



Software Design Document for the School Of The First District

Mohamed Ashraf, Farida Sherif, Farah Ashraf, Basma Tarek, Nada

Nady

Supervised by: Dr. ...

June 8, 2022

1 Introduction

1.1 Purpose

This Software Design Document provides the design details of the management system made for Khatwa w Basma charity organization.

1.2 Scope

The basic architecture is client-web, the basic pages will be HTML and PHP. The admin will have control over the file system (delete, update, add). other user types will have specific permissions related to their user-type.

1.3 Overview

Sections of this document contains:

Section 2: System Overview

Section 3: System Architecture

Section 4: Data Design

Section 5: Component Design

Section 6: Human Interface Design

Section 7: Requirements Matrix

Section 9: References

1.4 Definitions and Acronyms

Here are some Acronyms and their definition

Term	Definition
observer pattern	The observer pattern is a software design pattern in which an object, named the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods.
strategy pattern	In computer programming, the strategy pattern is a behavioral software design pattern that enables selecting an algorithm at runtime. Instead of implementing a single algorithm directly, code receives run-time instructions as to which in a family of algorithms to use.
Model View Controller (MVC)	A pattern in software design commonly used to implement user interfaces, data, and controlling logic.

2 System Overview

This system uses MVC, Strategy, Decorator, and Observer design patterns. Donors has the ability to donate and get verification notifications as well as specifying how the money will be spent.

3 System Architecture

3.1 Architectural Design

Architecture is client-server using the MVC design pattern to separate

the views from the model and controllers. It helps in separating the display and the data and allow modification in each data without affecting the others. It

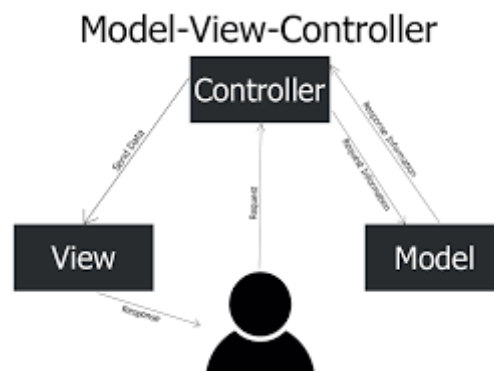


Figure 1: Architectural Design

3.2 Decomposition Description

MVC was used to separate file system read/write from the presentation layer of this project

3.3 Design Rationale

As this system was mainly a file system read/write system as well as a visualization system, we found that MVC was the most suitable architecture for this project.

4 Data Design

4.1 Data Description

The storage system of this project was a file system storage. The file class was used to make queries on the files using some query methods such as getting row by id, getting all rows. Then the row was converted from a string to an array using built-in PHP function called explode.

4.2 Data Dictionary

The data starts with the user info then donation info then donor's donation details There is also volunteer registrations and ledgers for the organization.

5 Component Design

In this section, we take a closer look at what each component does in a more systematic way.

The system is divided into several components that each serves as an essential core to the system's functionality:

Payment component

Registration

Volunteering

transactions

6 Human Interface Design

6.1 Overview of User Interface

To use the system the user must have an account. An account can be created by registering. A dashboard will be displayed dynamically for each different user type which contains some functionalities that varies per user-type. User will have a logout link in all pages to help them end their session and to be directed to the login page.

6.2 Screen Images

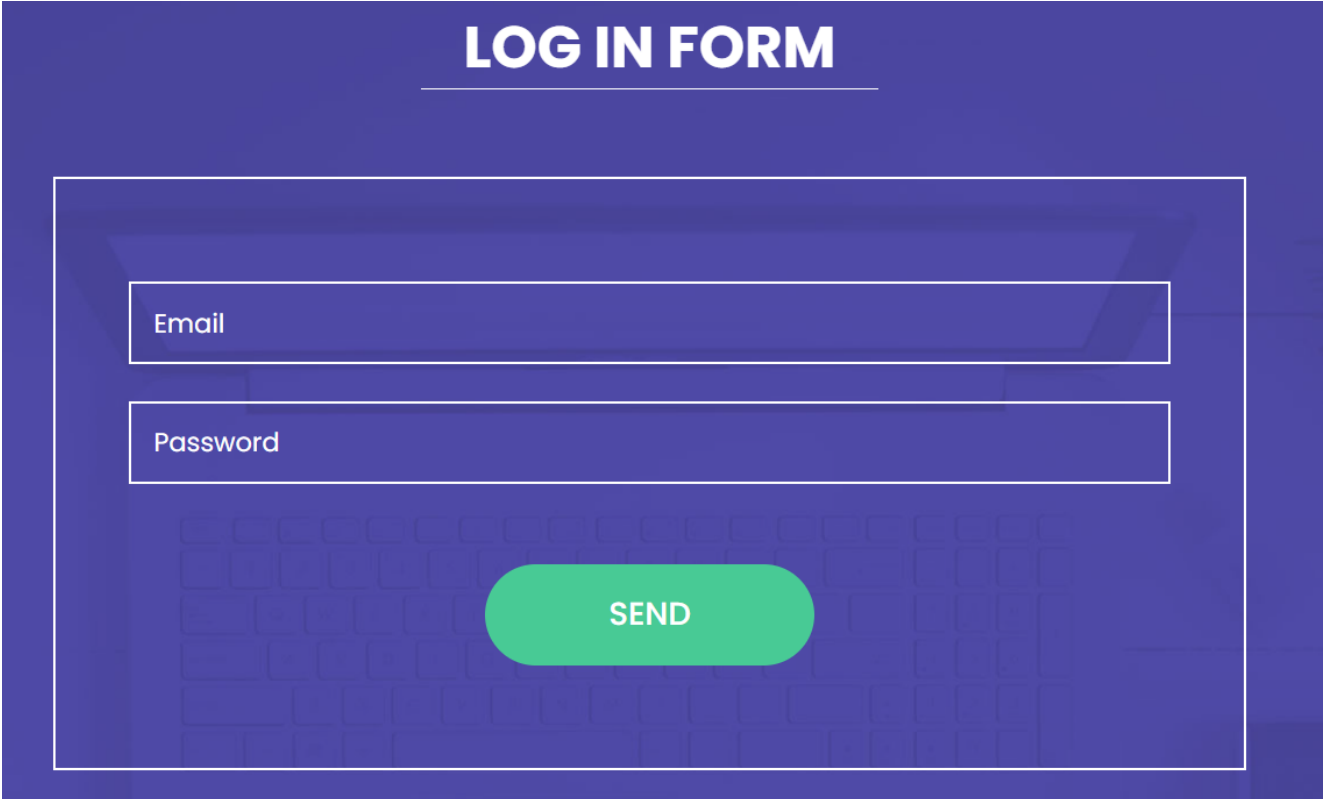
The image shows a login form on a dark blue background. At the top, the text "LOG IN FORM" is centered in white, bold, uppercase letters, with a thin white horizontal line underneath. Below this, there is a white-outlined rectangular box containing the form elements. Inside this box, there are two white-outlined rectangular input fields. The first field is labeled "Email" in white text. The second field is labeled "Password" in white text. Below these two fields, there is a green rounded rectangular button with the word "SEND" in white, bold, uppercase letters. The background of the entire page features a faint, stylized image of a laptop keyboard.

Figure 2: Login Page

USER NAME

USER AGE

USER EMAIL

USER PASSWORD

USER TYPE

**CONFIRM USER
PASSWORD**

Submit

Figure 3: Register Page

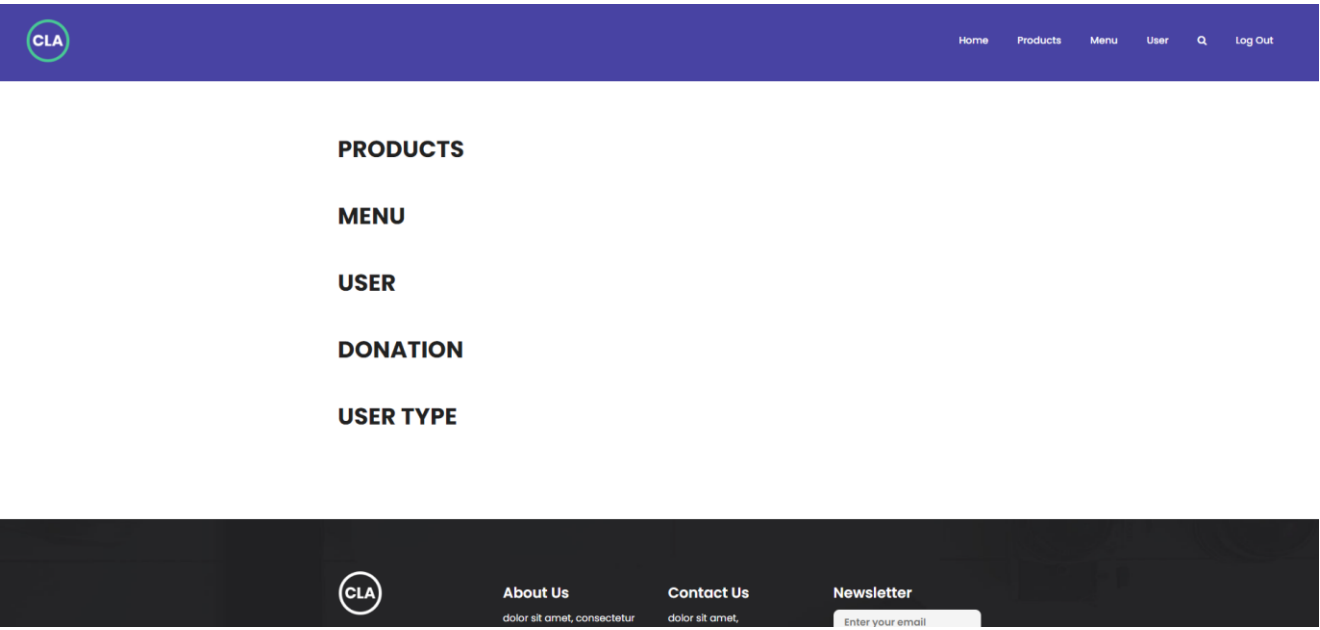


Figure 4: Admin Dashboard


DONOR ID

PAYMENT METHOD

TYPE OF DONATION

**AMOUNT OF
DONATION**

TIME OF DONATION

Submit

Figure 5: Donor Donation

6.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

7 Requirements Matrix

Provide a cross reference that traces components and data structures to the requirements in your SRS document. Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

8 References