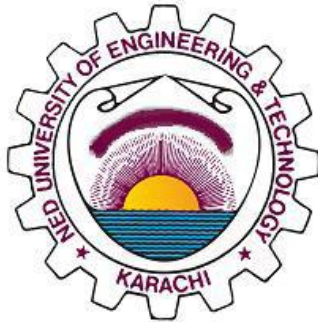


# UNDERGRADUATE FINAL YEAR PROJECT REPORT

*Department of Software Engineering*

NED University of Engineering and Technology



## Zakat Collection App

Group Number: 5  
Batch: 2019 – 2023

### Group Members:

- ◆ Abdul Hannan SE-19041
- ◆ Raja Hashir SE-19046
- ◆ M Daniyal Aamir SE-19049
- ◆ Saad Ashraf SE-19095

**Approved by**

Engr. Sidra Masood

*Lecturer*  
(Project Advisor)

## Author's Declaration

We declare that we are the sole authors of this project. It is the actual copy of the project that was accepted by our advisor(s) including any necessary revisions. We also grant NED University of Engineering and Technology permission to reproduce and distribute electronic or paper copies of this project.

Signature & Date	Signature & Date	Signature & Date	Signature & Date
Abdul Hannan SE-19041 <a href="mailto:hannan4204661@cloudu.edu.pk">hannan4204661@cloudu.edu.pk</a>	Raja Hashir SE-19046 <a href="mailto:mehmood4204792@cloudu.edu.pk">mehmood4204792@cloudu.edu.pk</a>	M Daniyal Aamir SE-19049 <a href="mailto:aamir4204608@cloudu.edu.pk">aamir4204608@cloudu.edu.pk</a>	Saad Ashraf SE-19095 <a href="mailto:ashraf4204654@cloudu.edu.pk">ashraf4204654@cloudu.edu.pk</a>

## **Abstract**

Poverty and hunger are some of the most pressing issues faced by societies worldwide. In many countries, charitable organizations collect Zakat, a form of obligatory charity in Islam, from individuals to provide assistance to those in need. However, the conventional receipt system used for door-to-door Zakat collection is highly susceptible to fraud and mismanagement, causing donors to lose trust in charitable organizations. To address this issue, we propose an automated system to streamline the record-keeping and auditing procedures for Zakat collection.

The system aims to enhance the transparency and credibility of charitable organizations by establishing trust among donors, minimizing fraud, and enabling proper auditing and funds management. It comprises five user classes: Super Admin, Admin, Office Secretary, Worker/Collector, and Donor. Each class has specific functions and privileges, and the system operates on a client-server architecture with a backend hosted on Firebase and two frontends using React.js and React Native.

The proposed system offers numerous benefits, including reduced collection costs, secure donation records, and real-time donation confirmations. By providing an automated and paperless solution, the system can eliminate delays in donation confirmations and establish better marketing strategies for organizations. Ultimately, the system's implementation aims to increase the number of donations for public welfare and contribute to the eradication of poverty and hunger.

## Acknowledgments

First, we would like to express our profound gratitude to almighty Allah for blessing us with the strength to achieve the set targets of our project.

The completion of the targets of our project could have never been possible without the participation and coordination of each and every member of the group. Here is a little effort to show our deepest gratitude to **Engr. Sidra Masood** (Project Internal Supervisor) despite being extremely bound to her duties served her best to guide and supervise us. Besides the Project Internal, we would like to appreciate **Mr. Akif Surya** (Project External Supervisor), too, to invest his precious time in communicating and guiding us with the project requirements.

The task achievement is the result of the efforts of all the people mentioned above. Our deepest regards to all the lecturers of the Software Engineering Department, NEDUET

# Contents

<b>Author's Declaration .....</b>	<b>2</b>
<b>Abstract.....</b>	<b>3</b>
<b>Acknowledgments .....</b>	<b>4</b>
<b>List of Tables .....</b>	<b>Error! Bookmark not defined.</b>
<b>United Nations Sustainable Development Goals .....</b>	<b>8</b>
<b>Chapter 1 INTRODUCTION .....</b>	<b>1</b>
<b>1.1. Background Information.....</b>	<b>1</b>
<b>1.2. Problem Statement.....</b>	<b>1</b>
<b>1.3. Proposed Solution .....</b>	<b>2</b>
<b>1.4. Significance and Motivation.....</b>	<b>2</b>
<b>1.5. Objectives.....</b>	<b>2</b>
<b>1.6. Benefits.....</b>	<b>2</b>
<b>Chapter 2 LITERATURE REVIEW .....</b>	<b>4</b>
<b>2.1. Introduction.....</b>	<b>4</b>
<b>2.2. Current Situation of Zakat Collection in Pakistan .....</b>	<b>4</b>
<b>2.3. Importance of Zakat in Islam .....</b>	<b>4</b>
<b>2.4. Factors Affecting Zakat Collection .....</b>	<b>4</b>
<b>2.5. Existing Zakat Collection Systems .....</b>	<b>5</b>
<b>2.5.1. Alkhidmat Foundation's Zakat Collection System.....</b>	<b>5</b>
<b>2.5.2. Saylani Welfare Trust's Zakat Management System .....</b>	<b>5</b>
<b>2.5.3. Pakistan Bait-ul-Mal's Zakat Distribution System.....</b>	<b>6</b>
<b>2.5.4. Bait-us-Salam Welfare Trust's Zakat Management System.....</b>	<b>6</b>
<b>2.6. Comparison of Existing Systems .....</b>	<b>6</b>
<b>2.7. Research Summary .....</b>	<b>7</b>
<b>2.8. Conclusion .....</b>	<b>7</b>
<b>Chapter 3 REQUIREMENTS GATHERING.....</b>	<b>9</b>
<b>3.1. Introduction.....</b>	<b>9</b>
<b>3.2. Product Overview .....</b>	<b>9</b>

<b>3.3. User Classes and Characteristics</b>	10
3.3.1. General Secretary	10
3.3.2. Chairperson	10
3.3.3. Office Secretary	11
3.3.4. Worker	11
3.3.5. Donor	11
<b>3.4. Key Features</b>	11
3.4.1. General Secretary Module	12
3.4.2. Chairperson Module	12
3.4.3. Office Secretary Module	12
3.4.4. Worker Module	12
3.4.5. Donor Module	12
<b>3.5. Operating Environment</b>	13
<b>3.6. System Features</b>	14
3.6.1. General Secretary (Super admin) Module	14
3.6.2. Chairperson (Admin) Module	16
3.6.3. Office Secretary Module	19
3.6.4. Worker Module	22
3.6.5. Donor Module	25
<b>Chapter 4 SYSTEM ARCHITECTURE AND DESIGN</b>	27
4.1. Introduction	27
4.2. Software Architecture	27
4.2.1. Frontend System	28
4.2.2. Backend System	28
4.2.3. Database System	28
4.2.4. Third-Party Services	28
4.3. Database Design	28
4.3.1. Entity Relationship Model	29
A. <i>Entities</i>	29

<b><i>B. Attributes</i></b> .....	29
<b><i>C. Relationships</i></b> .....	31
<b><i>D. ER Diagram</i></b> .....	31
<b>4.3.2. Schema Design Components</b> .....	32
<b>4.4. Backend Design</b> .....	34
<b>4.4.1. Backend Design Components</b> .....	34
<b>4.4.2. General Scenario (Flow of Request-Response Cycle)</b> .....	36
<b>4.4.3. Authentication and Authorization</b> .....	38
<b>References</b> .....	41

## **United Nations Sustainable Development Goals**

The Sustainable Development Goals (SDGs) are the blueprint to achieve a better and more sustainable future for all. They address the global challenges we face, including poverty, inequality, climate change, environmental degradation, peace and justice. There is a total of 17 SDGs as mentioned below. Check the appropriate SDGs related to the project.

- ☐ No Poverty
- ☐ Zero Hunger
- ☐ Good Health and Well being
- ☐ Quality Education
- ☐ Gender Equality
- ☐ Clean Water and Sanitation
- ☐ Affordable and Clean Energy
- ☐ Decent Work and Economic Growth
- ☐ Industry, Innovation and Infrastructure
- ☐ Reduced Inequalities
- ☐ Sustainable Cities and Communities
- ☐ Responsible Consumption and Production
- ☐ Climate Action
- ☐ Life Below Water
- ☐ Life on Land
- ☐ Peace and Justice and Strong Institutions
- ☐ Partnerships to Achieve the Goals



# **Chapter 1**

## **INTRODUCTION**

### **1.1. Background Information**

Zakat is one of the five pillars of Islam and an important source of welfare for the poor and needy in countries like Pakistan. While many charitable organizations collect Zakat donations, there is still a significant population that is not being served. This indicates a need for improvements to increase the number of donations.

Several factors have been identified as contributing to this issue:

- The door-to-door collection system lacks the ability to establish trust among donors. Only 33% of donors submit their Zakat to NGOs, and they donate larger amounts than other donors. By increasing trust, the number of donations may rise.
- Fraud and corruption among workers collecting donations also impact donor trust making it difficult to reach fundraising targets. Controlling these practices can help increase the number of funds collected.
- The current manual receipt system is inefficient, with provisional receipts being unauthentic and official confirmation taking weeks. This demotivates donors and leaves room for improvement to encourage more donations.
- Many people in the country are not aware of or do not prefer to donate online. Most still donate with cash or checks. It is important to consider the needs and preferences of these individuals as well.

### **1.2. Problem Statement**

The conventional receipt system for collecting donations is highly susceptible to fraud and mismanagement, leading to a lack of trust between donors and charitable organizations. Donors are often left uncertain about the proper use of their contributions, which can cause many to show less interest in donating. As a result, the donations for public welfare, which are vital for eliminating poverty and hunger, fall short of the required amount.

### 1.3. Proposed Solution

The proposed solution is an automated system for recording and auditing door-to-door collections of Zakat by charitable organizations. The system aims to reduce the chances of fraud, establish trust among donors, and provide an efficient, secure, and transparent way of tracking donations. The system will minimize the cost of collection, eliminate long delays, and enable a proper audit of transactions to maintain checks and balances of funds.

### 1.4. Significance and Motivation

The significance of the proposed system lies in its ability to improve the relationship between donors and charitable organizations by providing transparency and accountability. The system will encourage donors to contribute more, and the automated and paperless process will make it easier for charitable organizations to collect and manage donations efficiently.

The motivation behind this project is to create a system that can help charitable organizations collect funds effectively and transparently, ultimately helping to eliminate poverty and hunger by ensuring that the collected funds are distributed appropriately.

### 1.5. Objectives

The objectives of the system are:

- To establish trust among donors.
- To enable donors to track their donations easily.
- To minimize the chances of fraud.
- To enable proper auditing of transactions and maintain checks and balances of funds.

### 1.6. Benefits

Overall, the system has the potential to improve the organization's ability to attract and retain donors, increase the impact of its fundraising efforts, and ultimately achieve its goals more effectively. Some of the key benefits are listed below:

- ***Improved donation management:*** The system will provide a centralized platform for

managing donations, making it easier to track and analyze donations, as well as monitor fundraising progress.

- ***Increased transparency:*** By providing donors with real-time access to information about their donations, the system can increase transparency and trust between the organization and its donors.
- ***Enhanced communication:*** The system will allow the organization to send promotional messages to potential donors and keep them informed about the latest campaigns and events.
- ***Streamlined operations:*** By automating many of the manual processes involved in donation management, the system can help reduce administrative burdens, save time, and increase efficiency.
- ***Better decision-making:*** With improved data analysis and reporting, the system can provide valuable insights into donor behavior, campaign effectiveness, and other key metrics that can help the organization make better-informed decisions.
- ***Increased security:*** The system can help to improve the security of the donation process by providing secure channels for donors to make donations and ensuring that all financial transactions are properly tracked and recorded.
- ***Scalability:*** The system can easily scale to meet the growing needs of the organization, whether that involves expanding to new regions or increasing the number of campaigns and events.

## **Chapter 2**

### **LITERATURE REVIEW**

#### **2.1. Introduction**

This chapter provides a literature review of existing research on Zakat collection systems, donation management, and related systems. This section discusses the current situation of Zakat collection in Pakistan, the importance of Zakat in Islam, and the factors affecting the collection process. The review also analyzes various solutions proposed to address these issues and provides a summary of the findings. The review also focuses on identifying the key features, advantages, and limitations of the existing systems, as well as identifying the research gaps that this project aims to address.

#### **2.2. Current Situation of Zakat Collection in Pakistan**

Zakat is an essential religious obligation in Pakistan, but there is a lack of awareness among people about its significance. According to [1], only 2.5% of Pakistanis pay Zakat, and the majority of them do not pay the correct amount. Furthermore, the manual Zakat collection system in Pakistan is outdated, inefficient, and prone to corruption, making it challenging to collect and manage funds transparently [2]. Additionally, the majority of Zakat donations in Pakistan are made in cash, making it difficult to track and manage them [3].

#### **2.3. Importance of Zakat in Islam**

Zakat is one of the five pillars of Islam, and its importance cannot be overstated. According to [4], the payment of Zakat is a religious obligation that is incumbent upon every adult Muslim who has the financial means to pay it. The payment of Zakat is also a means of purification and a source of reward from Allah [5]. Zakat is seen as an important source of welfare for the poor and needy in Islamic societies and is intended to help redistribute wealth more equitably [6].

#### **2.4. Factors Affecting Zakat Collection**

Several factors impact the collection of Zakat in Pakistan, including the lack of trust between donors and charitable organizations, the absence of a proper system for tracking donations, and

the preference for cash donations [7]. Additionally, the lack of transparency and accountability in the Zakat collection process, along with corruption and fraud, have a significant impact on the number of donations made [8]. According to [9], a lack of awareness and understanding about the Zakat system and the need for Zakat also contribute to the low collection rate.

## **2.5. Existing Zakat Collection Systems**

Several Zakat collection systems have been proposed to address the challenges faced by charitable organizations. According to [10], the traditional Zakat collection system in Pakistan is primarily manual and outdated. The system involves the door-to-door collection, with handwritten receipts being issued to donors. The system is prone to corruption, fraud, and mismanagement, making it challenging to track and manage donations effectively.

However, in recent years, various Zakat collection and management systems have been developed to facilitate the donation process and improve transparency and accountability. The following subsections describe some of the existing systems that have been implemented in Pakistan.

### **2.5.1. Alkhidmat Foundation's Zakat Collection System**

Alkhidmat Foundation is a leading charitable organization in Pakistan that collects Zakat, Sadqat, and other donations for the poor and needy. The organization has developed a Zakat collection system that uses biometric verification to ensure the authenticity of the donor and the donation amount. The system generates a unique QR code for each donor, which can be used to track the donation history and monitor the utilization of funds. [11]

### **2.5.2. Saylani Welfare Trust's Zakat Management System**

Saylani Welfare Trust is another leading charitable organization in Pakistan that uses a web-based Zakat management system to manage the collection and distribution of Zakat funds. The system provides an online portal for donors to make donations and allows the organization to monitor and track the funds in real time. The system also generates detailed reports on the collection and utilization of funds. [12]

### 2.5.3. Pakistan Bait-ul-Mal's Zakat Distribution System

Pakistan Bait-ul-Mal is a government agency responsible for the collection and distribution of Zakat funds in Pakistan. The organization has implemented a Zakat distribution system that uses a database to maintain the record of eligible beneficiaries and the amount of Zakat distributed to them. The system provides a transparent and efficient way of distributing Zakat funds to the needy. [13]

### 2.5.4. Bait-us-Salam Welfare Trust's Zakat Management System

Baitussalam is a charity organization that is officially registered in Pakistan. The organization offers several options for individuals who wish to donate Zakat to their cause. These options include making donations online through their website using a secure donation portal, transferring Zakat directly to their bank account, donating in cash at their office in Karachi, using mobile banking options such as Bank Islami Meezan Bank and Faysal Bank, utilizing door-step collection services with TCS and Leopards Courier Services as their collection partners, or making anonymous donations through the 1 Link Top Up method. [14]

Once you have chosen your preferred donation method, you can select your desired cause and make your donation accordingly.

## 2.6. Comparison of Existing Systems

Several studies have compared various Zakat collection systems to determine which is the most effective. According to [15], an electronic Zakat system is the most efficient method for collecting Zakat, providing a transparent and secure platform for donors to make donations. The study also found that electronic Zakat systems were more effective than traditional Zakat collection methods in increasing the number of donations made.

**Table 1: Comparison of Existing and Proposed Zakat Collection Systems**

Aspect	Manual System	Electronic System
<i>Trust and Transparency</i>	Limited transparency; Lack of trust	Increased transparency; Improved trust

<i><b>Efficiency</b></i>	The manual collection process is time-consuming	The automated collection process is more efficient
<i><b>Security</b></i>	The cash collection system is prone to fraud	The online donation system is secure
<i><b>Scalability</b></i>	Difficult to scale to meet growing needs	Easy to scale and customize for different organizations

## 2.7. Research Summary

In the literature review, the author discussed the various existing Zakat collection and management systems in Pakistan, including Alkhidmat Foundation's Zakat Collection System, Saylani Welfare Trust's Zakat Management System, and Pakistan Bait-ul-Mal's Zakat Distribution System. The review also identified the factors that affect Zakat donation behavior in Pakistan, such as religious beliefs, trust in charitable organizations, perceived effectiveness of donations, and social influence. Additionally, the review highlighted the effectiveness of using technology in Zakat management systems to improve transparency and accountability, as well as the impact of social media on increasing awareness and promoting Zakat donations. Overall, the review identified several research gaps that this project aims to address.

## 2.8. Conclusion

Research has shown that the current manual Zakat collection system in Pakistan is inefficient, outdated, and prone to corruption, making it challenging to collect and manage funds transparently. Several studies have proposed automated Zakat collection systems as a solution to address these issues. These systems provide a transparent, efficient, and secure platform for donors to make donations and enable charities to manage donations more effectively. Comparisons between existing systems have shown that electronic Zakat systems are the most effective method for collecting Zakat, providing a transparent and secure platform for donors to make donations. The importance of Zakat in Islam cannot be overstated, as it is one of the five pillars of Islam and is seen as an important source of welfare for the poor and needy in Islamic societies. Factors affecting the collection of Zakat in Pakistan include the lack of trust between donors and charitable organizations, the absence of a proper system for tracking

donations, and the preference for cash donations. Lack of awareness and understanding about the Zakat system and the need for Zakat also contribute to the low collection rate. Automated donation management systems can help reduce the administrative burden associated with manual donation management, making it easier for charities to collect and manage donations. Overall, the use of an electronic Zakat system can potentially increase the number of donations made and improve the efficiency, transparency, and accountability of the Zakat collection process in Pakistan.



## **Chapter 3**

# **REQUIREMENTS GATHERING**

### **3.1. Introduction**

This chapter provides a concise overview of the Zakat Collection System, focusing on the project perspective, user classes, key features, operating environment, and functional requirements. It serves as a foundational understanding of the system's needs and objectives, ensuring alignment between stakeholders and the development team. By documenting these requirements, the chapter ensures clarity and coherence among stakeholders, users, and the development team, guiding subsequent stages of design, development, and testing.

### **3.2. Product Overview**

Our product is a “Zakat Collection System”. The system replaces the conventional systems of zakat collection and management by automating the record-keeping and receipt system with an app that tracks the activities in real-time. As far as the users are concerned, the main criteria for their needs are record management and retrieval. However, the system infrastructure comprises a hierarchy of personnel that collectively contribute to the system journey i.e. the flow/cycle of various operations. Hence, they play a pivotal role in determining the requirements of the system. The requirements for each user class, however, will be discussed later in this chapter.

The system allows donors to make requests for donations. The requests go to the relevant chairperson depending upon the area from where the donor has requested for donation. The chairperson, on the other hand, has the privilege to decide if the request should be approved by verifying the details entered by the donor on his/her own. Supposing that the request is approved, then the donation is entitled as a pending donation and now it can be displayed to each worker entity designated in that area for collection activities.

Apart from only approving the requests, the chairpersons can also add new pending collections on behalf of donors, because many of the donors don't directly use the app. They generally ask

to donate through phone calls or paying a physical visit. In this case, the chairperson can add pending records on behalf of donors (either already registered, or unregistered).

As far as workers are concerned, it is already mentioned that they will be shown with all pending donations in the area. They can accept these, based on their interest, proximity, and convenience, which means that once they have accepted a record, then they are responsible to go to the provided address, collecting the money, and then marking the donation as collected. The thing to note here is that just as the donation is marked as collected, the donor will receive the confirmation through a notification, which provides him/her a level of confidence that the record has been kept centrally, on the organization level.

Workers can also report new collections because on paying a visit, they may find other donors who are interested to donate without having made a request in the first place. In this case, the workers can still collect their zakat, and fill the required details. If a donor exclusively, is not having an account, the worker first registers him, and then reports the collection.

In addition, concerning about the on-premises collection, the office secretaries are working as the reporting entity. They can directly add collection records, and also send promotional messages in the long run, to inform them about the current conditions, donation campaigns, and relief programs in order to motivate them to donate.

### **3.3. User Classes and Characteristics**

#### **3.3.1. General Secretary**

A member of the executive body/office bearers, which is generally known as the general secretary will act as the super admin. The super admin will have the maximum rights and privileges, and will assign the area-wise admins.

#### **3.3.2. Chairperson**

An admin is typically the chairman of a particular region assigned by the super admin. The admin has the authority over the operations involved in his/her own particular region which include collection of funds, and ability to monitor the collections.

### 3.3.3. Office Secretary

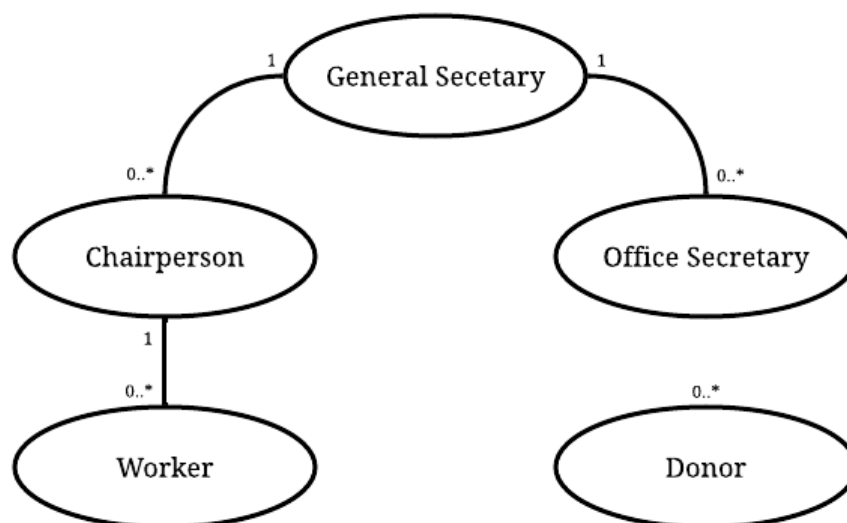
Office secretary is a full-time paid employee of the organizations who handles daily operations involved inside the premises of the office.

### 3.3.4. Worker

A worker is a trusted person of the organization, who has the responsibility to safely collect the funds from the donors and report it to the regional chairman.

### 3.3.5. Donor

Donor is any person who wishes to donate money to the organization.



**Figure 1** – User classes (Hierarchy of personnel)

Logically and practically, there is only one general secretary, acting as a super admin, privileged to manage the on-premises office secretaries and area-wise chairpersons. The chairpersons working independently can manage the workers below them. The donors are not part of the hierarchy as they are residing externally, but they are the most important part of the operations.

## 3.4. Key Features

On a broader level, the system is divided into five modules with respect to its functions.

### **3.4.1. General Secretary Module**

- Login with credentials
- Create/remove admin accounts.
- View the status and details of all donations.

### **3.4.2. Chairperson Module**

- Login with credentials.
- Create/remove worker accounts.
- Add a new pending donation with details.
- View the status and details of regional donations only.

### **3.4.3. Office Secretary Module**

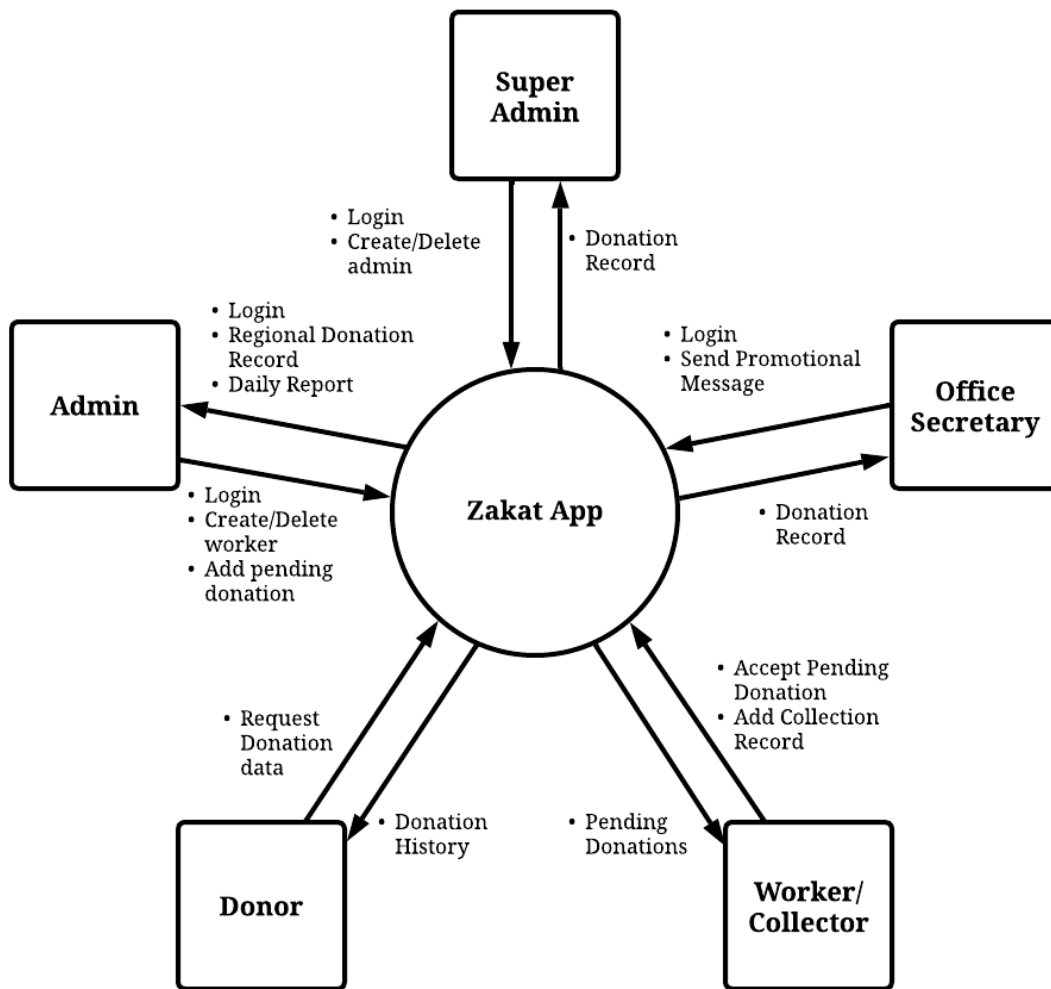
- Login with credentials.
- View the status and details of all donations.
- Add the new records of the donations collected at the office.
- Send promotional messages for different campaigns that encourage the donor to donate.

### **3.4.4. Worker Module**

- Login with credentials.
- View all pending donations added by the respective admin.
- Accept a pending donation.
- Submit the record of the collected donation through form
- Submit a new and non-existing donation record with details.

### **3.4.5. Donor Module**

- View the details of all the past donations made by the donor.
- Get the instant confirmation of the currently made donation.



**Figure 2 – Context Level Diagram**

The figure shows the various entities and the flow the information between the system and the entities. The arrow heads typically delineate the direction of information flow. We can also say that this diagram shows the entities, and their inputs and outputs.

### 3.5. Operating Environment

The operating environment refers to the specific hardware, software, and infrastructure requirements necessary for the proper functioning and deployment of a system or software application. It encompasses the technological components and dependencies that are essential for the system to operate effectively.

In the context of the Zakat Collection System, the operating environment of the system consists of the following:

- The system will be following a client-server architecture, meaning that the backend will be hosted as a web service, which will be integrated with the frontend/s (the user application).
- The backend will consist of a Node/Express monolith app, with MySQL as database.
- There will be two frontends for each respective user. They are:
  - For admin and office secretary, a simple web application will be built using React.js, which will be supported by all the modern web browsers.
  - The web app will be hosted on Vercel, a cloud-based platform for hosting frontends with high performance instantly by using CI/CD pipeline.
  - For the chairperson, worker and donor, a native mobile application will be built using React Native, with cross-platform support for IOS and Android.

## 3.6. System Features

This section explains the functional requirements of the system in a detailed a modular manner.

### 3.6.1. General Secretary (Super admin) Module

#### 1. Create Chairperson/Office Secretary Account

<b>Description</b>	The super admin can create the accounts of the admin or office secretary. No one has the right to become an admin without the consent of the super admin. This is intended to ensure security by limiting the privilege to a single person.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Super admin is logged in
<b>Stimuli</b>	The identity information of the admin/office secretary
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	A new admin account is created

<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the super admin to enter the new admin or office secretary's information.</li> <li>2. The system should validate the form, to check for correct, unique, and complete information.</li> <li>3. The system should show the error fields if the entered form is invalid.</li> <li>4. If the entered form is valid, the system should process the request and create a new admin or office secretary record in the database.</li> <li>5. The system should indicate the success message if the new admin or office secretary has been created.</li> <li>6. The system should send the login credentials of the new admin or office secretary to his/her provided email.</li> </ol>
--------------	---

## 2. Delete Admin Account

<b>Description</b>	The super admin can delete the accounts of the admin or office secretary. This right is also limited only to the super admin for security.
<b>Priority</b>	Medium
<b>Pre-Conditions</b>	Super admin is logged in
<b>Stimuli</b>	The ID of the admin or office secretary to remove
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	The selected user's entry is soft deleted from the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the super admin to search for and select the admin or the office secretary to delete.</li> <li>2. The system should trigger a message to indicate the status of the request as successful or failed.</li> </ol>

## 3. View Donation Records

<b>Description</b>	The super admin must be able to view the transactional records of all the donations with filters. This is intended to help the general secretary monitor the collection process and make decisions.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Super admin is logged in
<b>Stimuli</b>	Optionally the search query
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	The selected user's entry is soft deleted from the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the super admin to view the history of donation records in the form of a table, with totals.</li> <li>2. The system should provide the necessary search filters such as the timespan, admin, etc.</li> </ol>

### 3.6.2. Chairperson (Admin) Module

#### 1. Create Worker Account

<b>Description</b>	The admin can create the accounts of workers. Only the admins have the right to provide user accounts to workers of their respective regions.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Admin is logged in
<b>Stimuli</b>	The identity information of the worker
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	A new worker account is created
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the admin to enter the new worker's information.</li> </ol>



	<ol style="list-style-type: none"> <li>2. The system should validate the worker creation form, to check for correct, unique, and complete information.</li> <li>3. The system should show the error fields if the entered form is invalid.</li> <li>4. If the entered form is valid, the system should process the request and create a new worker record in the database.</li> <li>5. The system should indicate the success message if the new worker has been created.</li> </ol>
--	--

## 2. Delete Worker Account

<b>Description</b>	The admin can delete the accounts of the worker of his own area only. This right is also limited only to the respective admins only, for security.
<b>Priority</b>	Medium
<b>Pre-Conditions</b>	Admin is logged in
<b>Stimuli</b>	The ID of the worker to remove
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	The selected worker's entry is soft deleted from the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the admin to search for and select the worker to delete.</li> <li>2. The system should trigger a message to indicate the status of the request as successful or failed.</li> </ol>

## 3. Add a Pending Donation

<b>Description</b>	In order to let the workers update with the donations to collect, the admin can add a record in a database collection for pending donations in a particular region.
<b>Priority</b>	High

<b>Pre-Conditions</b>	Admin is logged in
<b>Stimuli</b>	The donor's identity, contact, and address information.
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	The new pending donation record is updated in the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the admin to enter the donor identity, contact, and address information through a form.</li> <li>2. The system should validate the form, to check for correct, unique, and complete information.</li> <li>3. The system should show the error fields if the entered form is invalid.</li> <li>4. If the entered form is valid, the system should process the request and create a new pending donation record in the database.</li> <li>5. The system should indicate the success message if the new pending donation record has been created.</li> </ol>

#### 4. View Regional Donation Record

<b>Description</b>	The admin must be able to view the transactional records of only the corresponding regional donations with filters. This is intended to help the regional chairman monitor and report the collection process in their area.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Admin is logged in
<b>Stimuli</b>	Optionally the search query
<b>Response</b>	The relevant regional donation records
<b>Post Conditions</b>	-

<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the super admin to view the history of donation records in his/her region only, in the form of a table, with totals.</li> <li>2. The system should provide the necessary search filters such as the timespan etc.</li> </ol>
--------------	--

## 5. Generate Report

<b>Description</b>	The admin should be able to generate a daily report to show the stats and important information about the collections done, in order to report it to the executive body (the upper management).
<b>Priority</b>	Medium
<b>Pre-Conditions</b>	Admin is logged in
<b>Stimuli</b>	The date to generate the report for.
<b>Response</b>	The daily report
<b>Post Conditions</b>	-
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The admin should be able to easily input the date whose report has to be generated.</li> <li>2. The system should validate the date before proceeding with generation of report.</li> <li>3. If the date is ahead of the current date, then the system should trigger with an error message.</li> <li>4. On requesting, the system will generate and display the report.</li> <li>5. The report generated should be downloadable as a PDF file.</li> </ol>

### 3.6.3. Office Secretary Module

#### 1. Add Donation Record

<b>Description</b>	The office secretary might collect donations inside the premises of the office. The system should facilitate him/her to update the record of those donations to the application.
<b>Priority</b>	Medium
<b>Pre-Conditions</b>	The office secretary is logged in
<b>Stimuli</b>	The donation information.
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	The new donation record is updated in the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the office secretary to enter the donor information and the amount of money donated.</li> <li>2. The system should validate the form, to check for correct, and complete information.</li> <li>3. The system should show the error fields if the entered form is invalid.</li> <li>4. If the entered form is valid, the system should process the request and create a new collection record in the database.</li> <li>5. The system should indicate the success message if the new donation record has been created.</li> <li>6. The system must send an acknowledgment message to the donor's contact with the amount and complete details for his/her record.</li> </ol>

## 2. View Donation Records

<b>Description</b>	The office secretary must be able to view the transactional records of all the donations with filters. This is intended to help him/her monitor the collection process and make decisions.
<b>Priority</b>	High
<b>Pre-Conditions</b>	The office secretary is logged in

<b>Stimuli</b>	Optionally the search query
<b>Response</b>	The relevant donation records
<b>Post Conditions</b>	-
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the office secretary to view the history of donation records in the form of a table, with totals.</li> <li>2. The system should provide the necessary search filters such as the timespan, admin, etc.</li> </ol>

### 3. Promotional Messaging

<b>Description</b>	The office secretary should also be able to send promotional messages to the donors through the system. These messages may be related to any donation campaign, or they might be for a specific group of individuals who have already donated to the organization, in order to ask them to donate again. This is beneficial in collecting maximum funds.
<b>Priority</b>	Middle
<b>Pre-Conditions</b>	Office secretary is logged in
<b>Stimuli</b>	The message body and the group of recipients
<b>Response</b>	The status showing if the message was sent
<b>Post Conditions</b>	All the recipients must receive the message.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should provide an interface to the office manager where he/she can type the message body and select the group of individuals to send the message.</li> <li>2. The system should also provide dynamic message generation through a single template formatted with some variables.</li> <li>3. There should be validation to ensure if the message body is not empty, and there is at least one recipient.</li> </ol>

	4. After submitting, the system must deliver the message successfully to the contacts of all the respective individuals selected.
--	---

#### 3.6.4. Worker Module

##### 1. View Pending Donations

<b>Description</b>	The pending donations added by the regional chairman (admin), will be available to all the collectors working in that particular area, in order to keep them updated about the donations which are yet to be collected, along with contact and other related information.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Worker is logged in
<b>Stimuli</b>	The worker requests for pending donations
<b>Response</b>	The listing of currently pending donations.
<b>Post Conditions</b>	-
<b>Steps</b>	1. The worker application should have an interface to request for, and view all the pending donations in the underlying region with details.

##### 2. Accept Pending Donation

<b>Description</b>	It is necessary that only one worker is heading towards collecting the donation from a donor. It requires a signal from the worker to let the system know that he has accepted the pending donation.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Worker is logged in
<b>Stimuli</b>	The donor presses a button to accept the pending donation from the listing.

<b>Response</b>	An alert message indicating that the pending donation has been accepted.
<b>Post Conditions</b>	The pending donation record is updated in the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. From the list of the pending donations, the worker should be able to click on a button to accept any particular donation.</li> <li>2. On clicking, the database record of the pending donation should be updated with the collecting worker set to the ID of the current worker.</li> <li>3. The system should trigger a success/failure message.</li> <li>4. In case of success, the pending donation will be removed from the list of pending donations and now the worker should be able to see them in a separate list, containing those pending donations which he has accepted.</li> </ol>

### 3. Add Collection Record of Pending Donation

<b>Description</b>	On reaching the address of the donor, the worker will collect the amount of zakat (or any funds), and then fill out a form through the application. The donor will not proceed with his/her donation unless he/she has got the confirmation message, which is not possible without this step.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Worker is logged in
<b>Stimuli</b>	The pending donation, and the amount of donation
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	The new donation record is updated in the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The worker can select the current pending donation he is going to collect from a list or menu, and put the amount of donation in a form and submit it.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The system should validate the form, to check for correct, and complete information.</li> <li>3. The system should show the error fields if the entered form is invalid.</li> <li>4. If the entered form is valid, the system should process the request and create a new collection record in the database.</li> <li>5. The system should indicate the success message if the new donation record has been created.</li> <li>6. The system must send an acknowledgment message to the donor's contact with the amount and complete details for his/her record.</li> </ol>
--	--

#### 4. Add New Donation Record

<b>Description</b>	There may be some individuals who directly pay their donation amount to the worker, and their record is not already present in the database. Collecting donations from them is also important and required in a safe and secure manner like other donors. Therefore, this feature allows workers to update their record as well.
<b>Priority</b>	High
<b>Pre-Conditions</b>	Worker is logged in
<b>Stimuli</b>	The donation information
<b>Response</b>	Success/Error message
<b>Post Conditions</b>	The new donation record is updated in the database.
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The system should allow the worker to enter the donor information and the amount of money donated.</li> <li>2. The system should validate the form, to check for correct, and complete information.</li> <li>3. The system should show the error fields if the entered form is invalid.</li> <li>4. If the entered form is valid, the system should process the request and create a new collection record in the database.</li> </ol>



	5. The system should indicate the success message if the new donation record has been created. 6. The system must send an acknowledgment message to the donor's contact with the amount and complete details for his/her record.
--	---

### 3.6.5. Donor Module

#### 1. Track Past Donation Records

<b>Description</b>	This feature is intended to allow the donors to track their past donation history for their own records.
<b>Priority</b>	High
<b>Pre-Conditions</b>	The donor has donated at least once
<b>Stimuli</b>	The donor ID
<b>Response</b>	The past donation data of the donor.
<b>Post Conditions</b>	-
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The donor shall be able to submit his/her donor ID which is present in every acknowledgment message, identifying him/her.</li> <li>2. The system should generate an OTP for that donor and send it to the donor's contact.</li> <li>3. The donor should then be asked to provide the correct OTP to proceed.</li> <li>4. If the entered OTP is correct, the donation record of the donor will be queried and displayed to him/her.</li> </ol>

#### 2. Request to donate

<b>Description</b>	This feature is intended to allow the donors to make a donation request seeking approval from the chairperson
<b>Priority</b>	High

<b>Pre-Conditions</b>	The donor has an account in place and logged in
<b>Stimuli</b>	The donor ID, address, amount
<b>Response</b>	Donation request created
<b>Post Conditions</b>	-
<b>Steps</b>	<ol style="list-style-type: none"> <li>1. The donor shall be able to submit his/her donor ID which is present in every acknowledgment message, identifying him/her.</li> <li>2. The system should generate an OTP for that donor and send it to the donor's contact.</li> <li>3. The donor should then be asked to provide the correct OTP to proceed.</li> <li>4. If the entered OTP is correct, the donation record of the donor will be queried and displayed to him/her.</li> </ol>

## Chapter 4

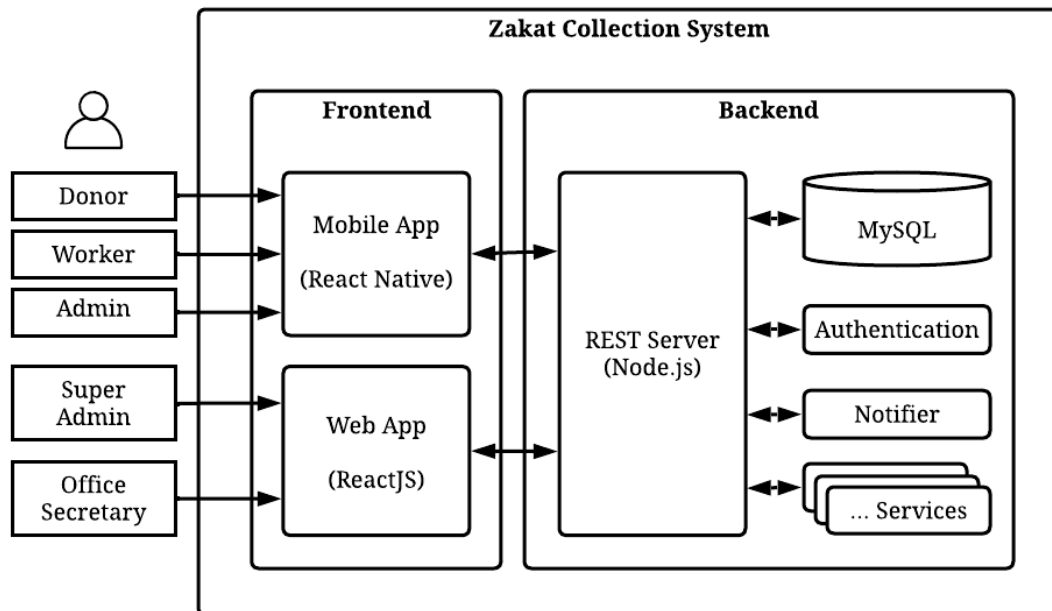
### SYSTEM ARCHITECTURE AND DESIGN

#### 4.1. Introduction

This chapter outlines the key components, modules, and functionalities of the system, along with the logical and physical structure. It discusses the database design, security measures, and user interface considerations. It also addresses the integration of technological components and third-party services, if applicable.

#### 4.2. Software Architecture

The proposed system architecture for the Zakat Collection System is designed to provide a comprehensive and efficient solution. The system consists of various modules and services, catering to five different user classes: Worker, Chairperson, Donor, General Secretary, and Office Secretary, as discussed.



**Figure 3** - Top Level Architecture

Overall, the proposed system architecture combines the use of React Native and ReactJS for frontend development, a monolithic REST Node Express backend, a MySQL database for data management, and various services to enhance user experience and communication. This architecture aims to provide a reliable, scalable, and user-friendly solution for the effective collection and management of Zakat funds.

#### **4.2.1. Frontend System**

The front end of the system is developed using React Native, a cross-platform framework, to create a mobile app for the Worker, Chairperson, and Donor user classes. This allows for a seamless user experience across multiple mobile platforms. Additionally, a web app developed using ReactJS is provided for the General Secretary and Office Secretary User classes, ensuring accessibility from desktop browsers.

#### **4.2.2. Backend System**

To streamline the backend development and ensure a cohesive architecture, a single backend system is implemented. This backend is developed as a monolithic REST Node Express application, providing a robust and scalable foundation for the entire system. The backend handles various functionalities, such as user authentication, data processing, and communication with the database.

#### **4.2.3. Database System**

For data management, a MySQL database is employed to store and manage the system's data effectively. MySQL is a widely used relational database management system known for its stability, reliability, and data integrity. It enables efficient querying and retrieval of information, essential for the smooth operation of the Zakat Collection System.

#### **4.2.4. Third-Party Services**

In addition to the core components, the system incorporates various services to enhance its functionality. Real-time notifications are implemented to keep users informed about important updates and events. Email and SMS alert services are integrated to enable communication with users, ensuring timely and relevant information delivery. These services contribute to improving user engagement and overall system efficiency.

### **4.3. Database Design**

The database design is a critical component of the Zakat Collection System, enabling efficient storage, retrieval, and manipulation of data. It encompasses the organization of data into logical

structures, relationships between entities, and implementation of data security measures.

A well-designed database ensures the accurate representation and management of system entities such as donors, transactions, and beneficiaries. It facilitates effective data storage and retrieval, enabling the system to capture and maintain essential information.

Data security is a vital consideration in the database design, with access controls, encryption, and user authentication implemented to protect sensitive donor information and ensure data confidentiality.

#### **4.3.1. Entity Relationship Model**

The Entity-Relationship (ER) model is a conceptual data model used to represent the structure and relationships of data within a database. It provides a visual representation of the entities (objects or concepts), their attributes (properties), and the relationships between entities. The ER model is widely used in database design and serves as a foundation for creating a well-structured and organized database schema.

##### **A. *Entities***

In the Entity-Relationship (ER) model, entities represent real-world objects or concepts that have distinct attributes and can be uniquely identified. Entities are the building blocks of the ER model and serve as the basis for constructing relationships between them. Each entity within the model corresponds to a table in a relational database.

In our system, our primary data layer comprises 3 broad entities: *User*, *Area*, and *Donation*.

##### **B. *Attributes***

In the Entity-Relationship (ER) model, attributes represent the specific properties or characteristics of an entity. They describe the data elements associated with an entity and provide further details about the entity's characteristics or qualities. Attributes are the building blocks of data within the ER model and are used to define the structure and content of the database.

In our system, the attribute definitions of each entity are given a following:

- **User**

<i><b>id:</b></i>	The unique identifier of a user
<i><b>createdAt:</b></i>	The time at which the user is registered.
<i><b>updatedAt:</b></i>	The time of last user update.
<i><b>firstName:</b></i>	The first part of full name.
<i><b>lastName:</b></i>	The last part of full name.
<i><b>email:</b></i>	Optional but unique for each user. Used for alerts and messages.
<i><b>emailOTP:</b></i>	Updated each time the user changes email. Used for verification.
<i><b>emailVerified:</b></i>	Specifies if the email is verified
<i><b>role:</b></i>	Specifies the class of user.
<i><b>phone:</b></i>	Unique and required. Used for alerts and messages.
<i><b>cnic:</b></i>	Unique and required. Used for identification.
<i><b>password:</b></i>	Encrypted password hash (SHA1).
<i><b>active:</b></i>	Is <b>true</b> if the user is not removed.
<i><b>areaId:</b></i>	ID of assigned area, empty otherwise.
<i><b>assignedAt:</b></i>	Time of worker area assignment.

- **Area**

<i><b>id:</b></i>	The unique identifier of an area
<i><b>createdAt:</b></i>	The time at which the user is created.
<i><b>updatedAt:</b></i>	The time of last area update.
<i><b>areaName:</b></i>	The unique and required name of the area.
<i><b>chairpersonId:</b></i>	The id of user (chairperson), the area is assigned to.
<i><b>assignedAt:</b></i>	Time of chairperson area assignment.

- **Donation**

<i><b>id:</b></i>	The unique identifier of a donation
<i><b>createdAt:</b></i>	The time at which the donation is created.
<i><b>updatedAt:</b></i>	The time of last donation record update.
<i><b>areaId:</b></i>	The id of area in which the donation is done.
<i><b>donorId:</b></i>	The id of donor making the donation, if registered, else empty
<i><b>amount:</b></i>	The amount of money donor wants to pay in rupees.
<i><b>address:</b></i>	The address where the collection is to be made.

<b><i>status:</i></b>	Indicates donation as requested, pending, accepted, collected
<b><i>refName:</i></b>	The name of donor if not registered, else empty.
<b><i>refPhone:</i></b>	The phone of donor if not registered, else empty.
<b><i>workerId:</i></b>	The id of worker that accepts/collects a donation.
<b><i>requestedAt:</i></b>	The time at which the donor requests for donation, if applicable.
<b><i>approvedAt:</i></b>	The time at which the donation was approved, if applicable.
<b><i>acceptedAt:</i></b>	The time at which the donation was accepted, if applicable.
<b><i>collectedAt:</i></b>	The time at which the collection was done.

### ***C. Relationships***

In an Entity-Relationship (ER) model, relationships represent the associations or connections between entities. They define the way entities interact with each other and establish the logical connections between different parts of the database. Relationships play a vital role in capturing the dependencies and interactions between entities, providing a comprehensive view of the data model.

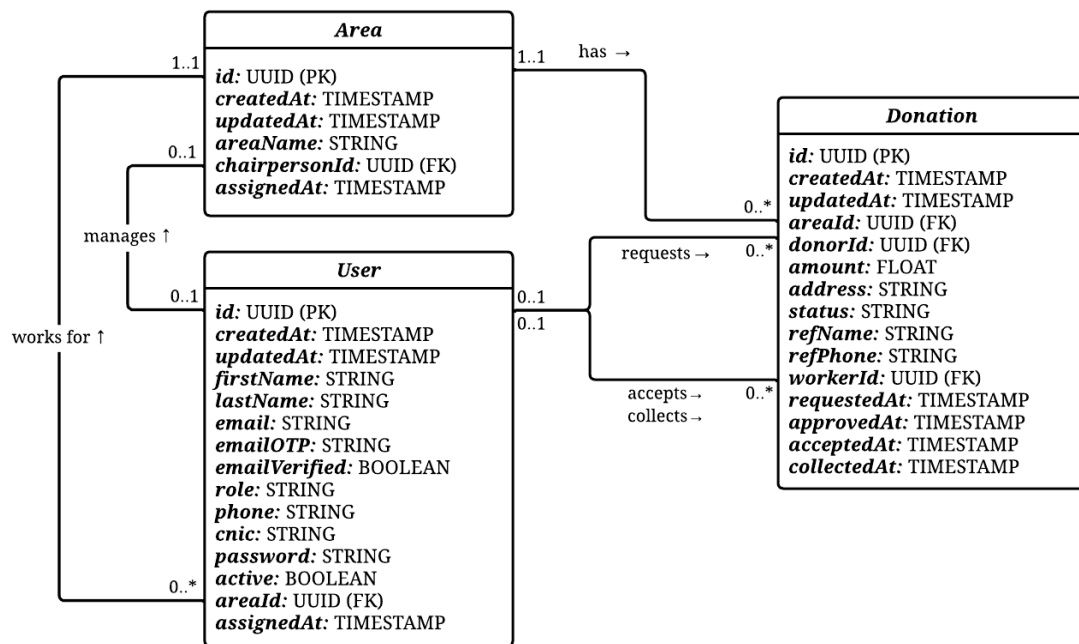
In our system, the entity relationships are listed as below:

- At most one chairperson manages an area and an area is managed by at most one chairperson
- A worker works for one and only one area. Many workers can work in an area
- An area has many donations
- A donation is made by only one donor. Donors can request multiple donations
- A donation is accepted/collected by only one worker. Workers can take many donations.

### ***D. ER Diagram***

An Entity-Relationship (ER) diagram is a graphical representation of the entities, attributes, relationships, and cardinality involved in a database system. It is a visual tool used to illustrate the structure and organization of data within a system, showcasing the entities and their relationships. The entities are represented by rectangular boxes, having titles as names of entities, and inside the boxes, the attribute definitions are depicted. The relationships are

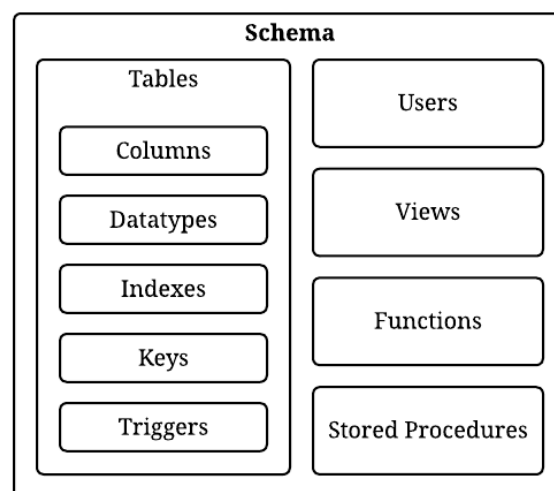
represented by joining the entities by edges, showing the relationship and cardinalities.



**Figure 4** – Entity Relationship Diagram (ERD)

The rectangular containers representing **entities**, the identifiers in PascalCase in the title representing entities names, the identifiers in camelCase representing **attributes**, followed by data types. The edges representing **relationships**, with identifiers, arrow heads showing the direction of relationship, can multiplicities (cardinalities and participations) on both sides.

#### 4.3.2. Schema Design Components



**Figure 5** – Key components of Schema design



The details of components of schema for our system are as follows:

### ***A. Tables***

A table represents a collection of related data entities, with each entity represented as a row and each attribute represented as a column. Tables define the structure, data types, and constraints for storing data in a tabular format. Tables have:

- ***Columns***: In a database table, a column represents a specific data attribute or field. It defines the individual data elements that make up a row in the table. Each column has a name and a specific data type.
- ***Datatypes***: A datatype defines the type of data that can be stored in a column of a database table. It specifies the nature and format of the data, such as numeric, string, date, or Boolean values.
- ***Indexes***: An index is a database structure that improves the retrieval performance of data from a table. It is created on one or more columns of a table to allow faster data access based on specific search criteria.
- ***Keys***: A key is a column or set of columns that uniquely identifies records in a table. Primary keys are unique identifiers assigned to each record and enforce the uniqueness of values in a specific column or combination of columns. Foreign keys establish relationships, linking records based on matching keys in different tables.
- ***Trigger***: Triggers contain a set of SQL statements or procedural code that is executed when the triggering event occurs. They are used to enforce data integrity, implement business rules, perform auditing, or maintain data consistency within the database.

### ***B. Users***

Users in a database refer to individuals or entities that interact with the database system. Each user is assigned a unique identifier and has specific access rights and permissions that determine their level of interaction and control over the data. Users can perform various operations, such as querying, inserting, updating, and deleting data, based on their authorized privileges.

### ***C. Views***

A view in a database is a virtual table that is derived from one or more existing tables or views. It represents a logical subset of data from the underlying tables, presenting a tailored and

customized perspective of the data to the users. Views are created based on predefined queries and can be used to simplify complex data access, restrict data visibility, and provide a consistent and simplified interface for users.

#### ***D. Functions***

Functions in a database are named, reusable blocks of code that perform a specific task and return a value. They are used to encapsulate a set of instructions or calculations that can be invoked from SQL statements or other functions. Functions can accept parameters, process data, and return a computed result. They enhance the flexibility and modularity of database operations by enabling the reuse of code and facilitating complex data manipulation and calculations.

#### ***E. Stored Procedures***

Stored procedures are sets of precompiled SQL statements and procedural code that are stored in the database. They provide a way to encapsulate a sequence of database operations into a single unit, which can be executed and reused. Stored procedures can accept input parameters, perform data manipulations, execute complex business logic, and return results or output parameters.

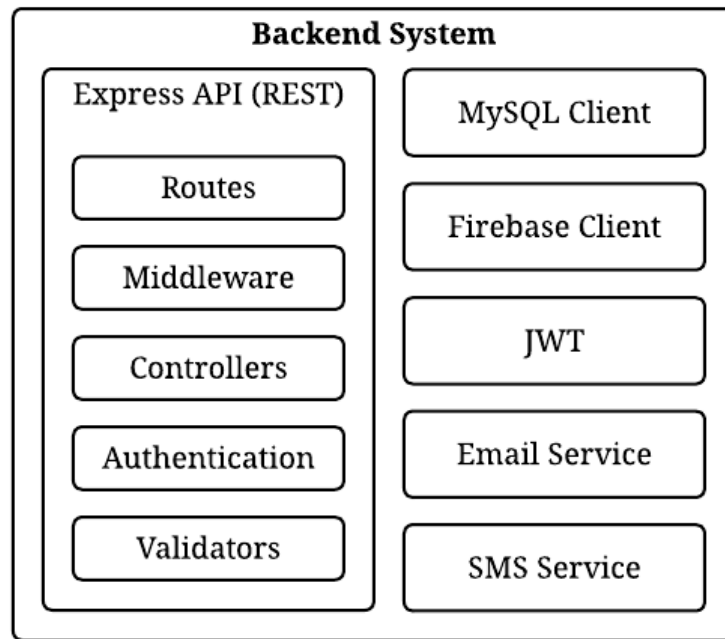
### **4.4. Backend Design**

The backend design of the Zakat Collection System will be based on a RESTful server monolith application developed using Express.js, a popular Node.js web framework. This design choice offers a scalable and efficient solution for handling API requests and managing server-side logic.

#### **4.4.1. Backend Design Components**

The backend architecture will incorporate several key components and concepts to ensure a robust and secure system. These components include routes, middleware, controllers, validators, database operations, authentication, authorization, and integration with external services.

The building blocks of our backend system are as follows:



**Figure 6** – Components of Backend Design

The figure shows both the internal and external components of the backend system. The internal components are those that are key features of the REST framework used. However, there are other services and APIs to connect the monolith with database, and user of other libraries and third-party services help fulfillment/completion of business logic.

### ***A. Routes***

The application will define various routes that map to specific API endpoints. Routes determine the URL paths for different functionalities of the system, enabling the handling of HTTP requests. On a high level, our app will have routers of base path: **/auth**, **/area**, **/donation**, **/user**.

### ***B. Middleware***

Middleware functions will be utilized to intercept and process incoming requests before they reach the corresponding route handler. Middleware can perform tasks such as request validation, authentication, authorization checks, error handling, and logging.

### ***C. Controllers***

Controllers will handle the actual business logic of the system. They receive requests from routes, interact with the necessary services and data sources, and generate appropriate

responses. Controllers help separate the concerns of handling HTTP requests from the core business logic implementation.

#### ***D. Validators***

Request validation will be implemented using validators to ensure the integrity and validity of incoming data. Validators validate request payloads, query parameters, and request headers, reducing the chances of processing invalid or malicious data.

#### ***E. Database Client***

A dedicated singleton DB class will be developed to handle database operations. This class will utilize a database client mysql, to interact with the underlying database. It will provide methods for data retrieval, storage, updates, and deletion, ensuring efficient and secure access to the data.

#### ***F. Authentication and Authorization***

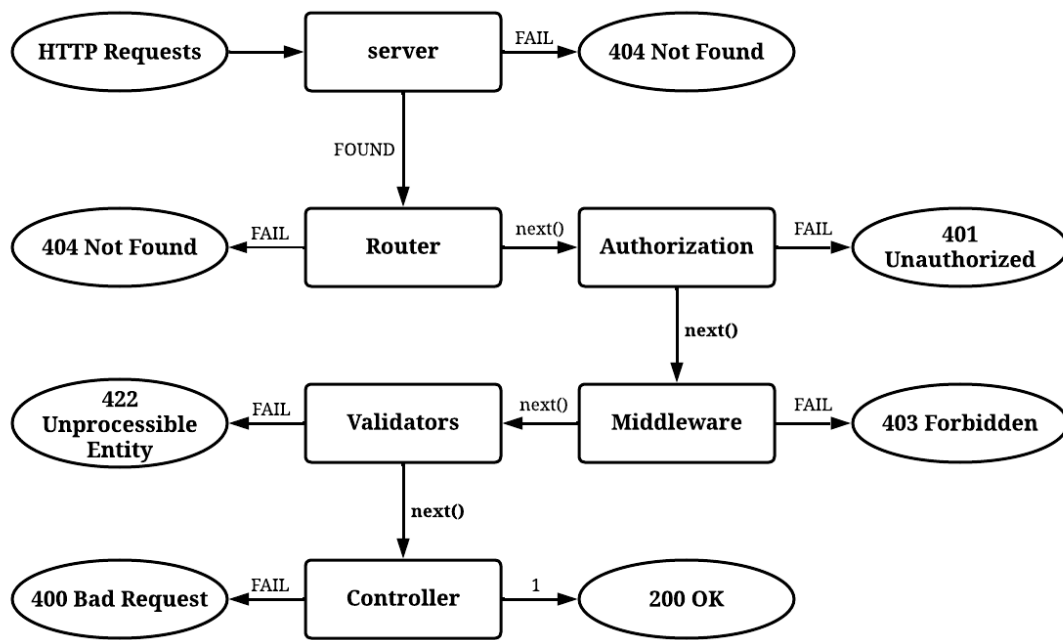
JSON Web Tokens (JWT) will be employed for authentication and authorization purposes. JWT tokens will be issued upon successful authentication and used to authenticate subsequent requests. Authorization checks will be implemented to ensure that only authorized users can access certain routes or perform specific actions.

#### ***G. External Services***

The backend will integrate with external services, such as Firebase Admin and SMS APIs, to facilitate additional functionality. Firebase Admin integration can be utilized for features like push notifications, and real-time updates. SMS APIs can enable sending notifications and alerts via SMS to relevant stakeholders.

### **4.4.2. General Scenario (Flow of Request-Response Cycle)**

In the request-response cycle of the backend for the Zakat Collection System, the following scenario has been selected:



**Figure 7** – Scenario for Request-Response Cycles

- The HTTP request is received by the app component, which acts as the entry point for the backend application.
- The app component checks if there is a router that matches the route specified in the request. If a matching route is found, the request proceeds to the next step. Otherwise, a 404 Not Found response is returned to indicate that the requested resource is not available.
- The router component processes the request and checks if there is an endpoint defined that matches the specific route of the request. If a matching endpoint is found, the request continues to the next step. Otherwise, a 404 Not Found response is returned.
- The request undergoes authentication to verify the identity and authorization of the user. If the authentication process succeeds, the request proceeds to the next step. Otherwise, a 401 unauthorized response is returned, indicating that the user is not authenticated.
- The request passes through middleware components that perform additional checks or operations based on specific criteria. These middleware components ensure that the request meets certain requirements or conditions. If the request fulfills the middleware

criteria, it proceeds to the next step. Otherwise, a 403 Forbidden response is returned, indicating that the user does not have permission to access the requested resource.

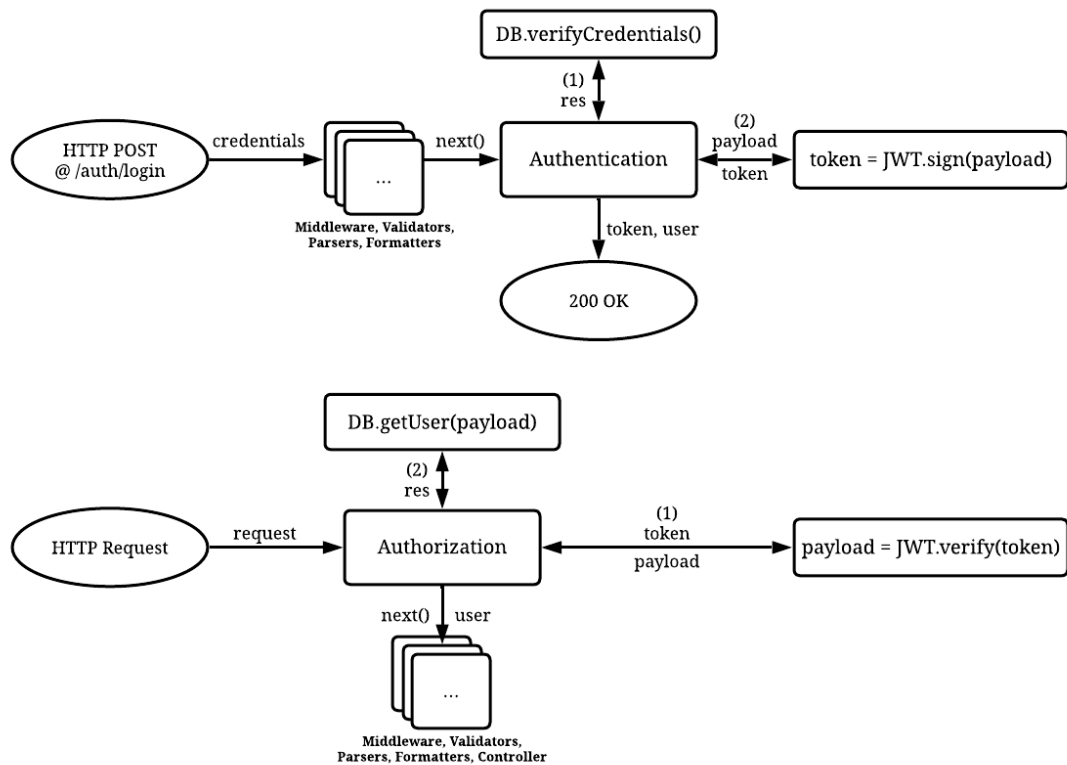
- The request is validated for integrity checks, ensuring that the provided data is valid and meets the specified requirements. If the validation process passes, the request proceeds to the next step. Otherwise, a 422 Unprocessable Entity response is returned, indicating that the request contains invalid data.
- The request reaches the controller component, which contains the business logic for processing the request. The controller performs the necessary operations based on the request's purpose and requirements. If the controller successfully processes the request, a 200 OK response is returned. However, if the controller encounters an error or failure, a 400 Bad Request response is returned, indicating that the request could not be fulfilled due to a client error.

#### **4.4.3. Authentication and Authorization**

Authentication is done to verify the identity of user through credentials. Authorization is done to provide controlled access to users. Both are a crucial part of secure systems.

For authentication, we will have an endpoint POST /auth/login, which will receive credentials from user's frontend, and the request will pass through validators to ensure the correct format of request, for security purposes. It also makes sure that the invalid requests do not put in an extra cost of making a database query where it wasn't actually needed. Supposing that the request finally reaches the login (authentication) controller. Here, the credentials are verified through a database query. If the credentials are correct, then a payload is taken from the user object and signed using JWT, to generate a token. The token is a one way string hash, which is generated using a secret key. This token and user object are sent to the user as response. The user's frontend should store the token and send in its authorization headers each time it makes a request on a protected route, using Bearer auth format.

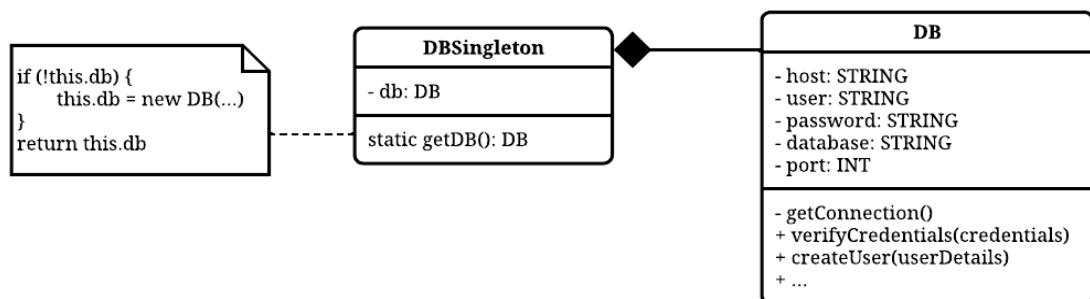
For authorization, it is assumed that the requests have a Bearer auth header, from which the token is separated and verified through JWT using same secret key as before. If the payload is returned back, it is used to get the user details from the database and the request is forwarded to upcoming functions.



**Figure 8** – Authentication and Authorization

#### 4.4.4. Database Client

A singleton class for database client will help get only one instance of database at a time. This class will be responsible for handling all the database calls, through methods. Each method will call a different query and the arguments needed for it, will be taken as method parameters.



**Figure 9** – DB Client Setup

The figure shows a simple class diagram that shows the interface of DB class that handles all the db operations in itself, and in addition to it, also uses a singleton pattern to ensure that there is a single instance of this class in the application.





## References

- [1] Mushtaq, F., & Ahmad, R. (2015). Assessment of Zakat Collection System in Pakistan: An Analysis of Problems and Prospects. *Global Journal of Management and Business Research: Finance*, 15(2), 39-48.
- [2] Hussain, M. (2019). Zakat Management System: A Case Study of Pakistan. *Journal of Islamic Accounting and Business Research*, 10(1), 28-43.
- [3] Qureshi, M. A., & Manzoor, F. (2016). Zakat Management in Pakistan: Issues and Challenges. *Journal of Islamic Thought and Civilization*, 6(1), 99-114.
- [4] Al-Quran 9:60
- [5] Bukhari, Volume 2, Book 24, Number 486
- [6] al-Qaradawi, Y. (1999). *Fiqh Al-Zakat*. Dar Al-Shorouk.
- [7] Rizvi, S. H., & Syed, T. M. (2018). Zakat Collection in Pakistan: An Analysis of Key Issues and Challenges. *Journal of Islamic Business and Management*, 8(1), 23-32.
- [8] Zaka-ur-Rab, M. (2018). Islamic Philanthropy and Development in Pakistan: A Case of Zakat Management. *Journal of Islamic Business and Management*, 8(2), 1-12.
- [9] Zafar, M. A. (2018). Zakat Collection in Pakistan: An Overview of Issues and Challenges. *Journal of Islamic Accounting and Business Research*, 9(3), 388-400.
- [10] Iqbal, M. (2017). Zakat Collection System in Pakistan: Issues and Challenges. *Global Journal of Management and Business Research: E Marketing*, 17(5), 9-16.
- [11] Alkhidmat Foundation. (n.d.). Zakat Collection System. Retrieved from <https://www.alkhidmat.com/zakat/>
- [12] Saylani Welfare Trust. (n.d.). Zakat Management System. Retrieved from <https://www.saylaniwelfare.com/zakat-management-system>
- [13] Pakistan Bait-ul-Mal. (n.d.). Zakat Distribution System. Retrieved from <http://www.pbm.gov.pk/>

- [14] Bait-us-Salam Welfare Trust. (n.d.). Zakat Management System. Retrieved from <https://www.baitussalam.org/ways-of-donations>
- [15] Malik, H. M., Nawaz, A., & Butt, T. A. (2018). E-Zakat: An Effective Way for Zakat Collection and Distribution in Pakistan. International Journal of Innovative Research in Computer and Communication Engineering, 6(8), 12534-12542.