Social delivery system for volunteers

Software Design Document

Kori Zohar 208897918 and Reut Uzan 315507160

Shenkar College
Faculty of Engineering

Date:__19/07/2023__

Content

1. **Introduction**

The purpose of this chapter is to represent the "Kudos" system, and the issue which the system provides an answer to.

1.1. System Overview

"Latet" (To Give) organization, is required to manage a wide system of deliveries and data bases that stores personal details of people in Israel.
Nowadays, "Latet" is not the direct supplier of the food baskets but depends on local associations to deliver them. The organization and local associations need to have many volunteers that shall supply the food buskets to the families in need, from different countries around Israel.
We will develop a system for managers, that shall optimize managing volunteers database. It will have an easy and accessible interface for each regional manager at "Latet". A system that shall save resources and time for both "Latet" organization and the volunteers.

1.2. Problem Description

- There is no platform that connects between volunteers to the regional managers from the organization.
- There is no accessible platform that allow managers to track volunteers applications, status and issues or requests.

1.3. Goals

- The system shall manage a connection between manager and volunteers via volunteering application requests and Inquiries interface.
- The system shall manage a data base with required personal details for delivery volunteers and manage volunteers status.

1.4. Scope

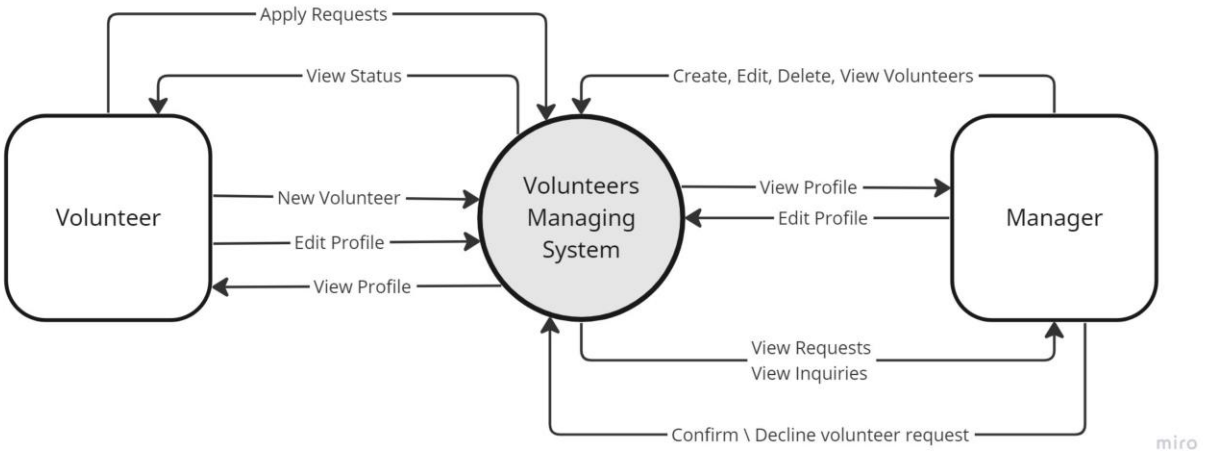
The project is about community and volunteering.

1.5. Definitions and Acronyms

1. Manager – a regional manager, assigned on behalf of the association, to manage the local basket deliveries of a specified area.
2. Volunteer – any person who is volunteering at the association at delivering food baskets.

2. System Architecture – System Context Diagram

The system context diagram, provides a “top-level” view, it focus on how external entities interact with Kudos system.



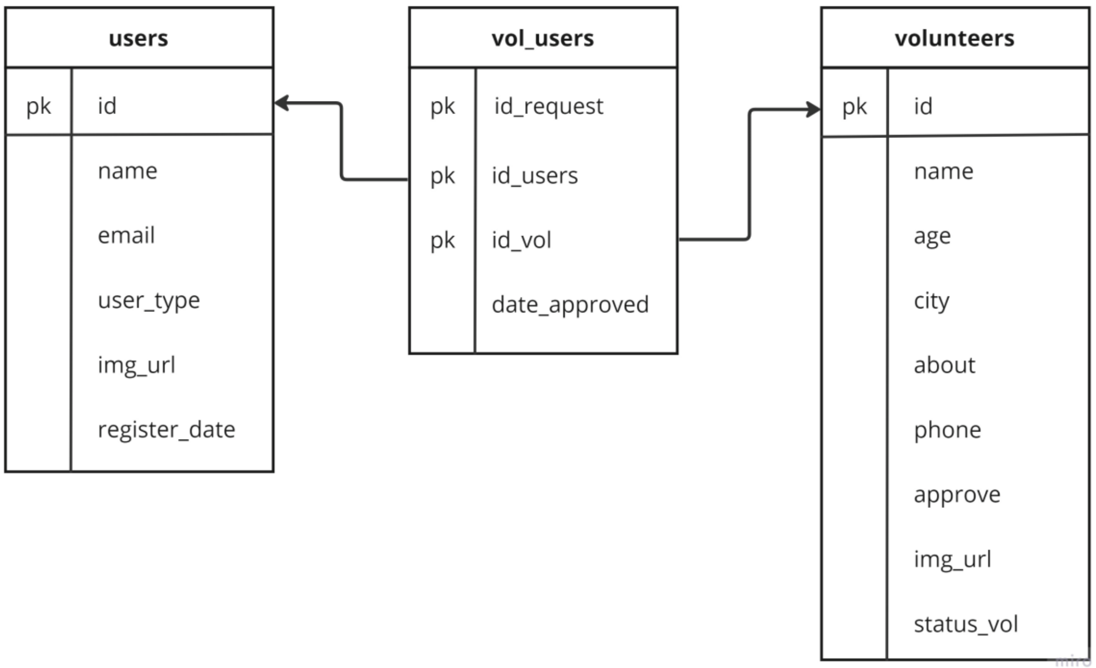
3. System Design

The following section shall focus on the main objects at the system, the main CRUDs and permissions.

3.1. Data Design - Database Description

Database Description- include 3 tables and JSON structure.

3.1.1. Database:



3.1.2.**CRUD for Volunteer main object:**

1. **Create**: Create a new Volunteer (apply request)

```
$insertQuery = "INSERT INTO tbl_208_vol (id, name , age, city, phone,
about,approve,statusVol) VALUES ('$nextId', '$name', '$age', '$city', '$phone', '$about',
'$approve',  '$statusVol')";
```

2. **Read**: Show all volunteers, show only one volunteer

```
$query  = "SELECT * FROM tbl_208_vol order by name";
$query  = "SELECT * FROM tbl_208_vol WHERE id=".$_GET["id"];
```

3. **Update**: Confirm volunteer, and update it’s status

```
if($editv == "approve"){
$query = "UPDATE tbl_208_vol SET approve = 1, statusVol ='Offline' WHERE
id=".$_GET["id"];
```

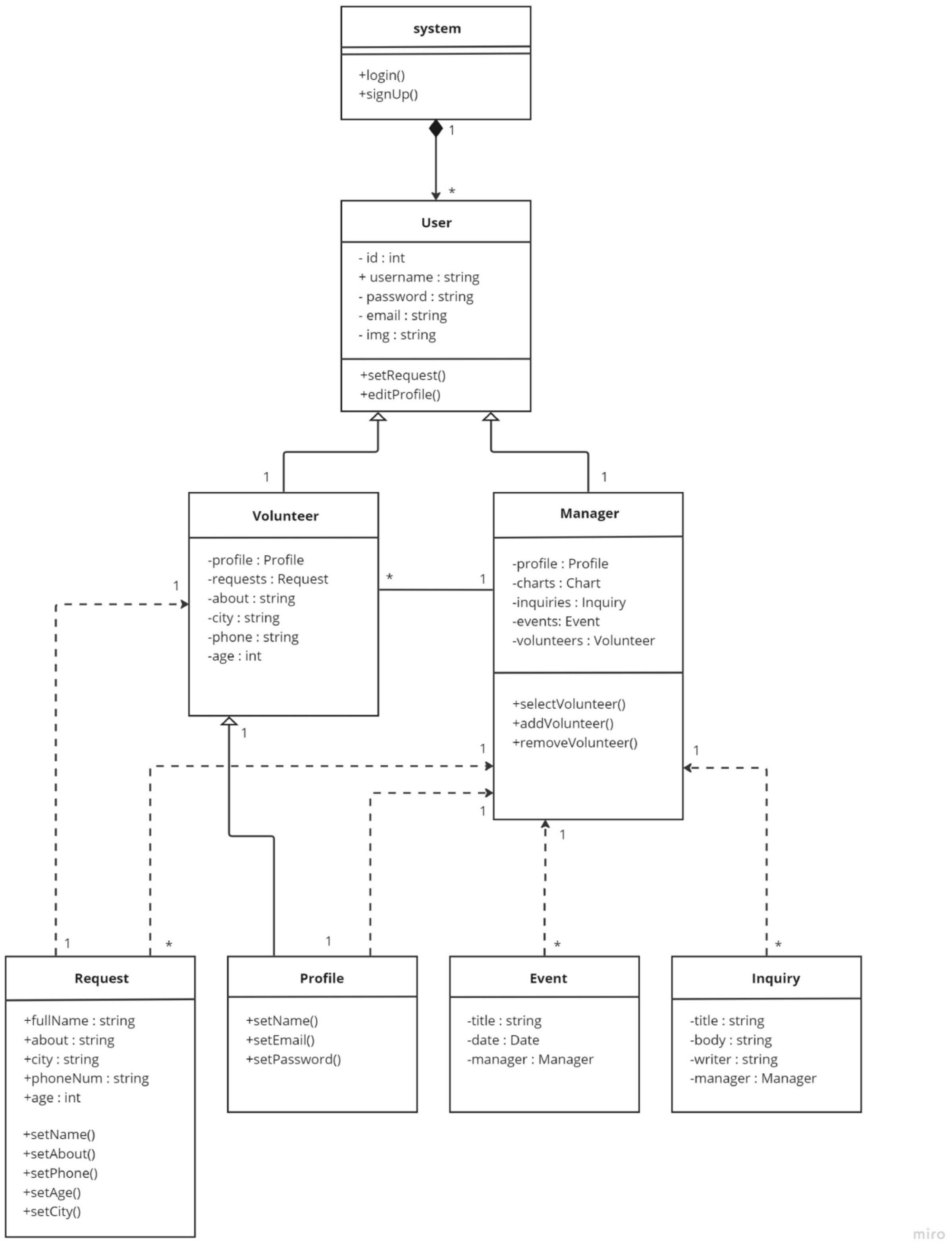
4. **Delete**: Decline volunteer, and delete it from the volunteers list

```
if($editv == "decline" || $editv == "delete"){
$query = "DELETE FROM tbl_208_vol WHERE id=".$_GET["id"];
```

3.1.3. **Json:**

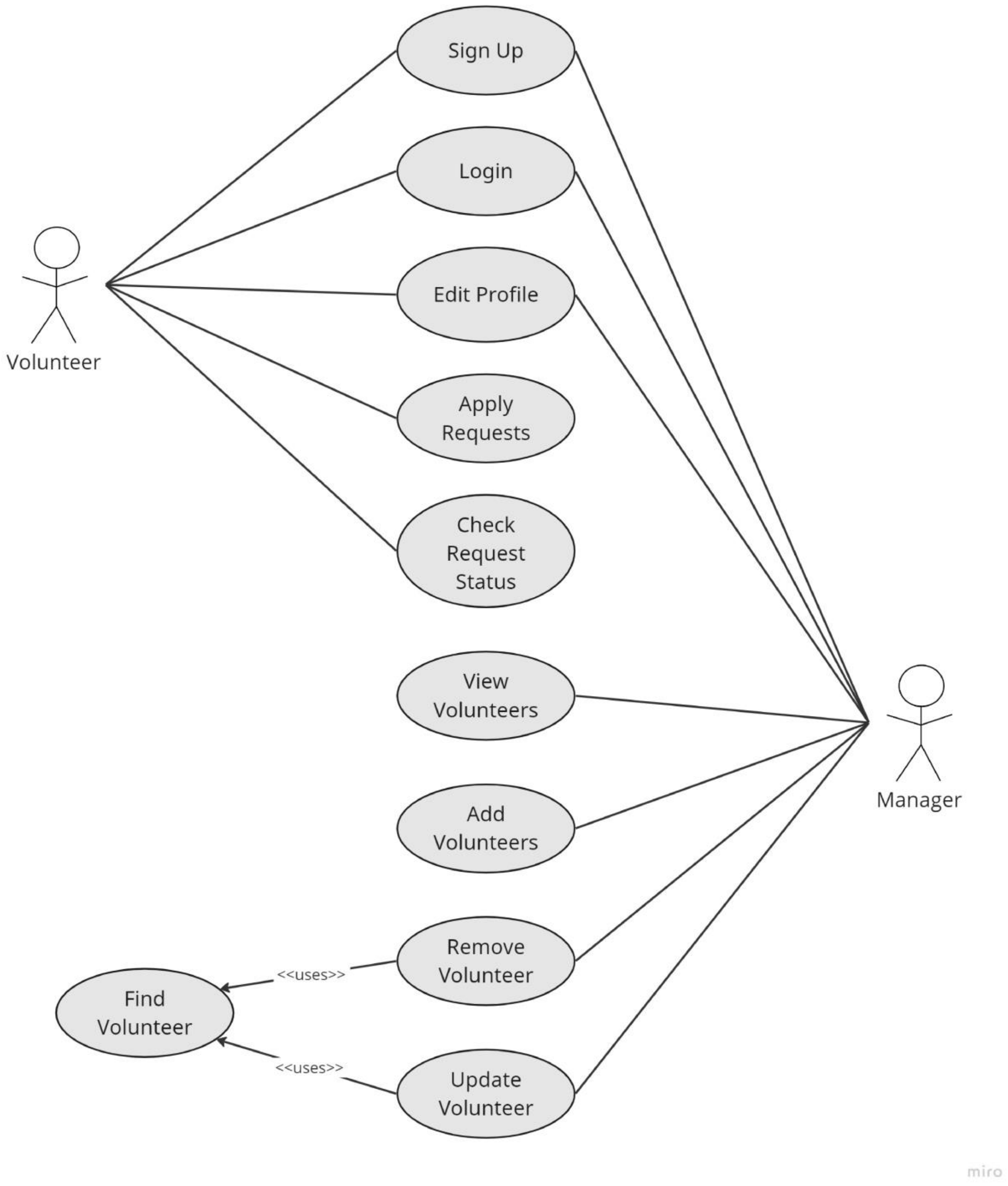
```
[
  {
    "header": "Volunteer Availability",
    "message_body": "I'll be available to assist with food basket deliveries on Saturday morning.",
    "writer_name": "John Smith"
  },
  {
    "header": "Collaboration Opportunity",
    "message_body": "Our association would like to partner with yours for a joint food drive event next
month.",
    "writer_name": "Emma Johnson"
  },
  {
    "header": "Donation Inquiry",
    "message_body": "We have excess non-perishable food items that we would like to donate to your
association.",
    "writer_name": "Sarah Thompson"
  },
  {
    "header": "Request for Assistance",
    "message_body": "We need help delivering food baskets to elderly individuals who are unable to pick them
up.",
    "writer_name": "Michael Rodriguez"
  },
  {
    "header": "Volunteer Recruitment",
    "message_body": "We are looking for enthusiastic volunteers to join our food delivery team.",
    "writer_name": "Jessica Lee"
  },
  {
    "header": "Donation Request",
    "message_body": "We are organizing a fundraising event and would appreciate any contribution towards food supplies.",
    "writer_name": "Mark Thompson"
  },
  {
    "header": "Collaboration Proposal",
    "message_body": "Our association specializes in providing hygiene products. Let's discuss how we can work together to support families in need.",
    "writer_name": "Emily Davis"
  },
  {
    "header": "Volunteer Training Session",
    "message_body": "We will be conducting a training session next week to ensure volunteers are equipped with the necessary skills for food basket deliveries.",
    "writer_name": "Alex Morgan"
  },
  {
    "header": "Request for Dietary Restrictions",
    "message_body": "Please inform us if any recipients have specific dietary restrictions or allergies to ensure we provide suitable food baskets.",
    "writer_name": "Sophia Garcia"
  },
  {
    "header": "Sponsorship Opportunity",
    "message_body": "Our organization is interested in sponsoring your association's efforts. Let's explore how we can contribute.",
    "writer_name": "Daniel Kim"
  }
 ]
```

3.2. **Structural Design** - Class Diagram

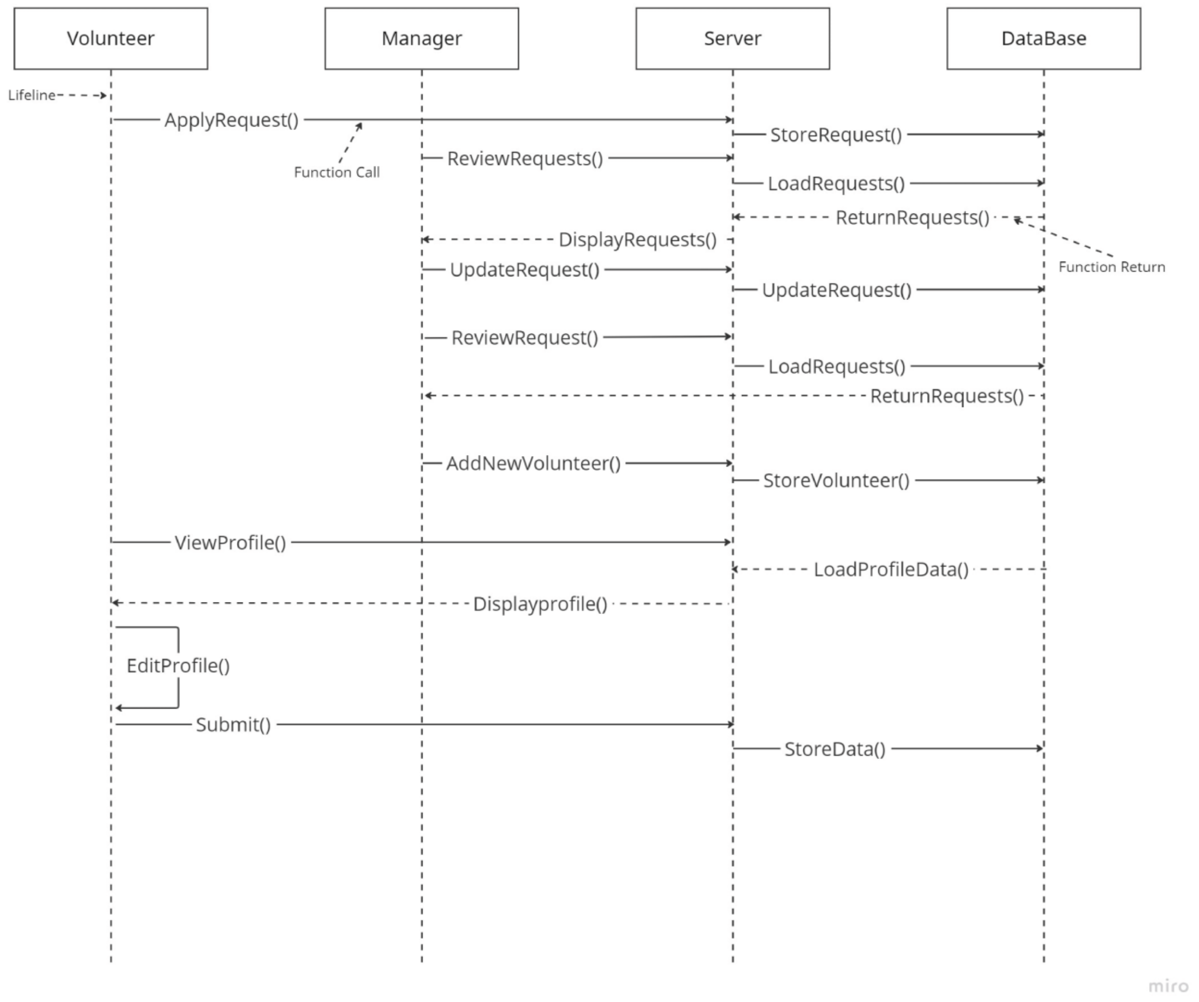


3.3. Interactions Design

3.3.1.Use Cases

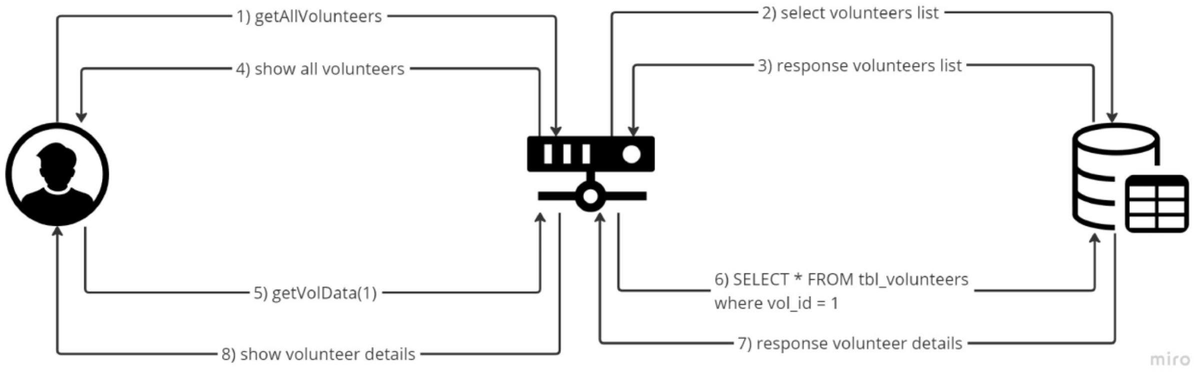


3.3.2.Sequence Diagram



3.4. **Software Architecture Pattern**

The following software architecture pattern represents the relationship between Client, Server and Database. It defines the actions being performed while Manager enters the system and can review his volunteers.


1) getAllVolunteers
4) show all volunteers
2) select volunteers list
3) response volunteers list
5) getVolData(1)
8) show volunteer details
6) SELECT * FROM tbl_volunteers where vol_id = 1
7) response volunteer details
miro

4. **Verification and Validation**

This chapter shall discuss the V&V tests that will be done at the project.

1. On Login – verify valid email and password are redirecting user to Index.php page.
2. On login – verify invalid email and password are alerting to user regarding incorrect data.
3. On Signup – verify that the system validates the inserted email and password according to the valid format.
4. After a successful login, it is possible to go from every page to every page.
5. All users have functionality based on permissions: admin or user.
6. After creating a new user, it is inserted into the database, to the correct table.
7. Test that the system generates a unique user id for each new volunteer
8. After creating a new volunteer request, volunteer's details are entered into the database.
9. Test that the system generates a unique volunteer id for each new volunteer
10. After edit user's profile, the changes are inserted properly into the correct database.
11. After edit volunteer's status, the changes are inserted properly into the correct database.
    a. If user has approved volunteer, volunteer's status is updated.
    b. If user has declined volunteer, volunteer is deleted from database.
12. Performance test, what happens when there are more then X(100 for example) volunteers.
13. Permissions, test that only the Admin, manager, have access to change other volunteers data.
14. Acceptance Testing – testing the system with real users, to verify that it meets the needs and requirements.