



# Department of Computer Science

## Database Systems

<b>Course Instructor:</b> Mehwish Ghafoor	<b>Batch:</b> 22
<b>Session:</b> Spring	<b>Semester:</b> 4th

## Final Report

### “Charity Management System”

Project Partners	
Name	Roll Number
Bazil Suhail	BSCS_22072
Muhammad Rehman	BSCS_22018
Abdullah Masood	BSCS_22054
Ahmad Saleem	BSCS_22078



## Project Overview:

The **"Charity Management System"** named **"AISAAR"** is a open source charity network which aims to provide a centralized platform for charity franchises to manage their operations, including *franchise management, there registration, donor management, fundraising campaigns, volunteer coordination along with specific project planning, proposal and there creation, monitoring complains of volunteers*. The system will streamline administrative control, enhance donation control, and improve transparency in charity operations.

## Functionalities:

*Creation and management of franchises, projects, volunteer, campaigns, and beneficiaries.*

- o Admin registering different franchises holding multiple projects, volunteers and campaigns as well as beneficiaries.
- o Proper authentication system for Franchises, Volunteers, Donors as well as the Admin
- o Volunteer proposing different projects and getting approved from individual franchises.
- o Tracking of donations made by donors to specific projects and campaigns.
- o Maintaining the record of Donor's donations(Transactions made)
- o Monitoring fundraising campaigns associated directly to franchises.
- o Publishing Top Donor's testimonials on Home Page.
- o Monitoring Top volunteer who have completed most projects.
- o Receiving complaints from volunteers and resolving their problems from Admin.
- o Managing beneficiaries enrollment, status and skills they are acquiring within franchises.
- o Maintaining record of completed Projects and franchises.
- o Maintaining the filtration of beneficiaries with respect to their status of enrollment in each franchise(orphan, widow, other).



## Required Collections with Documents

**Note:** All Collections listed below are the *ENTITES* in our *Project* while *Documents* are the expected attributes of those *ENTITES*

<b>Franchise:</b> <ul style="list-style-type: none"><li>• Franchise ID</li><li>• Name</li><li>• Description</li><li>• Location</li><li>• Contact Information □ Zip Code</li></ul>	<b>Projects:</b> <ul style="list-style-type: none"><li>• Project ID</li><li>• Volunteer ID</li><li>• Name</li><li>• Description</li><li>• Start Date</li><li>• End Date</li><li>• Goal Amount</li><li>• Collected Amount</li></ul>
<b>Volunteers:</b> <ul style="list-style-type: none"><li>• Volunteer ID</li><li>• Name</li><li>• Email</li><li>• Phone Number</li><li>• Address □ Tasks</li></ul>	<b>Beneficiary</b> <i>(under the organization):</i> <ul style="list-style-type: none"><li>• Member ID</li><li>• Name</li><li>• CNIC</li><li>• DOB</li><li>• Franchise ID</li><li>• Status</li><li>• Education Level</li><li>• Skills Acquired/Taught</li><li>• Health Status.</li></ul>
<b>Completed Projects:</b> <ul style="list-style-type: none"><li>• Project ID</li><li>• Volunteer ID</li><li>• Name</li><li>• Description</li><li>• Start Date</li><li>• End Date</li></ul>	<b>Completed Campaigns:</b> <ul style="list-style-type: none"><li>• Campaign ID</li><li>• Franchise ID</li><li>• Description</li><li>• Start Date</li><li>• End Date</li><li>• Goal Amount Status</li></ul>



<b>Donors:</b> <ul style="list-style-type: none"><li>• Donor ID</li><li>• Name</li><li>• Email</li><li>• Phone Number</li><li>• Donations</li></ul>		<b>Campaigns:</b> <ul style="list-style-type: none"><li>• Campaign ID</li><li>• Franchise ID</li><li>• Description</li><li>• Start Date</li><li>• End Date</li><li>• Goal Amount</li><li>• Collected Amount</li><li>• Status</li></ul>	
<b>Transaction History:</b> <ul style="list-style-type: none"><li>• ID <i>(can be of project and campaign)</i></li><li>• Amount</li><li>• Name <i>(of project campaign)</i></li><li>• idType <i>(whether donated to project or campaign)</i></li><li>• Date</li><li>• Time</li></ul>		<b>Proposed Projects:</b> <ul style="list-style-type: none"><li>• Project ID</li><li>• Volunteer ID</li><li>• Name</li><li>• Description</li><li>• Start Date</li><li>• End Date</li><li>• Goal Amount</li></ul> <p><i>(This is a temporary collection which will wait for the approval from the admin and store attributes back in projects,it has same attributes as projects)</i></p>	
<b>Top Volunteers:</b> <ul style="list-style-type: none"><li>• Projectscompleted</li><li>• displayName</li><li>• email</li><li>• idtype</li><li>• id</li></ul>	<b>Testimonials:</b> <ul style="list-style-type: none"><li>• displayName</li><li>• donations</li><li>• email</li><li>• feedback</li><li>• phoneNumber</li></ul>	<b>Complains:</b> <ul style="list-style-type: none"><li>• email</li><li>• displayName</li><li>• projectsCompleted</li><li>• phoneNumber</li><li>• title</li><li>• complain</li></ul>	



# Relationships Between Entities and there Cardinality / Updated Schema

## Franchise:

- A Franchise can have multiple projects.
- A Franchise can have multiple campaigns.
- A Franchise can have multiple beneficiaries enrolled
- A Franchise can have multiple Volunteers registered.

## Projects:

- A project can receive multiple donations.
- A project is volunteered by one Volunteer
- A project can be registered in Only One Franchise

## Volunteers:

- A volunteer can only administrator only One project.
- A volunteer can write many complains
- Single volunteer can belong to the top Volunteers
- A volunteer could register himself in one Franchise.

## Donors:

- A donor can make multiple donations to different projects.
- A donor can make multiple donations to different campaigns.
- A donor can have multiple transaction histories as per he makes
- A donor cam write multiple testimonials

## Top Volunteer :

- It can have many volunteer(top 5 only)

## Transaction History:

- Each transaction history's record belong to one Donor.



## **Campaigns:**

- A campaign belongs to one franchise.
- A campaign can receive multiple donations.

## **Beneficiaries:**

- Each beneficiary belong to only one franchise.

## **Proposed Projects:**

- A proposes project is proposed by one Volunteer
- A project can be proposes to Only One Franchise

## **Completed Projects:**

- A completed project can have multiple donations.
- A completed project has one Volunteer
- A completed project can belong to Only One Franchise

## **Completed Campaign :**

- A completed campaign can have multiple donations.
- A completed campaign has one Franchise.

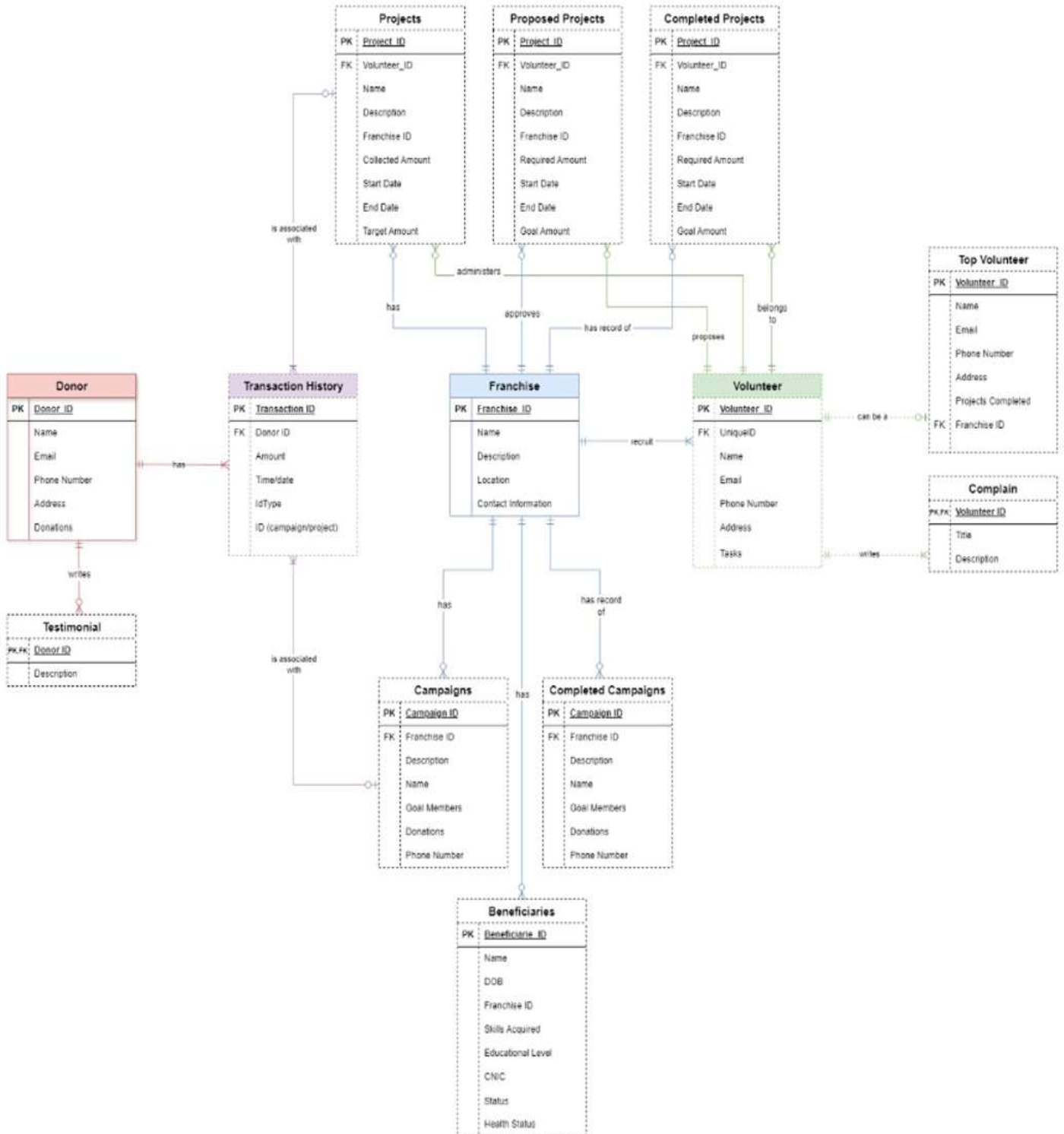
## **Testimonials :**

- Each testimonial belongs to one donor

## **Complains :**

- Each complain belongs to one Volunteer.

# Final ER-Diagram of Entities





# Technology Stack

As the database seems to realistically implemented we will be using “**REACT JS**” for front-end and for back-end, we will be using the “**FIREBASE**”

## REASON OF USING THEM:

- We have used **REACT JS** because of its powerful rendering components approach. With the potential for our project to encompass multiple pages, **React JS's** modular structure ensures that we can write concise code while rendering a greater number of components, thereby enhancing code reusability and maintainability.
- Reason of using **FIREBASE** for back-end is our project is realistically flexible and demands on time changes. Moreover **FIREBASE** is an online hosted database so our data will be preserved online and with **REACT JS** as front-end , users will be able to see changes on the spot.
- **Firebase** provides seamless integration with various authentication providers (like Google and email/password authentication), making it simpler to implement secure user authentication in our application. This is crucial for maintaining the security and integrity of user data.
- **Firebase's** real-time database and **Fire store** provide powerful synchronization capabilities. This is particularly useful for collaborative applications where multiple users might be interacting with the data simultaneously, ensuring that all users see the latest data without delay.





## Application's Features

Our application will be comprised of two sites at first. There will be Client site and a Admin portal.

### **Client Side:**

- At client site options will be there for Donor and Volunteer to register and login.
- At first there will be basic pages for the visitor showing some stats(total donations made, projects on going etc..) and listing some services.
- Donor donating greater than 100,000 will be able to write testimonials which will be listed onto the home page.
- For Donor after logging in he will be able to see his profile he has created, can edit it and also will be able to donate to projects and campaigns.
- He will also be able to see the transaction history listing to what project or campaign he has donated at what time and date.
- For volunteer after logging in , he will also be able to see his profile he has created ,can edit it.
- He will also be able to write complains to the Admin.
- Top volunteer contributing to most projects will also get listed.
- He also will be able to apply for different projects to different franchises which are registered (*will be listed*).

### **Admin Side(portal):**

- For admin side at first there is main admin. He has no record but take **him as the owner of the entire network who took the initiative to start this chain of charity network.**
- Admin will also be able to resolve complaints from different volunteers

He can only **Register franchises** and will be able to see all stats and figures going under the hood of his charity network's database.

- At second the franchises that will be registered by the admin himself will be listed.
- The owner (*user in our case or the one having control*) of the franchise will log into the account of that registered **Franchise Admin Panel**.
- He will then be able to approve projects from volunteer(*volunteer from client side will propose projects to listed registered franchises*).
- Franchise owner can also add Beneficiaries, can update and delete them.
- He will be able to create, update and delete campaigns.

## Visual Representation of Client And Admin Interaction

