue ID (integer)
     + Check in date (date)
   * Hotspot object
     + Hotspot ID (unique integer)
     + Venue ID (integer)
2. Client-Server Interactions
   * Get map markers
     + GET request: Get all map markers for hotspots from the database
     + Response: Map markers saved onto the current Vue instance and rendered onto the map
   * Log in user
     + POST request: Convert log in credentials into a JSON object and post it to the server
     + Response: User is logged in and redirected to the user page if correct; otherwise, the website informs the user of errors
   * Sign up user
     + POST request: Convert sign up information into a JSON object and post it to the server
     + Response: New user object is created in the database; otherwise, the website informs the user of errors
   * Complete sign up with Google
     + POST request: Convert remaining sign up information into a JSON object and post it to the server
     + Response: New user object is created in the database; otherwise, the website informs the user of errors
   * Submit check in code
     + POST request: Convert code into a JSON object and post it to the server
     + Response: User activity is recorded into the database; otherwise, the website informs the user of errors
   * Get history
     + GET request: Get the logged in user’s recorded activity
     + Response: Activity data is saved onto the current Vue instance and shown on the history table
   * Get user data
     + GET request: Get the logged in user’s details
     + Response: User details are saved onto the current Vue instance and shown on the user profile page
   * Update user data
     + POST request: Convert the user’s updated information into a JSON object and post it to the server
     + Response: The user’s information is updated in the database; otherwise, the website informs the user of errors
   * Log out
     + POST request: Delete the user’s session data
     + Response: The user is redirected to the index page

Design

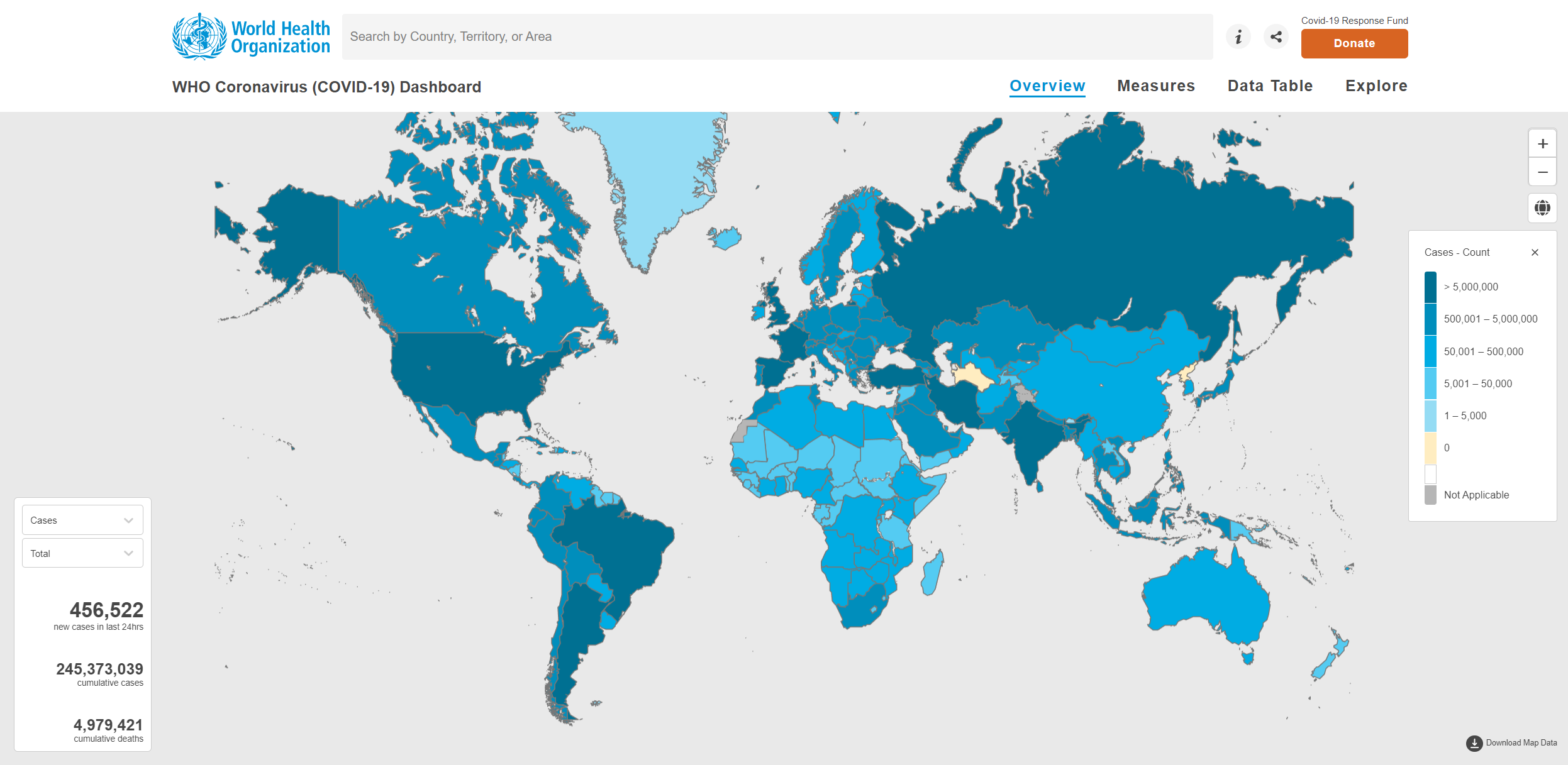
* + Index page
    - Header bar
      * Name/logo on the left side
      * Log in button on the right side
      * Sign up button on the right side
    - Main space
      * Homepage
        + Map showing all hotspots (places of high COVID-19 activity)
    - Footer bar
      * Credits to the Bootstrap framework on the left side
      * Social media and GitHub on the right side
  + Sign up page
    - Sign up form
      * Sign up with Google button at the top
      * Text fields for required information (full name, username, password, email, etc.)
      * Sign up button to submit information to the server
    - Log in area
      * Log in button to redirect the user to the log in page (in case they went to the wrong page)
  + Log in page
    - Log in form
      * Continue with Google button at the top
      * Text fields for required information (username and password)
      * Log in button to submit information to the server
    - Sign up area
      * Sign up button to redirect the user to the sign up page (in case they went to the wrong page)
  + User page
    - Header bar
      * Name/logo on the left side
      * Home button on the left side
      * Check in button on the left side
      * History button on the left side
      * User profile button on the left side
      * Log out button on the right side
    - Main space
      * Homepage
        + Map showing all hotspots (places of high COVID-19 activity)
      * Check in page
        + Text field to submit code
        + Submit button to submit code to the server
      * History page
        + Table showing all recorded activity (where the user has gone based on the submitted codes)
      * User profile page
        + Edit profile button to turn on edit mode (while edit mode is off)
        + A summary of user details (username, email, date of birth, etc.) (while edit mode is off)
        + Cancel button to turn off edit mode and ignore changes (while edit mode is on)
        + Save button to turn off edit mode and save new changes (while edit mode is on)
        + Text fields for each user detail (while edit mode is on)
    - Footer bar
      * Credits to the Bootstrap framework on the left side
      * Social media and GitHub on the right side

Database Schema



Research

1. The World Health Organization (WHO) COVID19 Page



* Simple, easy-to-read design
* Clear world map with necessary information
* Graphs for more in-depth information

1. The Centers for Disease Control and Prevention (CDC) COVID19 Page

A screenshot of a computer

Description automatically generated with medium confidence

* Cluttered, hard to navigate
* Data is minimal and hard to interpret (just numbers or small graphs)
* Mostly static information at the top, latest updates at the bottom, requires scrolling down for COVID-19 news, inconvenient

1. Vietnam’s Ministry of Health COVID19 Page

Graphical user interface, website

Description automatically generated

* Toggle between Vietnam data and global data
* Latest important news at the top
* In-depth table showing confirmed cases and changes in cases per Vietnamese city
* Bar chart showing number of infections, recovered cases, and deaths in the past week along with the average of each category
* Additional news at the bottom