SYMBIOSIS INSTITUTE OF TECHNOLOGY

A DBMS PROJECT REPORT ON

A SIMPLIFED NGO ENGAGEMENT SYSTEM

SUBMITTED BY:

SOUMILI BISWAS (22070122216)

SRISHTI PARULEKAR (22070122219)

TANVI SONTAKKE (22070122234)

UNDER THE GUIDANCE OF

PROF. VANI HIRAMANI

DEPARTMENT OF COMPUTER SCIENCE

Index

INTRODUCTION

Finding the right NGO to volunteer for takes up a lot of time and effort. Sifting through endless lists, navigating unreliable sources and lacking connections can make the search frustrating and time consuming. This can be especially true for service-learning students who have specific requirements for their volunteer experience.

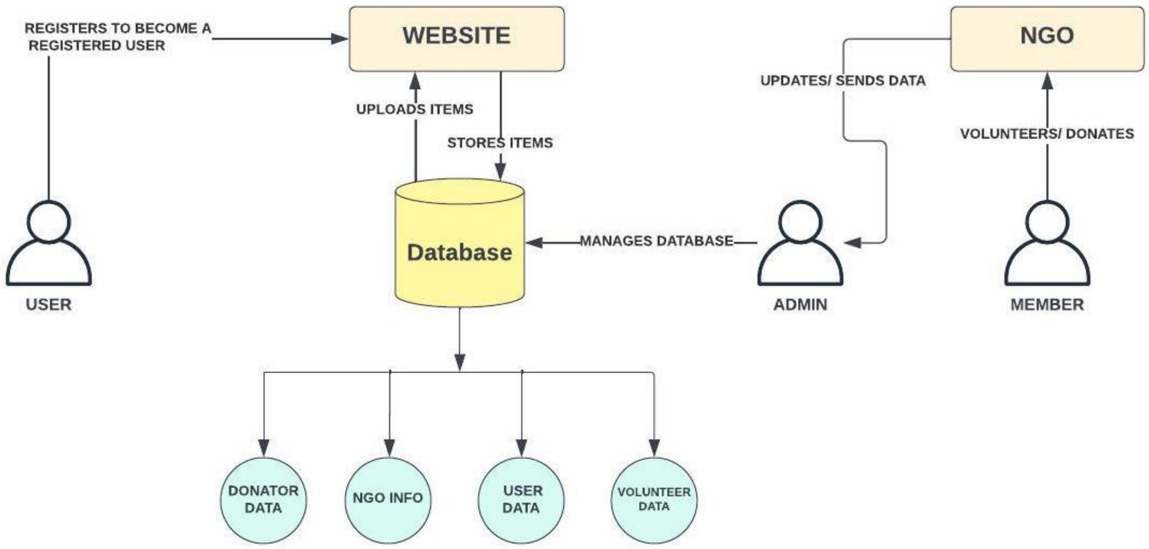
This is where our NGO searching, registration, and donation system comes in. It streamlines the process of finding the right NGO by providing a centralized platform that houses comprehensive information on the various NGOs in the city chosen by the user. Our website allows the user to directly sign up for volunteer opportunities offered by the chosen organization or simply donate to them, immensely saving traveling time and endless inquiries.

The platform filters NGOs based on specific causes the user is passionate about, for example, education, poverty, food security or various other social issues. This targeted approach lets the user identify the right NGO that aligns with the user's values and interests, saving valuable time and effort.

PROBLEM STATEMENT

We aim to develop an online platform that makes the process of searching, registering and donating to NGOs everywhere, a seamless user experience. Our platform will act as a medium for anyone who wishes to contribute to a good cause by eliminating cumbersome searches and registration processes. We emphasize on the user-friendly nature of our platform, where a simple click translates into a significant act of philanthropy.

SYSTEM ARCHITECTURE



MODULES

1. **User Module:**
   - This module is used to browse through the website or register to become a member and create profiles.
   - A user can search for NGOS based on location and cause.
   - A user can view detailed information about NGOs.
   - A user can also register for volunteering opportunities or making donations.
2. **Website module:**
   - This module enables users to create an account and join the platform.
   - The module is used to track and manage volunteers and donors.
   - This module provides navigation to detailed NGO profiles and enables filtering them by location, cause and online/offline opportunities.

- It saves member, donor and volunteer profiles.

3. **NGO Module:**
    - This module enables NGOs to provide information about their organization such as requirements and preferred skills in volunteers to the admin.
    - This module enables NGOs to review and approve volunteer applications.
4. **Admin Module:**
    - This module allows the admin to manage website content and the database (to update information and maintain data accuracy).
    - This module enables user to track and manage volunteers.
5. **Member Module:**
    - This module allows the member to volunteer or donate to an NGO, based on their preference.
6. **Security and Authentication Module:**
    - This module implements user authentication and authorization mechanisms.
    - It protects user and NGO data from unauthorized access.
    - This module facilitates secure online donations to NGOs.

FUNCTIONAL REQUIREMENTS

1. **User Authentication and Profile Management:**

- Implementation of a secure sign-up page requiring essential credentials such as username and password, along with other necessary information.
- Development of a login page for user authentication.

2. **User Profile Display:**

- Integration of a feature enabling the display of user profiles.

3. **Search and Filtering Functionality:**

- Incorporation of search and filtering mechanisms for users to narrow down NGO options based on causes and geographical locations.

4. **Volunteer Engagement:**

- Provision for volunteers to sign up for events organized by selected NGOs after applying search filters.
- Creation of volunteer profiles, facilitating connection with event activities.

5. **Donation Management:**

- Inclusion of a platform where donors can contribute to specific donation drives or directly to NGOs following search filtering.
- Integration of a secure Payment Gateway to facilitate financial transactions between donors and NGOs.

6. **Donor and Volunteer Profiles:**

- Creation of profiles for both donors and volunteers.
- Donors can contribute to their preferred NGOs, and volunteers can actively engage in events.

ENTITIES, RELATIONSHIPS AND ATTRIBUTES

TABLE:

Entity	Relationship	Entity	Cardinality
NON_REGISTERED_USER, **REGISTERED_USER**	*is-A*	**USER**	**(Specialization)** **1: 1**
REGISTERED_USER	*is-A*	**DONOR,** **VOLUNTEER**	**(Generalisation)** **1: 1**
DONOR	*donates to*	**NGO**	**1: N**
VOLUNTEER	*volunteers to*	**NGO**	**1: N**

ATTRIBUTES

1. **REGISTERED_USER:**
    - Registered_UserID
    - Registered_User_Name
    - Gender
    - Password
    - Username

2. **DONOR:**

   - DonorID
   - NGO_ID
   - Register_UserID

3. **VOLUNTEER**

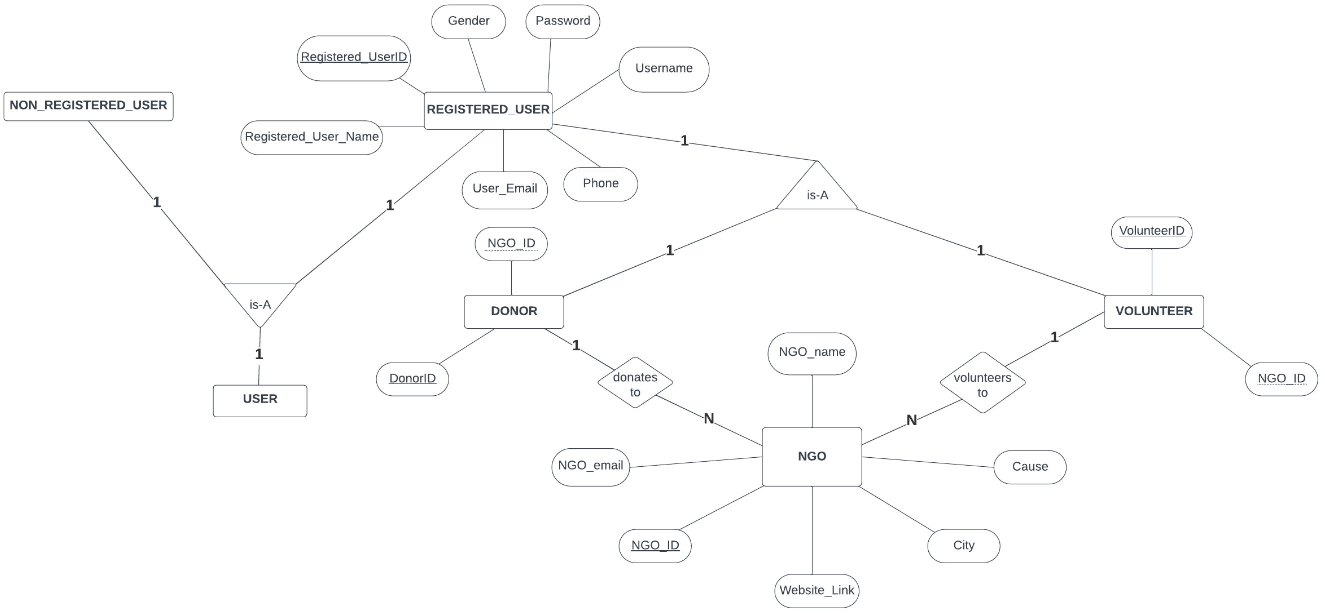
   - VolunteerID
   - NGO_ID
   - Register_UserID

4. **NGO**

   - NGO_ID
   - NGO_Name
   - Cause
   - City
   - Website_Link
   - NGO_email

5. **NON_REGISTERED USER**

ER DIAGRAM



RELATIONAL SCHEMA

1. Registered User

Registered_user_ID	Registered_user_name	Username	password	Gender	Email	phone

- **Primary Key:** Registered_user_ID
- **Foreign Keys:** Null
- **Candidate Keys:** Username**,** Registered_user_ID
- **Alternate Keys:** Username

2. is-A

VolunteerID	DonorID	*NGO_ID*	*Registered_UserID*

- **Primary Key:** VolunteerID, DonorID
- **Foreign Keys:** NGO_ID, Registered_UserID
- **Candidate Keys:** VolunteerID, DonorID
- **Alternate Keys:** Null

3. NGO

NGO_ID	Website_Link	City	Cause	NGO_Name	NGO_email

- **Primary Key:** NGO_ID
- **Foreign Keys:** Null
- **Candidate Keys:** NGO_ID, Website_Link
- **Alternate Keys:** Website_Link

APPLICATION OF CODD'S RULES

1. INFORMATION RULE

- According to this rule, all the information in the relational database must have its logical representation in the form of values in the tables.
- Each value in the table is unique and has no conflict or ambiguity and can only be represented in any one way.
- For example, all the information about the members, NGOs and volunteers is stored in the form of database tables in MySQL.

2. GUARANTEED ACCESS RULE

- According to this rule, every data element from the database must be logically accessible by resorting to a combination of table name, column name, and primary key.
- For example, this rule is followed in our database where every value can be accessed logically with the help of primary and foreign keys.

3. SYSTEM TREATMENT OF NULL VALUES

- This rule states that the database must support a representation of missing or inapplicable information. These NULL values must be independent of datatype.
- For example, our database handles cases where information might be absent such as optional contact details or missing qualifications. However, for

primary keyed attributes like IDs, our system makes sure that we do not leave it NULL and assign some value to it.

4. ACTIVE OR DYNAMIC ONLINE CATALOGUE BASED RELATIONAL MODEL

- This rule states that logical data structure and physical storage structure should be separate and the system must support an online, inline relational catalogue that is accessible to authorized users by means of their regular query language.
- This rule is followed in our database as we have a query language to check all the constraints and data in the database.

5. COMPREHENSIVE DATA SUBLANGUAGE RULE

- This rule states that the system must support at least one relational model language that has a linear syntax and can be used both interactively and within application programs. It must support data definition operations, such as DML, DCL, security and integrity constraints and transaction management operations.
- If the database allows access to data without any help of this language, then it is considered as a violation.
- In our project, the admin manages and updates the database using these query languages.

6. VIEW UPDATING RULE

- This rule states that all views that are theoretically updatable must be updated by the system.
- The visual EER representation and the system architecture give a visual representation or depiction on how the admin manages website content and updates information in the database which is achievable practically using SQL database, ensuring the view updating rule.

7. HIGH-LEVEL INSERT, UPDATE, AND DELETE:

- This rule states that the system must support set-at-a-time insert, update, and delete operators.

- Members can register, volunteer, or donate all ‘set-at-a-time’ involving high-level insert, update, and delete operations on the database keeping in mind all the constraints.

8. PHYSICAL DATA INDEPENDENCE

- This rule states that the changes to the physical level must not require a change to an application based on the structure.
- For example, if a user wishes to switch his/her operating system from Android to Apple, his/her data remains intact in the application providing a level of physical data independence.

9. LOGICAL DATA INDEPENDENCE

- This rule states that the changes to the logical level must not require a change to an application based on the structure.
- Changes in the logical structure of the database (e.g., modifying member profiles) should not affect user interactions, maintaining logical data independence.

10. INTEGRITY INDEPENDENCE

- This rule states that integrity constraints must be specified separately from application programs and stored in the catalogues.
- The admin module allows for maintaining data accuracy, data intactness or truthfulness of data.
- For example, if the admin wants to make changes to a table having the referential key, it should not change the data of other tables contributing to integrity independence.

11. DISTRIBUTION INDEPENDENCE

- This rule states that the distribution of portions of database to various locations should be invisible to the users of the database.
- The distribution of user information to NGOs maintains confidentiality, ensuring that users are not informed about the specifics of the distribution

process. The data distribution strictly adheres to established security protocols.

12. NON-SUBVERSION RULE

- This rule states that if a system uses a low-level or different language to access the database system other than SQL, it should not bypass or subvert data integrity.
- However, if it is a SQL language it can manipulate and store the data. Our model abides by the rule as mentioned in the security and authentication module.