COMMUNITY-BASED DONATION SYSTEM

A Course Project Report in the course

Software Engineering

Computer Science & Engineering

By

18K41A0505	AREEFA
18K41A0520	G. NIKHITHA
18K41A0527	K. HARSHITH
18K41A0528	K. PRAGATHI
18K41A0529	K. RISHITHA

Under the Guidance of

Dr. D. Kothanda Raman

Assistant Professor, Department of CSE

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

S.R.ENGINEERING COLLEGE(A),ANANTHASAGAR, WARANGAL

(Affiliated to JNTUH, Accredited by NBA)

May, 2021.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the **Software Engineering - Course Project** Report entitled **"**Community-Based Donation System**"** a record of bonafide work carried out by the student Areefa, G.Nikhitha, K.Harshith, K.Pragathi, K.Rishitha bearing Roll No(s) 18K41A0505, 18K41A0520, 18K41A0527, 18K41A0528, 18K41A0529 during the academic year **2021-22** in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

Lab In-charge **Head of the Department**

INDEX

1. **WEEK-1**
   - **1.1** Requirement Identification (Problem statement, Proposed system) 1
   - **1.2** Categorization of requirements ---------- 2
   - **1.3** Requirements (FR's and NFR's) ---------- 2-3
     - **1.3.1** Functional Requirements
     - **1.3.2** Non-Functional Requirements

2. **WEEK-2**
   - **2.1** Use case Diagram ---------- 4-5

3. **WEEK-3**
   - **3.1** Class Diagram ---------- 6-7
   - **3.2** Effort Estimation ---------- 8

4. **WEEK-4**
   - **4.1** Activity Diagram ---------- 9-10

5. **WEEK-5**
   - **5.1** Statechart Diagram ---------- 11-12

6. **WEEK-6**
   - **6.1** Sequence Diagram ---------- 13-16

7. **WEEK-7**
   - **7.1** ER Diagram ---------- 17-18

8. **WEEK-8**
   - **8.1** Data Flow Diagram ---------- 19-20

9. **WEEK-9**
   - **9.1** Component Diagram ---------- 21-22
   - **9.2** Deployment Diagram ---------- 23-24

10. **Sample Template of the website.** 25-31

WEEK 1

1.1. REQUIREMENT IDENTIFICATION

PROBLEM STATEMENT:

To build a website that helps in creating a medium between communities and the people such that people can contribute their donations for the camps being held by communities.

Problems with the present system:

Many communities are not getting the properly required donations because they are not able to reach the people who are willing to donate.

Through social media, people can see the advertisements about the donation camps, but they are not ready to donate because they are not able to trust them as they can be fake.

Thus, there is a need to bridge this gap between the communities and the donators by building a system that can encourage the donators to contribute their part and communities to do the charity camps with ease.

PROPOSED SYSTEM DESCRIPTION:

Our main aim is to create a web portal for the community's involvement to display their camps that are going to be conducted by them in our portal where the users could see them and contribute their part by donating through our portal to the communities.

Objectives:

- Overcoming the existing problems in the present systems, we need to make sure that the communities can schedule the camps and send their requests to approve over the website. All the details of the camps must be correctly displayed over the website once, the admin approves.
- The donators and the communities must be able to log in.
- The donators must be able to view the camps and the communities conducting the camps.
- The donators must be able to take part in camps by donating where they must be given the options to donate.
- All the communities should be able to get the funds required for the camps and the acknowledgment of the camps must be displayed over the website.

Thus, on a whole, the proposed donation system helps in involving the chain of communities in organizing the camps and get their donations from the users over the same website. All the camps would be genuine and will be published only once they are verified and approved!

1.2. CATEGORIZATION OF REQUIREMENTS:

Based on the target audience or the subject matter, requirements can be classified into different types, as stated below:

- **User requirements:**
  They are written in natural language so that both customers can verify their requirements have been correctly identified.

- **System requirements:**
  They are written involving technical terms and specifications and are meant for development or testing terms.

1.3. REQUIREMENTS:

1.3.1. Functional Requirements:

From the brief description of our proposed system, we are going to note the requirements for the basic functionality of the system.

Register:

All the communities must be the first to get registered to schedule camps over the website. Their identity will be verified and approved by the admin. In the same way, the users can register or sign up with their Google accounts or any user identities.

During registration, username and password are stored in the database.

Login:

A login button must be displayed for the registered users to login into their profiles or website for donating. In the same way, the communities need to log in to schedule camps and enter data.

Approve camps:

The camps that are requested by the communities to schedule over the website, must be first approved by the admin. This helps to avoid fake camps.

Search Camps:

All the camps that have been approved must be sorted based on their schedule and the ongoing active camps must be displayed first to the donators to view the camps.

Status of Camps:

The status of the displayed camps must be portrayed over the website for the donators to view.

Communities involved:

All the details corresponding to the communities must be displayed on the camps list.

Upcoming Camps:

Even the upcoming camps can be approved to be displayed over the website.

Donation Process:

The users must be able to donate through different ways like Online or offline at community or pre-booking of donation over the phone.

Contact Us:

The website must also contain the details about the contact and email of the admin or the website holders for FAQ.

1.3.2. Non Functional Requirements:

Security:

The login must be secure and the donation gateway must redirect to the secure payment means.

Usability:

The communities must not face any difficulty in scheduling camps and the donators should have an easy view over browsing the camps over the website.

Availability:

It should be available 24×7.

Efficiency:

It should take less time for access.

Scalability:

The application should be upgraded by the increase in the number of camps by communities and even the donators.

Reliability:

When there is any failure it should give the notification and insights of fault.

WEEK - 2

2.1 USE CASE DIAGRAM:

Use case diagram is a platform that provides a custom understanding for the end-users, developers, and that domain experts. It is used to capture basic functionality i.e., use cases and the users of those available functionalities, i.e., actors from a given problem statement,

CASE STUDY: COMMUNITY BASED DONATION SYSTEM

The Community Based Donation System deals with creating a system that helps in coordinating between the users as per the requirements.

In this case, the users are generalized into 2 categories – Donors and the communities and of course there is Admin involved as it is a website.

	Use Case	**Description**
1	Register	Allows users to register with the website and create their accounts to login.
2	Login	Authenticates the user to access their profiles.
3	Children data upload	Communities upload data of the children that include BMI data and children's identity.
4	Organize camps	Communities organize camps by including the funds required and all the camp details.
5	Check Communities	The donors check the camps on the website. They check the profiles and might check camps.
6	Check current camps	The status of funds reached will be included and may extend to the children's data.
7	Donate	The Payment process will be depicted with the different payment modes generalized.

2.1. Use Case Diagram

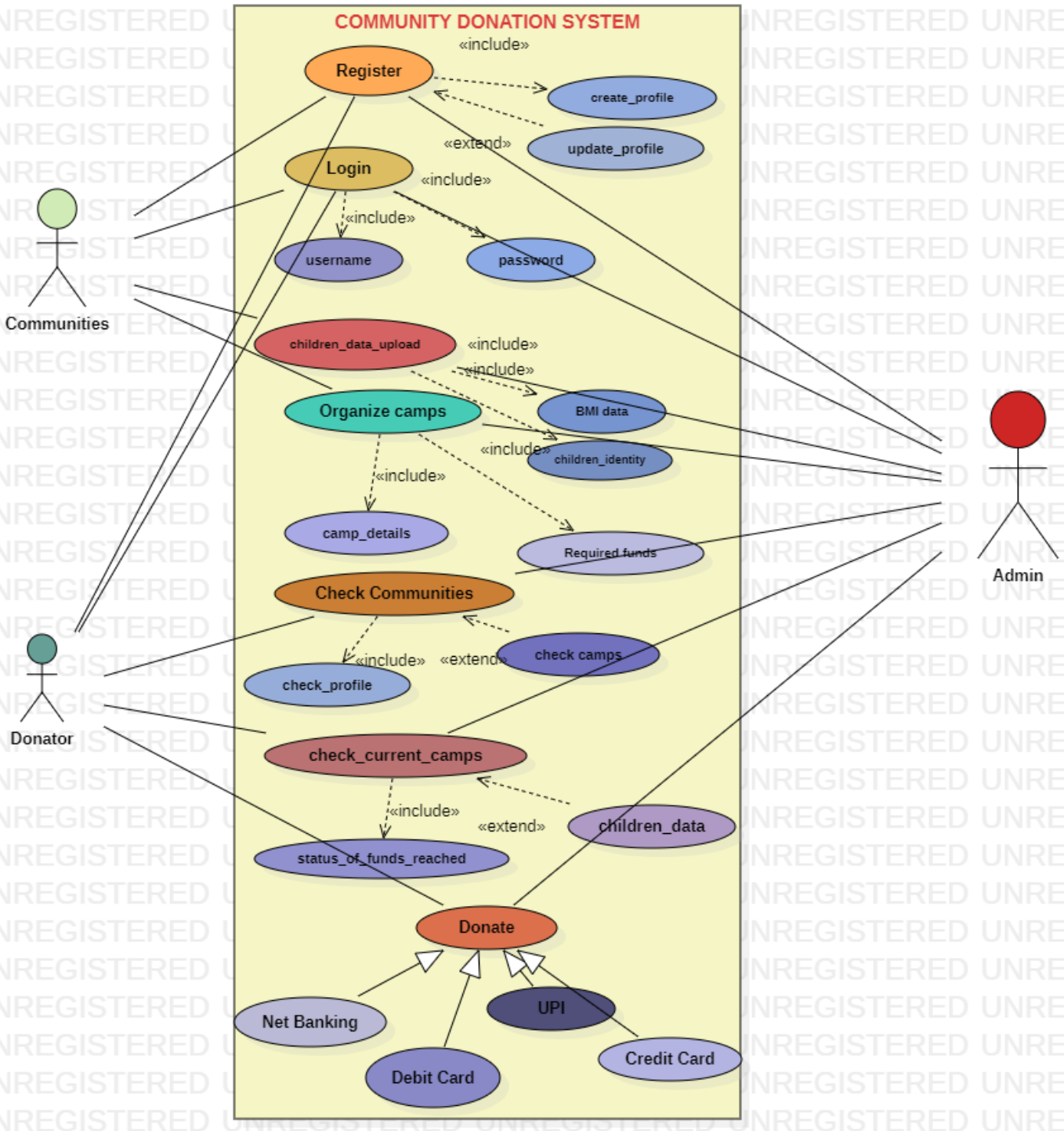


Fig 1: Use Case Diagram – Community Based Donation System

WEEK 3

3.1. CLASS DIAGRAM

In software engineering, a class diagram in the **Unified Modeling Language (UML)** is **a type of static structure diagram** that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagrams

1. Shows static structure of classifiers in a system.
2. Diagram provides a basic notation for other structure diagrams prescribed by UML.
3. Helpful for developers and other team members too.
4. Business Analysts can use class diagrams to model systems from a business perspective.

CASE STUDY: COMMUNITY BASED DONATION SYSTEM

As per the problem statement, there are different classes like: admin, user, add_new, camps, donate proceed.

Every class consists of the corresponding attributes and the operations involved.

All the attributes and the respective operations are represented as shown in the class diagram below.

- + denotes public attributes or operations.
- - denotes private attributes or operations.

The relationship between the clasees involved are:

Aggregation: This relationship exists between the CDS (our website) and the User class. If the website doesnot exist even, the users still be existing. So, this relation is depicted.

Composition: The relation where if the whole system be removed then the dependent class doesnot exist. In our case, if the website is removed, there will be no matter of existence of admin.

Generalization: An abstract class here is User and there are 2 sub classes – Community and Donator.

Dependency: The operations done by users are validated and accepted only by Admin. Thus, the User class via add_new class is depends on Admin class as shown.

Interface: The communities if they want to organize camps, they need to implement the camps_setup interface and the donator if he/she want to donate, must implement the donation interface as show.

Class Diagram

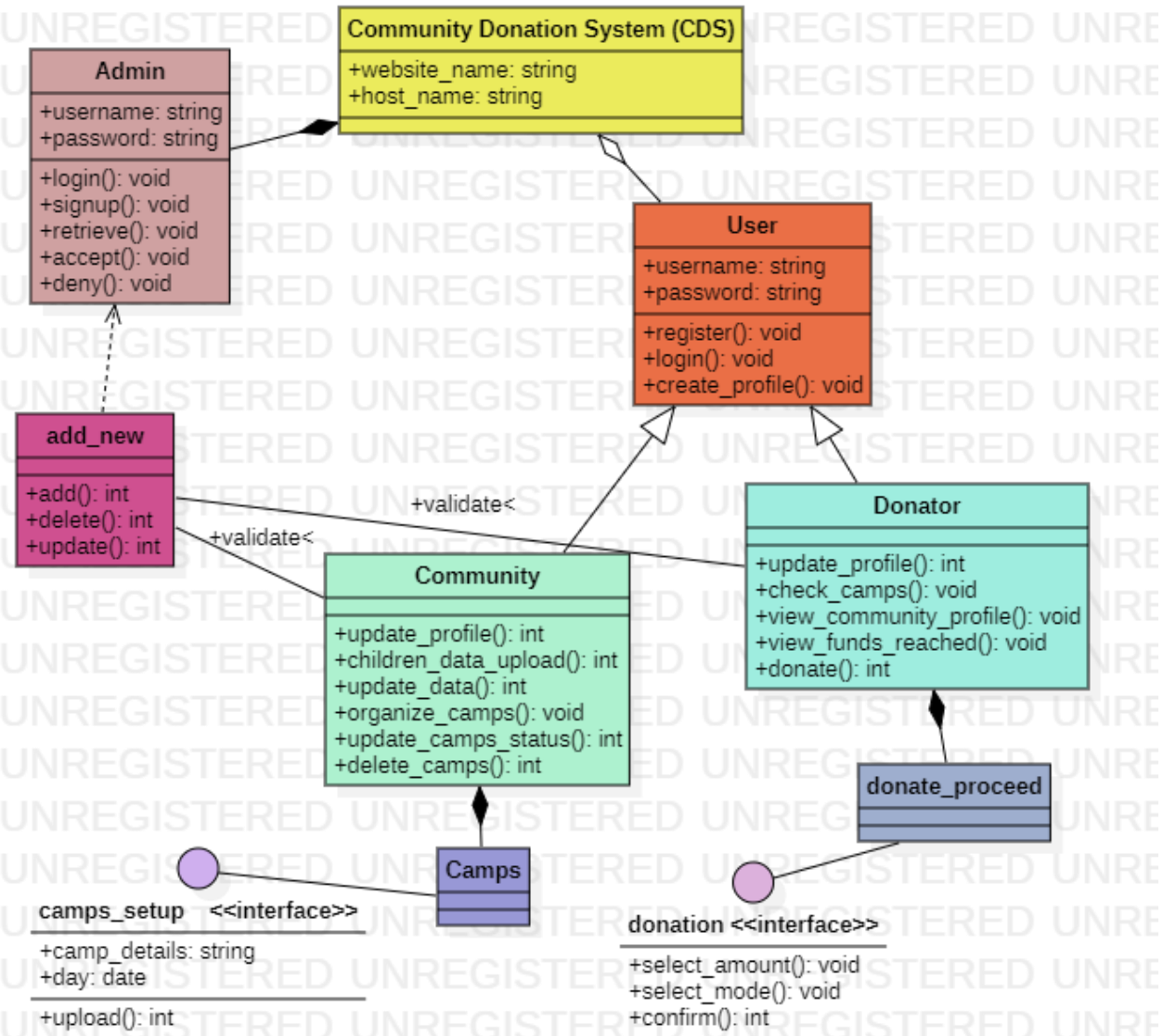


Fig 2: Class Diagram – Community Based Donation System

3.2. EFFORT ESTIMATION

Using COCOMO and based on the team size (small) and experienced (high), the concerned project could be categorized as 'organic'. The experts, based on the prior experience, suggested that the project size could roughly be around 2-50KLOC. This would serve as the basis for estimation of different project parameters using basic COCOMO, as shown below:

Effort (E) = a * $(KLOC)^{bb}$

Development time (Tdev)= $C_b(E)^d$

People required P=E/D for a, b, c, d are Coefficients Values.

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

For semi detached category of project a, b, c, d values are 2.8, 1.20, 2.5, 0.32 respectively.

bb, db - coefficients

Effort = 3.0 * $(150)^{1.12}$ PM = 821.0005 PM

Tdev = 2.5 * $(821.0005)^{0.35}$ Months = 26.17 Months

Effort Estimation Adjustment Factor (EAF)

Determined by multiplying all chosen weights. EAF lies in range of 0.9 – 1.4

Here, EAF = 1.2 (Approx)

Effort (corrected) = Effort * EAF

= 821.0005 * 1.2 = 985.2 PM

Tdev (corrected) = 2.5 * $(\text{Effort corrected})^c$

= 2.5 * $(985.2 \text{ PM})^{0.38}$ = 27 Months (Approx)

People required = E/D= 35.31 PM

WEEK 4:

4.1. ACTIVITY DIAGRAM:

Capturing the dynamic view of a system is very important for a developer to develop the logic for a system. State chart diagrams and activity diagrams are two popular UML diagram to visualize the dynamic behavior of an information system.

Activity diagram are constructed from a limited number of shape, connected with arrows. The most important shape types:

- Eclipses represent actions;
- Diamonds represent decisions;
- Bars represent the start(split) or end(join) of concurrent activities;
- A black circle represents the start(initial node) of work flow;
- An encircled black circle represents the end (final node).

Arrows run from the start towards the end and represent the order in which activities happen.

CASE STUDY – COMMUNITY BASED DONATION SYSTEM

Starting from the very top, the user starts to check the camps over the website. The communities upload the camps and the details.

If the camps are available, the user selects the camps from those displayed. If the user wants to donate, he/she selects the card. If user don't have card to pay, he/she go with the UPI payment. Now they select the amount to be donated. If their account has that balance, they don't get any error and if user confirms, the payment will be success and completed and reaches to finish. In all other cases, where the condition is no, they reach to the finish state as shown in the diagram below.

4.1. Activity Diagram

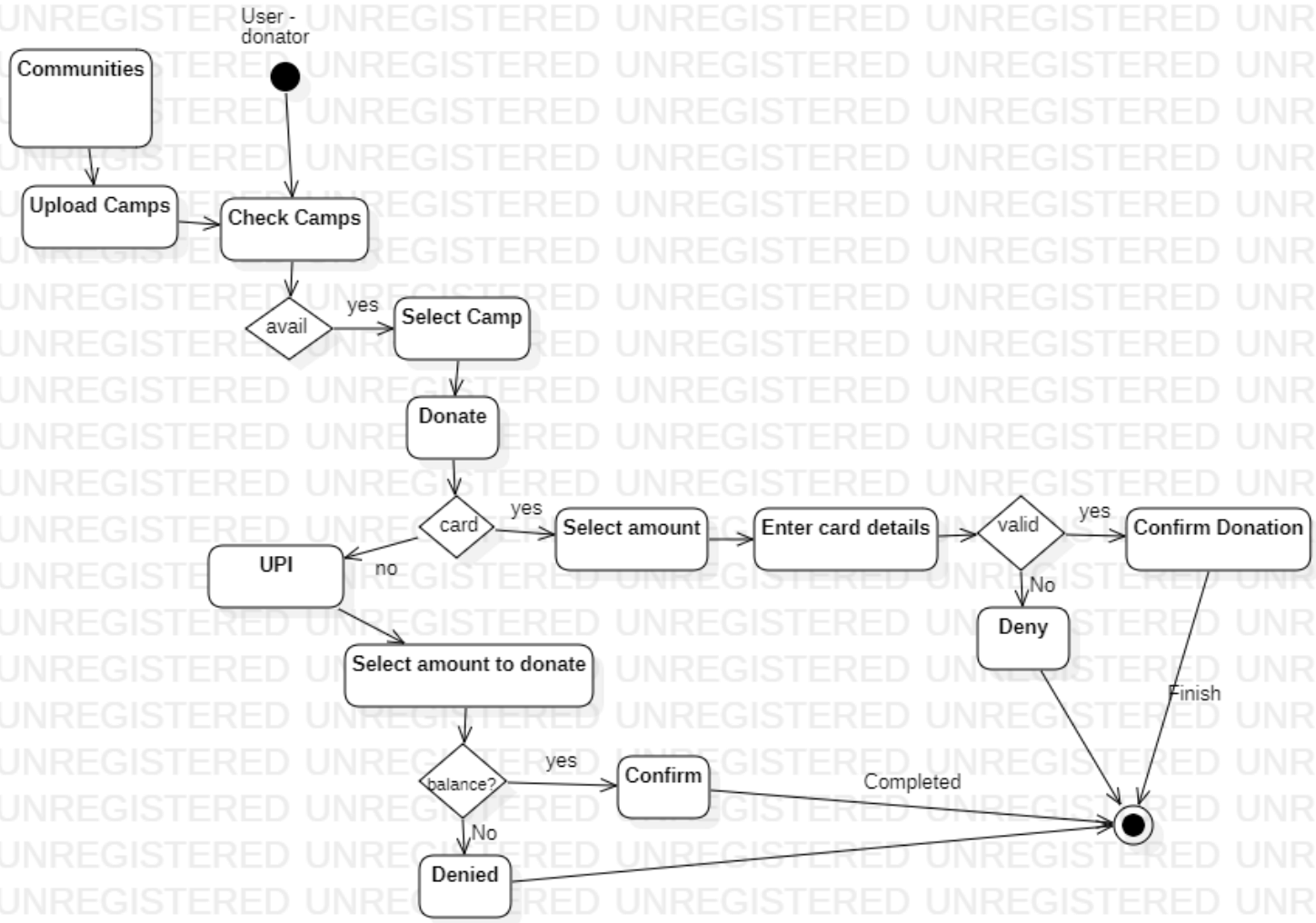


Fig 4: Activity Diagram – Community Based Donation System

WEEK 5

5.1. STATE CHART DIAGRAM

A Statechart diagram is used to describe the states of different objects in their life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement accurately.

A statechart diagram shows the behavior of classes in response to external stimuli. Specifically, a state diagram describes the behavior of a single object in response to a series of events in a system.

State chat diagrams are very important for describing the states.

States can be identified as the condition of the objects when a particular event occurs.

Before drawing a state char diagram, we should clarify the following points-

- Identify the important objects to be analyzed.
- Identify the states.
- Identify the events

.

CASE STUDY:

Starting right from the very top, the user needs to register to login.

Login is the entry-level activity as shown in the diagram where the login is validated and if it is a success only, the website profile appears otherwise it redirects back to the login page.

The website shows the details of the camp that has been uploaded by the communities as their activity. The users can select the camps and if they don't want to donate they can exit the activity.

If they want to donate, they proceed to the next activity where they do the donation that involves making payment and after successful payment, they get notified. The account balance is checked by the user. At any point, users can log out and exit as shown.

5.1. State Chart Diagram

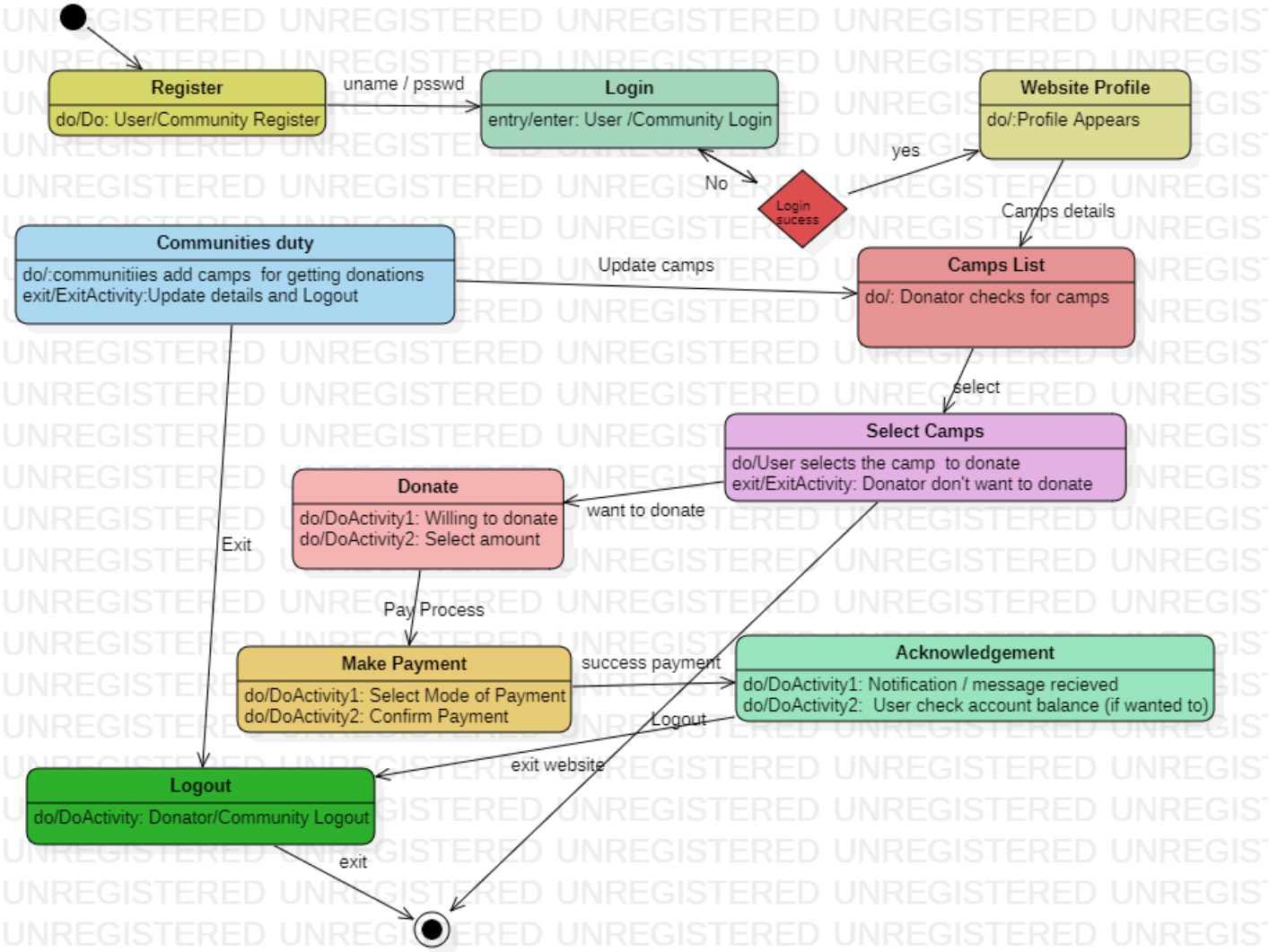


Fig 5: State Chart Diagram – Community based Donation System.

WEEK 6

6.1. SEQUENCE DIAGRAM

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Purpose of a Sequence Diagram:

1. To model high-level interaction among active objects within a system.
2. To model interaction among objects inside a collaboration realizing a use case.
3. It either models generic interactions or some certain instances of interaction.

Classes are the structural units in the object-oriented system design approach, so it is essential to know all the relationships that exist between the classes, in a system.

All objects in a system are only interacting with each other utilizing passing messages from one object to another. The sequence diagram shows these interactions with the time ordering of the messages.

Sequence diagram shows these interactions with the time ordering of the messages.

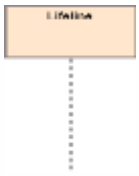
Notations of a Sequence Diagram:

Actor:

A type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data). Anyone how to use to perform a certain function in the system is actor.

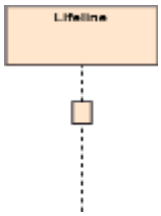
Lifeline:

A lifeline represents an individual participant in the Interaction.



Activations:

A thin rectangle on a lifeline) represents the period during which an element is performing an operation. The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively



Messages:

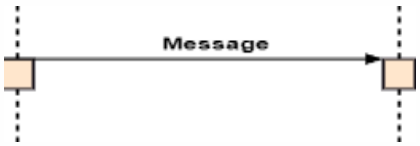
The messages depict the interaction between the objects and are represented by arrows. They are in the sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines.

Following are types of messages enlisted below:

1. **Call Message**

A message defines a particular communication between Lifelines of an Interaction.

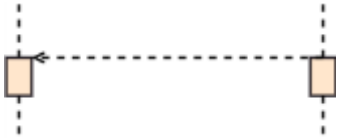
Call message is a kind of message that represents an invocation of operation of target lifeline



2. **Return Message**

A message defines a particular communication between Lifelines of an Interaction.

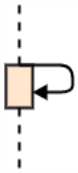
Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.



3. Self Message

A message defines a particular communication between Lifelines of an Interaction.

Self message is a kind of message that represents the invocation of message of the same lifeline.



CASE STUDY:

Donators and communities first need to register and their registration data stored from the system to the Database.

Next time when they login, their data validates and a reply message comes from the database over the home screen as a success.

If the user wants to update the profile, sends the message and a reply message appears as updated.

If communities want to upload data of camps, they send a message and a reply message follows as shown.

If the user requests the current camps, the curr3ent camps are fetched from the database and shown on the website.

The payment mode selected and payment status is sent as acknowledgment in reply message as shown once confirmed.

6.1. Sequence Diagram

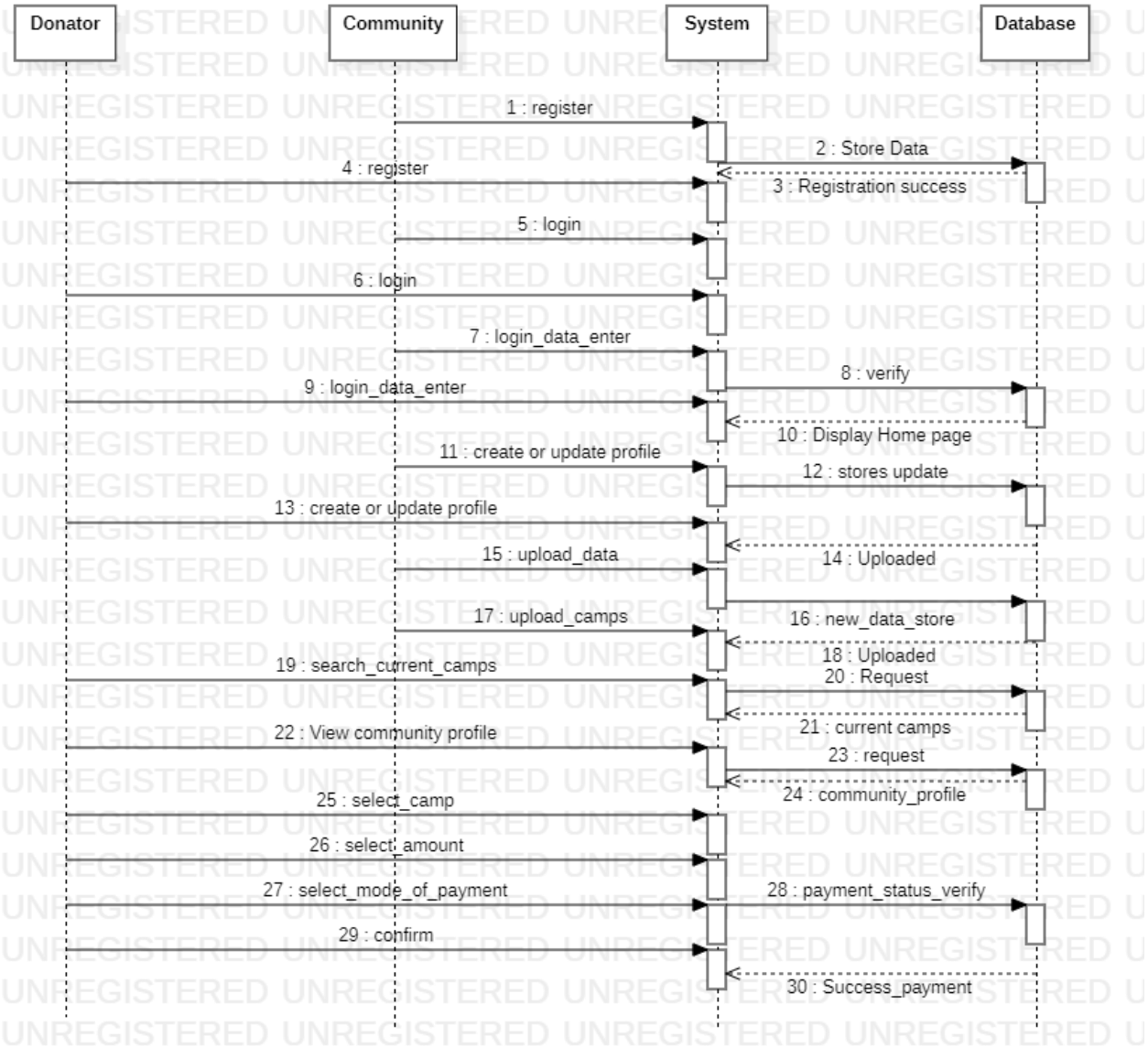


Fig 6: Sequence diagram – Community-based donation system.

WEEK 7

7.1. ER DIAGRAM

A robust database backend is essential for a high-quality information system. Database schema should be efficiently modeled, refined and normalized.

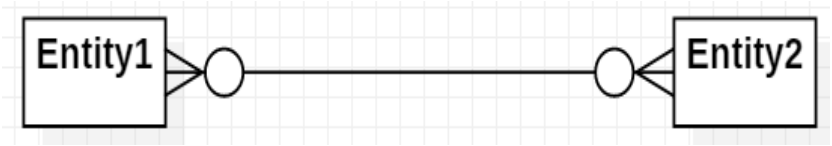
It is a systematic way of describing. ER diagram is created based on three concepts entities attributes and relationships.

- Entities are the things for which we want to store information. An entity is a person, place, thing or event. Entities are represented with rectangle shape.
- Attributes are the data we want to collect for an entitiy. Attributes are represented with the shape of elipse.
- Relationships describe the relations between the entities.

One-to-One Relationship: When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship.

One-to-Many Relationship: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship.

Many-to-Many Relationship: When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship.



CASE STUDY SCENARIO:

Admin, Community, and the donator have their login credentials. One user has only one login in the database. Therefore the relationship is One to One. The attributes and the primary keys are represented in the ER diagram below.

Admin can view multiple profiles and approve. Hence the relationship is One to Many. The users have only one profile and can update at a time only once. Hence, the relationship is One to One.

Users can check at a time multiple camps and can donate in the camps not limited to only one camp donation. Hence, One to Man. At a time, user can donate in payment portal only one, Hence, One to one relationship exists as shown. The communities can club and schedule multiple camps. Hence multiple communities scheduling or partnering with multiple camps. Hence, Many to Many relations.

Admin approves multiple camps & multiple donations. Hence the relationship that exists is One to Many.

7.1. ER diagram:

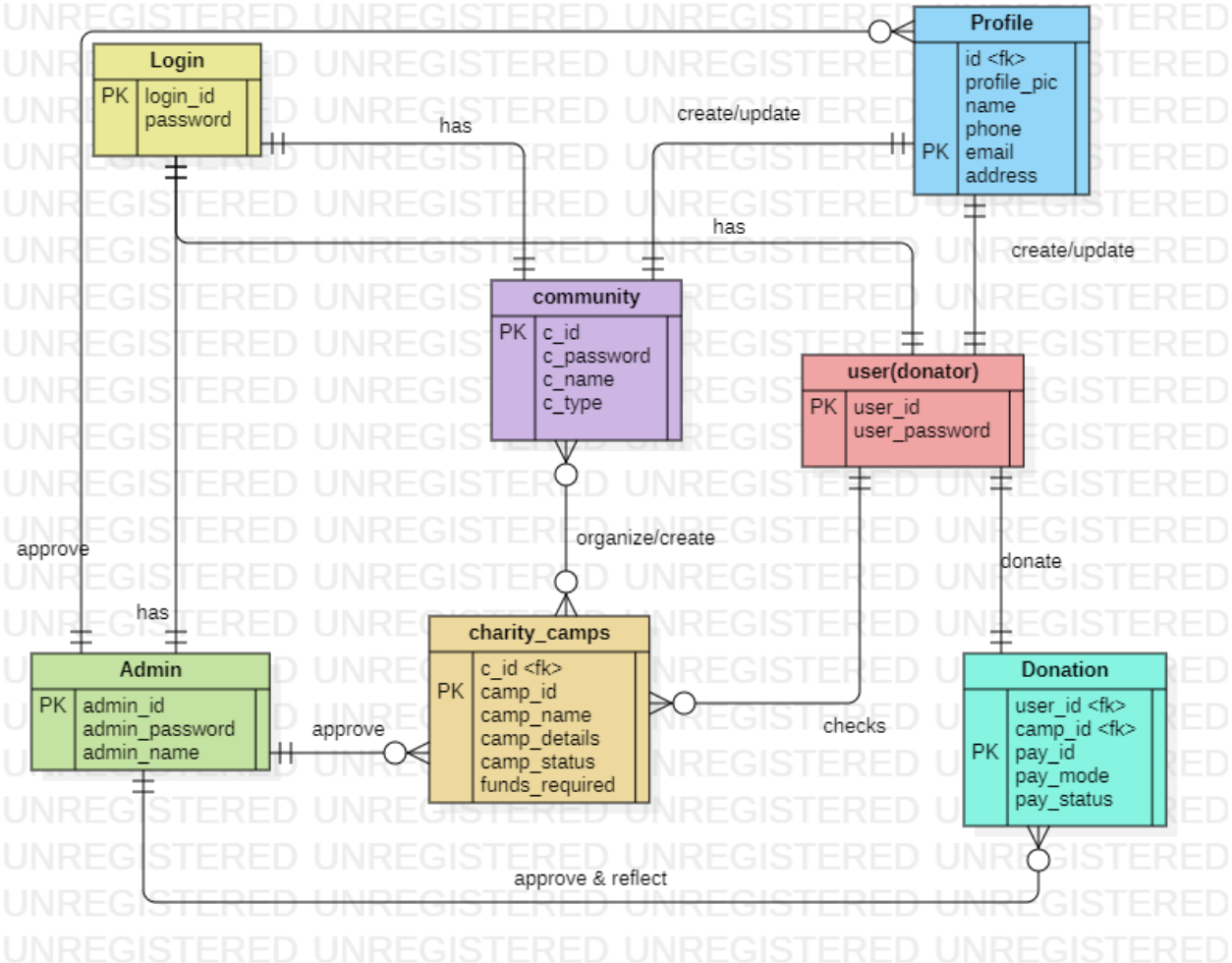


Fig 7: ER diagram – Community Donation System

WEEK 8

8.1. DFD DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

CASE STUDY SCENARIO:

The user or the communities once, registering need to login into the website. The communities must be able to create the camps. Once, approved by the admin, those camps and their details will be displayed over the website and the donators will be able to reach out to the camps by checking them over the website and donate. The donations are approved and acknowledged by the user.

All this flow is displayed in the below DFD diagrams in detailed.

DFF LEVEL 0:

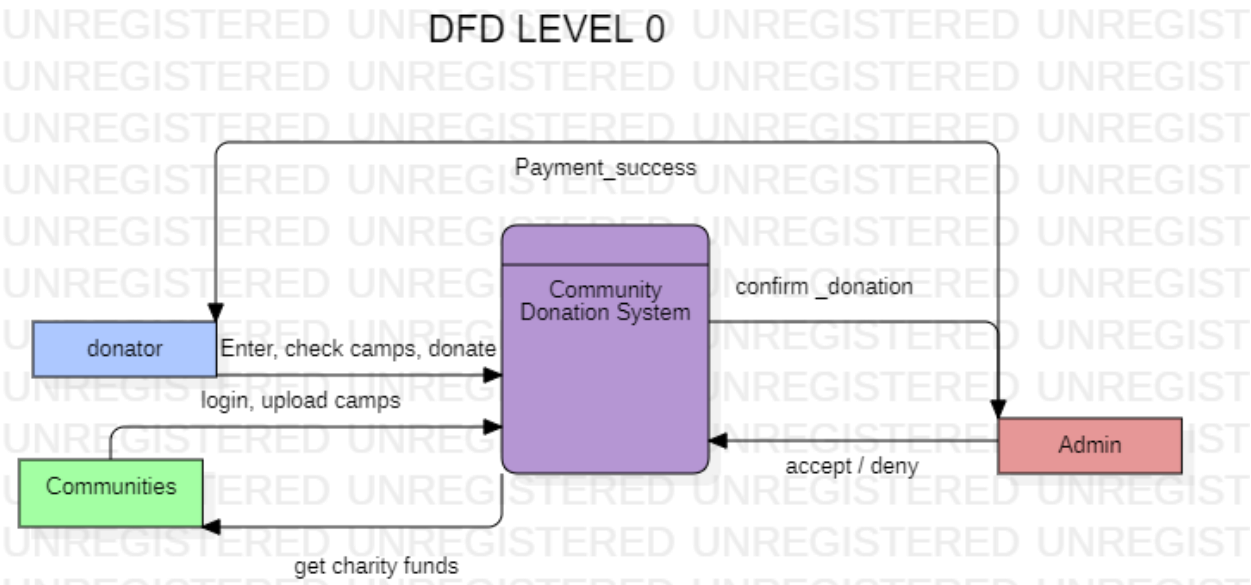


Fig 8.1: DFD Level 0

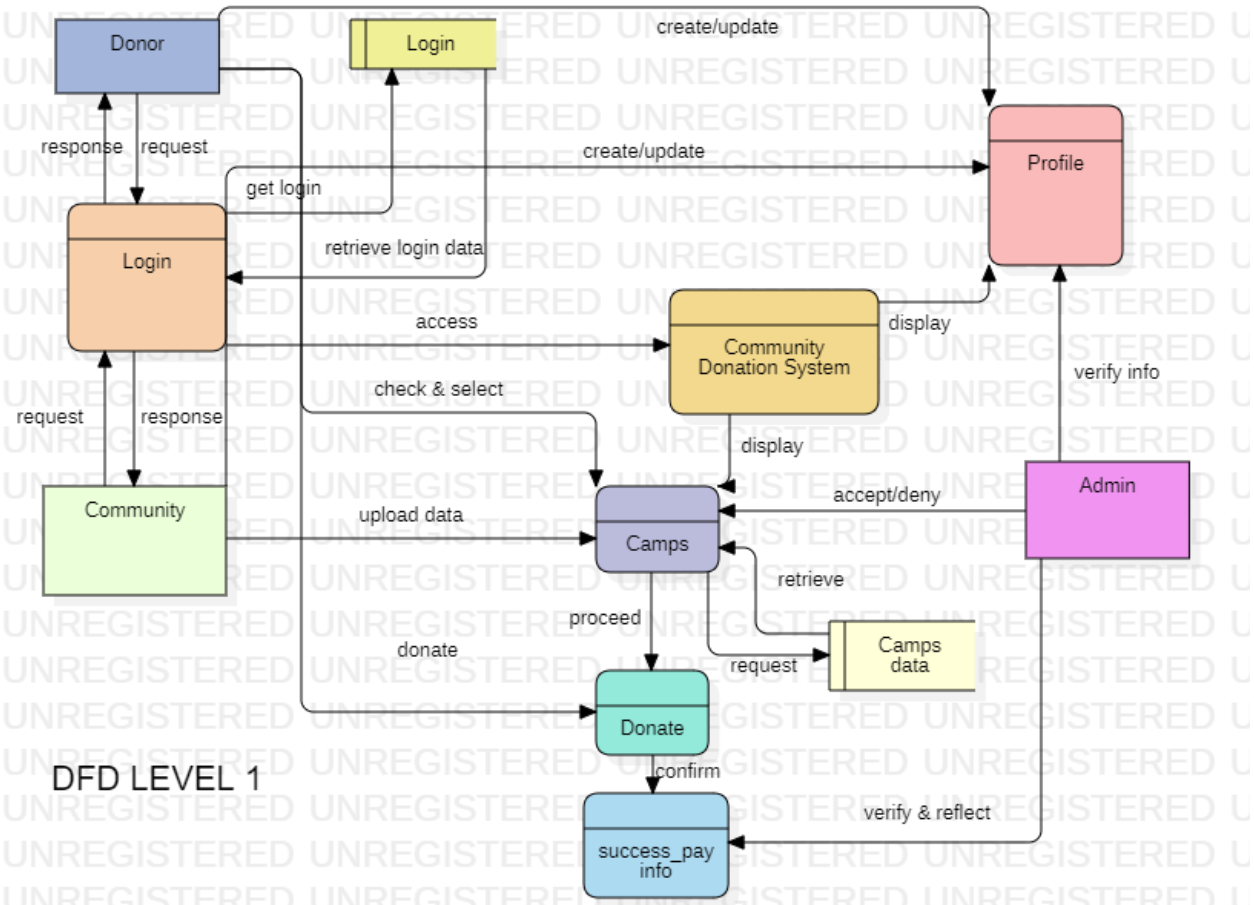


Fig 8.ii. DFD Level 2

WEEK 9

9.1. COMPONENT DIAGRAM

The component diagram is represented by figure dependency and it is the graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by a boxed figure. Dependencies are represented by communication association.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams also used to make an executable collection of diagrams are used to represent the whole.

Component diagrams can be used to:-

- Model the components of a system.
- Model the database schema.
- Model the executable applications.
- Model the system's source code

CASE STUDY – Component Diagram

According to the proposed system, the user component has subcomponents as donator and community.

The users can get the registration over the registration component and the details of credentials will be inputted through an interface implemented over to the User DB component.

The camps component details will be interfaced with the database component which has Camps DB. The data of camps approved will be inputted over the Camps DB.

The artifacts corresponding to the Database components give details about the attributes and the operations involved.

The donation process depends on the donation component where the user is redirected to the new portal of the Payment system component as shown.

The whole Community donation system is dependent on all these components.

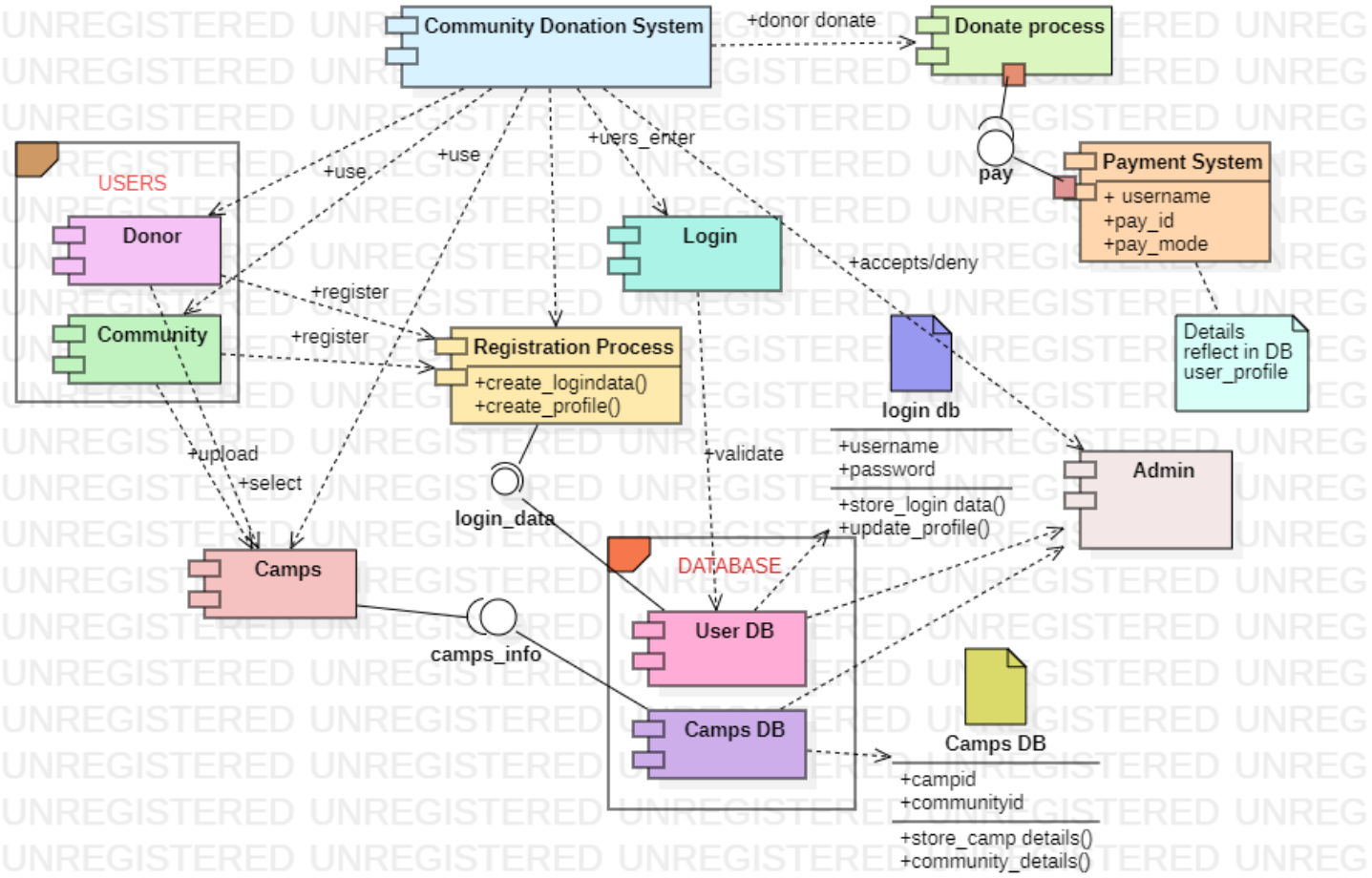


Fig 9.1. Component Diagram

9.2. DEPLOYMENT DIAGRAM

Deployment diagram issues the configuration of run time processing modes and the components that live on them.

Deployment decrease kind of structure diagram used to modeling the physical accepts of an object-oriented system.

Deployment diagrams can be used:

- To model the hardware to the topology of a system.
- To model the embedded system.
- To model the hardware details for a client/server system.
- To model the hardware details of a distributed applications for forward and reverse engineering.

Deployment Diagram – Case study

All the components specified in the Component diagram are now embedded into the nodes and the deployment relationships are shown in this diagram.

The whole model of the embedded system consists of:

Donor Device:

The node of the device by which the donor component accesses the website.

Community Device:

The node of the device by which the communities access the website.

Web Browser: The platform on which the website is accessed over the devices.

User Interface: By which the users can communicate with the website.

The other nodes include – **Register/Login system.**

Payment node: The Payment system is embedded as a separate node.

The **Database Servers** with the embedded components are displayed in the diagram below.

The **Admin Pane node** consists of an admin component with all the functionalities to handle.

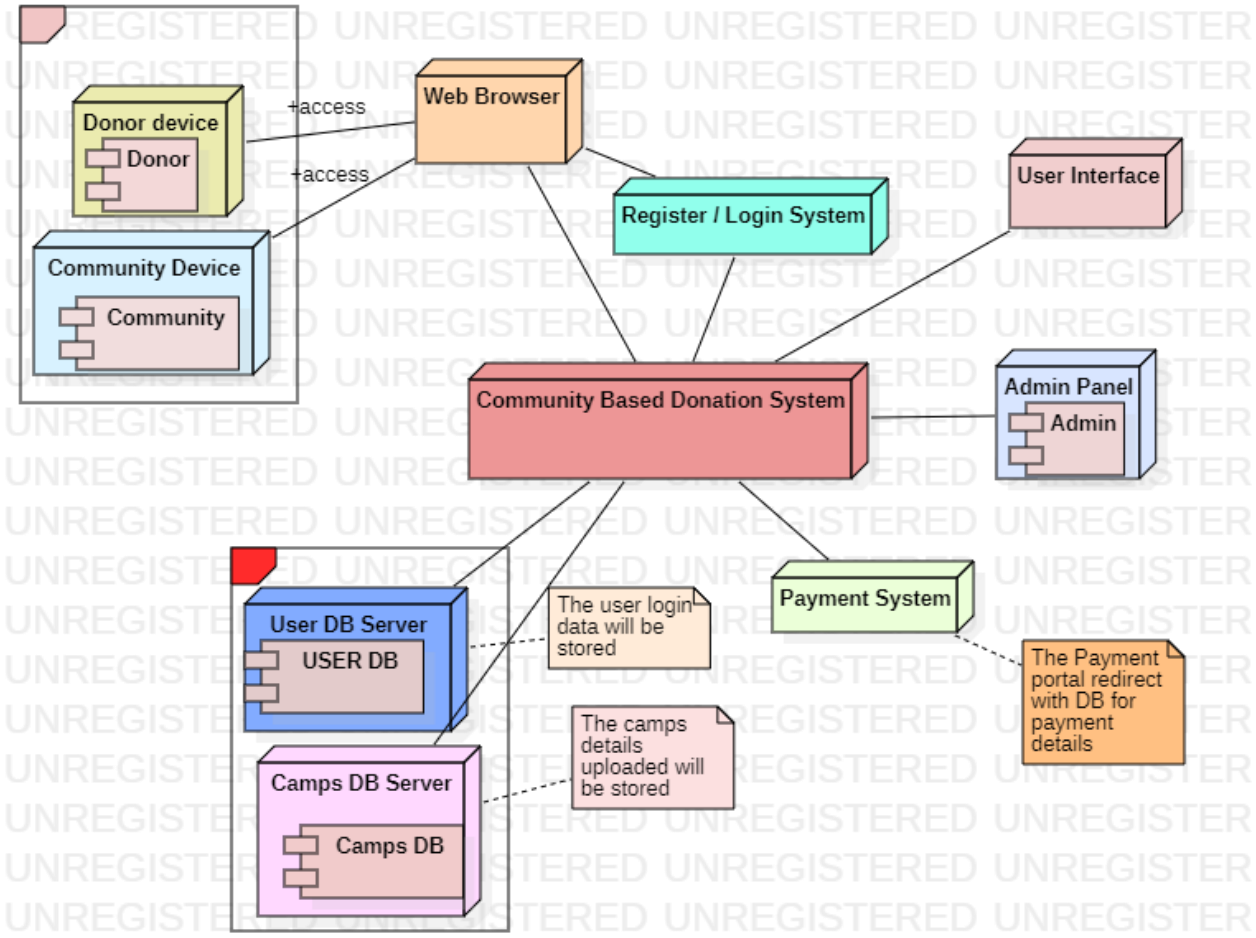


Fig 10: Deployment Diagram

BASIC FRONT END

11.Warangal – Community Based Donation System (Proposed Model)

These are the basic front end designed by our team. This template prototype is designed by us on WiX platform. (for demo) - https://areefahnk.wixsite.com/my-site-1

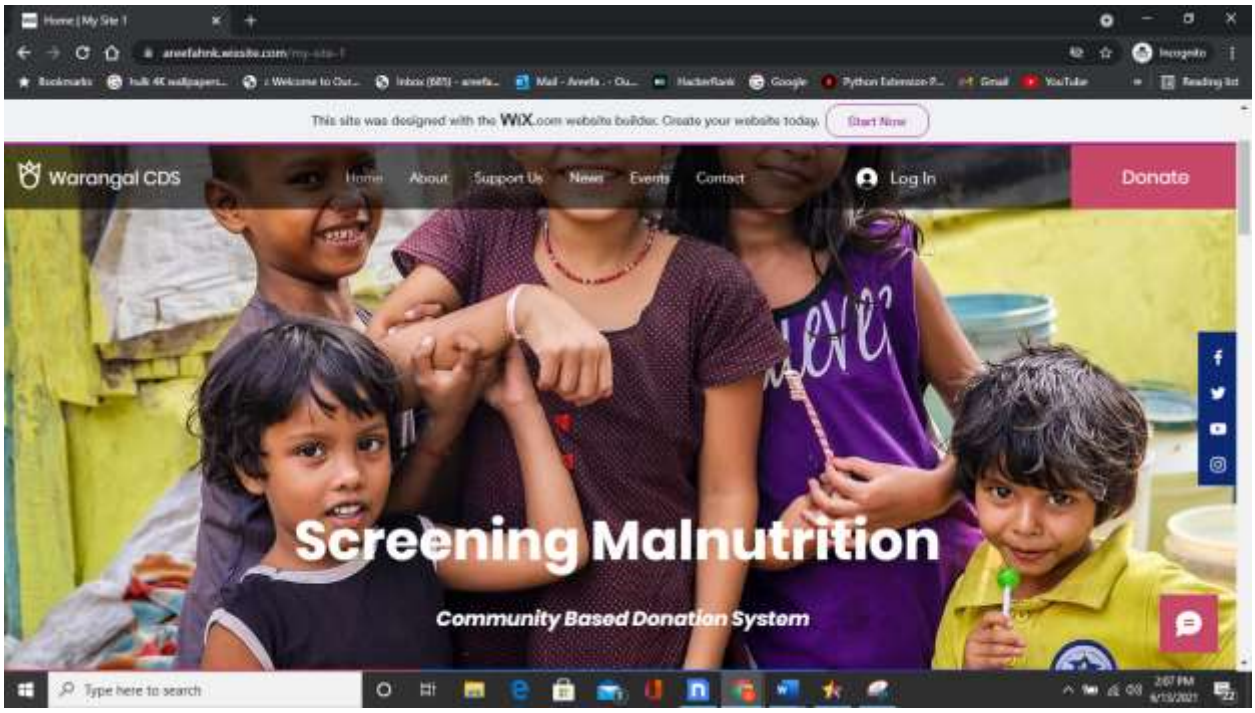


Fig 10.i. Basic Layout Home

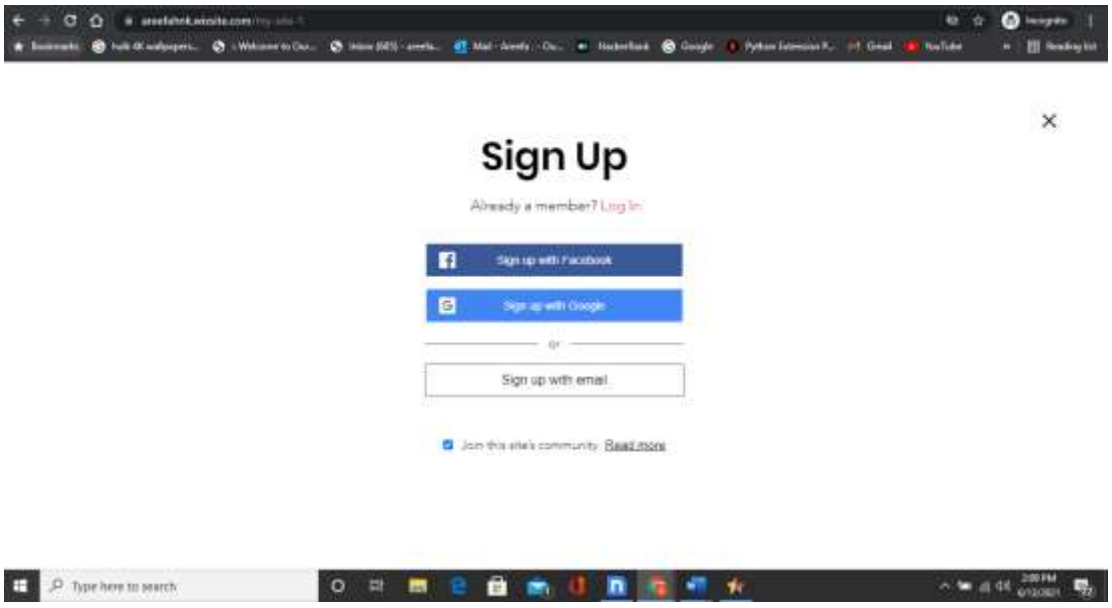


Fig 10.ii. Community / Donator Sign Up / Login

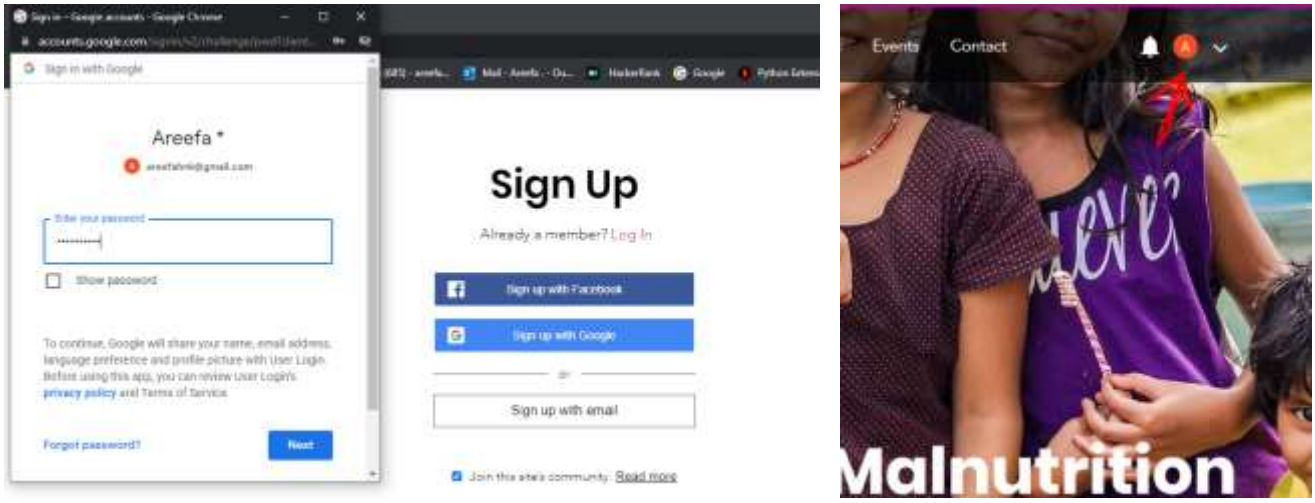


Fig 10.iii. Successful SignUp and Login

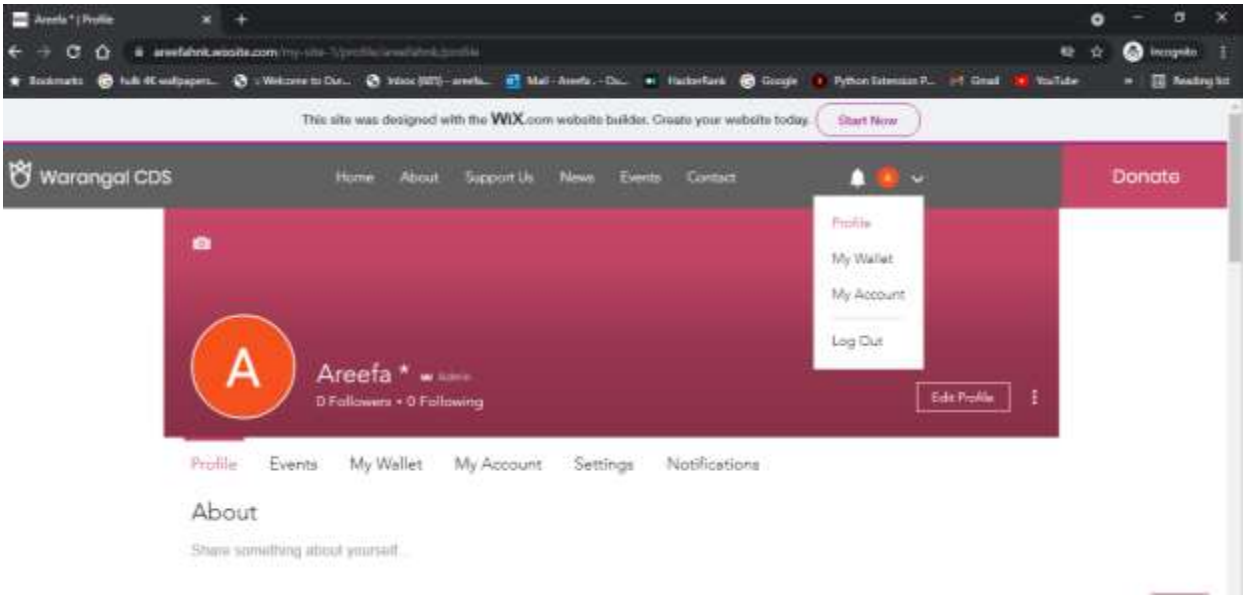


Fig 10.iv. Admin Profile

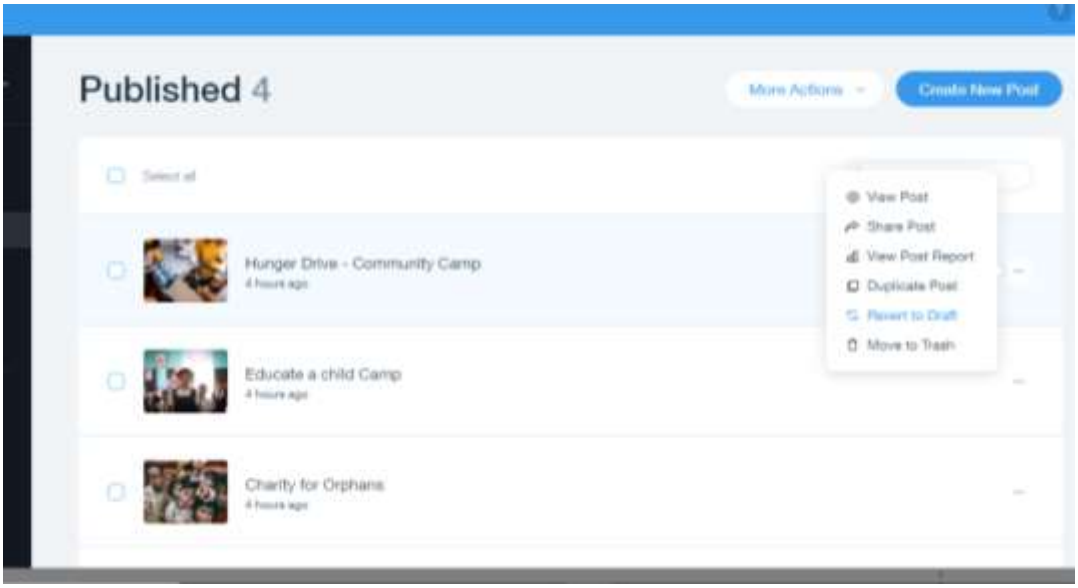


Fig 10.v. Admin Approves or rejects the camps (created by communities)

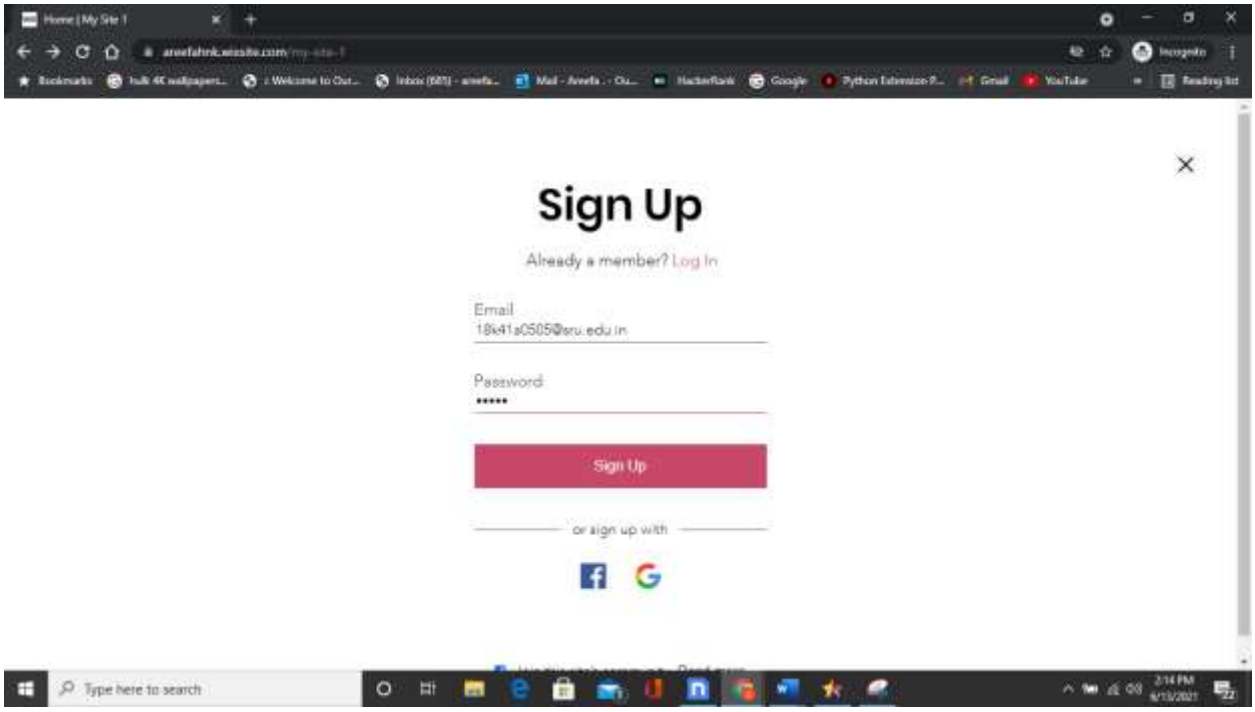


Fig 10.vi. User / Community Sign Up

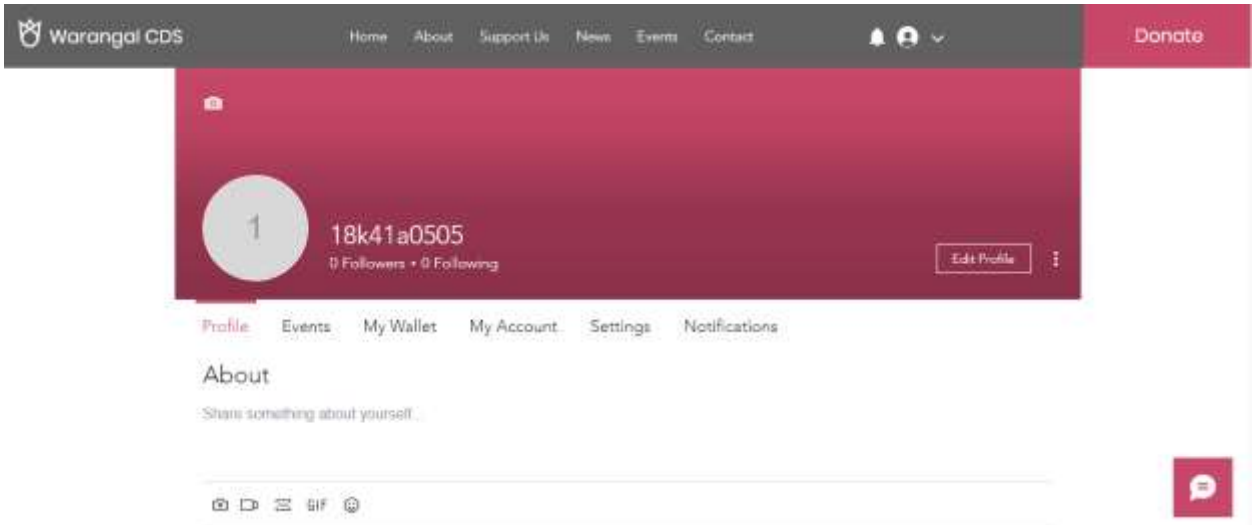


Fig 10.vii. User Profile

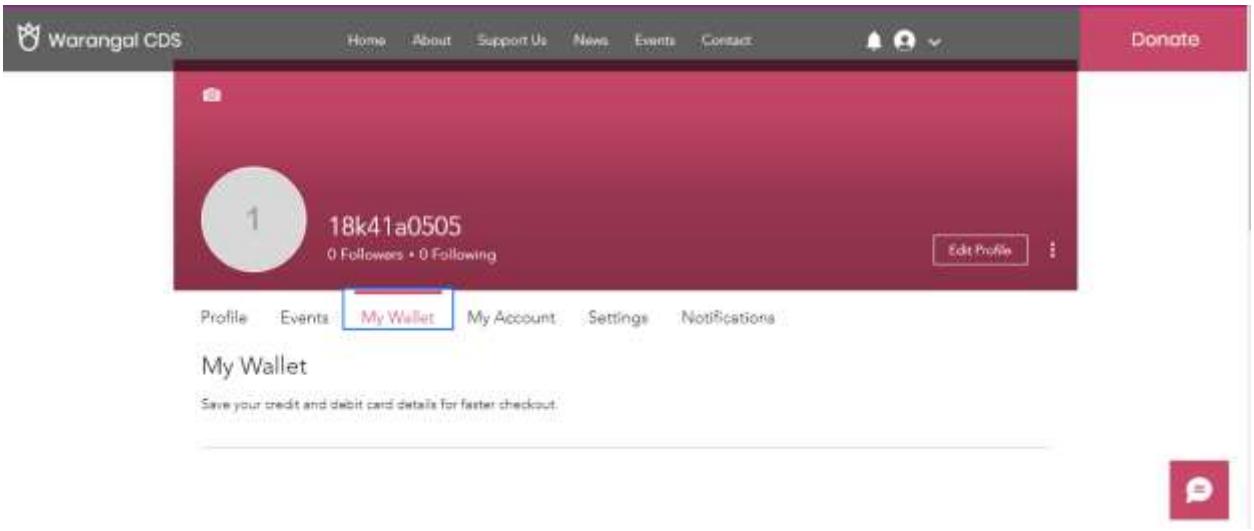


Fig 10.viii. User Wallet

Fig 10.ix. Ongoing Scheduled Camps by Communities

(Users can select camp from this button and donate)

Fig 10.x. Upcoming Camps to donate

Fig 10.xi. Camp & Community details

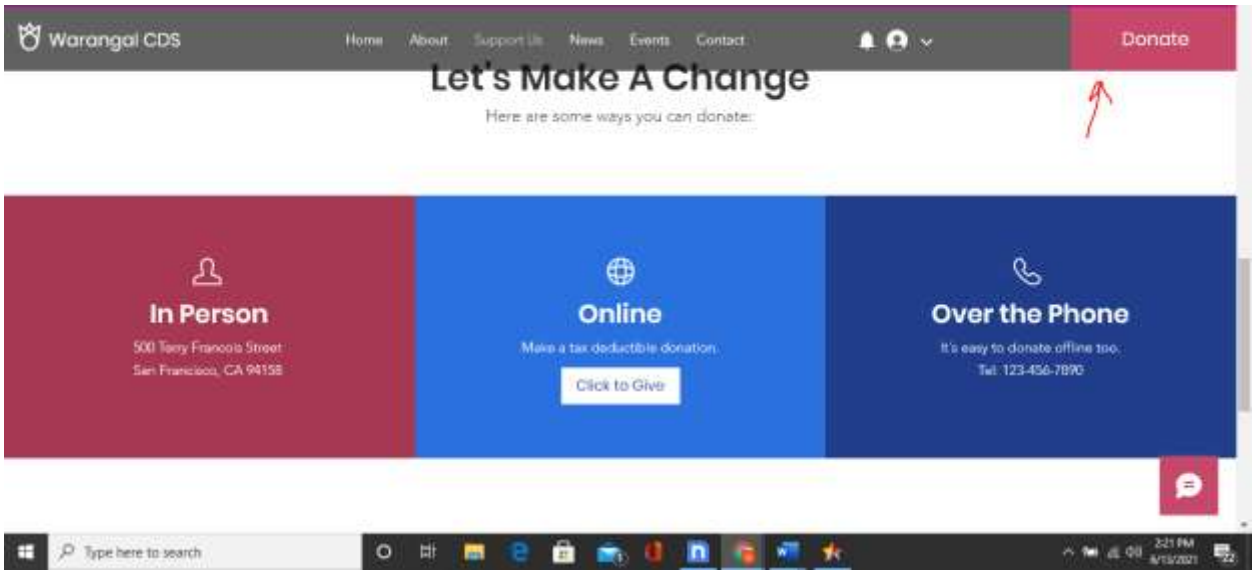


Fig 10.xii. Donation Payment Options

Fig 10.xiii. Contact Details

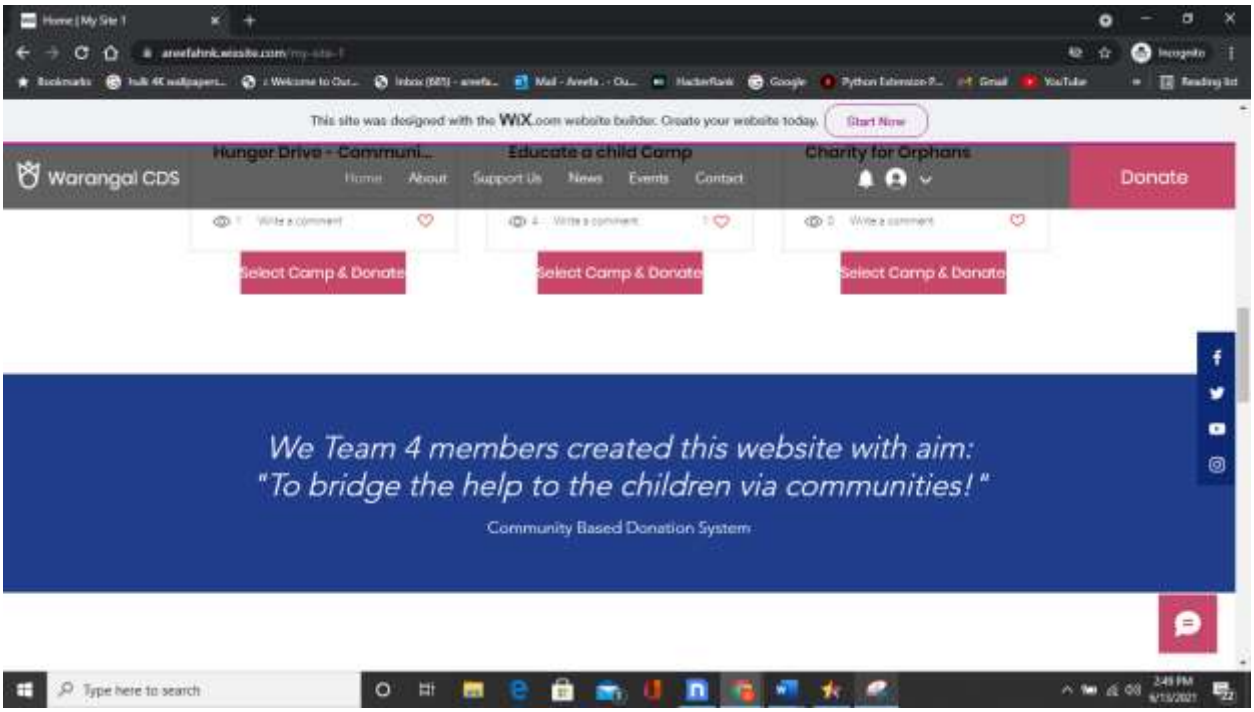


Fig 10.xiv. Page view