SENG 244 - Object Oriented Software Engineering
Software Analysis Report (SAR)

NGO Aid Operations Management System : NGO-AOMSYS
AidFlow

Group Members

Yasin Yıldırım - 210204005	1
Mehmet Algül - 210204004	2
Mustafa Eymen Aslan - 210201004	3
Miraç Avcı – 220205060	4
Tahsin Cemal Sakin – 220205018	5

1) Introduction:
In response to the significant challenges faced by NGOs in effectively distributing aid to those in need, a new tool called the NGO-Aid Operations Management System (NGO-AOMSYS) is currently in development. Think of it as a digital assistant tailored for NGOs operating within national borders.

NGO-AOMSYS aims to simplify the process for donors to register and contribute, for volunteers to offer their time and skills, and for individuals seeking aid to apply for support. It's akin to bringing all stakeholders together in a virtual space to ensure assistance reaches those who need it most.

Behind the scenes, operational coordinators will utilize NGO-AOMSYS to manage aid requests, plan operations, and ensure resources are utilized efficiently. For administrators, it functions as a control center where they can oversee users and access important information to make informed decisions.

Envision NGO-AOMSYS as a helping hand, leveraging technology to enhance the efficiency and transparency of aid distribution, ultimately making a positive impact on the lives of those most in need.

2) Overview:
Non-governmental organizations (NGOs) have a hard job of delivering help to people in different places who really need it. It's super important to make sure resources, like stuff or money, get to where they need to go.

NGO-AOMSYS is the system that helps NGOs manage their operations. It's like a digital assistant made for NGOs working within a country's borders.

2.1) Main Task - 1: Bringing Donors and Volunteers Together

NGO-AOMSYS lets you link up with people who want to help you. You can join the system and give whatever you want, and choose where your donation goes. Volunteers also sign up, sharing info about themselves and when they are free. Once they are all set up, volunteers just have to wait for the admins approval before they start assisting.

2.2) Main Task - 2: Helping People Apply for Aid and Keeping Operations Smooth

NGO-AOMSYS simplifies the process for people who require assistance to apply. They fill out a form on the internet, telling about their family and what kind of help they need. While backstage, coordinators use the system to handle requests, plan operations, and make sure resources get where they need to go smoothly.

So, the system also uses some cool math stuff to decide when and how to do the aid work based on how much money and people are giving and volunteering. Coordinators can watch over everything using a dashboard, where they see numbers and make smart choices about how to help people the most.

3. Functional Requirements

User Registration:

Donors, volunteers, and indigent people must be able to register to the system.

Donors must be able to log in, select a donation type, specify the amount or quantity, and view their donation history.

Volunteers must complete a profile form with personal details, availability, transport capability, and geographic region preferences.

Indigent people must fill out an application form with details of their household situation and support needed.

Communication:

Enable communication channels between donors, volunteers, indigent individuals, and administrators for inquiries, updates, and feedback.

Notifications should be sent to relevant parties when aid requests are approved, donations are made, or schedules are updated.

Reporting and Analytics:

The system should provide reporting functionalities to track donation trends, volunteer activities, aid allocations, and impact assessment.

Generate analytics reports to evaluate the efficiency and effectiveness of aid operations.

Administrator Approval:

Administrators must be able to review and approve volunteer registrations.

Aid Management:

Operation coordinators must be able to view and schedule aid requests.

Allocation and distribution of aid materials should be managed, including collecting donations from donors and delivering aid to those in need.

Algorithm-based Scheduling:

The system should employ algorithms to analyze donation status, aid requests, and volunteer availability to propose optimal schedules for aid operations.

Publicity Events:

The system should support organizing publicity events to raise donations when resources are below a certain threshold.

Messaging functionality for sending emails and SMS messages to inform registered users and increase awareness.

4. Nonfunctional Requirements

Security:

The system must have appropriate security measures to protect user data and transactions.

Performance:

The system should be able to handle a significant number of simultaneous users effectively.

Usability:

The system interface should be user-friendly for donors, volunteers, coordinators, and administrators.

Reliability:

The system should be reliable and available for users to access when needed.

Scalability:

The system should be designed to scale and accommodate increased users, donations, and aid requests over time.

Mobile Accessibility:

Design the system to be responsive and accessible on mobile devices to cater to users who prefer to use smartphones or tablets.

Optimize the user interface for mobile screens for easy navigation and use.

Accessibility:

Ensure the system is accessible to users with disabilities by following accessibility standards and providing features like screen readers, keyboard navigation, and alternative text for images.

Consider multilingual support to cater to users from diverse linguistic backgrounds.

5. System Models

5.1 Use Case Model

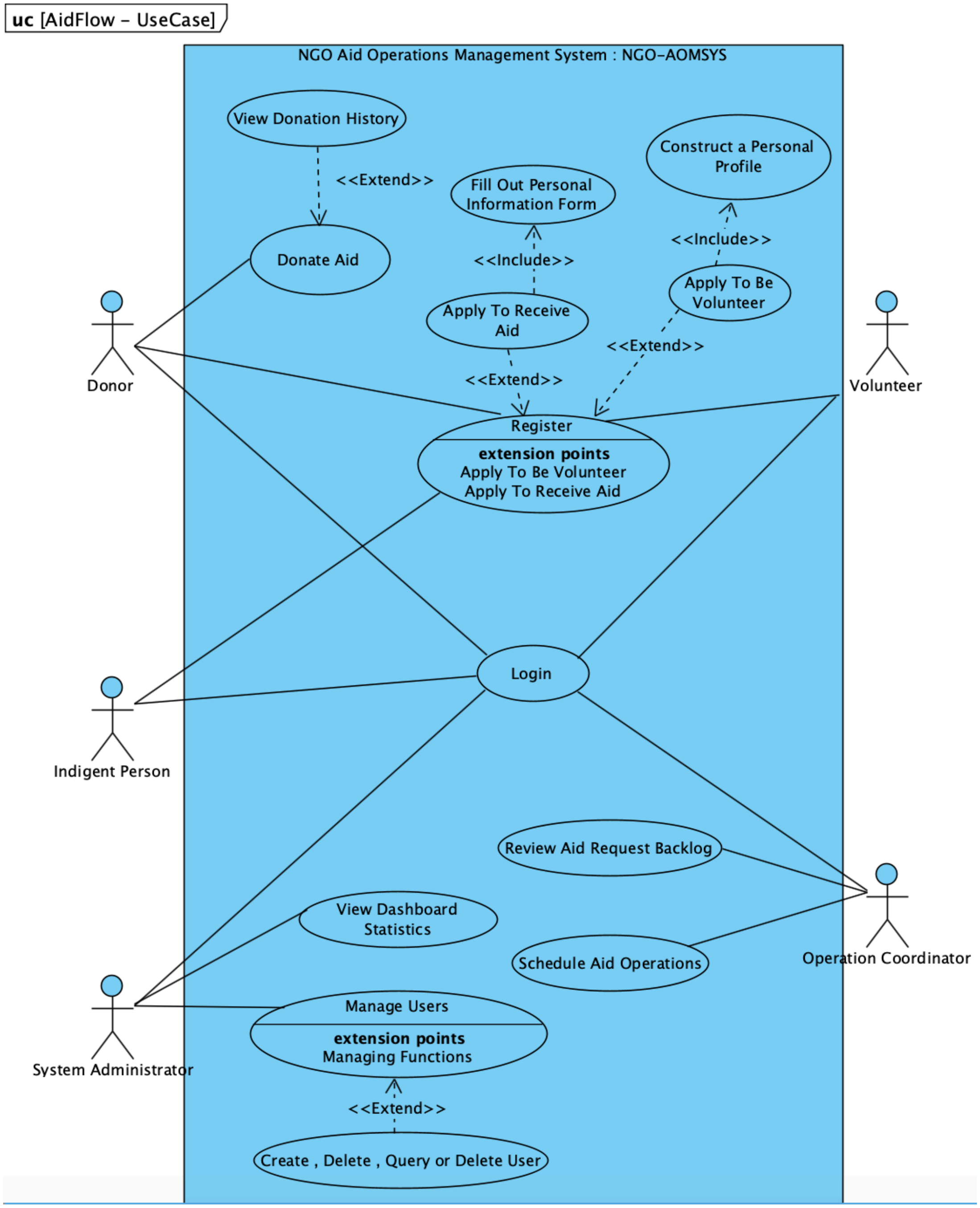


Figure : Main Use Case Diagram

5.1.1 Use Case Description Table

Table : “Request for aid” Use Case Description

Use Case Name	Request for aid
Use Case ID	1
Primary System Actor	Indigent Person
Definition	After registering, indigent person requests for aids
Precondition	The user must be logged in.
Trigger	The user clicks on the “Apply to Receive Aid” button.
Basic Path	1. User chooses the “Apply to Receive Aid” button in the main menu. 2. User fills in a form providing personal information(monthly income, number of people in the household, number of children, educational status of each child etc.) 3. User sends the form to the system.
Alternative Path	The user can fill out the form without registering, but more detailed information is requested including Id.
Exceptional Case	-
Postcondition	The user's request must be accepted and meet the conditions

5.1.2 Use Case Description Table

Table : “Approval of volunteering” Use Case Description

Use Case Name	Approval of volunteering
Use Case ID	2
Primary System Actor	Administrator
Participating Actor	Volunteer
Definition	Admin of the system confirms volunteer application after inspection
Precondition	Presence of a volunteer application
Trigger	The button named “Manage Users” is clicked to inspect applications.
Basic Path	1. Administrator chooses the user to view or edit details. 2. The administrator accesses the user's application for the purpose of inspection. 3. The administrator checks whether the applicant meets the rules and requirements 4. Once an applicant has successfully completed the application process, their application will be approved.
Exceptional Case	In case the applicant does not comply with the volunteering conditions, the application will be rejected.

5.2 Dynamic Models

Activity Diagrams

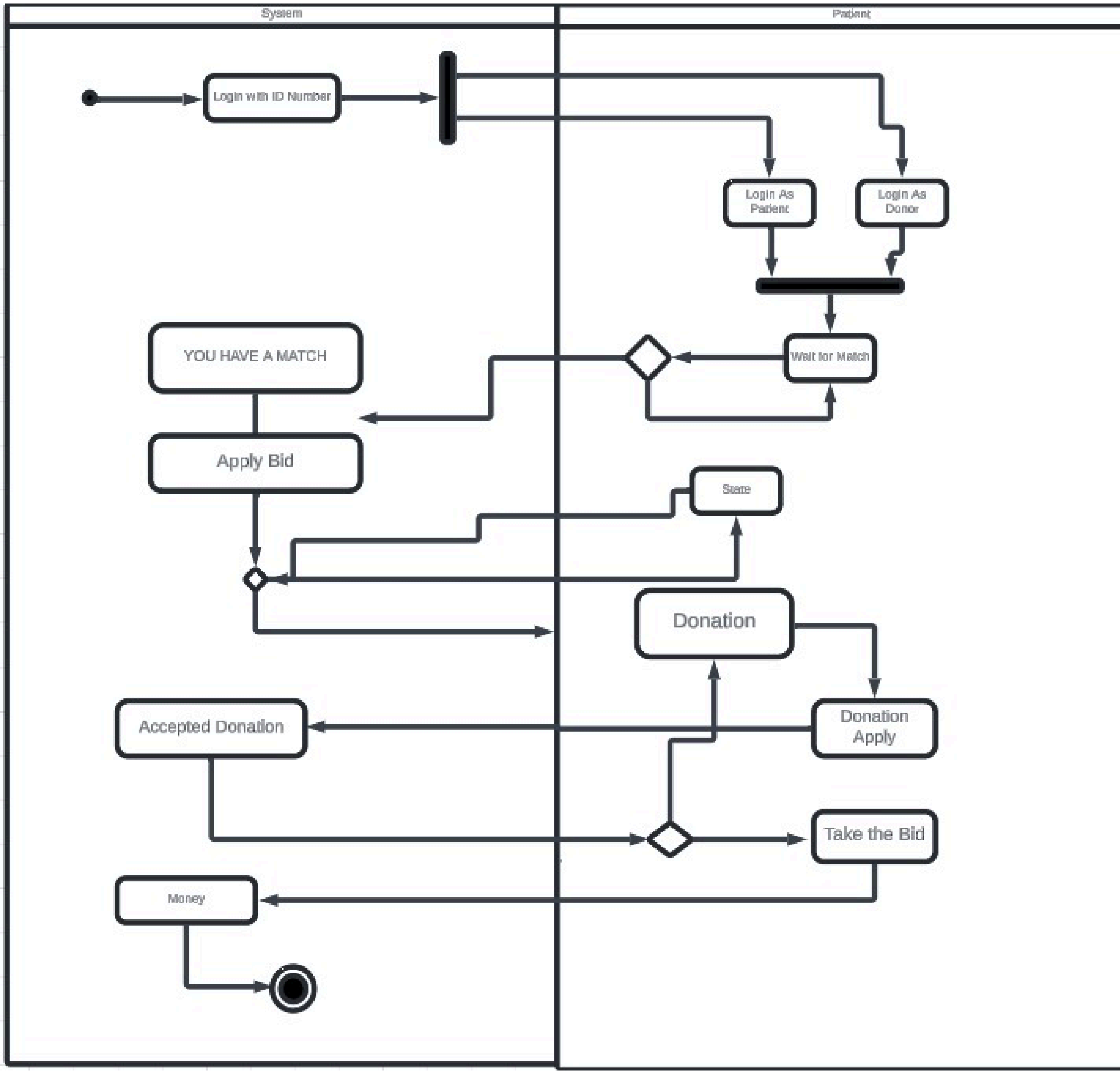


Figure : Activity Diagram

1. *State Diagrams*
    1. *The Game Object State Transitions*

Figure 3: State Diagram for the Game Object

a. Initial State

Every game element is initialized to the default values.

b. Land Selection

This is the state where the players select their lands one by one. When no land is left, advance to the next state.

c. Remaining Soldiers Placement

In this state players will place their soldiers that are left after the land selection is over. This state and the previous state serves as a pre-game to the main game. When the soldiers that are given for his phase run out advance to gameplay.

d. Gameplay

This state is where the main game of RISK is played out. Some classic rules and some newly introduced rules apply. Advance to the next state when one player conquers the whole map.

e. Declare Winner

Declares the winner then returns to the initial state so that the game can be played again if desired. Advances to final state otherwise.

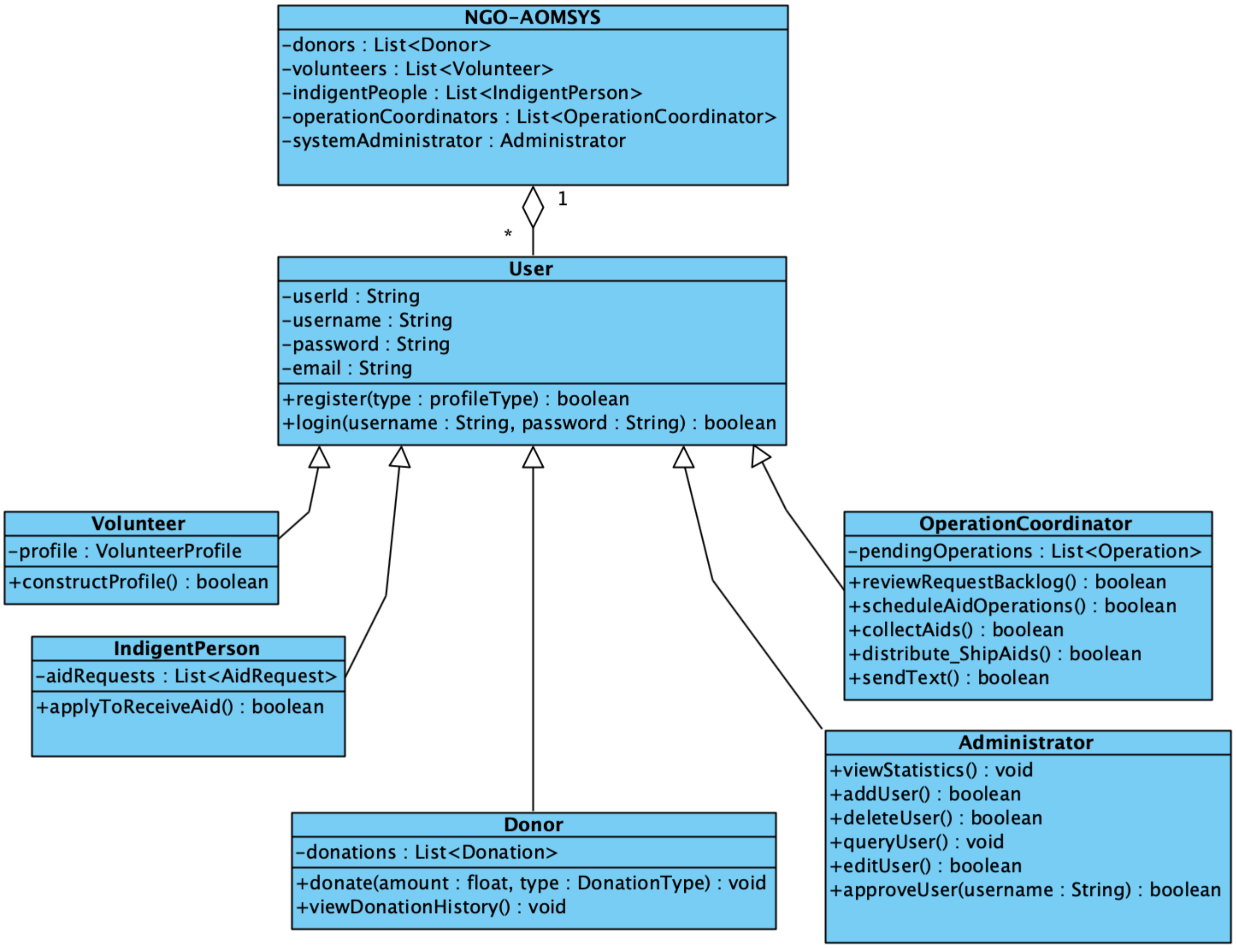
f. Final State

Game has ended and there is no desire for a restart.

*An example is provided above… Add your other **high-level** state diagrams here*

5.3 Object and Class Model

Figure : Class Diagram


NGO-AOMSYS
-donors : List<Donor>
-volunteers : List<Volunteer>
-indigentPeople : List<IndigentPerson>
-operationCoordinators : List<OperationCoordinator>
-systemAdministrator : Administrator
1
*
User
-userId : String
-username : String
-password : String
-email : String
+register(type : profileType) : boolean
+login(username : String, password : String) : boolean
Volunteer
-profile : VolunteerProfile
+constructProfile() : boolean
IndigentPerson
-aidRequests : List<AidRequest>
+applyToReceiveAid() : boolean
Donor
-donations : List<Donation>
+donate(amount : float, type : DonationType) : void
+viewDonationHistory() : void
OperationCoordinator
-pendingOperations : List<Operation>
+reviewRequestBacklog() : boolean
+scheduleAidOperations() : boolean
+collectAids() : boolean
+distribute_ShipAids() : boolean
+sendText() : boolean
Administrator
+viewStatistics() : void
+addUser() : boolean
+deleteUser() : boolean
+queryUser() : void
+editUser() : boolean
+approveUser(username : String) : boolean

5.3.1 NGO-AOMSYS Class

The NGO-AOMSYS class is the main superclass that runs the entire NGO Aid Operations Management System. It has lists of donors, volunteers, people who need help, and people who run the operations. It also has the system administrator who manages the system. This class is the main hub for all system functions and interactions. It handles user registration and management, donation processing and aid request handling, aid operation scheduling, and system-wide statistics generation. As the system's backbone, the NGO-AOMSYS class keeps aid operations running smoothly and ensures effective aid management for those in need.

5.3.2 User Class:

The User class is a general class that acts as a superclass for different types of users. It includes attributes such as ID, username, password, and email for users. Other types of user classes are inherited from this class.

5.3.3 Donor Class:

The Donor class represents individuals or entities who contribute donations to the NGO-AOMSYS system. It contains a list of donations made by the donor. The class includes a method donate(amount: float, type: DonationType) to allow donors to make donations of a specified amount and type,and a method viewDonationHistory() which individuals may observe their donation history

5.3.4 IndigentPerson Class:

The IndigentPerson class represents individuals or households in need of aid. It contains a list of aid requests made by the indigent person. The class includes a method requestAid(request: AidRequest) to allow indigent persons to request aid by providing relevant information such as monthly income, household size, etc.

5.3.5 Volunteer Class:

The Volunteer class represents individuals who offer their time and services to support aid operations. It contains a profile for each volunteer, including personal details, profession, availability, etc. The class includes methods approveVolunteer(), rejectVolunteer(), and updateProfile() to manage volunteer registrations and profiles.

5.3.6 Administrator Class:

The Administrator class represents the system administrator responsible for managing user accounts and system statistics. It includes methods manageUsers() to manage user accounts and generateStatistics() to generate summary statistics for the system.

5.4 Sequence Diagram

In this sequence diagram we see the start of the game. The actor who is the player, chooses if he/she wants to be the host or join an already existing game. After that if the actor is the host he decides the number of players and starts the game. When the game is started a series of object creations occur. After all the object creations the game is ready to start and a special game part that lets the actors to distribute their army and choose their initial lands occur that is controlled by the class Start Up. This part is not involved in the main game play as it is not in the main loop of the game and the mechanics are different. Therefore, it is a special state and is considered separately. It can be seen by the sequence diagram that there is no main Class that calls all the others to create a more simple game control. This simplifies the control of the operations and will enable simpler coding.

1. Other Sequence diagrams

An example is provided above… Add you other high-level sequence diagrams here

5.5 User Interfaces

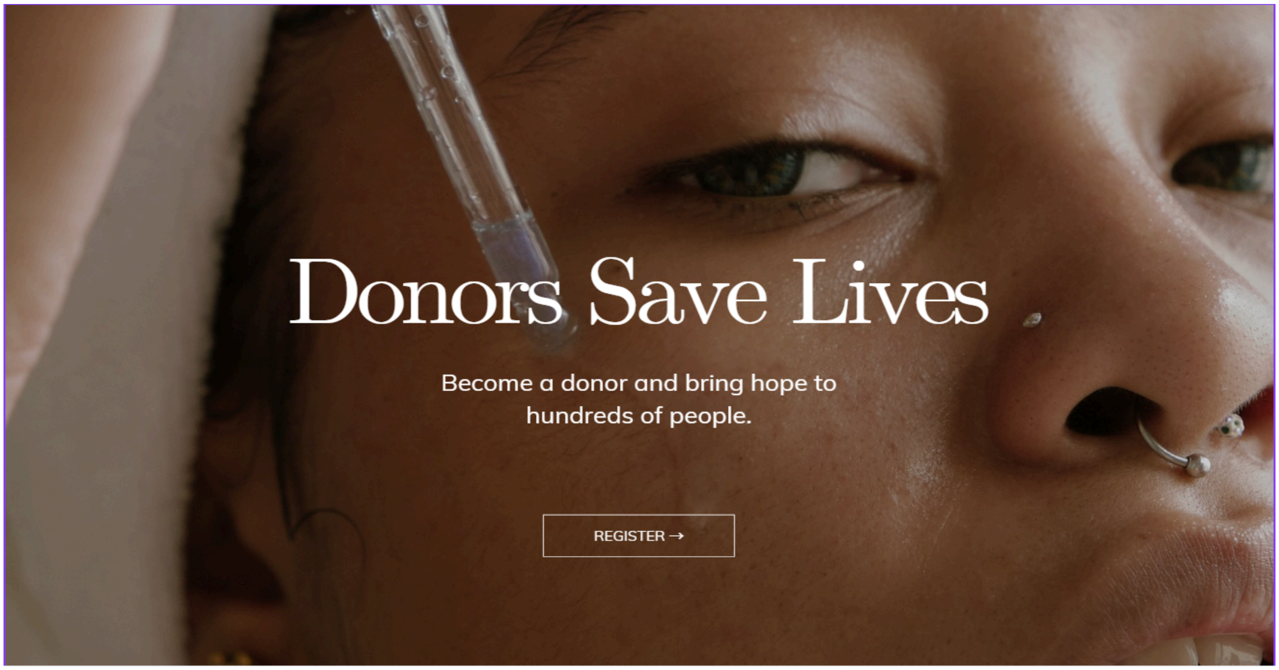
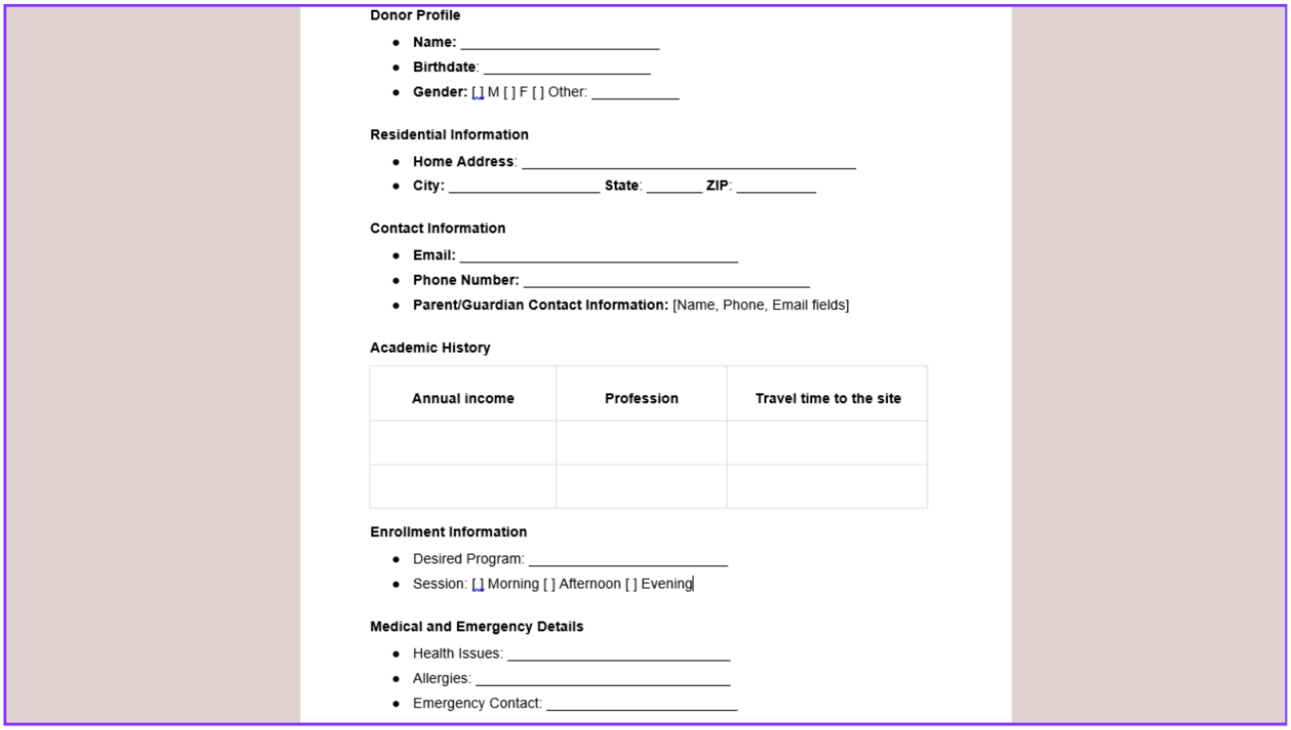


Figure : UI of the Main Menu Screen



Donor Profile

- **Name:** _________________________
- **Birthdate**: _____________________
- **Gender:** [ ] M [ ] F [ ] Other: ___________

Residential Information

- **Home Address**: ____
- **City:** ___________________ **State**: _______ **ZIP**: __________

Contact Information

- **Email:** ___________________________________
- **Phone Number:** ____________________________________
- **Parent/Guardian Contact Information:** [Name, Phone, Email fields]

Academic History

Annual income	Profession	Travel time to the site

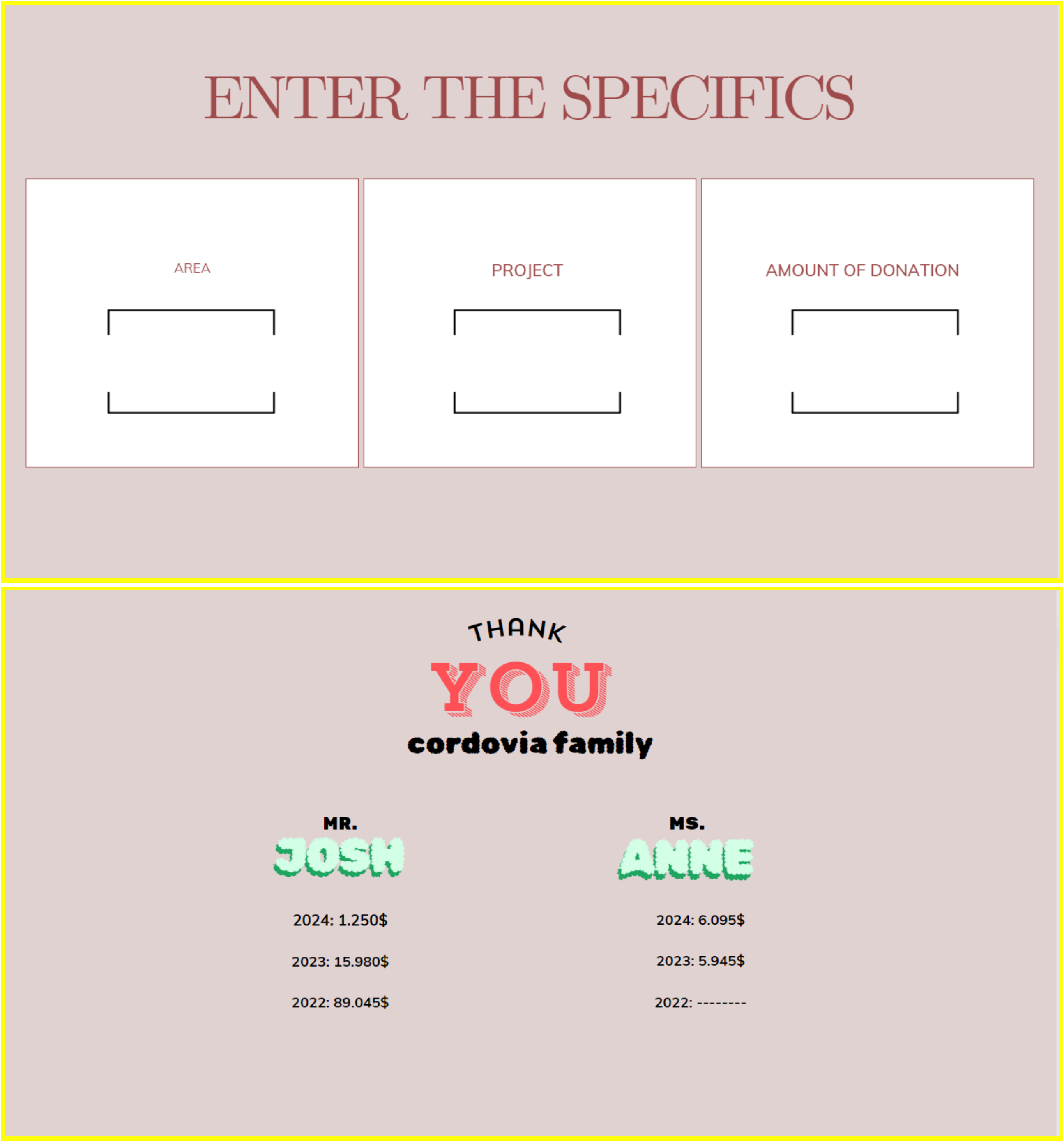
Enrollment Information

- Desired Program: _________________________
- Session: [ ] Morning [ ] Afternoon [ ] Evening

Medical and Emergency Details

- Health Issues: ____________________________
- Allergies: ________________________________
- Emergency Contact: ________________________

Figure : UI of the Filling Form Screen



ENTER THE SPECIFICS

AREA

PROJECT

AMOUNT OF DONATION

THANK
YOU
cordovia family

MR.
JOSH

2024: 1.250$

2023: 15.980$

2022: 89.045$

MS.
ANNE

2024: 6.095$

2023: 5.945$

2022: --------

Figure : UI of the Donation Screen