# User Requirements

* Donor Management:

1. The system should be able to allow users to add, update, and delete donor profiles.
2. Donor profiles should include personal information (name, contact details, career information), donation history, and event participation.
3. The system should allow users to view donor details and interactions.

* Communication Management:

1. Users should be able to record communication with donors (emails, phone calls, meetings, etc.).
2. The system should allow users to tag the communication with relevant categories.
3. Users should be able to set reminders for follow-up communications based on previous interactions.
4. Communication history should be searchable by donor, date, or type of communication.

* Donation Tracking:

1. Users should be able to record, update, and track donations made by donors, including amount, date, and type (monetary or in-kind).
2. The system should allow users to track recurring donations and generate donation receipts or thank-you notes.
3. Users shall be allowed to analyze the donation trends based on total donations, recurring frequency of donations.

* Event Management:

1. The users will be able to create an event, update and manage them accordingly.
2. Event information to capture: date, location, description, and guest list
3. In events, it needs to record whether the donor participates, and the kind of participation-donor attended or guest speaker or volunteer
4. Events allow inviting guests, reminders, and thank-you note sending

* Reporting & Analytics:

1. Users should be able to develop reports on donor activities regarding the total donations and events they have participated in.
2. The system should provide the analytics to track trends such as which events have the highest engagement of donors or which donors create the biggest value.

* Search and Filter:

1. In this regard, the system should allow users to search donors, donations, events, and communications based on set criteria such as donor name, event date, and amount of donation.
2. The users should have the ability to filter this list based on engagement level, frequency of donations, frequency of events, among others.

* User Access & Security:

1. The system shall have the ability to manage user roles that may include but are not limited to admin, event coordinator, fundraiser.
2. Users shall have access to only view or edit information given their role.
3. The system shall ensure donor data is kept securely and adhere to data privacy regulations like GDPR.

# System Requirements

* Software Requirements:

Operating System: The operating system should be compatible with Windows, macOS, and Linux.

Database: A relational database management system such as MySQL, PostgreSQL, or SQL Server to store data about donors, communications, donations, and events.

Web Browser: The system should also support modern web browsers such as Chrome, Firefox, Edge, Safari for access through a web interface.

Backend Framework: A web application framework like Django (Python), Ruby on Rails, or Laravel (PHP) that will handle user requests, database operations, and business logic.

Frontend Framework: HTML, CSS, JavaScript, and frameworks such as React or Angular for building a responsive and user-friendly interface.

API Integration: The system should have API capabilities for third-party integrations, such as email platforms for communication and payment gateways for donations.

Hardware Requirements:

Server: A web server (e.g., Apache, Nginx) to host the application.

Storage: Sufficient disk space to store donor data, communication logs, event files (e.g., images, PDFs), and donation records. This may vary depending on the size of the donor base.

Backup: A reliable backup system for database and file backups to ensure data integrity and recovery.

Security Requirements:

Authentication: Secure login mechanism using passwords, with support for two-factor authentication (2FA).

Data Encryption: All sensitive data, such as donor information and donation amounts, should be encrypted both in transit (using HTTPS) and at rest.

Access Control: RBAC ensures that users can only access the data and functionalities they are authorized to use.

Performance Requirements:

The system should support multiple concurrent users-for example, event coordinators and fundraisers-accessing the system without significant performance degradation.

The database should provide fast query support for searching and filtering in donor, donation, event, and communication data.

The system should be able to scale up to a growing donor base and event participation.

Usability Requirements:

The interface should be intuitive and user-friendly, with minimal training required for new users.

The system should provide mobile-friendly interfaces for users who need to access it from smartphones or tablets.

The system should clearly notify and remind users of follow-up communications and upcoming events. Backup and Recovery:

Automated daily backups for all donor, event, donation, and communication data.

Disaster recovery procedures to ensure the system can be restored in case of a failure.

Compliance Requirements:

The system shall comply with data protection laws and regulations, such as GDPR and CCPA, to ensure donor privacy.

This shall record who accesses or modifies the sensitive data, and such logs need to be periodically reviewed.