

Assignment Details: Design Document

CHARITY AND ONLINE DONATION

Team - 09:

ROHITH VEDANTAM - PES2UG21CS437

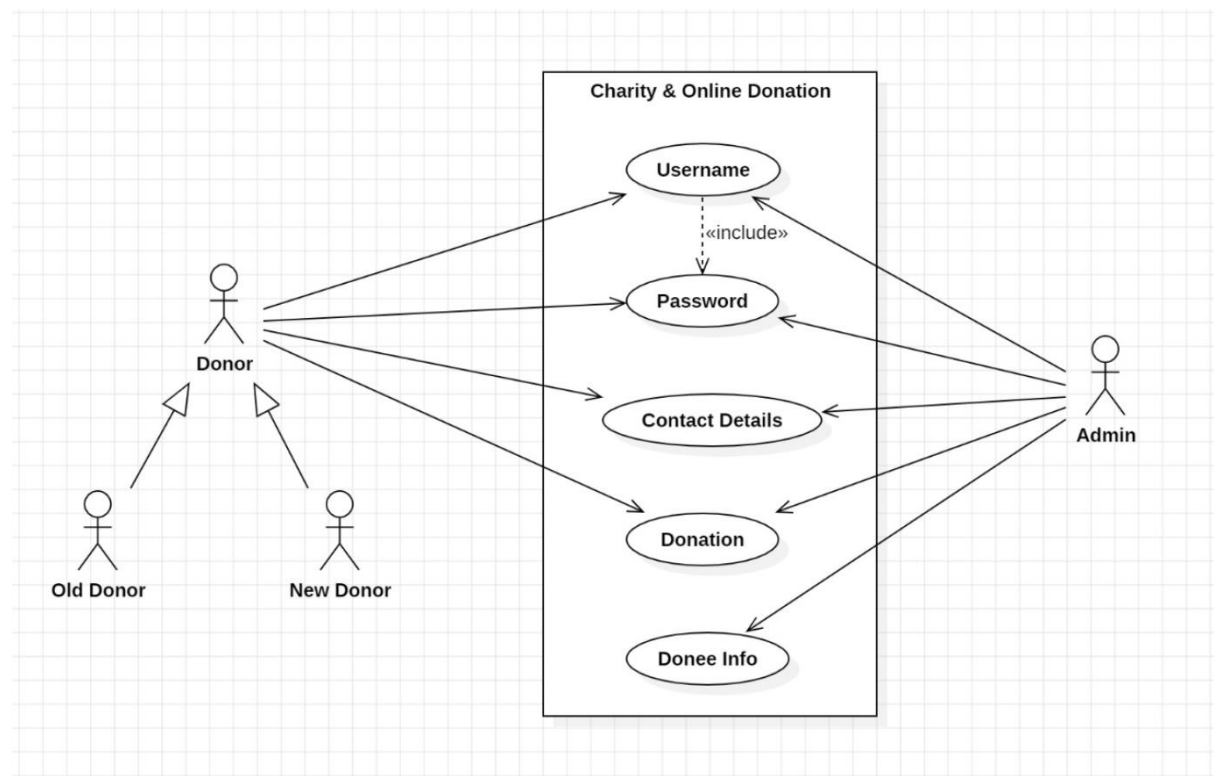
SAHIL KAMATE - PES2UG21CS453

SUPRITH M P - PES2UG22CS823

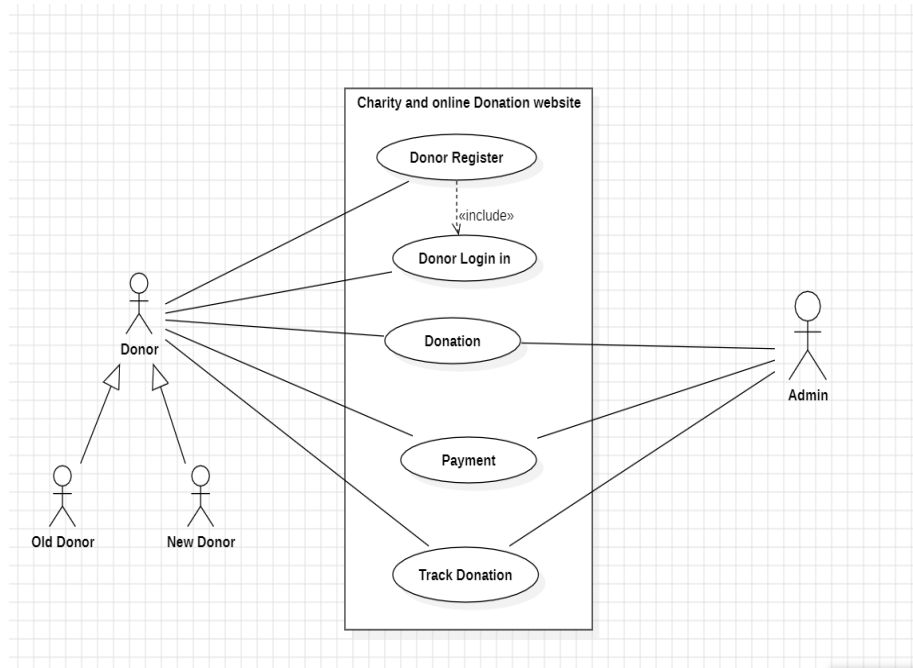
SHASHANK K N - PES2UG21CS490

1. Use case diagrams

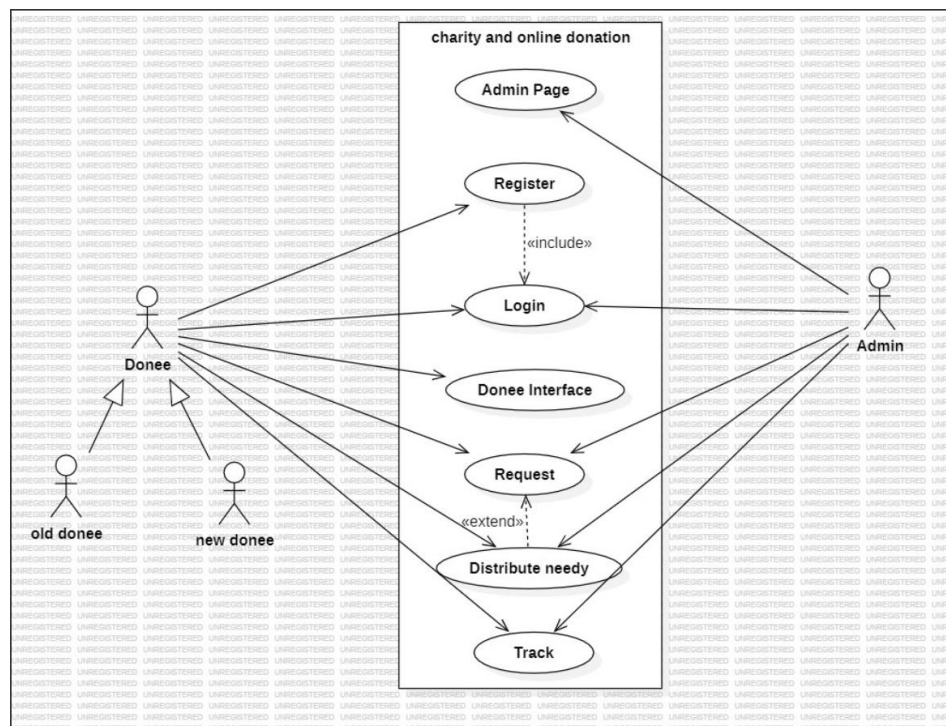
Home use case: Rohith Vedantam



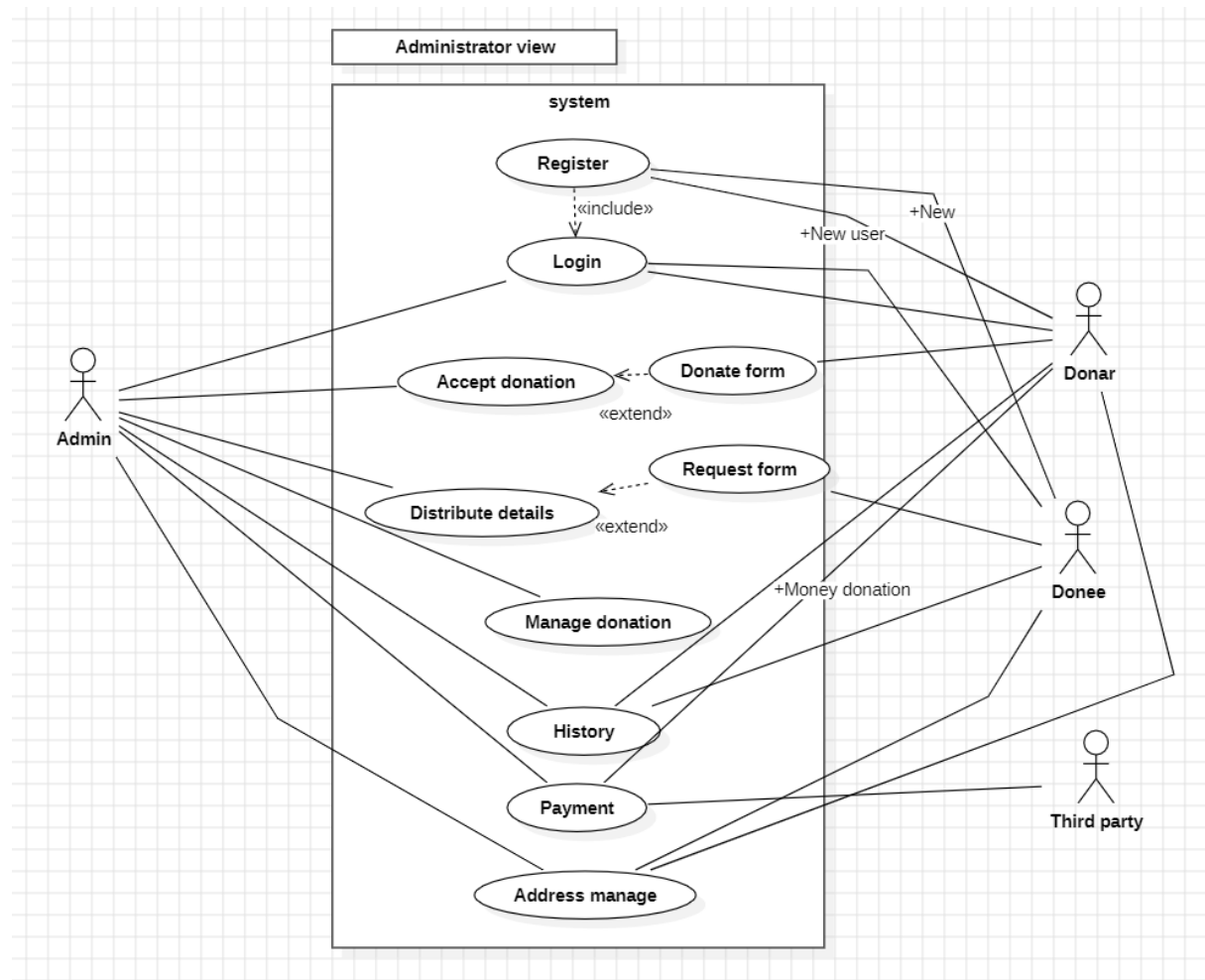
Donar use case: Sahil Kamate



Donee use case : Suprith M P

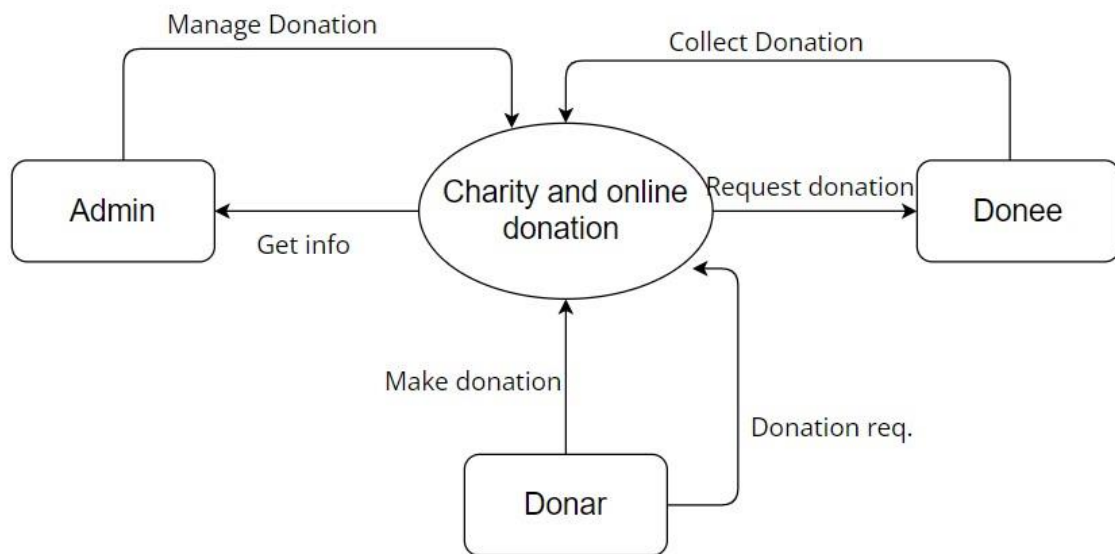


Administrator use case: Shashank K N

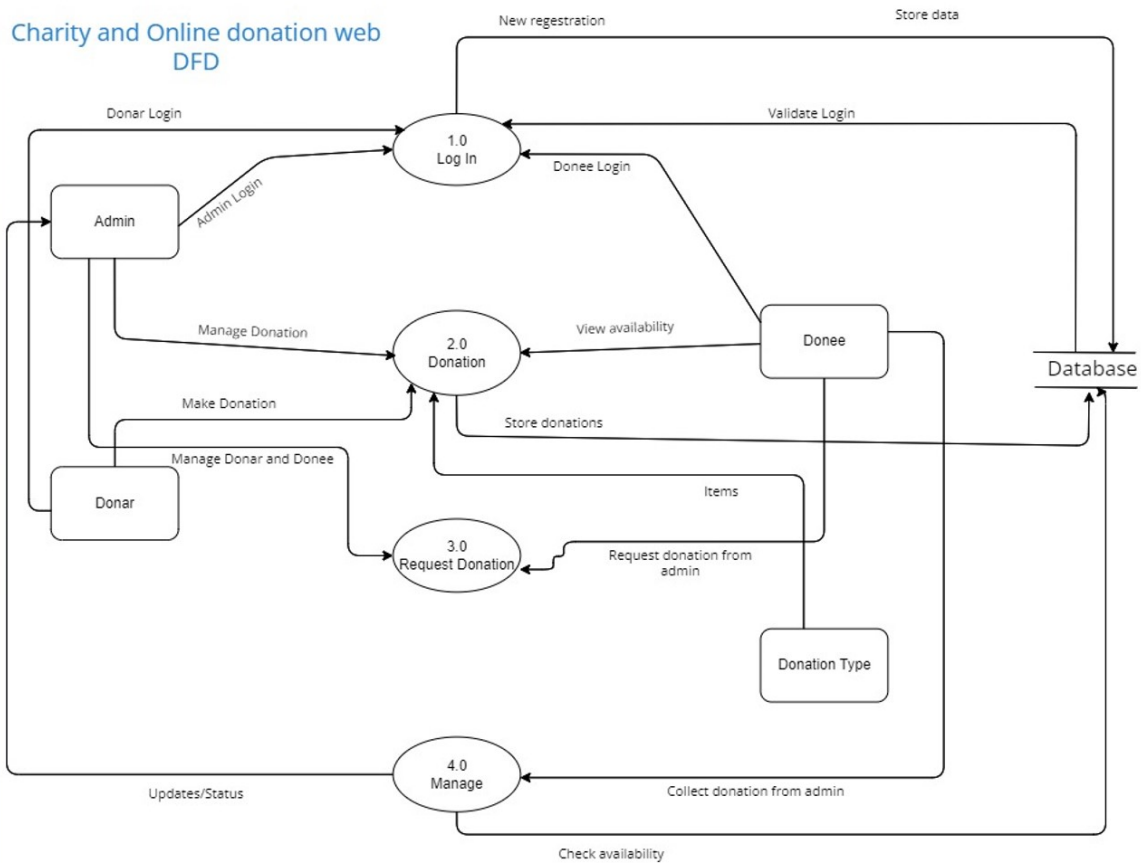


2. Incorporate DFDs:

- Develop a Data Flow Diagram (DFD) for your project:
- **Level 0 (Context Diagram):** This is the top level of the DFD, which provides a bird's eye view of the system. It should include external entities and how they interact with the main system.



- Level 1:** This expands the main system from Level 0 and shows its main functions. It should contain processes, data stores, and data flow among them.



3. **Architectural Style Integration:**

- Choose and design an architectural style for your project (e.g., Layered, Client-Server, Microservices, Peer-to-Peer, Service-Oriented).

Service Oriented architecture

- Describe and justify the chosen architectural style for your system. Explain how this style is suitable for your system's requirements and how it will benefit the project in terms of scalability, maintainability, performance, etc.

Description:

1. Modularity and Scalability:

In a charity and online donation system, there may be various functionalities and services such as user management, donation processing, analytics, and communication. SOA allows you to break down the system into loosely coupled, independently deployable services. This modularity makes it easier to scale individual components as needed, which is essential for handling varying levels of donation traffic.

2. Maintainability:

SOA promotes reusability and maintainability by breaking the system into smaller, manageable services. This separation of concerns allows for easier updates and maintenance of individual services without affecting the entire system.

3. Flexibility and Interoperability:

Charities often need to integrate with third-party services for payment processing, user authentication, and communication channels. SOA enables easy integration with external services, as each service can have well-defined and standardized interfaces (APIs).

4. Performance:

By distributing the workload across different services, you can optimize the performance of your system. You can deploy critical services on high-performance servers while allocating resources as needed for less critical services.

5. Security:

Security is paramount in an online donation system. By encapsulating security measures within individual services and controlling access through well-defined APIs, you can enhance the security of your system.

Justification:

A charity and online donation web system involves multiple interrelated functions, including user management, payment processing, fundraising campaigns, and reporting. These functions can be modularized into services like user authentication service, payment processing service, and campaign management service. A SOA can help you manage these diverse services efficiently, and it offers the flexibility to integrate with third-party services and adapt to changing requirements.
