

# Requirements and Specification Document

## *1. Purpose*

The purpose of this desktop-based Charity Donation System (CDS) is to provide an efficient, user-friendly platform for donors, fundraisers, and administrators to participate in and manage charitable activities. The system will streamline donation processes and improve transparency and reporting for stakeholders.

## *2. Scope*

The Charity Donation System will be developed as a standalone desktop application in Java. The system will facilitate user interactions while managing various charity campaigns and donation activities. It will have the following functionalities:

### **2.1. Functional Scope**

- **Donor Module:**
  - Browse available donation campaigns.
  - Make secure donations to selected campaigns.
  - View donation history.
- **Fundraiser Module:**
  - Create, update, and manage fundraising campaigns.
  - Track donations and monitor fundraising progress.
- **Administrator Module:**
  - Manage user accounts (create, update, and deactivate accounts).
  - Approve or decline new fundraising campaigns.
  - Generate and view donation reports.

### **2.2. Non-Functional Scope**

- **Performance:** System will support a reasonable number of concurrent users.
- **Usability:** A simple, user-friendly GUI for all user types.
- **Security:** Basic user authentication and data validation.
- **Data Integrity:** Ensure consistency between data entries.

### **2.3. Exclusions**

- No online payment gateway integration.
- No web or mobile-based versions of the system.

### 3. Definitions, Acronyms, and Abbreviations

Term	Definition
GUI	Graphical User Interface
CRUD	Create, Read, Update, and Delete operations
JDBC	Java Database Connectivity for database access

### 4. User Requirements

The system is designed to meet the following user requirements:

User Role	Requirement
Donor	Browse campaigns, donate to causes, view donation history
Fundraiser	Create and manage campaigns, monitor donation progress
Administrator	Manage user accounts, approve campaigns, generate reports

### 5. System Requirements

#### 5.1 Functional Requirements

- The system must allow users to register and log in with appropriate credentials.
- Donors must be able to browse campaigns and make donations.
- Fundraisers should have the ability to create, edit, and delete campaigns.
- Administrators should be able to generate detailed donation reports and manage users.

#### 5.2 Non-Functional Requirements

- **Security:** Ensure secure user authentication.
- **Performance:** Response time should be within acceptable limits for all operations.
- **Scalability:** Designed to allow future enhancements.
- **Data Storage:** Maintain user, campaign, and donation information in a relational database.

### 6. Constraints and Assumptions

#### Constraints:

- Must be a Java-based desktop application.
- Must use JDBC for database connectivity.
- No cloud or external APIs integration allowed.

### **Assumptions:**

- Users will access the system from their local machines.
- Java Runtime Environment (JRE) will be available on all target systems.
- A database server will be installed and accessible for data storage.

## ***7. Functional Architecture***

### **Overview:**

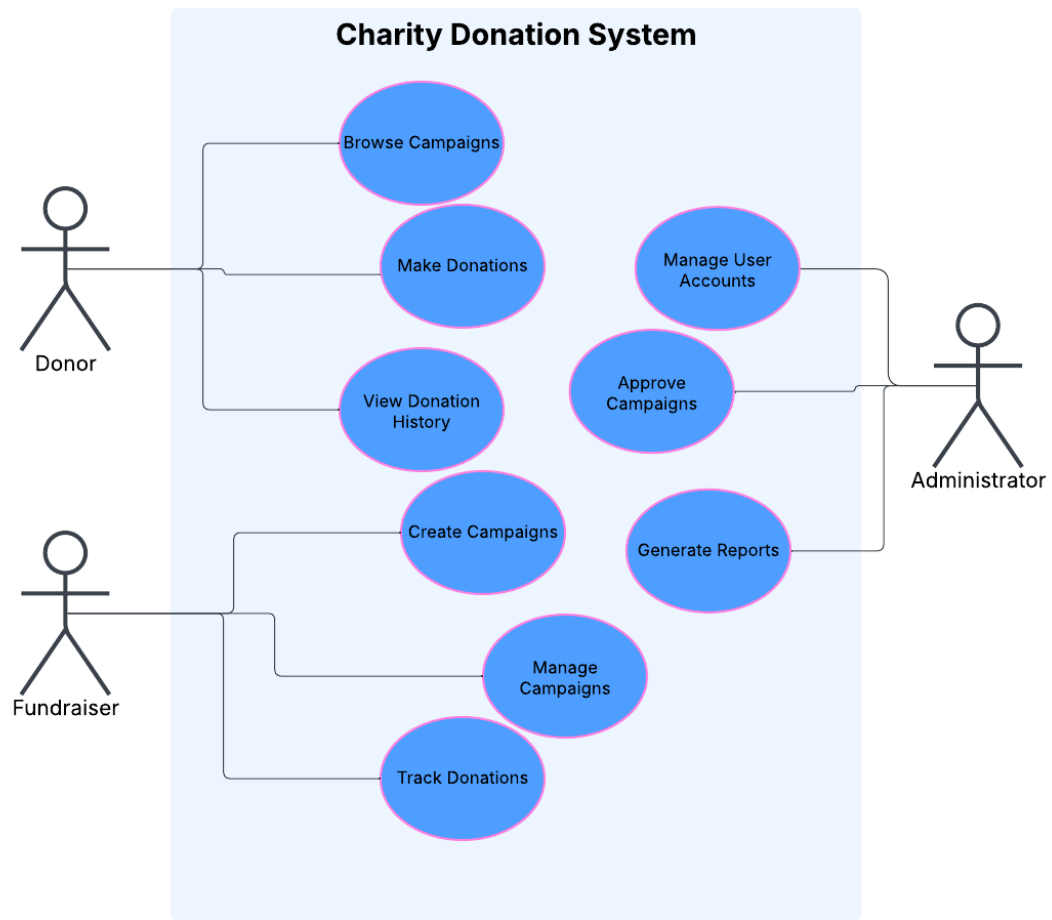
- The system will have three primary modules:
  - **User Module:** Manages user interactions.
  - **Campaign Module:** Handles campaign management for fundraisers.
  - **Reporting Module:** Provides reporting functionalities for administrators.

### **High-Level Flow:**

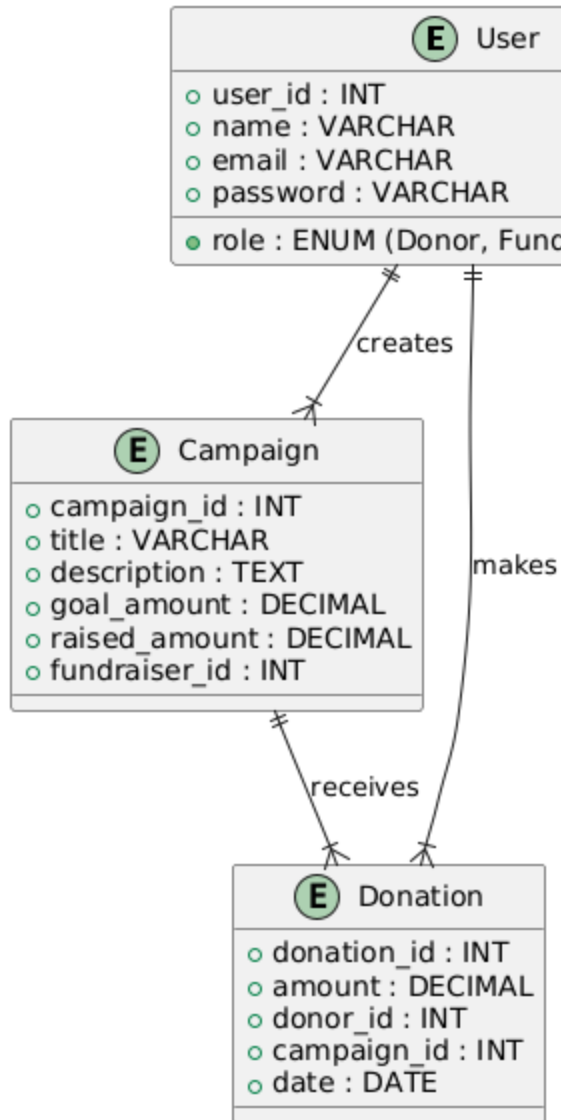
1. Users log into the system based on their role.
2. Donors view available campaigns and donate.
3. Fundraisers create and track their campaigns.
4. Administrators monitor the system and generate reports.

## ***8. System Models***

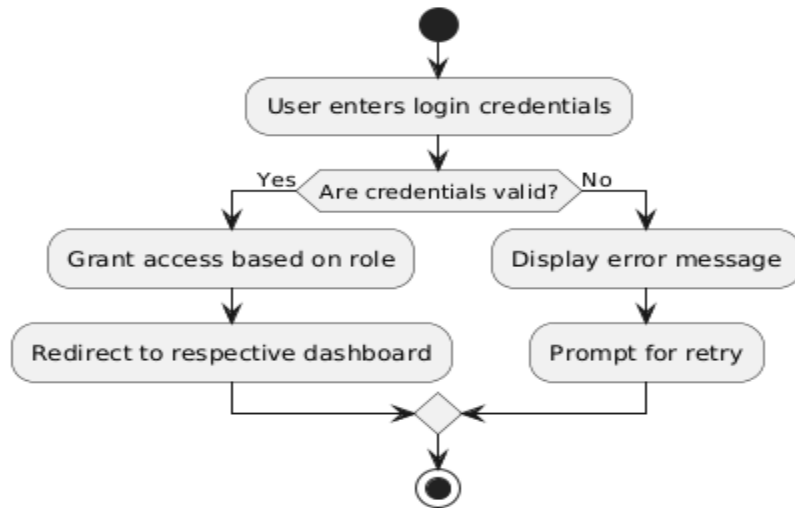
### **Use Case Diagram:**



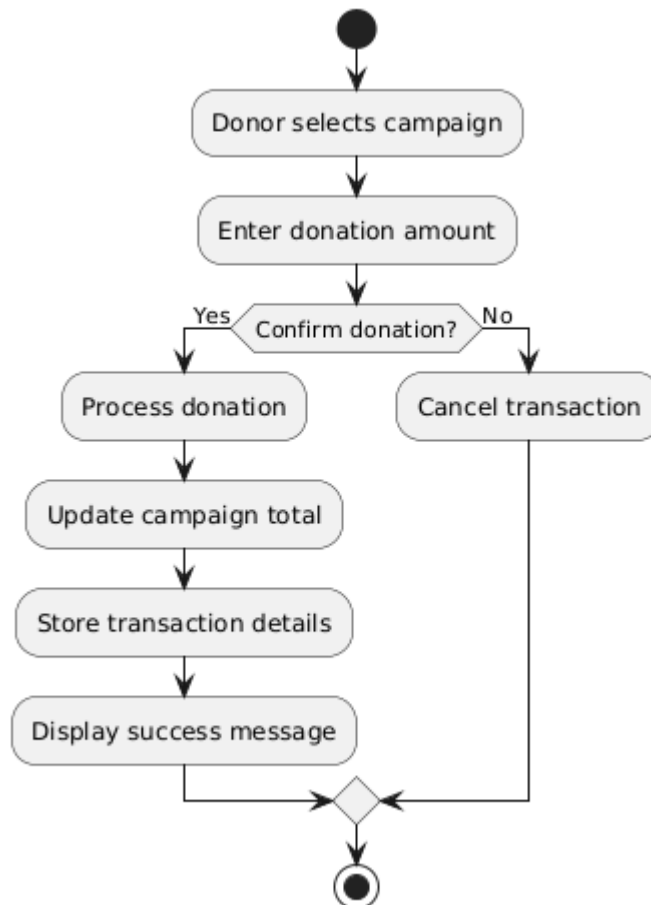
**Entity-Relationship Diagram:**



## User Login Flowchart



### Donation Process Flowchart



### 9. Testing Requirements

#### Validation:

- Ensure all user inputs are validated to maintain data integrity.

**Verification:**

- Verify that each system function matches the requirement specifications.

**Test Scenarios:**

1. Verify that donors can browse and donate to campaigns.
2. Ensure fundraisers can create and manage campaigns.
3. Confirm administrators can generate system reports.

***10. Future Enhancements***

- Integration with online payment gateways.
- Cloud-based and mobile application versions.
- Advanced reporting and analytics.