SRS DOCUMENTAION

2023-2024

CHARITY PROJECT

كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

THIS FORM IS FOR BOTH THE GENERAL & MEDICAL INFORMATICS PROGRAMMES
SE-I COURSE PROJECT (PHASE 1 COVER SHEET)
Discussions Scheduled for Week 8 *(more details will be announced later)***.**

- Print 1 copy of this cover sheet and attach it to a printed copy of the documentation *(SRS, … etc.)*. You must also submit softcopies of all your documents *(as PDFs)*; details will be announced later.
- Please write all your names in Arabic.
- Please make sure that your students' IDs are correct.
- Handwritten Signatures for the attendance of all team members should be filled in before the discussion.
- Please attend the discussion on time *(announced separately)*, late teams will lose 5 grades.

Project Name: Charity Organization
Team Information *(typed not handwritten, except for the attendance signature)*:

Grading Criteria:

5 Items		Grade	Notes
1. Updated Use-case Diagrams & Descriptions, Activity Diagrams, Object Diagrams, Sequence Diagrams, System Architecture	**2**		
2. Collaboration/Communication Diagrams	**2**		
3. Class Diagram *(3 versions)* **1) An initial version based on the requirements and Use-Case/Activity diagrams. [Submitted in phase 1]** **2) An intermediate version based on the interaction diagrams.** **3) A final version, after applying the design patterns and any other modifications.**	**3**		
4. Three Mandatory Design Pattern Applied *(Including a typed description)*	**4**		
5. Front End Design for all Functions *(HTML, Bootstrap)*	**2**		
6. Implementation based on the submitted Requirements & Design. Should include at least 4 of the following modules (in addition of course to modules specific to your individual projects): **1) User Role Management Module.** **2) User manipulation Module** *(Login, Add / Delete / Update / Search, List).* **3) Controlling Resources Module** *(Rooms, Orders, Products, … etc.).* **4) Reservation and Rescheduling Module.** **5) Generating Reports Module** *(PDFs, … etc.).* **6) Sending Emails or Notifications Module.**	**7**		

Teaching-Assistant's Signature: _____

20

Introduction

Welcome to our charity management system! Our software solution is designed specifically for non-profit organizations to manage their charitable donations and donor base. With features such as electronic fund transfer, donation management, fundraising, donor management, and event management, our system makes it easy for organizations to streamline their operations and focus on their mission. Our key features include the ability to easily track donations and donors, send gift messages, monitor payment plans and cycles, set up recurring payment plans, and generate detailed

reports on donations. By using our software, non-profit organizations can automate their donor relations and financial processes, saving time and resources, and ultimately making a greater impact in their communities, our software solution is designed to be flexible and user-friendly, allowing non-profit organizations to efficiently manage their entire operation in-house. By automating the manual operation of managing fundraising campaigns, donors, and financial management, our system saves time for more critical focus areas. Our customizable reporting tools allow organizations to generate reports of any type, keeping them on track of their progress and fund management. Our centralized database feature creates a single location for storing information about volunteers, events, campaigns, pledges, trusts, and donors, eliminating the need for multiple sheets in various places. Our screening component helps track and predict donor giving habits, while our efficiency feature organizes campaigns and donations from various platforms, including social media, email campaigns, and event organization. Our key features include communication tools that filter donors, volunteers, and peers based on various criteria, allowing for easy personalization and email marketing. We also keep a track record of all communication and provide different opportunities to create and connect. By using our charity management system, non-profit organizations can streamline their operations, save time and resources, and ultimately make a greater impact in their communities.

Functional requirements

1- System shall manage donations that come from donor or from events and campaigns.

1.1 The system shall import online donor and donations records from PayPal or other services.

1.2 -The system shall Allow donors to select exactly how their donation should be used.

1.3 The system shall Create unique departments and track nonprofit program goals.

1.4 The system shall manage donations to people need these donations.

2- System shall send gift messages to donors after successful donation process.

2.1 System shall provide the ability to send gift messages allows donors to personalize their donations and communicate their appreciation or support to the recipient of the gift.

3- System shall record our main actors information (Admin, Sponsor,donors,volunteers) in addition to events and campaign in database.

3.1 System's database shall record event's name, date, and location.

3.2 System's database shall record campaign's name, date, and it's purpose (campaign's goal).

3.3 System's database shall record donor's name, mail information (username, password, mail), home address and donation category which donor donated to.

4- System shall set up recurring payment plans.

4.1 System shall monitor payment's plans and cycles for donors.

4.2 System shall distribute the payment plans to categories according to amount of price the donor will send.

4.3 System shall allow donor to add new recurring payment plan, delete it or update his payment plan (e.g. : In our system there are three payment plans (Gold-Silver-Bronze) with two create choices which are (Rank payment: which a recurring payment depend on amount of money ever certain time) and duration payment (depend on the duration of donation), donor can cancel one of them and subscribe on the other one . for example, cancel gold payment plan subscription and join silver payment plan.

5- System shall Generate Detailed Report to help admin to track system statistics and progress.

5.1 The System shall monitor donor profile and check his payment plans, donations and how many donations he made to send a gift message to him and check if there are a recurrent payment.

5.2 The System shall provide a dashboard for the admin contains needed details about recurring donors , fund transfer that come from donors (e-fund , campaign's fund or event's fund), and provide statistics about new donors and volunteers to know the hierarchy of organization).

5.3 The System shall generate reports for donors, sponsors and volunteers and assist admin to modify these reports if needed.

5.4 Database shall record the donation's report and start record it then distribute total donation on a table on it to distribute funds equally for needed people.

6- System shall provide advantage of screening such as wealth screening.

6.1 The system must provide wealth screening which is a way that organizations assess their donors' assets to learn how much they can give. This information informs how much your organization should request when making an ask and this type of screening is crucial in charity organizations.

6.2 The system has a component that helps to track and predict when and how the donors give and extend to highlighting their giving or donation habits.

7- System shall provide a rigid real-time communication between donor and system or even an admin and system.

7.1 The admin can filter donors, volunteers, peers based on various donations allows ease of communication or personalization.

7.2 The system sends marketing emails for the donors automatically.

7.3 The system shall establish groups for donors of same interests depending on their donations.

8- System shall provide an embedded fundraising tool (as ease of donation system) to help organization to grow financially on behalf of charity works.

8.1 The system shall provide Embeddable buttons on your website that will redirect to your donation page.

8.2 (Peer-to-peer): System shall allow you to launch campaigns where your supporters create teams and fundraise on organization's behalf.

8.3 (Crowdfunding): System shall allow your supporters to launch their own campaigns and fundraise on organization's behalf.

8.4 System shall provide an important feature which called (Zakat calculator), which help donor, admin and volunteer to access it to know how much donation can donor send even if it was a/an (gold zakat , money zakat , assets zakat,… etc.)

9- System shall record all donor's activities.

9.1 The system shall give objectives for donors and show them the impact of their donations.

9.2 Give your donors the ability to view and manage their giving history, recurring donations, receipts.

10- System shall track fundraising campaigns and their progress.

10.1 The system shall track campaigns and which donations resulted from them.

11- System shall track fund transfer process and ensure about fund transfer process completion.

11.1 System shall be making connection between other payment system as PayPal, visa, or any payment services easy to access by the donor.

12- System shall manage all charity events details according to location, time, and predict sponsors who interested in specific event.

12.1 The admin determines the number of volunteers and the number of sponsors.

12.2 According to the vacant places, the number of registered in the event shall be determined by the system.

12.3 System shall provide many sponsors to one event.

Non-Functional Requirements

1- System shall have a secure login system saved in a proper database server to save the security of the donors.

1.1 The system must provide a secure and reliable login process that allows admins to access the platform and their accounts.

1.2 The login process should require admins to enter their credentials, such as their username and password, and should verify that the credentials are valid before allowing access to the platform.

2- System shall be flexible to wide range of users , ease to use as user can interact with it easily without any training and responsive to all devices (Pc , Laptop , Mobile , etc.…)

2.1 The platform should be designed to accommodate a wide range of users, including donors, volunteers, and administrators, with different levels of access and different needs and preferences.

2.2 The system must be easy to use in different donation methods and variant E-fund ways.

2.3 The system should also be able to handle different types of donations, such as one-time donations, recurring donations, and donations in-kind.

3- System shall be efficient as it shall provide fast responses to each actor in the system with his/her needed message or transactions and work correctly without any error.

3.1 The system must be efficient and performant, with fast response times and minimal downtime.

3.2 The platform should be designed to handle many users and transactions simultaneously, without slowing down or crashing.

3.3 The system should be able to handle large volumes of data, such as donor information, campaign data, and transaction logs, and should provide tools and features to efficiently manage and analyze this data.

4- System shall provide a fast performance to make UX a rigid one.

4.1 System shall ensure that the charity platform can handle the expected load and traffic.

4.2 System must provide a fast and reliable experience for users. By providing a high-performing and reliable platform, the system can improve the overall user experience and encourage users to continue using and supporting the charity (such as make responsive and attractive landing page).

5- System shall be scalable (As well as system grows and number of donors and volunteers increase, the performance shall be fixed as much as possible)

5.1 The system must be designed to scale up or down as needed, to accommodate changes in user traffic, data volumes, or system requirements.

5.2 The platform should be able to handle increased load and traffic without compromising performance or reliability.

5.3 System should be able to support additional features or functionality as needed.

5.4 System shall help admin to search about users by user-type (donors, volunteers, or sponsors) retrieved from database or by username.

6- The System shall be easy to access and determine the right person who can access the data which he/she authorized to access it

6.1 The system shall provide a positive user experience for every user (better UI experience).

6.2 The system can improve the overall user experience and demonstrate the charity's commitment to diversity, equity, and inclusion.

6.3 System shall include a feature to help user to retrieve his password by sending him a password texted on his mail.

7- System shall provide security and prevent any unauthorized users to access admin's data for example by ordinary user.

7.1 The system must be designed to be secure and protect sensitive user information and data from unauthorized access, modification, or disclosure.

7.2 The system should implement authentication and authorization mechanisms to control access to sensitive data and functionality.

8- System shall provide real-time activities made by donor or by the admin especially e-fund processes.

8.1 The system must be designed to be responsive and provide fast response times for user interactions and requests.

8.2 The platform should be optimized for performance, with minimal delays or downtime, and should be able to handle high volumes of user traffic without degradation in performance.

9- System shall be maintainable as any error or crash by the system happen, software engineers or developers can manage it easily.

9.1 The system must be designed to be easily maintainable and adaptable to changes over time.

9.2 Software engineers shall ensure that the code is well-organized and well-documented, with clear comments and documentation to make it easy for developers to understand and modify.

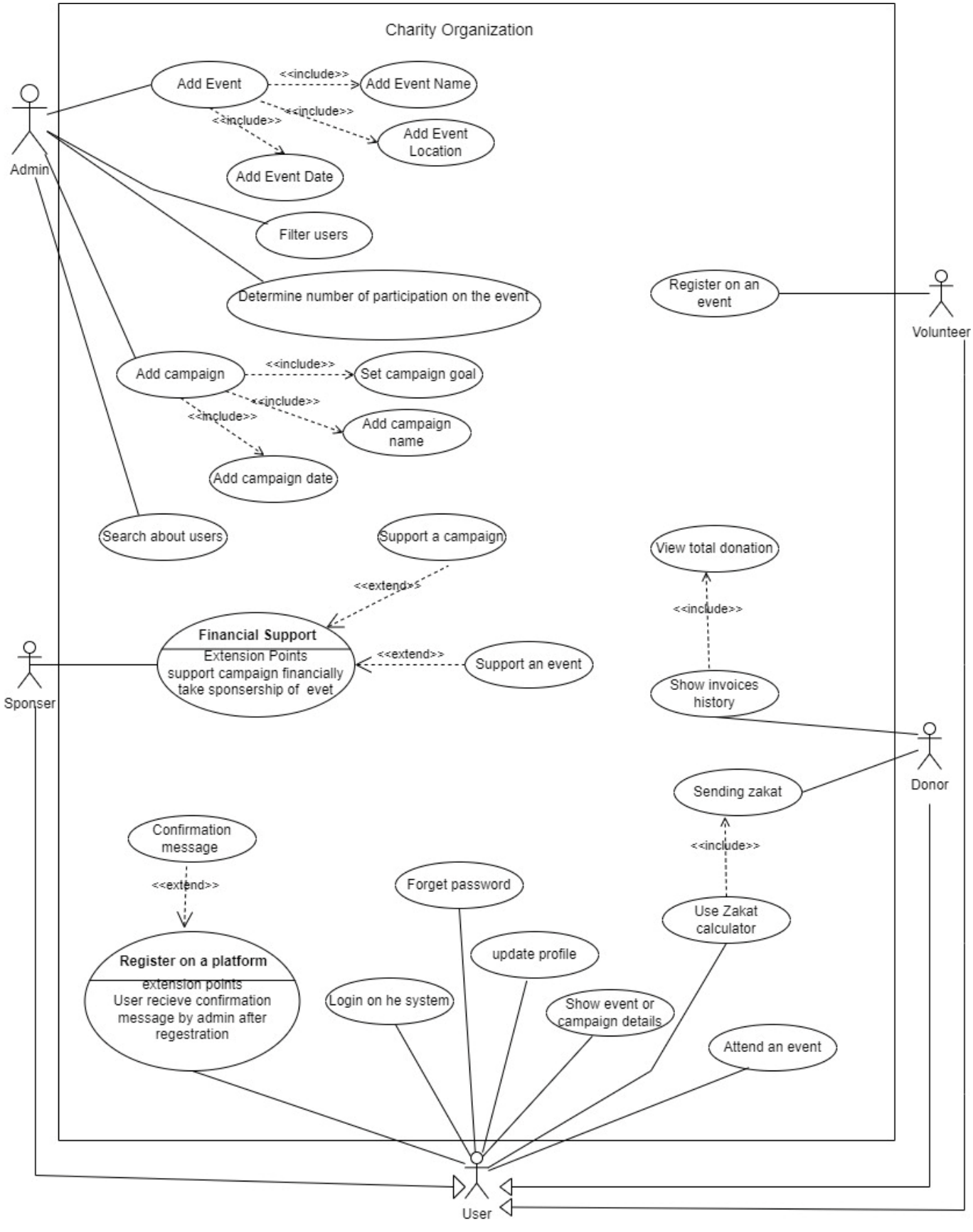
9.3 The platform should also be designed to support ongoing maintenance and updates, with the ability to easily test, debug, and deploy changes as needed.

10- System's marketing campaigns and events shall be accurate, specific, and detailed.

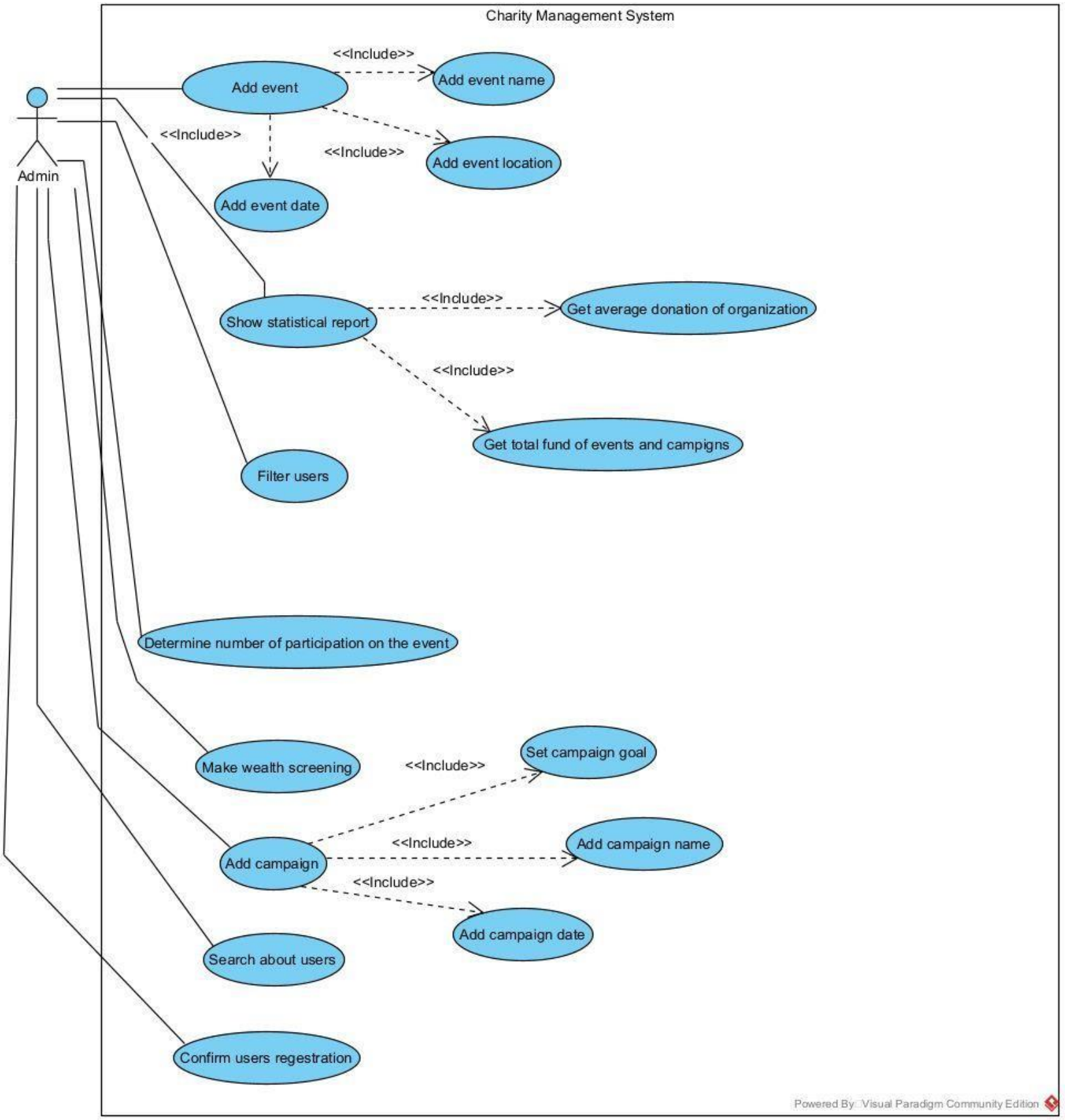
10.1 Event or campaign's goals shall be specific, measurable, attainable, and relevant to organization's overall mission and time based.

10.2 Event or campaign shall have specific start and end date, location, and name.

Use case diagram.
Use case diagram detailed description



Admin Use cases



Admin use cases:-

ID Number – Name:	Add event
Short description:	Admin want to initiate a charity event to attract participation or audience to support the organization.
Preconditions:	Admin organizes event for organization and determine name, location, start date, end date and number of people who will participate in this event/campaign.
Postconditions:	The event added on website and users start to register on it.
Error Situations – System State in the Event of an Error:	Admin doesn't determine or ignore one of essential attributes of event (name, location, date, or number of participations).
Actor(s) / Initiator(s):	Admin
Trigger:	Admins agreed to initiate an event/campaign.
Standard Process (Main Success Scenario):	1- Event initiated on the start time determined. 2- Event ended on the time determined.
Alternative Processes (Alternative/Other Scenarios):	1'- Event cancelled or may be postponed. 2' - Admin extends event duration or may be postponed it.

ID Number – Name:	Show statistical reports.
Short description:	Admin want to know number of campaigns or events initiated, number of users on the system (number of donors, volunteers, sponsors) and progress of campaigns.
Preconditions:	Admin want to know some statistics about the organization to determine the transactions happened on the website or progress of campaigns.
Postconditions:	The admin gets the mathematical statistic as he wants from website for a specific purpose successfully.
Actor(s) / Initiator(s):	Admin
Trigger:	Admin wants to get number of volunteers, sponsors and donors on the system , know the progress bar of every campaign he/she published or know total number of donation on the organization.
Standard Process (Main Success Scenario):	1- Admin wants to know number of volunteers, sponsors, or donors. 2- Admin want to know progress of certain campaign.
Alternative Processes (Alternative/Other Scenarios):	None

ID Number – Name:	Filter users
Short description:	Admin want to determine the activities of all users in the system to determine if he/she can remove it from the system or not.
Preconditions:	Admin displays the activities of all users on the system (for example a sponsor no longer provides the organization financially or cancel the partnership with organization so the admin shall remove him/her).
Postconditions:	The user (sponsor, volunteer, or donor) filtered successfully.
Actor(s) / Initiator(s):	Admin
Trigger:	Admins wants to filter system's users monthly.
Standard Process (Main Success Scenario):	1 – Admin filtered user successfully. 2 – User removed from database.
Alternative Processes (Alternative/Other Scenarios):	None.

ID Number – Name:	Get average donation of organization.
Short description:	Admin wants to know total fund of the organization.
Preconditions:	Admin want to display the total financial support of organization especially ordinary donation that sent by admin with visa.
Postconditions:	The database recorded the donations and start to distribute it.
Actor(s) / Initiator(s):	Admin
Trigger:	Admin want to know the total transaction and donations on the organization.
Standard Process (Main Success Scenario):	1- Admin want to know the total donation made by the organization. 2- The total donation updated in database as real-time action on the system to start record it to distribute this fund for needed people or save it for other donations or campaigns.
Alternative Processes (Alternative/Other Scenarios):	None.

ID Number – Name:	Get total funds of event or campaign.
Short description:	Admin want to display the total funds come from organization from event or campaign from the system's report.
Preconditions:	The donations from events or campaigns recorded successfully and added to database.
Postconditions:	The database recorded the needed donations successfully.
Actor(s) / Initiator(s):	Admin
Trigger:	Admin want to know the total transaction and donations made by participant in the event to know total funds come from this event/campaign.
Standard Process (Main Success Scenario):	1- Admin want to know the total donation made by the donor. 2- Admin want to know the most interested campaigns or donation's category by the donor
Alternative Processes (Alternative/Other Scenarios):	None

ID Number – Name:	Search about users.
Short description:	Admin want to search about user by his name

	or by the user type (Admin, Sponsor or Volunteer)
Preconditions:	Admin wants to search about specific user by name or wants to search about all users by type (For example search about all volunteers to select all their mails to send them a recommendation to join a specific event or to lead a specific event.)
Postconditions:	The admin search about specific user/users and send the needed message or handled specific action successfully.
Actor(s) / Initiator(s):	Admin

ID Number – Name:	Add campaign

Short description:	Admin want to set a goal for campaign and start to market for it and determine its start date, end date and its name.
Preconditions:	Admin wants to publish a campaign and set the marketing base goal for this campaign to attract donors and sponsors to donate to organization.
Postconditions:	The marketing base attract donors and sponsors to donate to organization and this cause exponentially increasing on donation according to campaign's goal
Actor(s) / Initiator(s):	Admin
Standard Process (Main Success Scenario):	1- Admin starts to select the appropriate goal for the campaign. 2- Admin set the name, start date, and end date for the campaign.
Alternative Processes (Alternative/Other Scenarios):	2'- Admin can cancel the campaign after publishment.

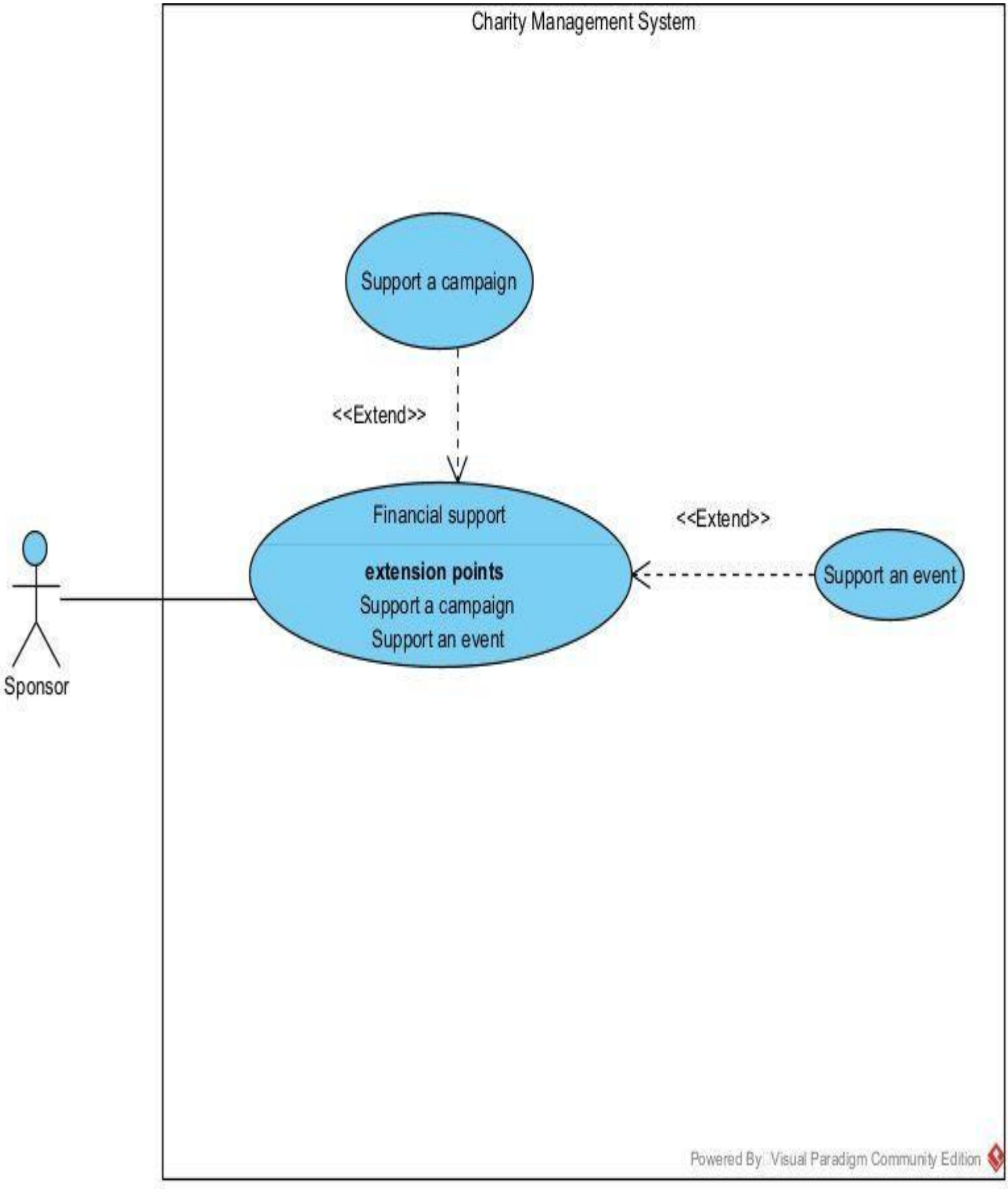
ID Number – Name:	Make wealth screening
Short description:	Screening: means following the admin donations and transaction to know the most interested donated category or campaign.
Preconditions:	Admin want to know the most donated category according to the interests of the users determined by wealth screening.
Postconditions:	Admin send a message for all users about the most donated campaigns and donations' category generally and the most donated category for specific user specially.
Actor(s) / Initiator(s):	Admin
Trigger:	Admin want to know the total transaction made by sponsor in system.

ID Number – Name:	Confirm user registration.
Preconditions:	New user starts to request to join the charity platform as sponsor, volunteer, or donor.
Postconditions:	Admin confirms his/her registration.
Actor(s) / Initiator(s):	Admin
Trigger:	New user starts to sign up on the system.

ID Number – Name:	Determine number of participations for the event

Preconditions:	Admin want to determine number of people who will participate in the event.
Postconditions:	The admin published the number of people for the event on the website and the available text.
Actor(s) / Initiator(s):	Admin
Trigger:	The admin publishes an event and start to announce about the available tickets.
Standard Process (Main Success Scenario):	1- Admin determines the number of tickets and it all reserved successfully.
Alternative Processes (Alternative/Other Scenarios):	1'- The number of participations exceed the number determined by the admin or several participations cancelled their registration for the event.

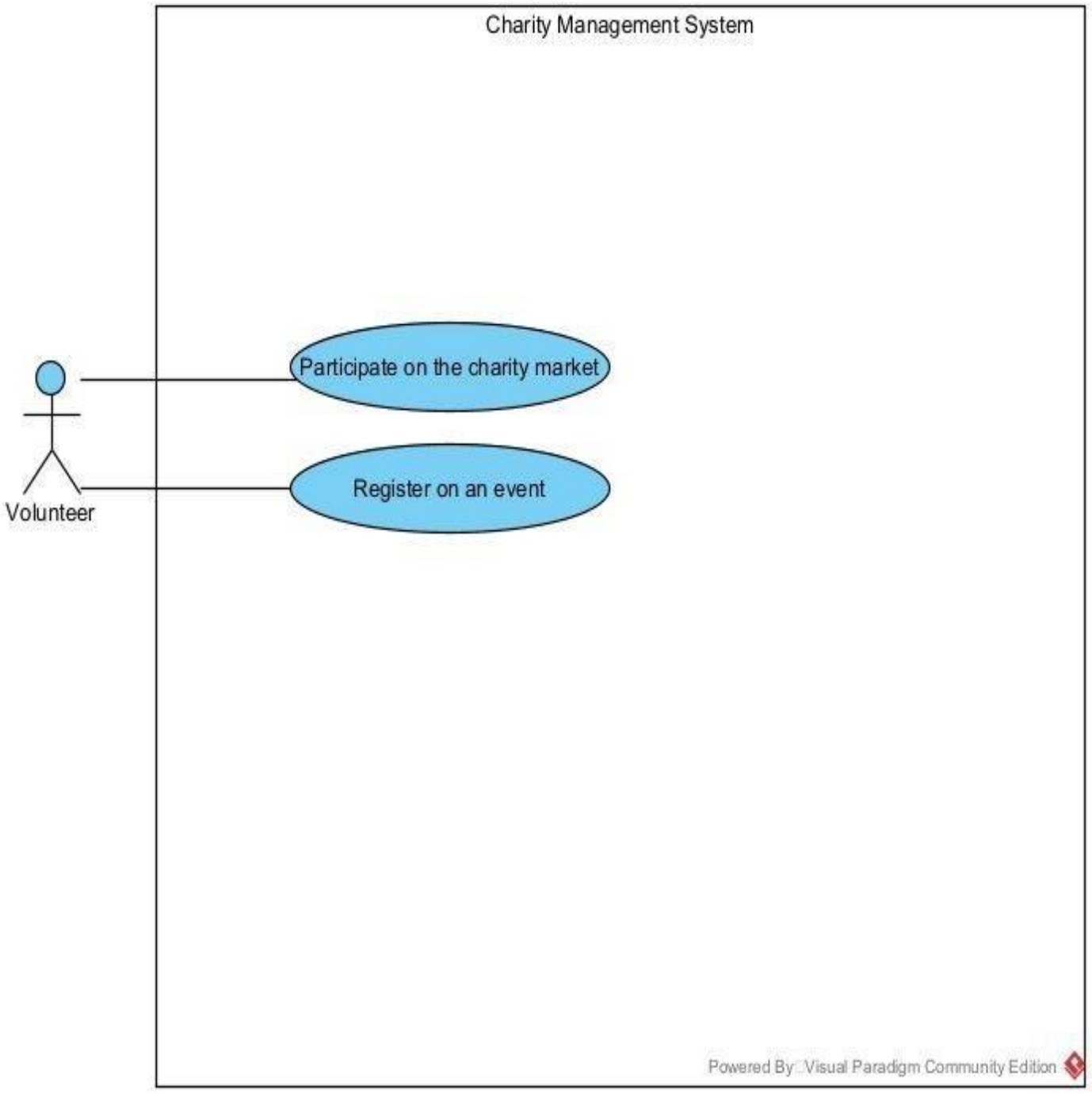
Sponsor use cases



Sponsor use cases: -

ID Number – Name:	Financial support
Short description:	Sponsor supports the organization financially to all the main parts of it (Support campaign, Support event or support by donation)
Preconditions:	Sponsor registers on website and start to send donation, recommend carrying an event or support a campaign financially
Postconditions:	The admin approved the sponsorship of specific sponsor to support an event financially.
Actor(s) / Initiator(s):	Sponsor
Trigger:	Sponsor starts to register on a website and start to recommend a sponsorship with the organization.
Standard Process (Main Success Scenario):	1- Sponsor sends a donation for organization. 2- Sponsor recommend handling a campaign financially or the whole marketing base. 3- Sponsor recommend carrying an event.
Alternative Processes (Alternative/Other Scenarios):	None.

Volunteer Use cases



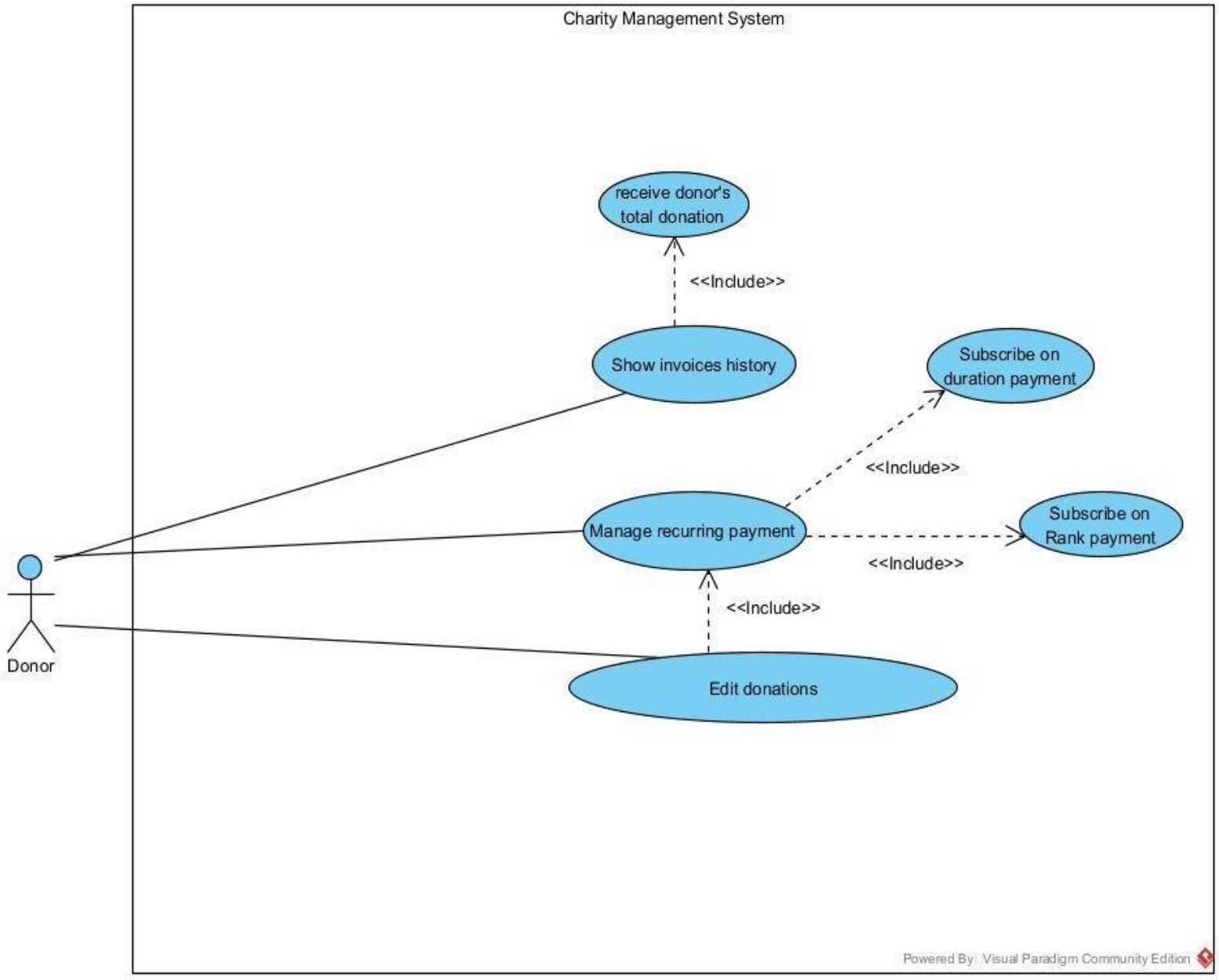
Volunteer use cases: -

ID Number – Name:	Register on an event
Short description:	Volunteers wants to join an event and want to assist on charity works for this event.
Preconditions:	Volunteer register on an event and start to know his role in the event.
Postconditions:	Volunteer attends the event successfully.
Actor(s) / Initiator(s):	Volunteer.

ID Number – Name:	Participate on charity market.

Preconditions:	Volunteer start registering to join charity markets.
Postconditions:	Volunteer joins one of charity markets.
Actor(s) / Initiator(s):	Volunteer.

Donor Use cases



Donor use cases: -

ID Number – Name:	Edit donations.
Short description:	Donor wants to modify the donation lists.
Preconditions:	Donor wants to modify or cancel his donations lists and plans including recurring payments plans.
Postconditions:	The list of donations and payment's plans successfully modified in the donor's report.
Actor(s) / Initiator(s):	Donor.
Trigger:	Donor wants to modify his payment plan (add, delete, update).
Standard Process (Main Success Scenario):	1- Donor wants to cancel his donation after he send it by his visa. 2- Donor wants to add a new payment plan or make a new donation. 3- Donor wants to update his payment.
Alternative Processes (Alternative/Other Scenarios):	None

ID Number – Name:	Manage recurring payments.
Short description:	Donor wants to modify the plan of recurring payments.
Preconditions:	Donor wants to add, update, or delete a payment plan.
Postconditions:	The transactions made by donor on a recurring payment plan updated on the system and donor's report.
Actor(s) / Initiator(s):	Donor.

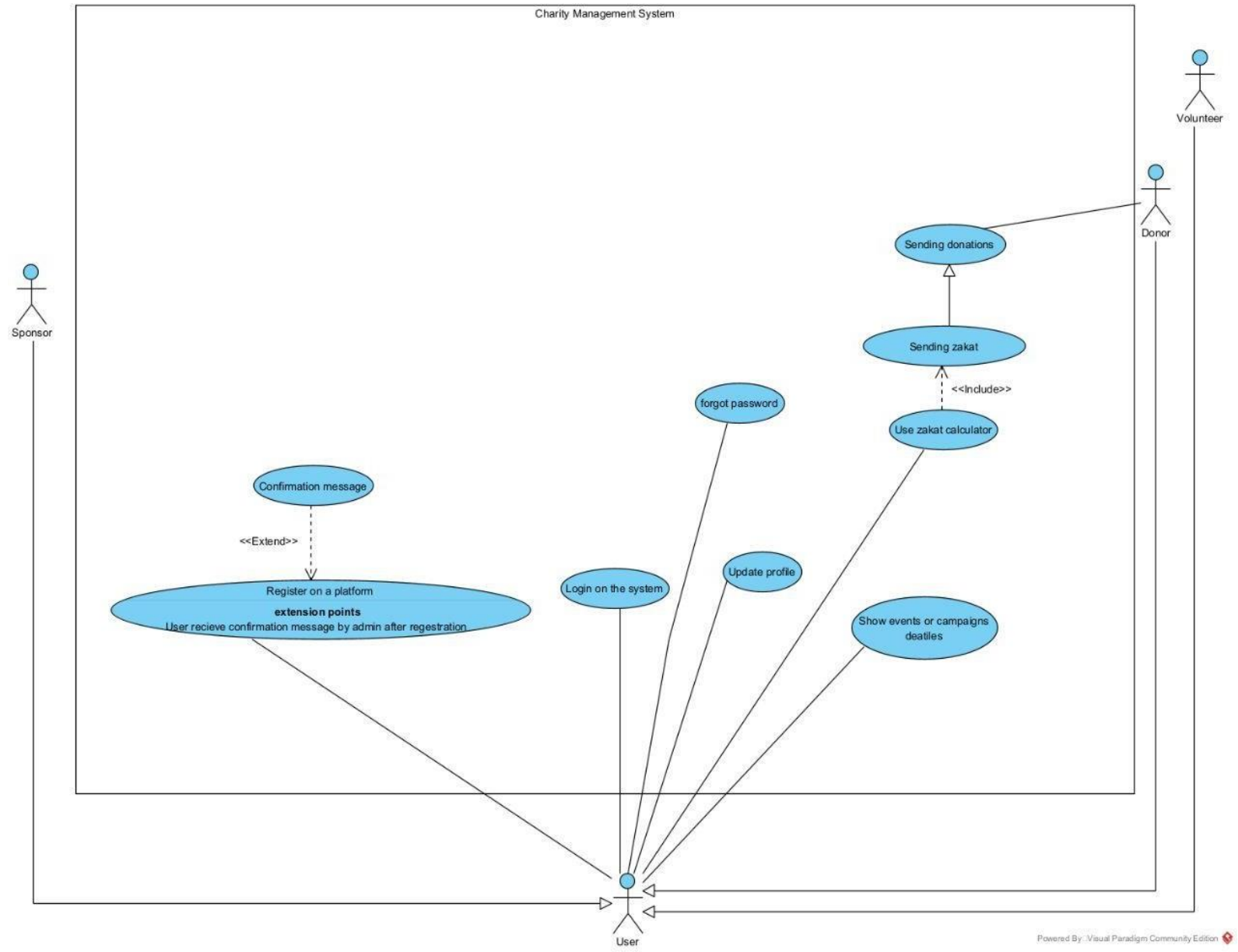
ID Number – Name:	Sending donations
Preconditions:	Donor wants to send donations to a specific campaign or donation's category.
Postconditions:	Donation sent to its destination successfully by offer a secure visa platform for the donors.
Actor(s) / Initiator(s):	Donor.

ID Number – Name:	Sending zakat

Short description:	Zakat: Is a special donation type sent to a specific donation's categories (gold donation, assets donations, ...etc.) with specific percentage of money (This use case generalized from sending donations).
Preconditions:	Donor wants to send zakat (Gold zakat for example) after he put amount of money comes from this gold and put it in zakat calculator to determine amount of money will be donated.
Postconditions:	Donation sent to its destination successfully by offer a secure visa platform for the donors.
Actor(s) / Initiator(s):	Donor.

ID Number – Name:	Show invoices history.
Short description:	Donor wants to receive a report
Preconditions:	Donor wants to show all his donation in invoice recorded by the system from table of reports in database.
Postconditions:	The invoice displayed to the donor successfully.
Actor(s) / Initiator(s):	Donor.
Trigger:	Donor wants to display all his/her transactions and donations for the organization.

User Use cases



User (Generalized actor for sponsor , donor and volunteer actors) use cases : -

ID Number – Name:	Register on the platform
Short description:	Users want to make an account on the organization platform.
Preconditions:	User adds his personal username, e-mail, and password.
Postconditions:	The information about user on signup saved successfully in login table on database.
Error Situations – System State in the Event of an Error:	User writes a wrong email format or restricted password length (too short password or too long one).
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).
Trigger:	User wants to join our charity platform.
Standard Process (Main Success Scenario):	1- User writes his personal information on signup page and admin send him a confirmation message.
Alternative Processes (Alternative/Other Scenarios):	1'- User write a wrong formats for mail or password as well as the system send to him/her an error message.

ID Number – Name:	Update profile
Preconditions:	User wants to change information in his profile (his username or to change his donation plan as a donor for example)
Postconditions:	Information changed in database successfully and updated to user even if he reloads his profile or sign out.
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).

ID Number – Name:	Login to the system.
Short description:	Users want to enter his/her account on the organization platform.
Preconditions:	The information about user retrieved from database to know that this information matches the data on login table or not.
Postconditions:	The user enters his/her profile successfully.
Error Situations – System State in the Event of an Error:	User writes his/her information wrongly.
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).
Trigger:	User wants to enter his/her profile.
Standard Process (Main Success Scenario):	1- User writes his information (mail or password) and enters his/her profile successfully.
Alternative Processes (Alternative/Other Scenarios):	1'- User write a wrong cardinalate as well as the system send to him/her an error message or even not have a pervious account on organization.

ID Number – Name:	Use zakat calculator
Preconditions:	User want to calculate a specific type of zakat to send his/her money as a zakat.

Postconditions:	User knows percentage of zakat and sent his donation to the system and admin starts to handle this zakat to donation's category which need this zakat.
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).

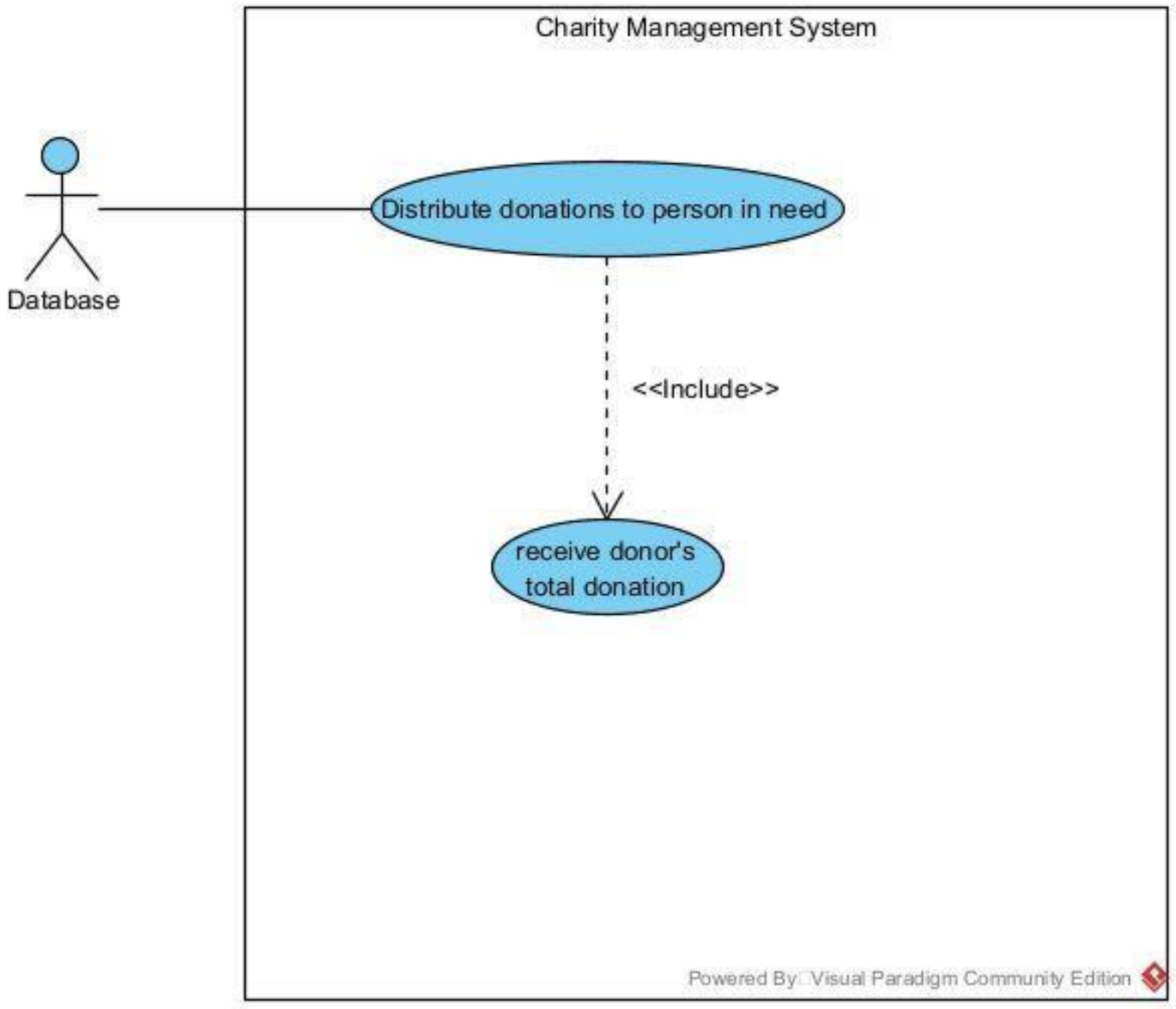
ID Number – Name:	Show event and campaign details
Preconditions:	Users view every event's or campaign's details including (Location, start date, end date, name or even number of participations for an event).
Postconditions:	Users start to make a various activity according to their type. for example , as a donor I can register on an event/campaign to attend it or donate to the campaign , as admin I can view every details on the website to make sure that the details published is a correct one , as volunteer I request to join the event/campaign to handle charity works happen in this event and finally as sponsor I can support event/campaign financially or on the organization marketing base , carry the event and make it sponsored by me or to attend the event.
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).

ID Number – Name:	Forgot password.
Preconditions:	User doesn't remember his password of his/her account.
Postconditions:	System sends him/her a mail including his/her password texted to retrieve it again to login to the system.
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).
Standard Process (Main Success Scenario):	1- System sends user mail texted on his/her mail to retrieve his password to enter his/her profile (System retrieves the password from signup database).
Alternative Processes (Alternative/Other Scenarios):	1'- System sends password on mail but he couldn't retrieve it as he/she never own this mail and prevent unauthorized user to enter the platform. (Providing security non-functional requirements attribute).

ID Number – Name:	Forgot password.
Preconditions:	User doesn't remember his password of his/her account.
Postconditions:	System sends him/her a mail including his/her password texted to retrieve it again to login to the system.
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).
Standard Process (Main Success Scenario):	1- System sends user mail texted on his/her mail to retrieve his password to enter his/her profile (System retrieves the password from signup database).
Alternative Processes (Alternative/Other Scenarios):	1'- System sends password on mail but he couldn't retrieve it as he/she never own this mail and prevent unauthorized user to enter the platform. (Providing security non-functional requirements attribute).

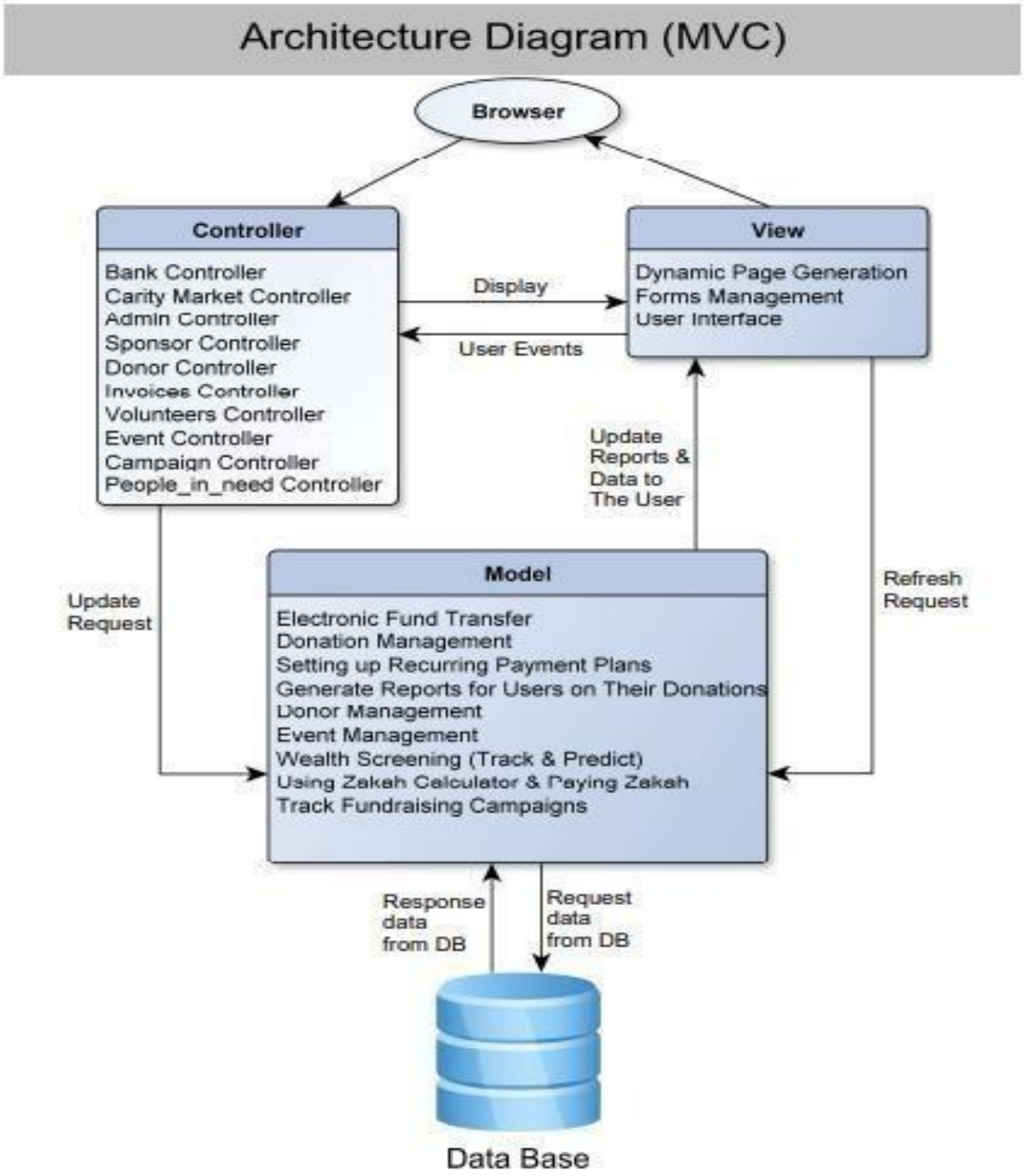
ID Number – Name:	Attend an event.
Preconditions:	User wants to attend an event and start to reserve a ticket.
Postconditions:	The user reserved on the event successfully and user waiting for the start date of the event to attend.
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).
Standard Process (Main Success Scenario):	1- Sponsor reserves on the event. 2- Volunteer registered to the volunteer's event form and wait for admin confirmation to attend and organize the event. 3- Donor reserves on the event.
Alternative Processes (Alternative/Other Scenarios):	1'- Sponsor already carry the event so his/her participation will be without a message of confirmation from admin. 2'- The maximum number of volunteers determined by the admin was reached .so, the system reject the volunteer's registration. 3'- The maximum number of attendances determined by the admin was reached .so, the system reject the donor's registration.

Database actor use cases



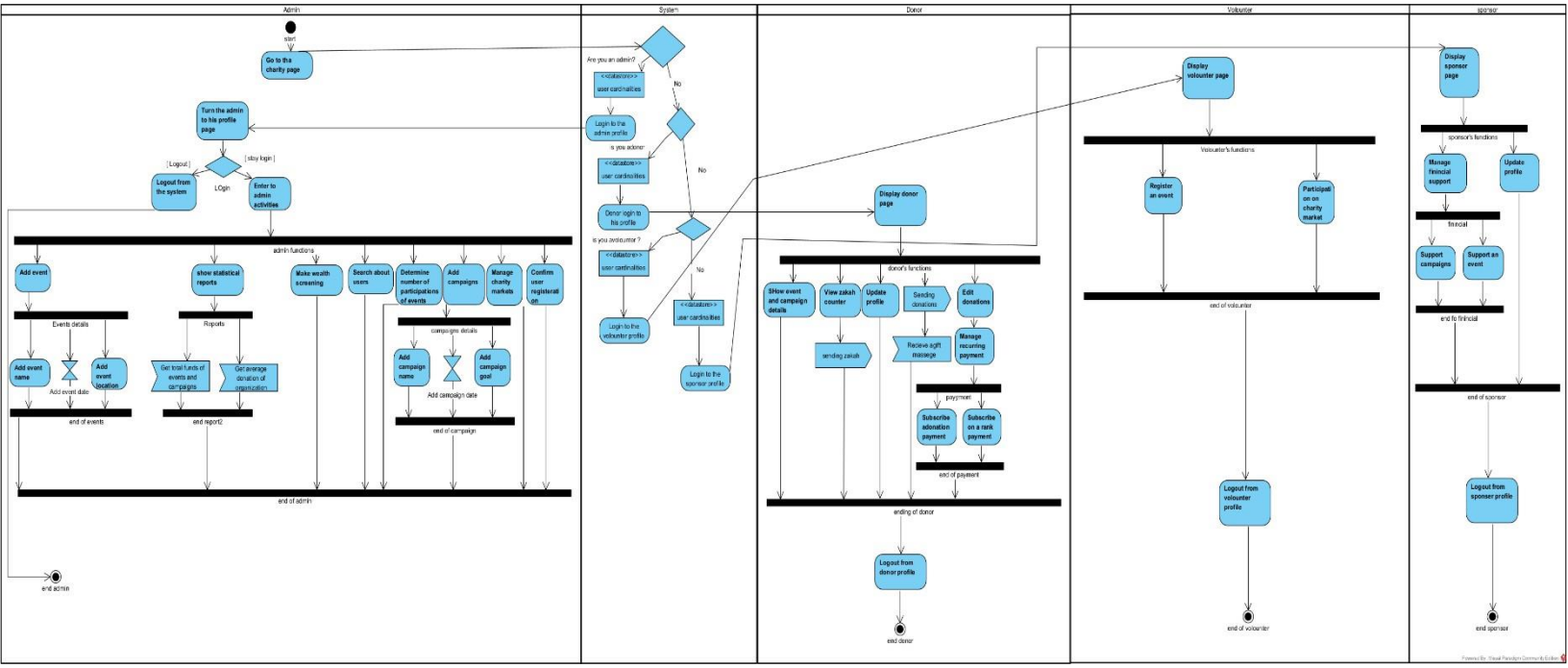
Database actor use cases : -

ID Number – Name:	Distribute donations on needed people.
Preconditions:	System passes the reports of donation to the database and database starts to distribute the donations that come on people on need equally.
Postconditions:	Donations distributed successfully for the people in need after receiving donation of certain volunteer from donor's report
Actor(s) / Initiator(s):	User (Admin, volunteer, sponsor, and donor).

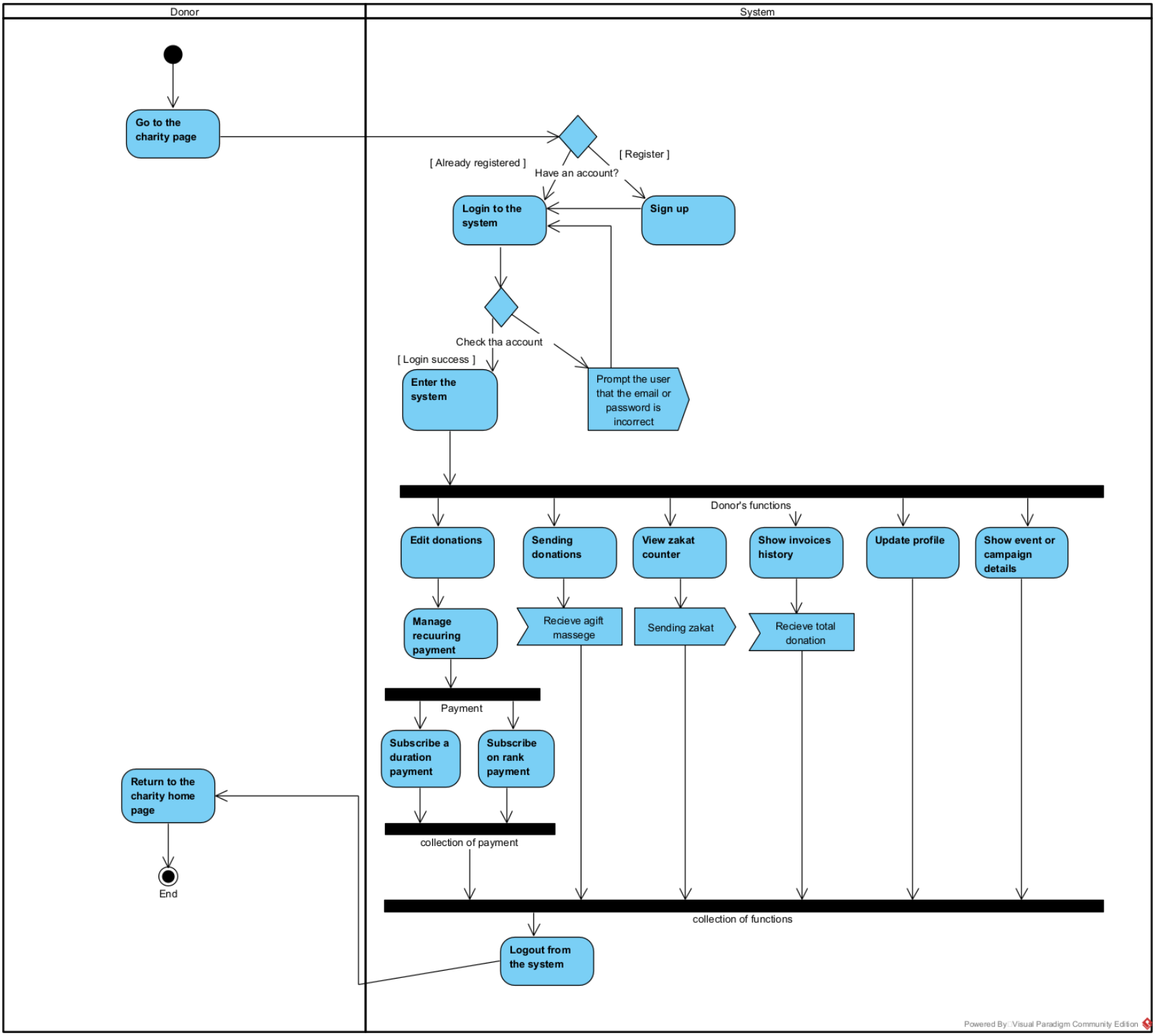

Architecture Diagram (MVC)
Browser
Controller
Bank Controller
Carity Market Controller
Admin Controller
Sponsor Controller
Donor Controller
Invoices Controller
Volunteers Controller
Event Controller
Campaign Controller
People_in_need Controller
View
Dynamic Page Generation
Forms Management
User Interface
Display
User Events
Update Reports & Data to The User
Model
Electronic Fund Transfer
Donation Management
Setting up Recurring Payment Plans
Generate Reports for Users on Their Donations
Donor Management
Event Management
Wealth Screening (Track & Predict)
Using Zakah Calculator & Paying Zakah
Track Fundraising Campaigns
Update Request
Refresh Request
Response data from DB
Request data from DB
Data Base

Activity diagrams

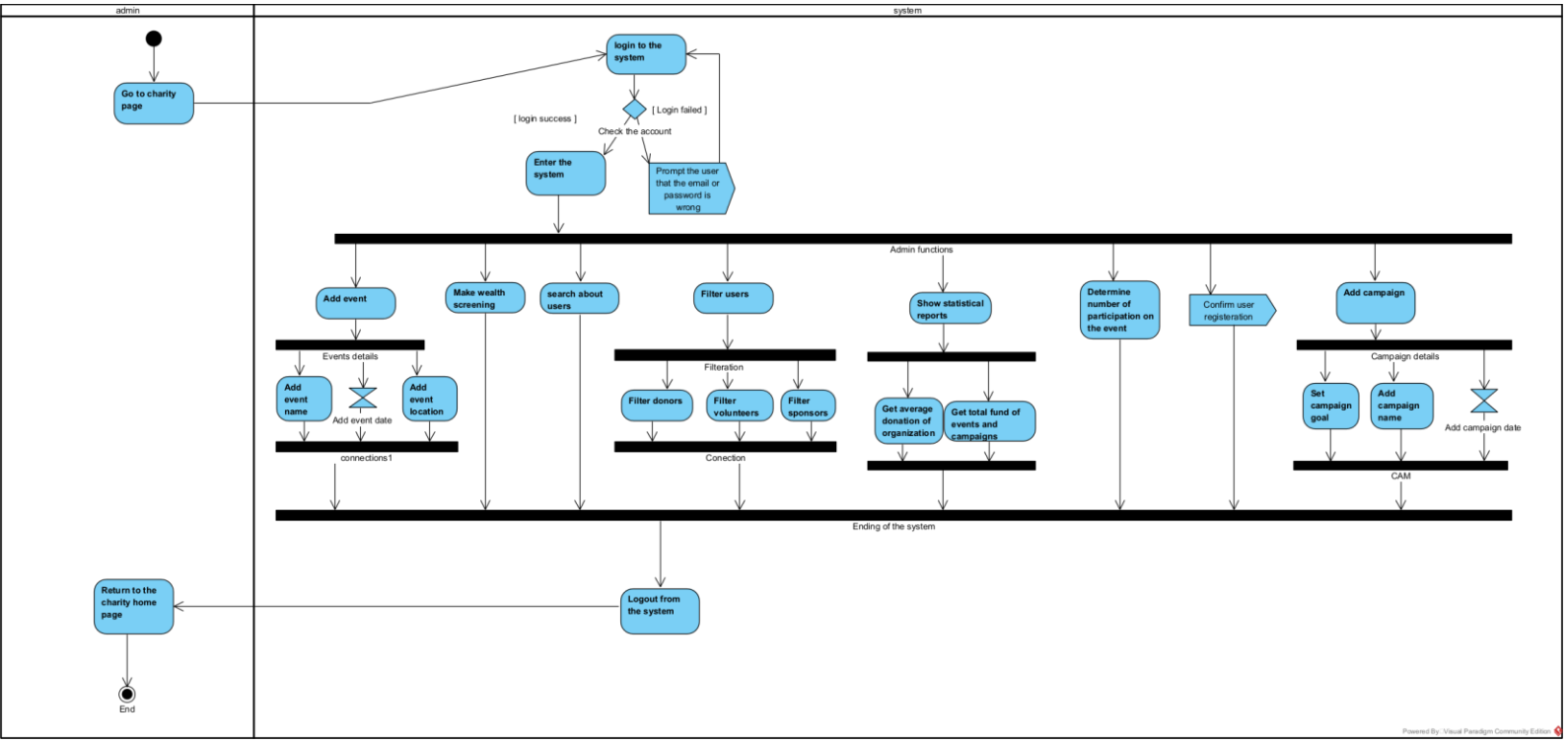
Whole system activity diagram



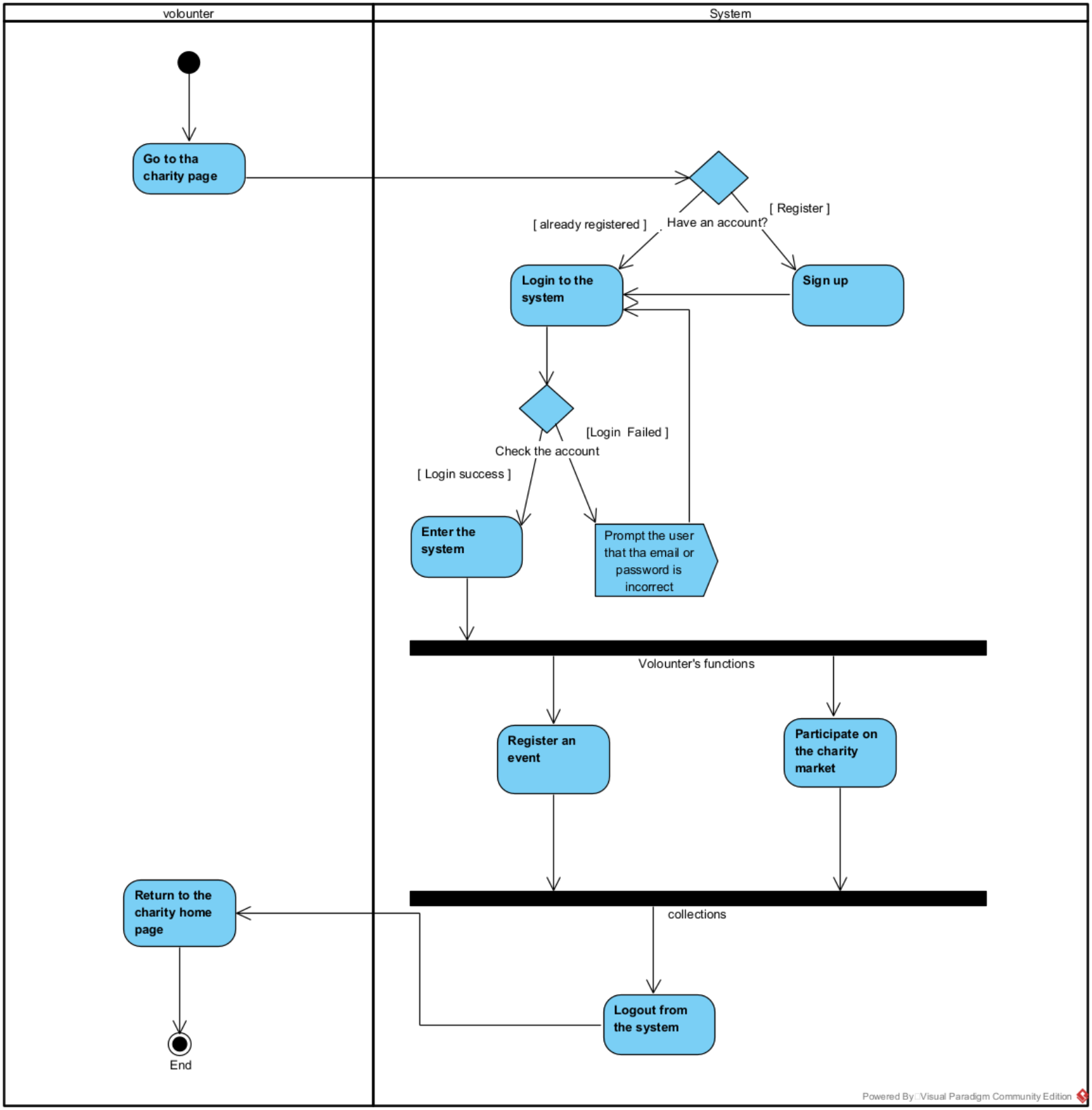
Donor activity diagram



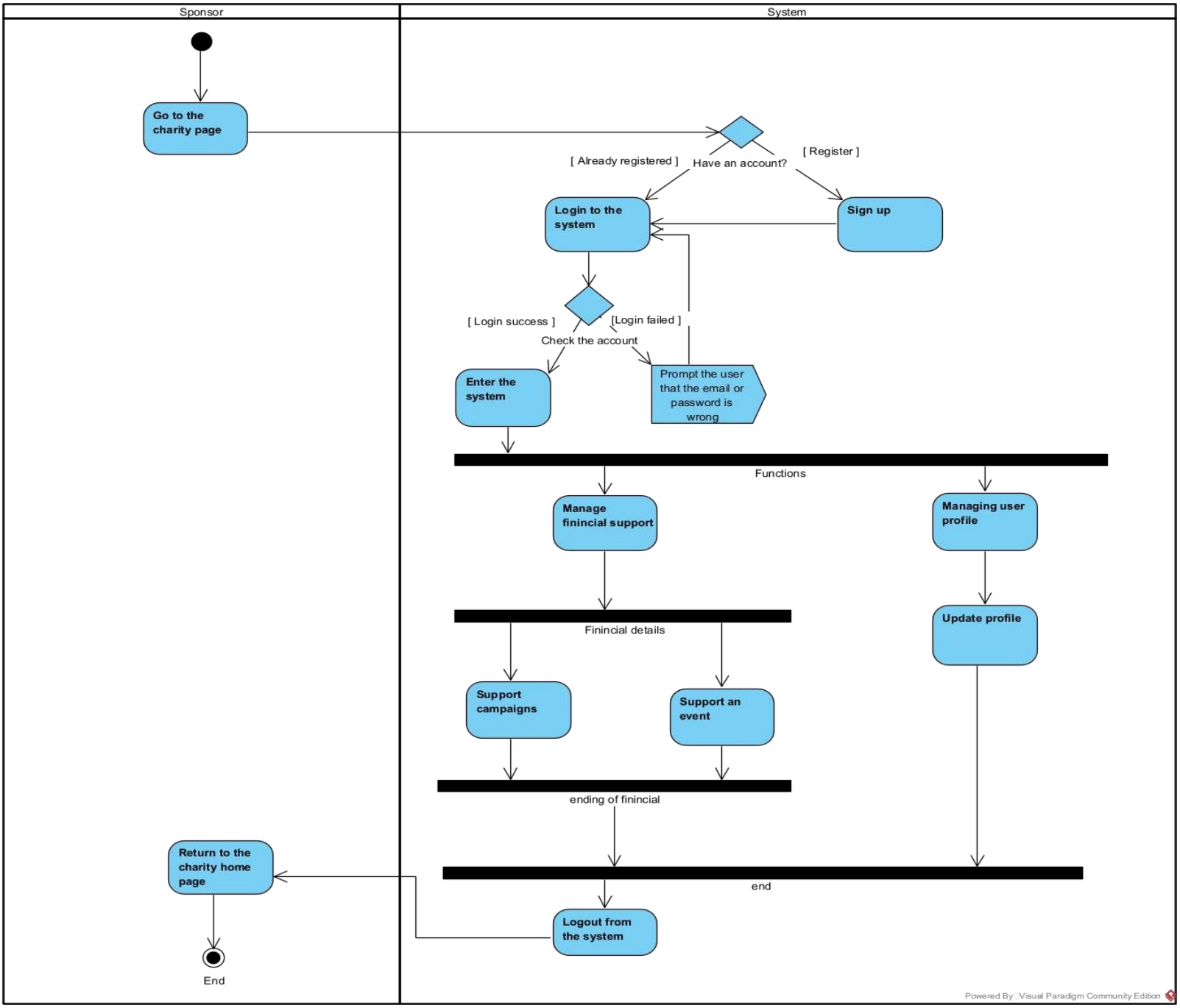
Admin activity diagram



Volunteer activity diagram

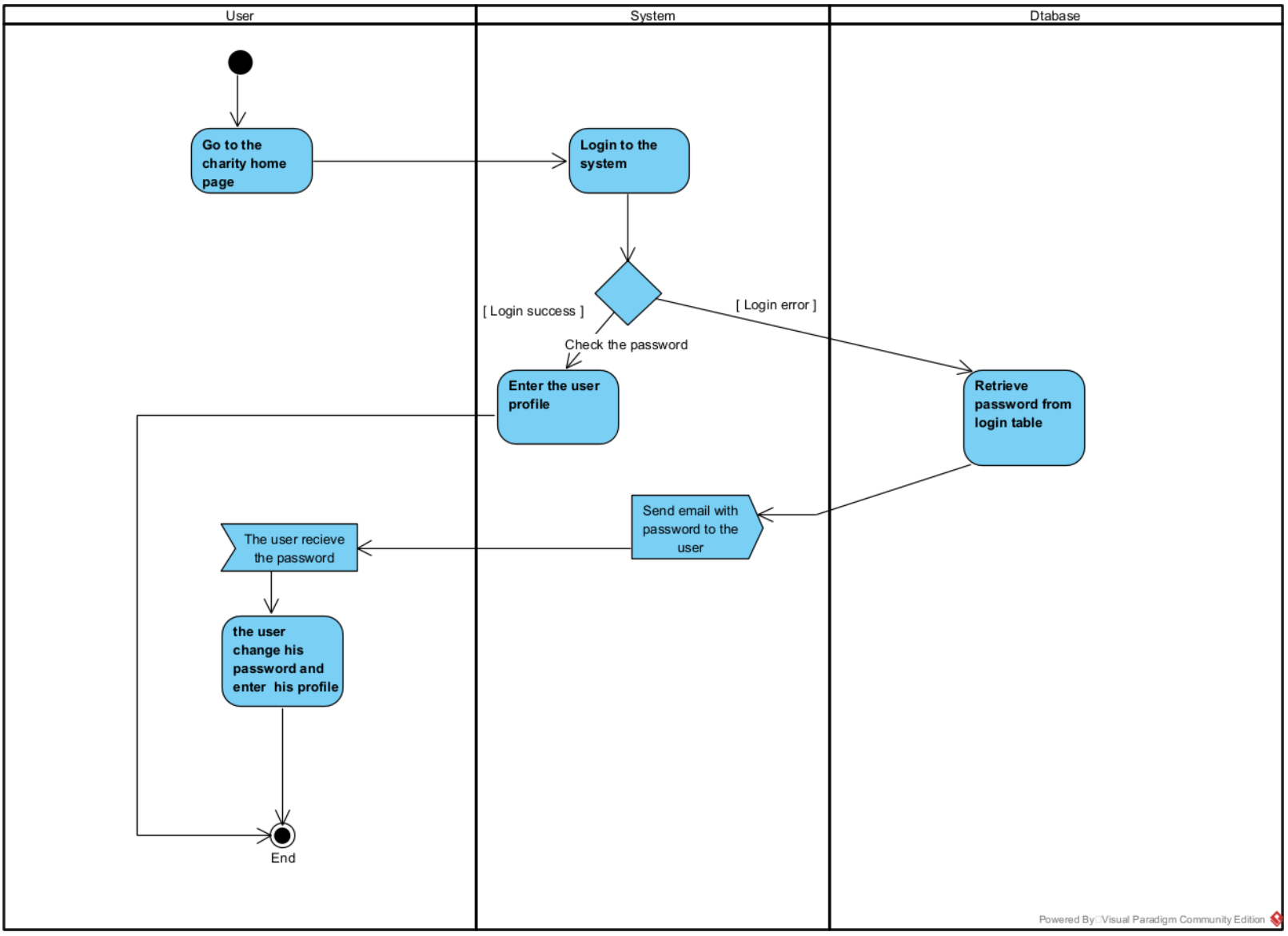


Sponsor activity diagram



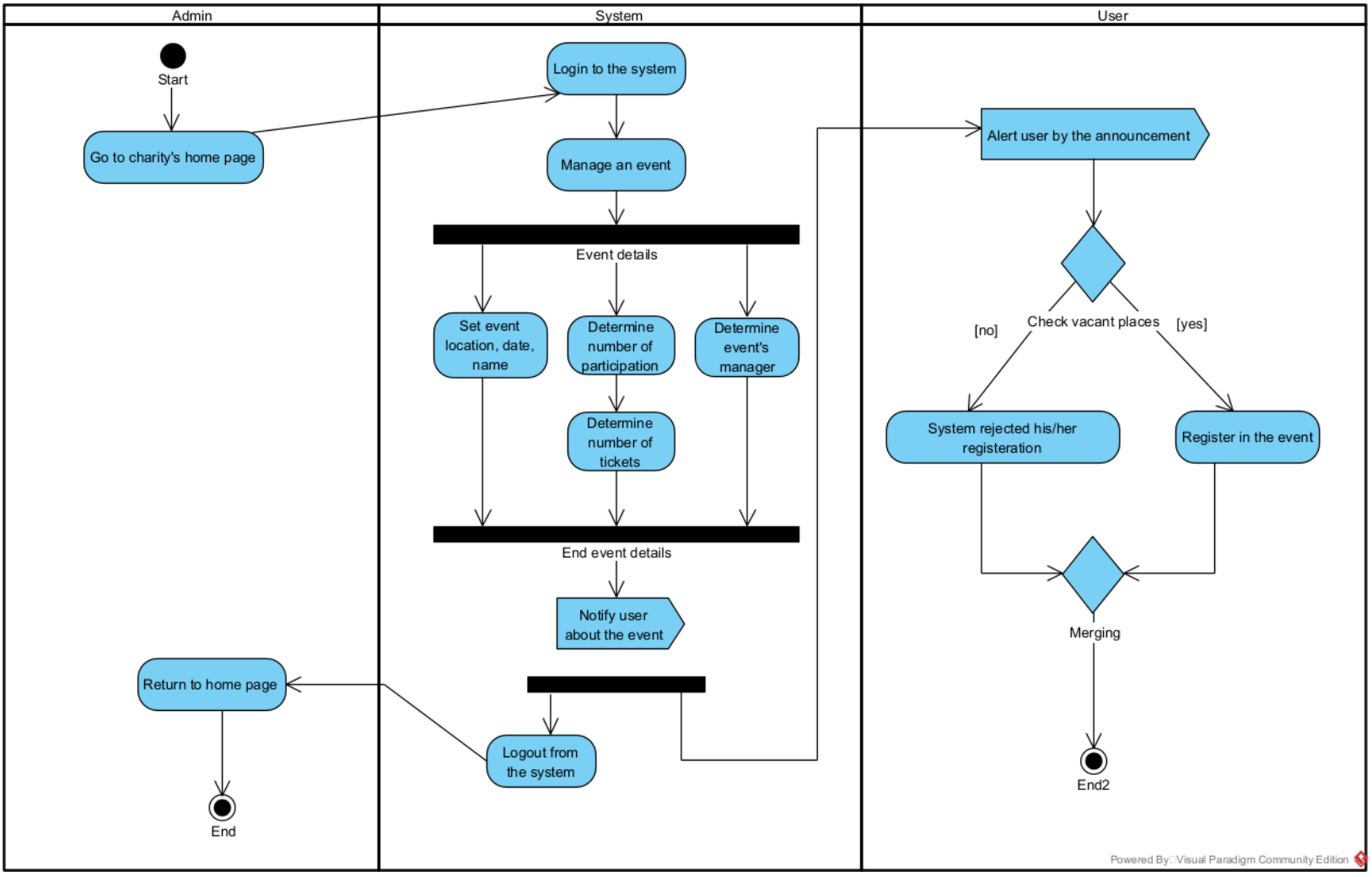
Forgot password activity diagram

Preconditions: - User forgets his password and need to retrieve it again.
Postconditions: - Admin sends the password to user texted on his mail.



Event participation activity diagram

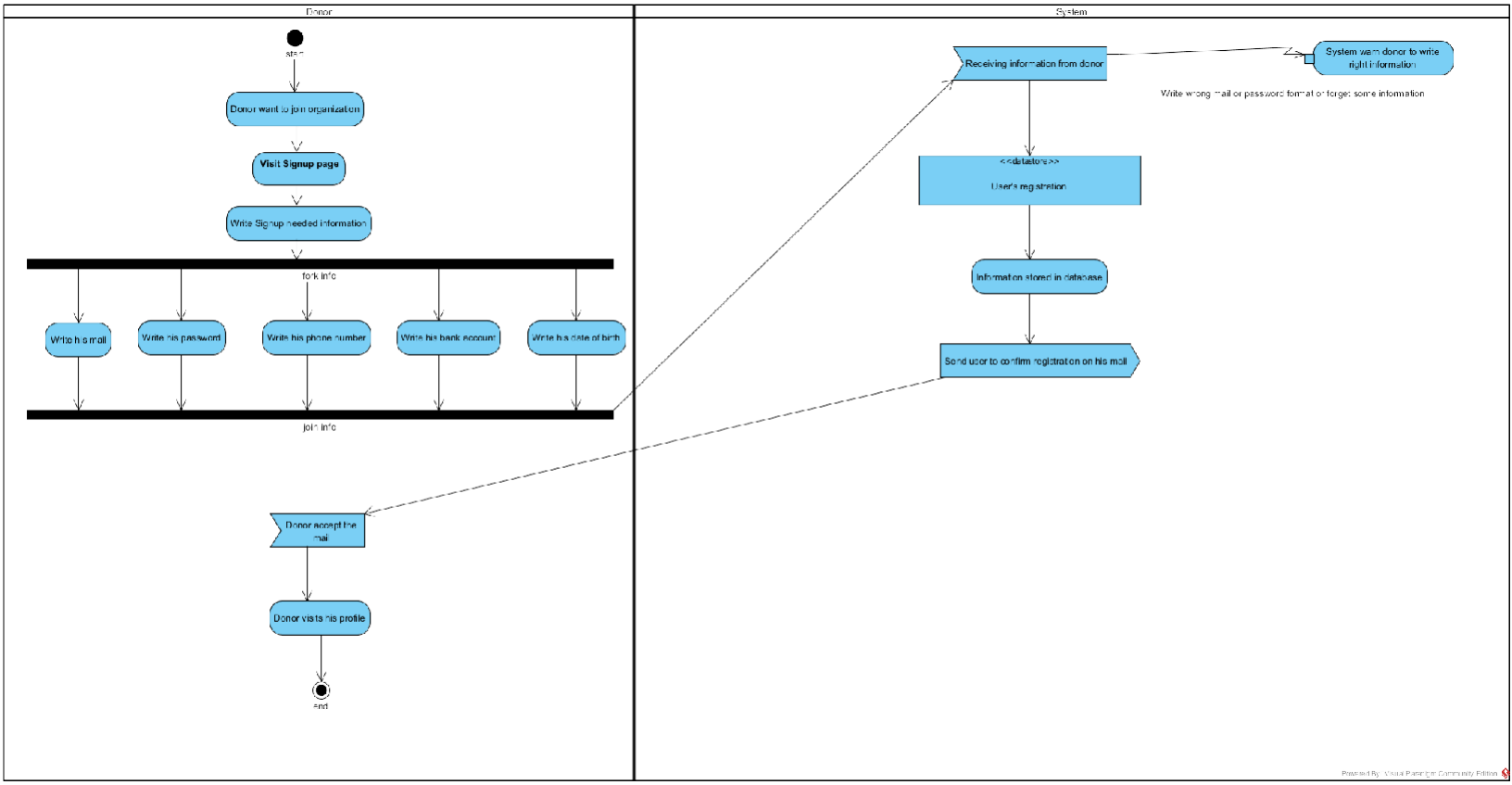
Preconditions: - Admin wants to know the number and percentage of total event's participation.
Postconditions: - Admin knows the number and percentage of participation and start take an action according to event handling.



Donor Signup

Preconditions: - Donor wants to join charity organization.
Postconditions: -

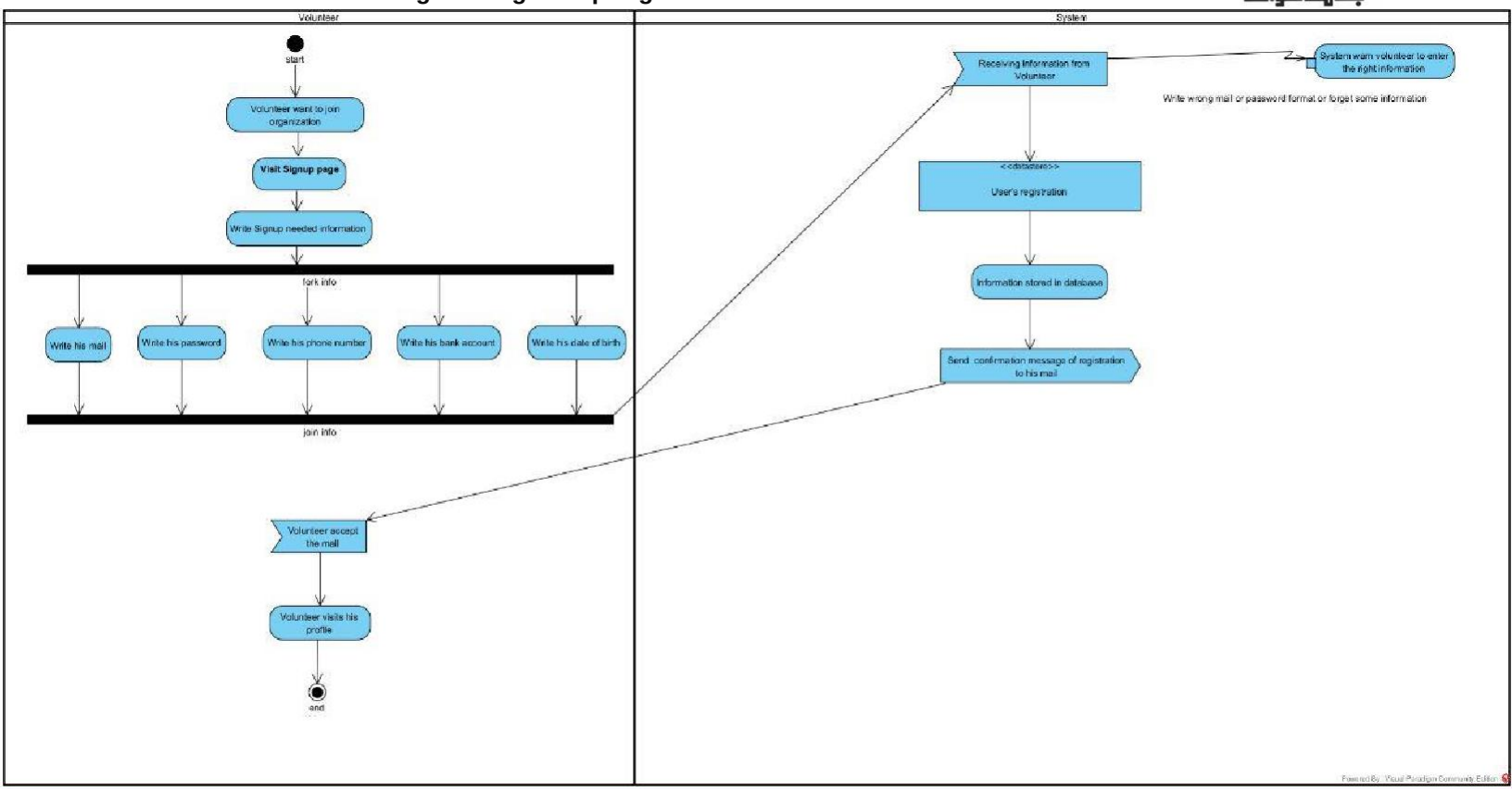
1- Donor joins our organization after admin approval.
2- Donor get rejected or deleted by the admin.



Volunteer Signup

Preconditions: - Volunteer wants to join charity organization.
Postconditions: -

3- Volunteers join our organization after admin approval.
4- Volunteer get rejected or deleted by the admin.



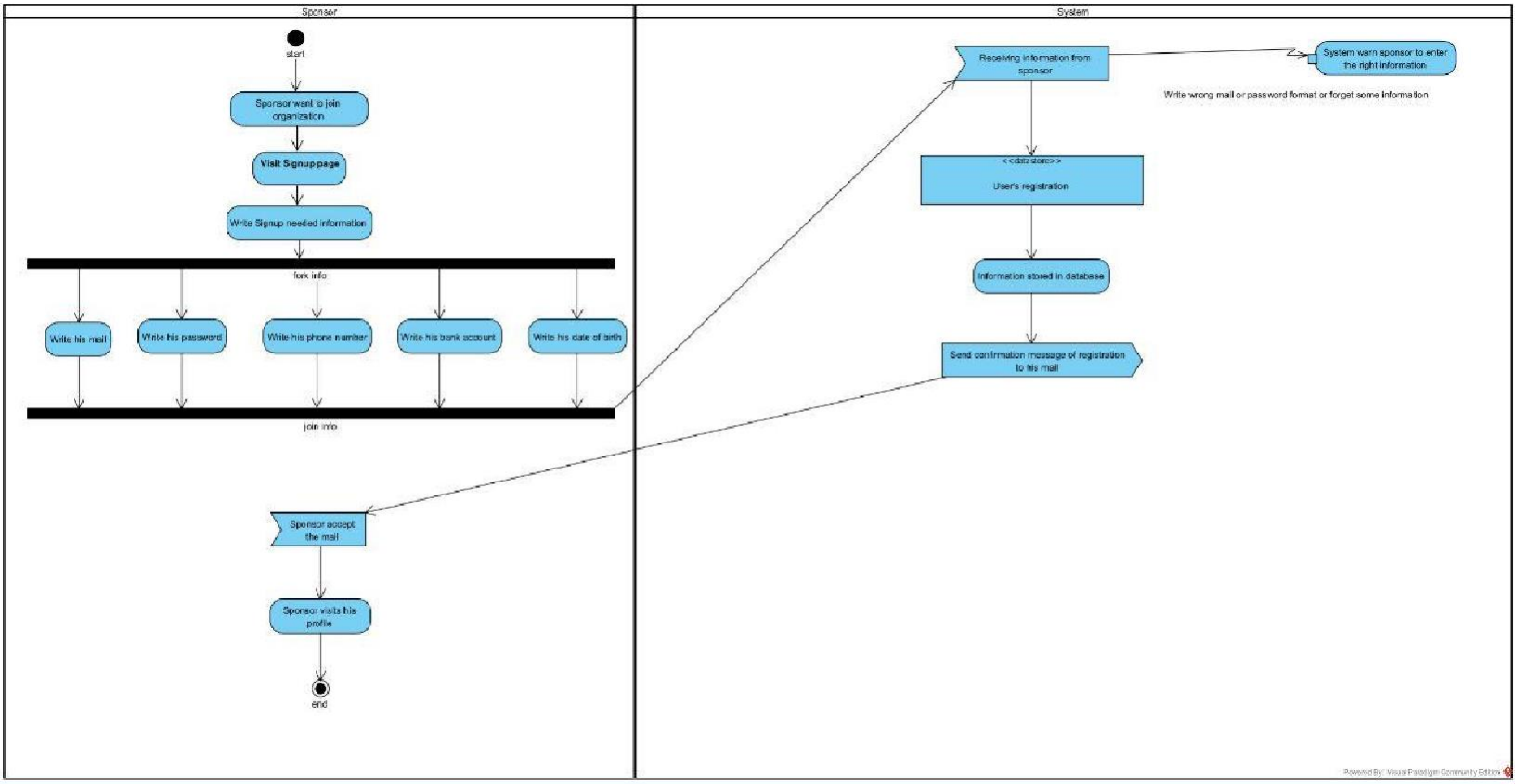
Sponsor Signup

Preconditions: - Sponsor wants to join charity organization.

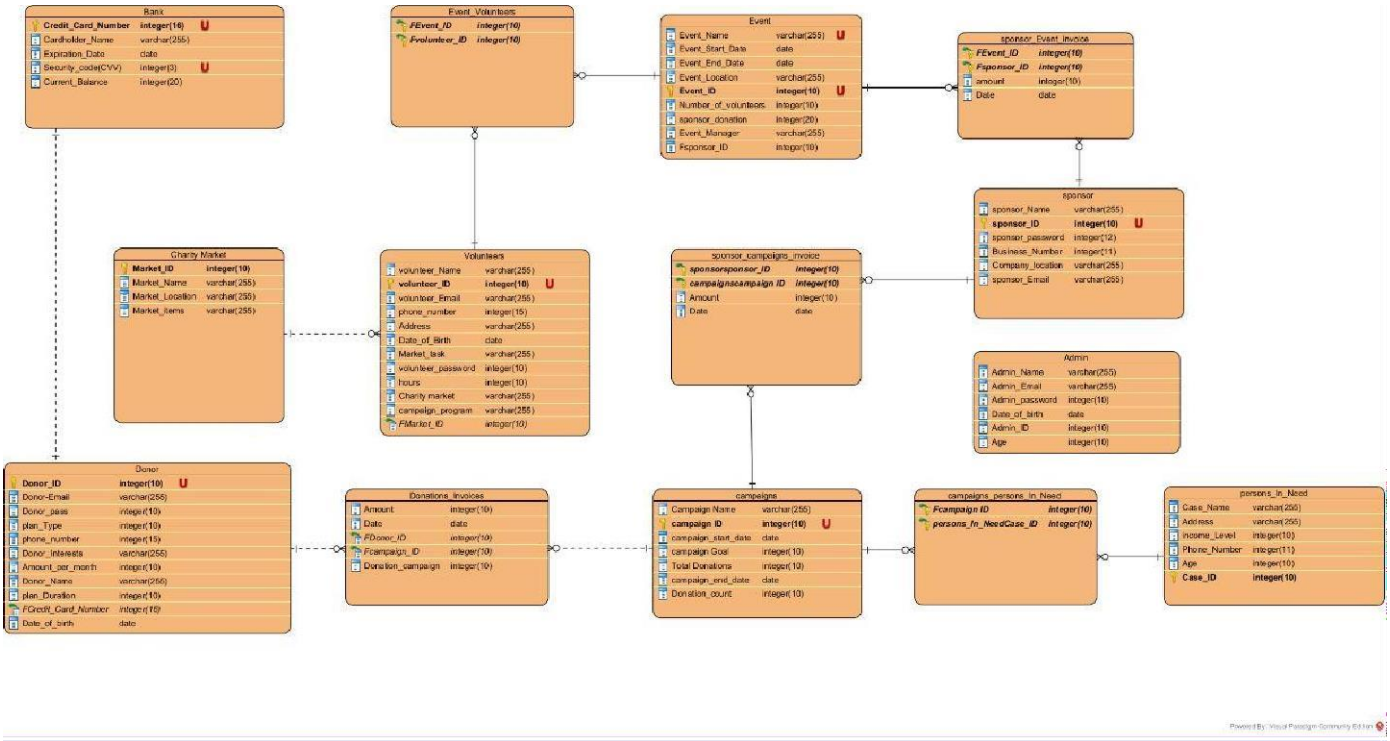
Postconditions: -

5- Sponsor joins our organization after admin approval.

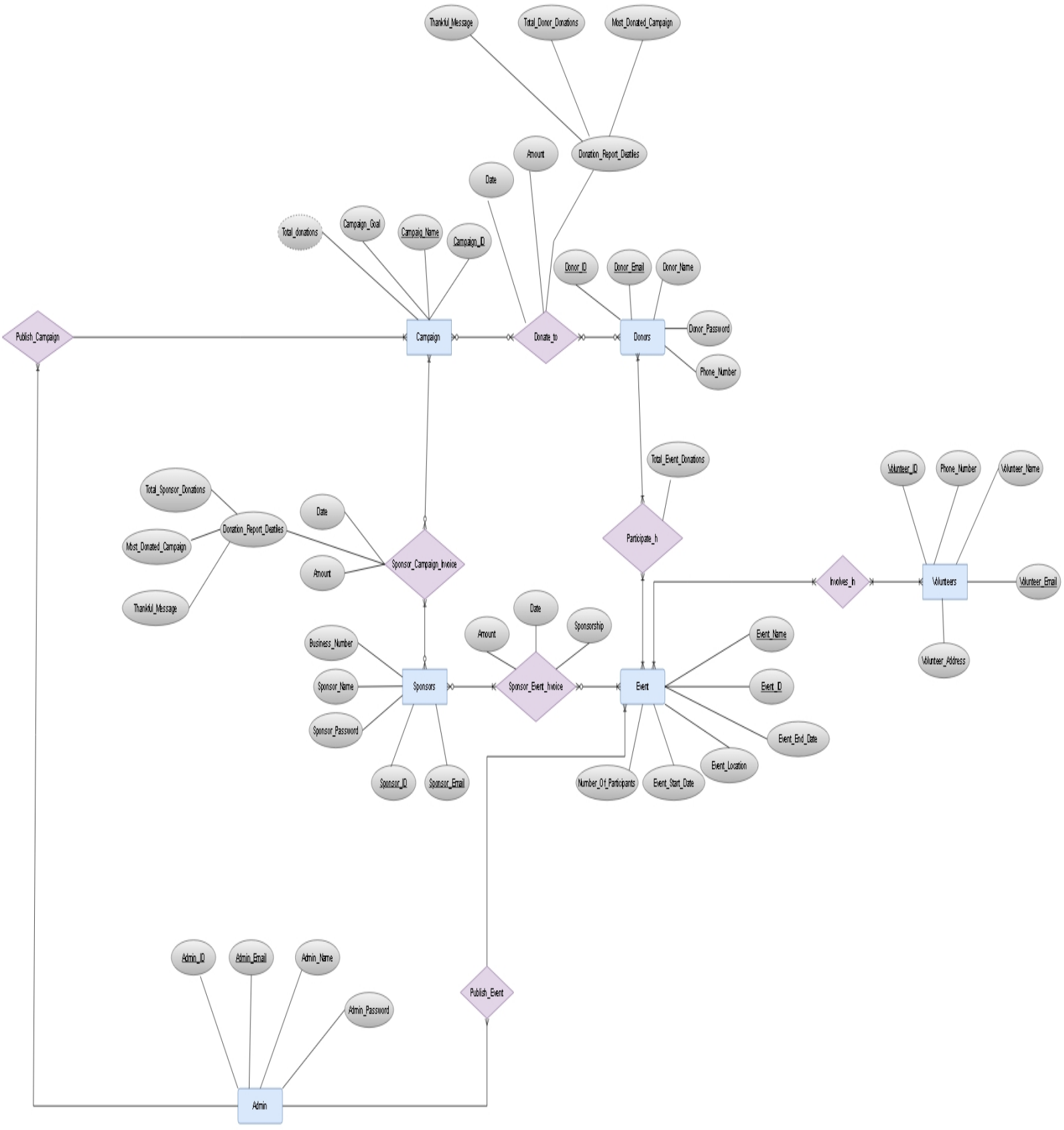
6- Sponsor get rejected or deleted by the admin.



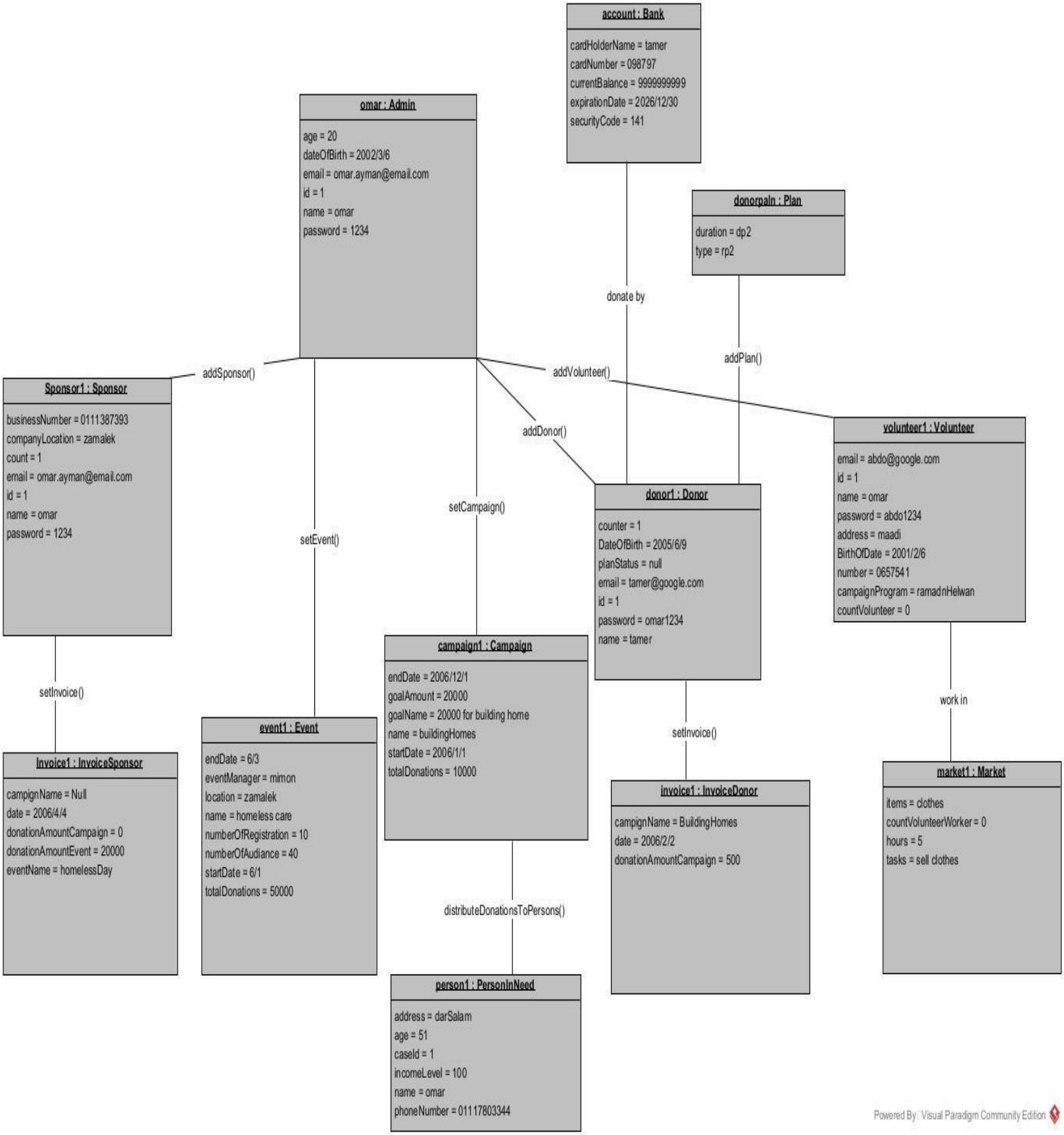
Database table



ERD diagram


Thankful_Message
Total_Donor_Donations
Most_Donated_Campaign
Donation_Report_Deatiles
Amount
Date
Total_donations
Campaign_Goal
Campaig_Name
Campaign_ID
Donor_ID
Donor_Email
Donor_Name
Publish_Campaign
Campaign
Donate_to
Donors
Donor_Password
Phone_Number
Total_Event_Donations
Total_Sponsor_Donations
Donation_Report_Deatiles
Most_Donated_Campaign
Thankful_Message
Date
Amount
Sponsor_Campaign_hvoice
Participate_h
Volunteer_ID
Phone_Number
Volunteer_Name
Involves_h
Volunteers
Volunteer_Email
Volunteer_Address
Business_Number
Sponsor_Name
Sponsor_Password
Sponsor_ID
Sponsor_Email
Sponsors
Amount
Date
Sponsorship
Sponsor_Event_hvoice
Event
Event_Name
Event_ID
Event_End_Date
Event_Location
Event_Start_Date
Number_Of_Participants
Admin_ID
Admin_Email
Admin_Name
Admin_Password
Publish_Event
Admin

Object diagram.

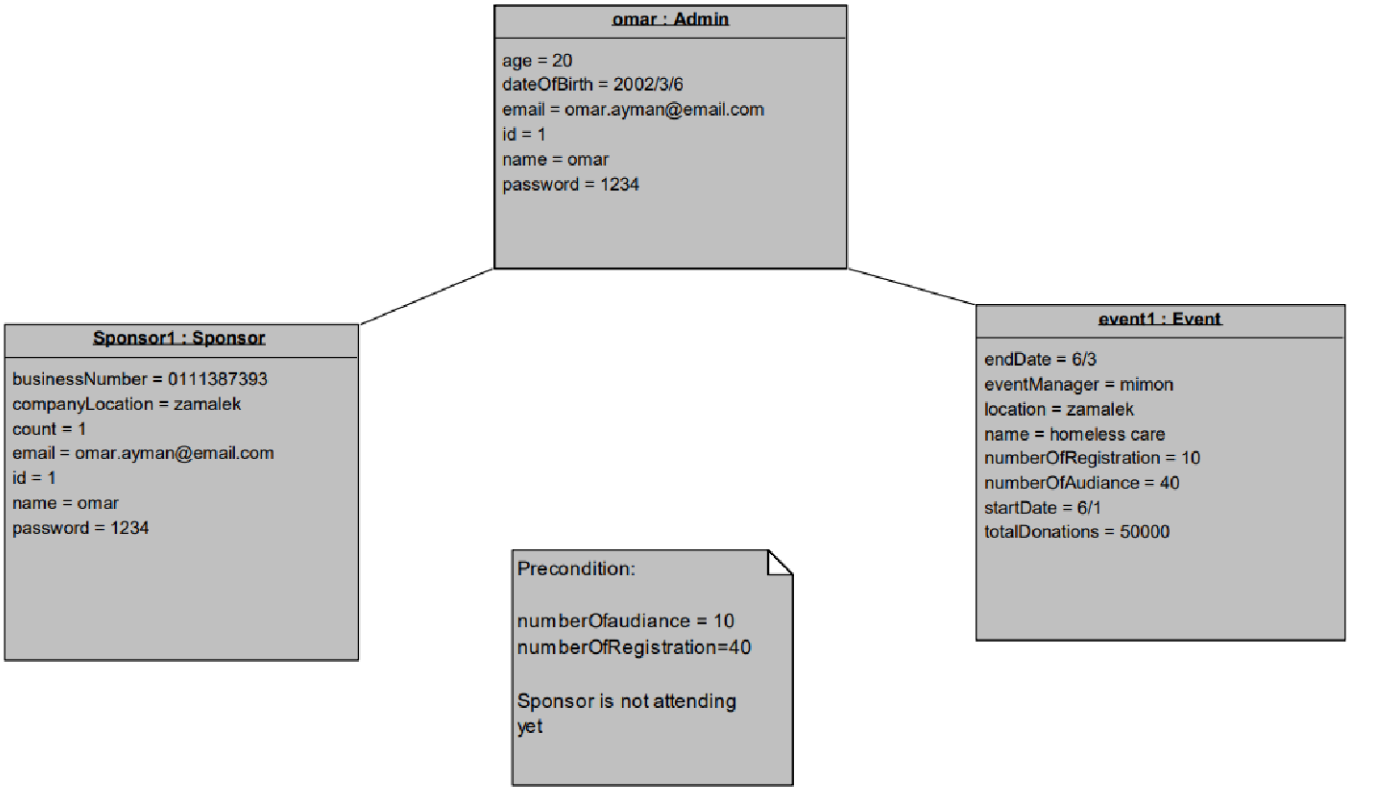


Post-conditions and pre-conditions of object diagram

Post-condition and pre-condition for attend event and check maximum registration: -

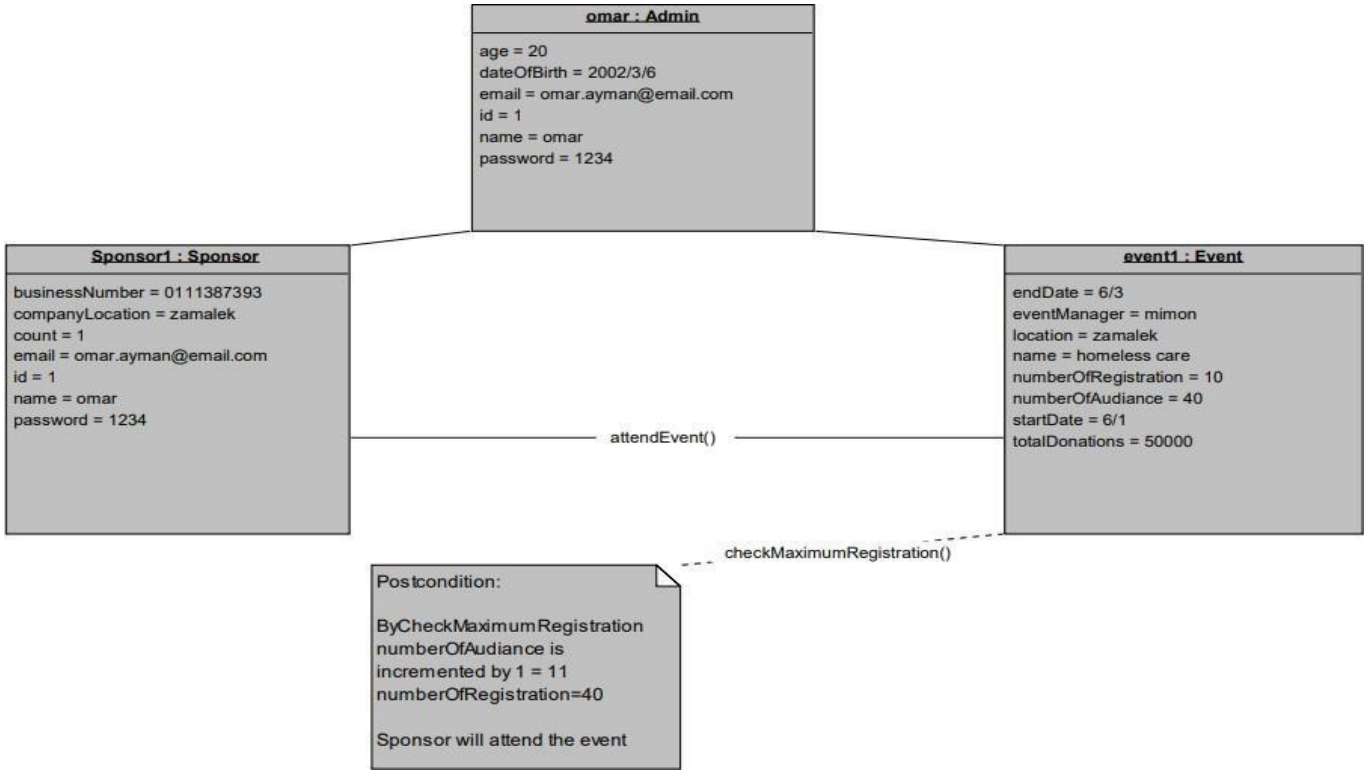
Precondition: -

AttendEvent() and checkMaxRegistration()
Precondition: there is a sponsor and event and the sponsor want to attend the event



Postcondition

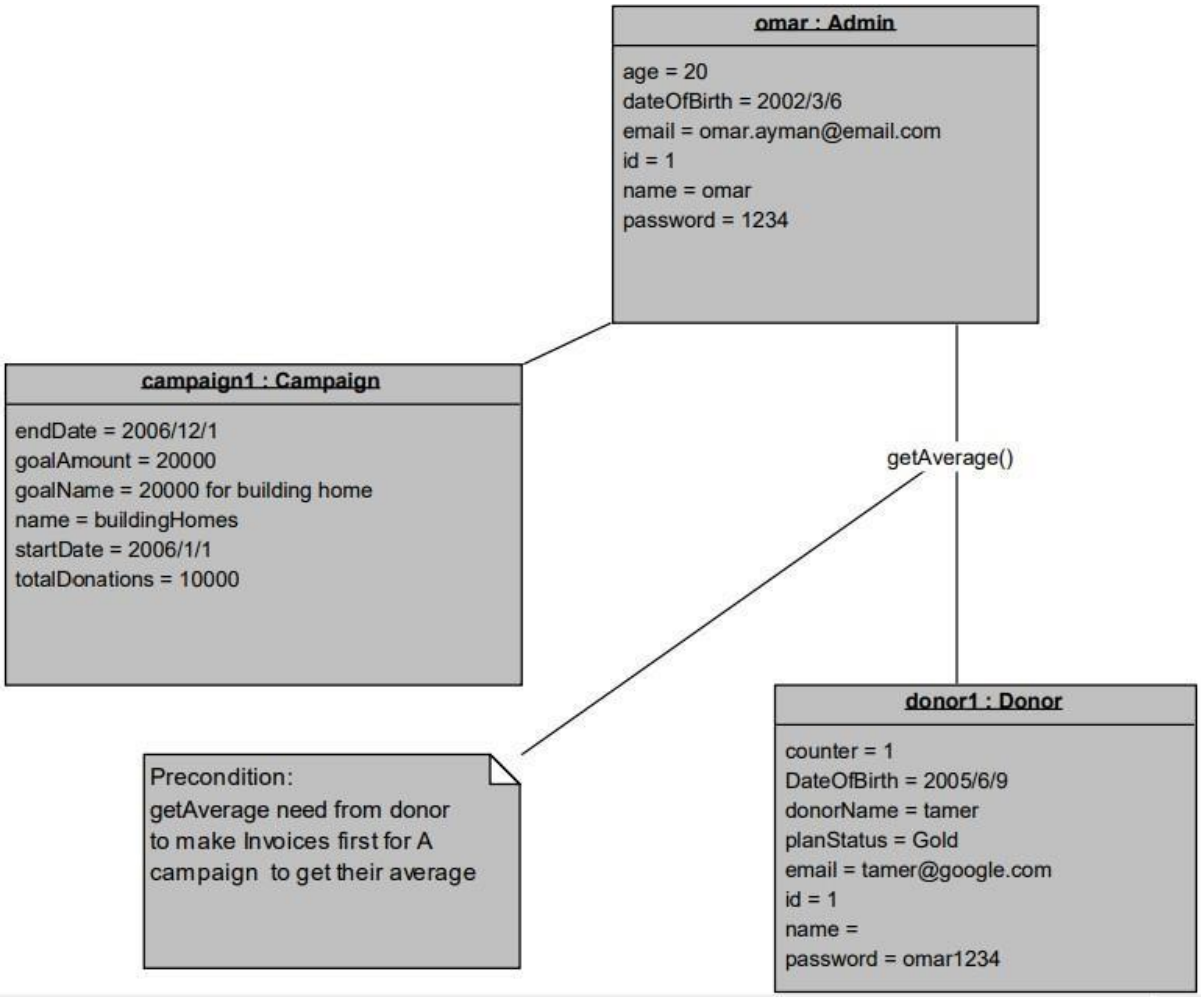
AttendEvent()and checkMaxRegistration()
Postcondition: sponsor attends the event and the number of registrations increases and checks that it's still less than the max number of registrations in that event.



Pre-condition and Post-condition for donor to get average donation: -

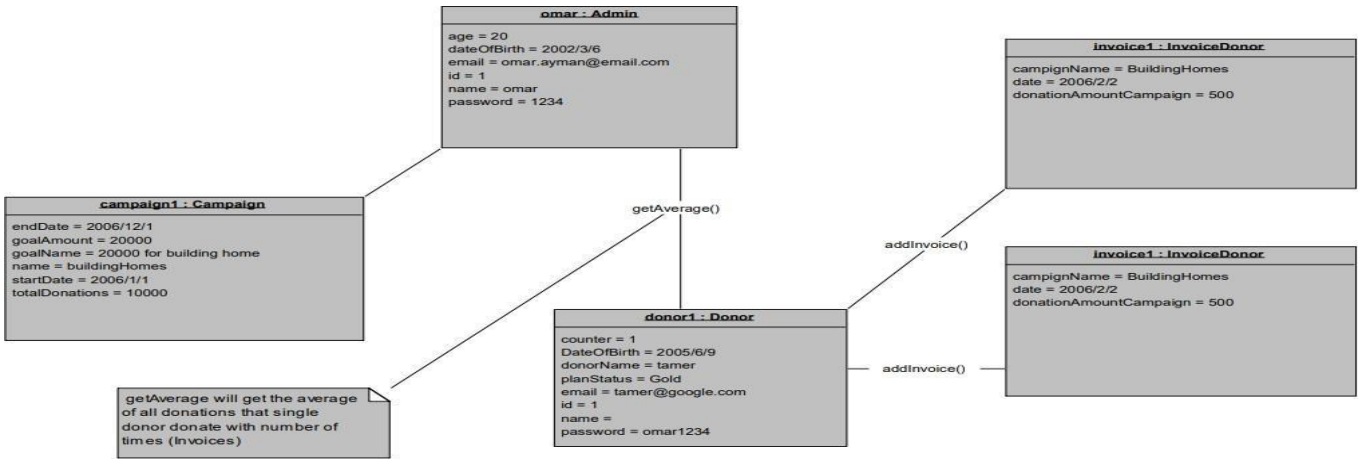
Precondition: -

Get Average ()
precondition: the donor donates to a campaign.



Postconditions: -

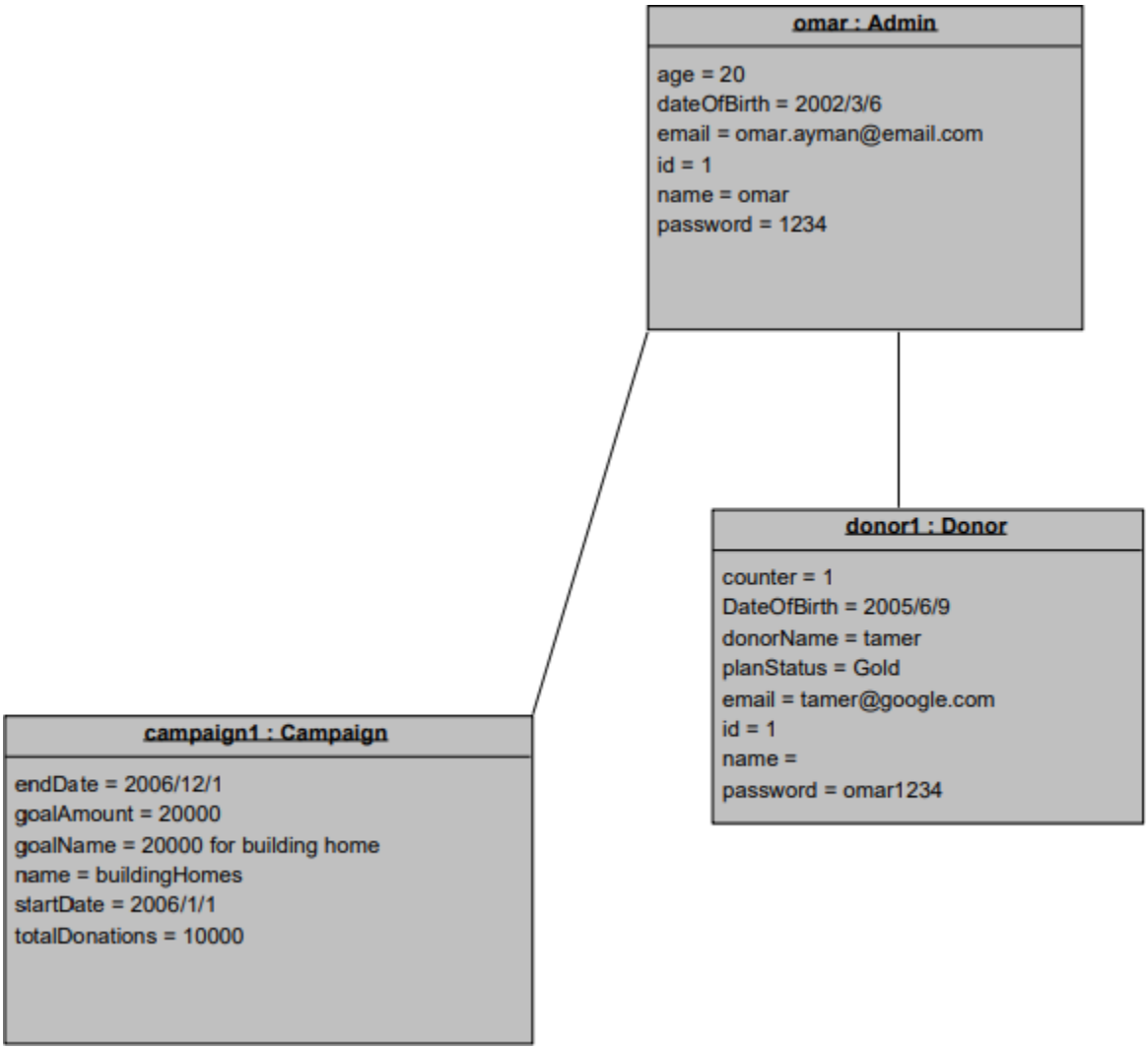
getAverage()
postcondition: donor invoices have been made and put in the history report, the system makes a for loop on all these invoices to get an average for the donations the specific donor can make



Pre-condition and Post-condition for Person in need: -

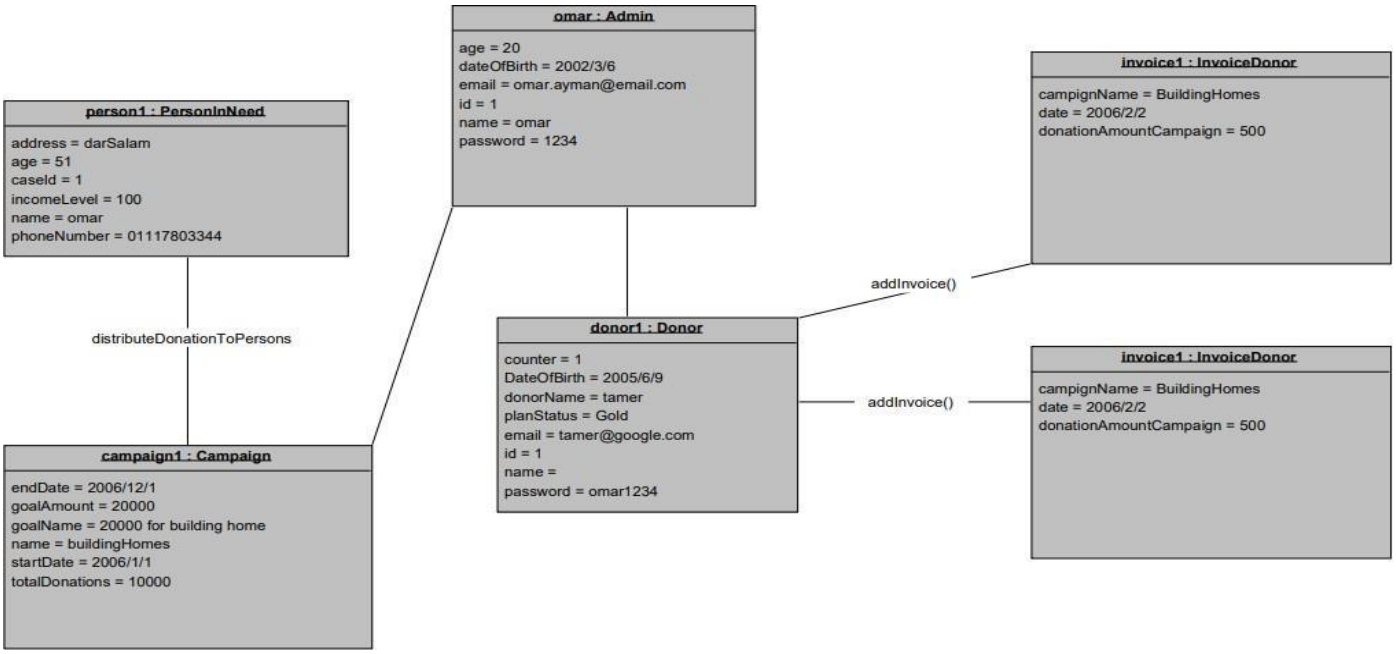
Precondition: -

distributeDonationsToPersons ()
precondition: the donor makes a donation to a campaign.



Post-condition: -

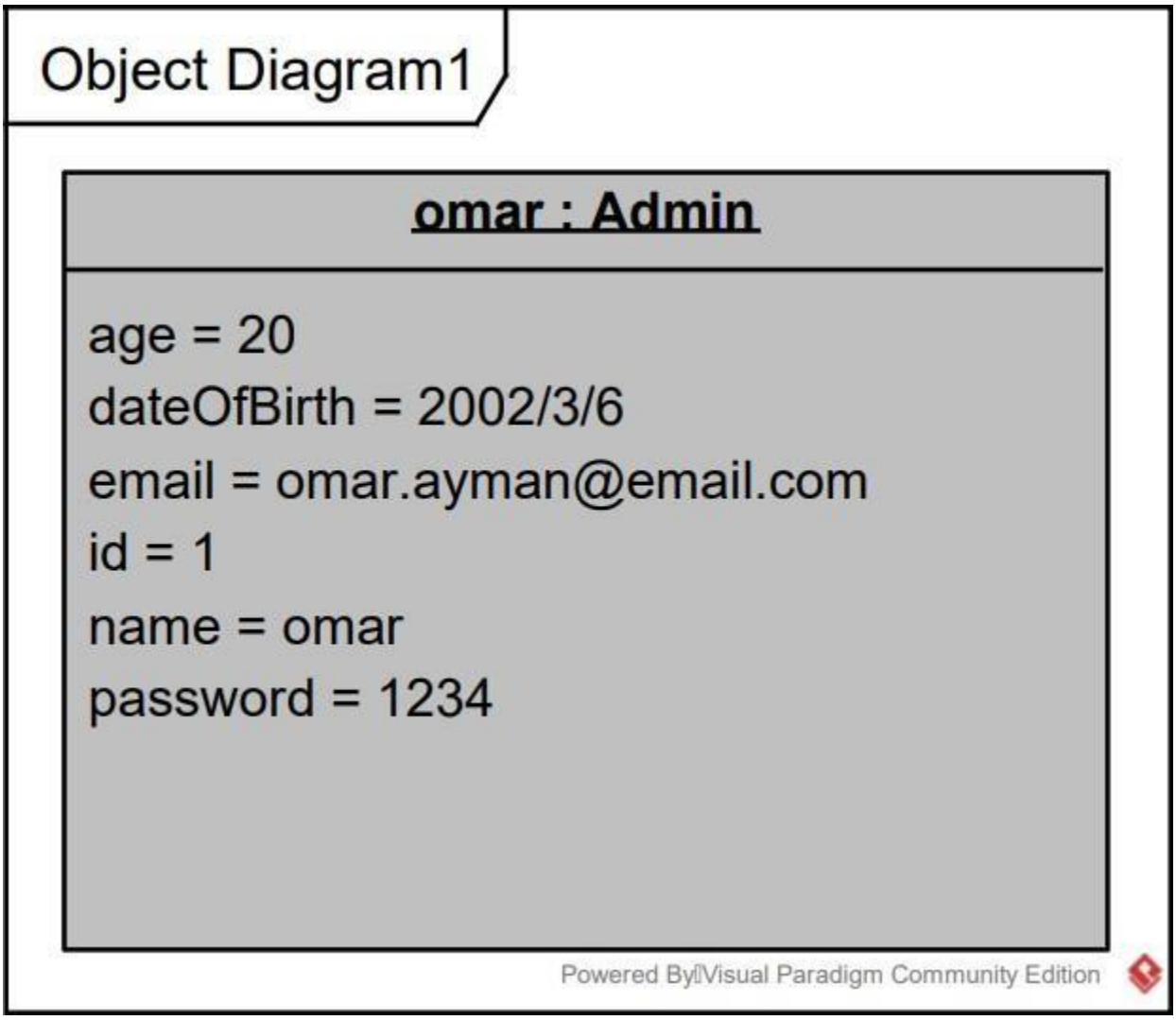
distributeDonationsToPersons ()
postcondition: an invoice has been made for this donation and using the campaign name, the donated money is distributed through an algorithm to the people in need



Post-condition for adding (sponsor, volunteer, event or campaign): -

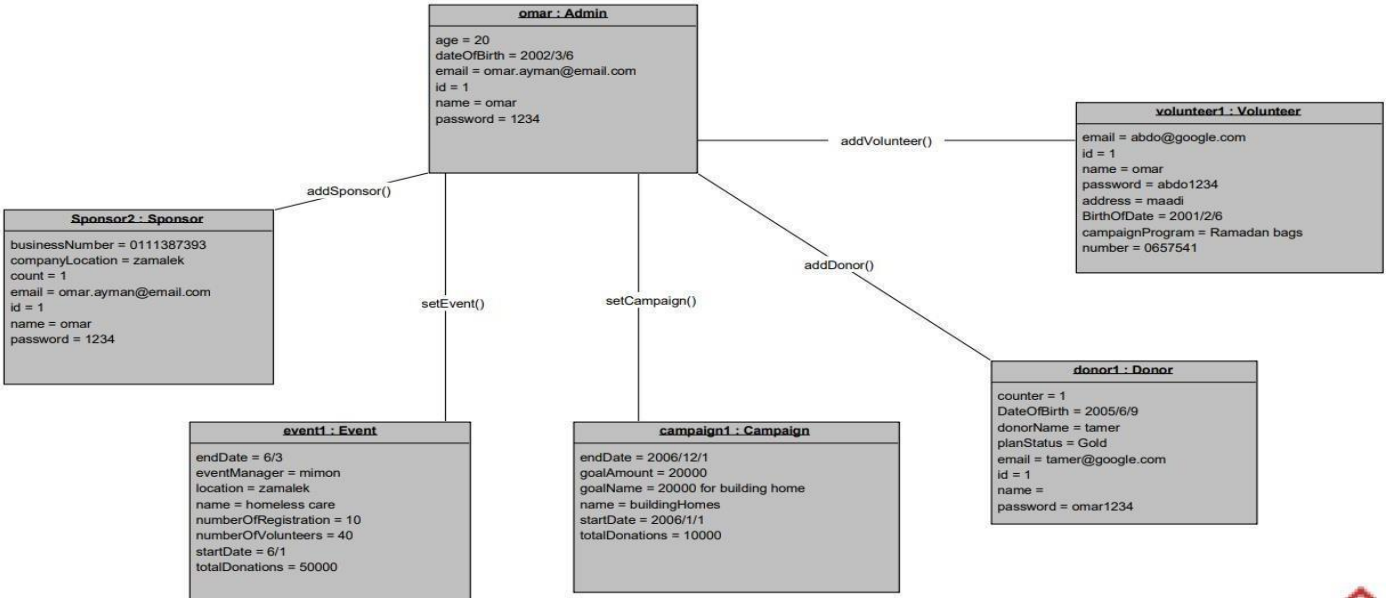
Precondition: -

Add {sponsor-Event – campaign -donor - volunteer}
Precondition: admin is the only one who can access the system and every other actor should make a request to be registered.



Postcondition: -

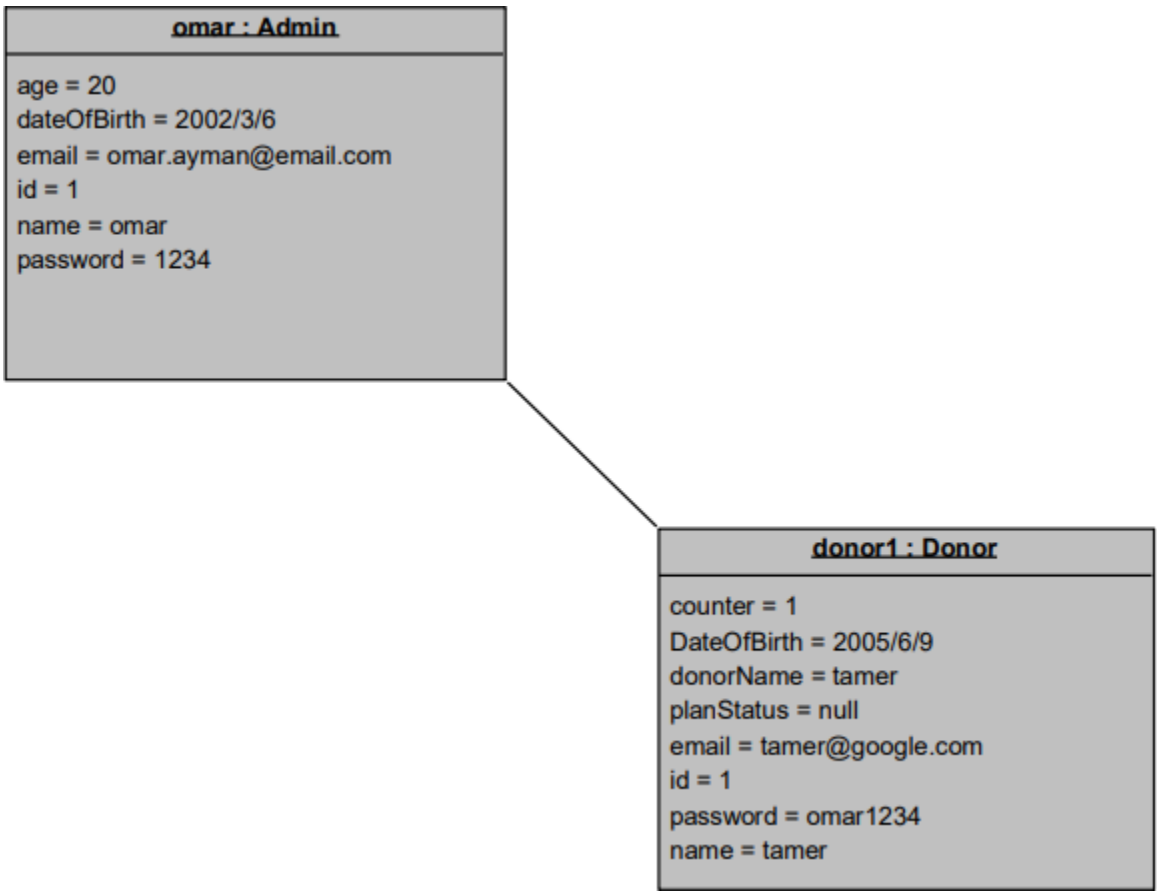
Postcondition: after using add function, the user registered on the system and can use it.



Pre-condition and Post-condition for add plan: -

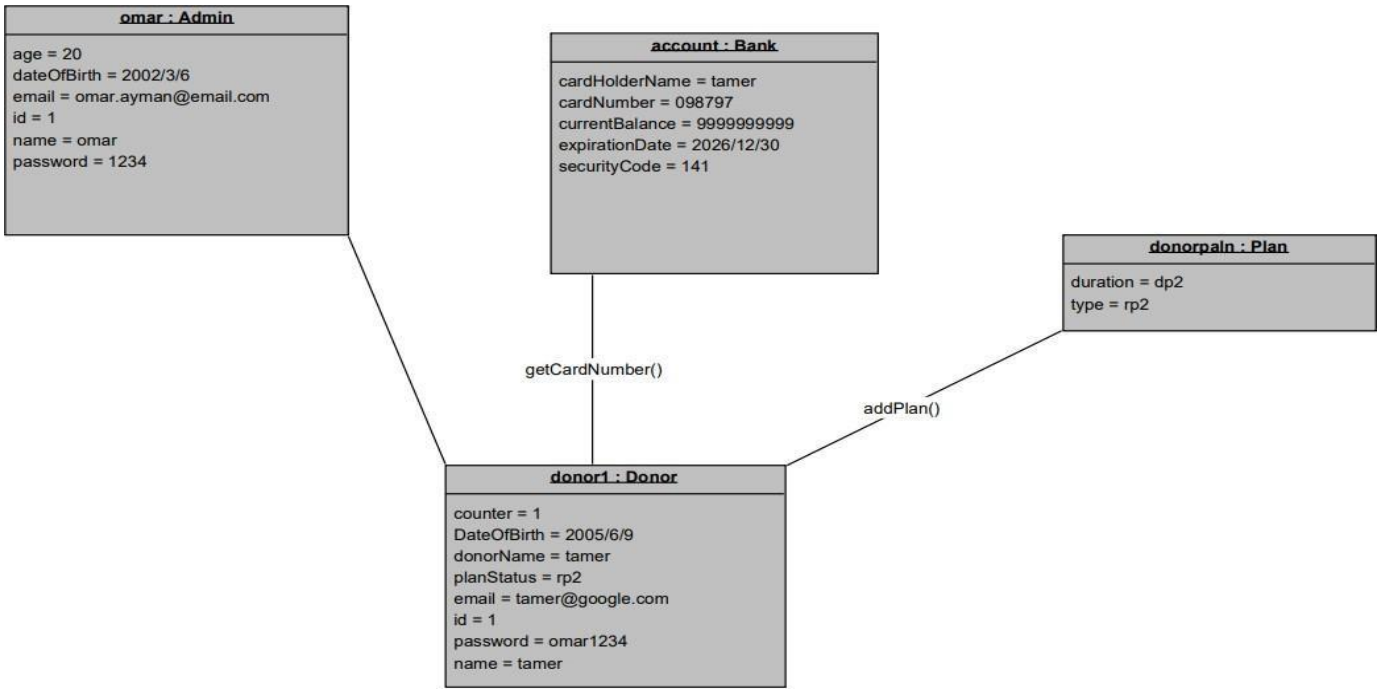
Precondition

Addplan()
Precondition: a donor has been registered into the system



Postcondition

Addplan()
Postcondition: the donor selected a recurring payment with the selected amount and the subscription package.

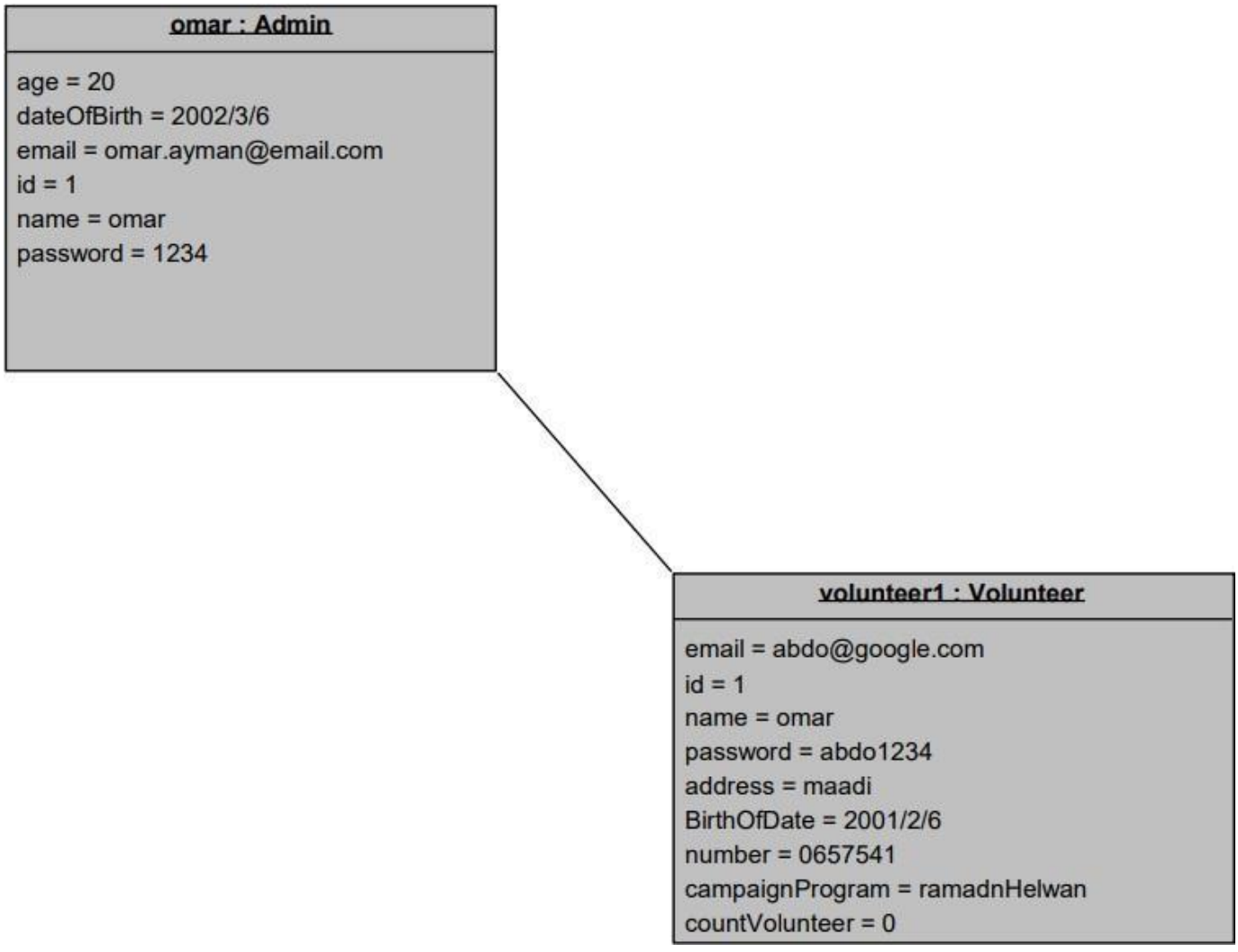


Precondition and Post-condition for work in market: -

Precondition: -

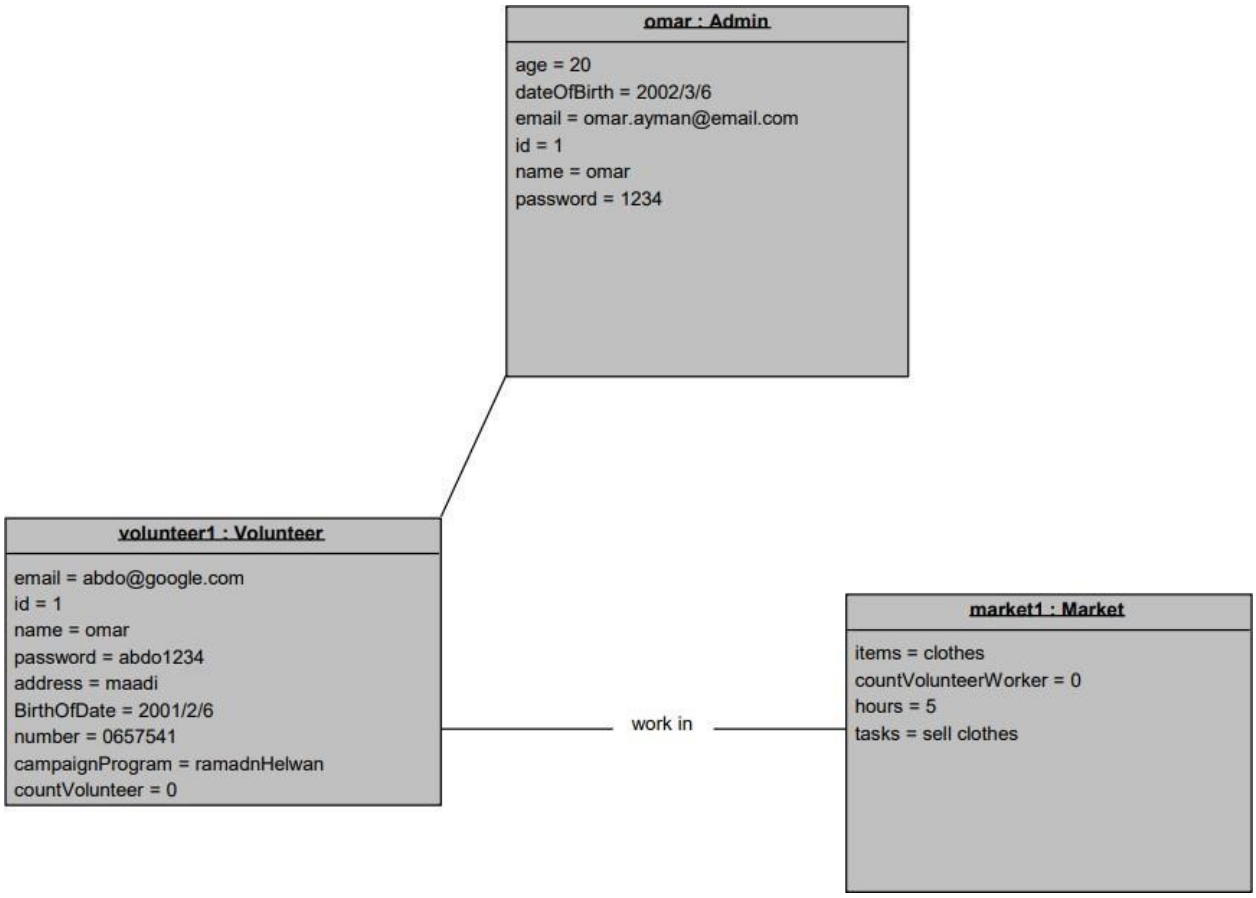
WorkInMarket()
Precondition: an object of volunteer and market has been made

---



Postcondition: -

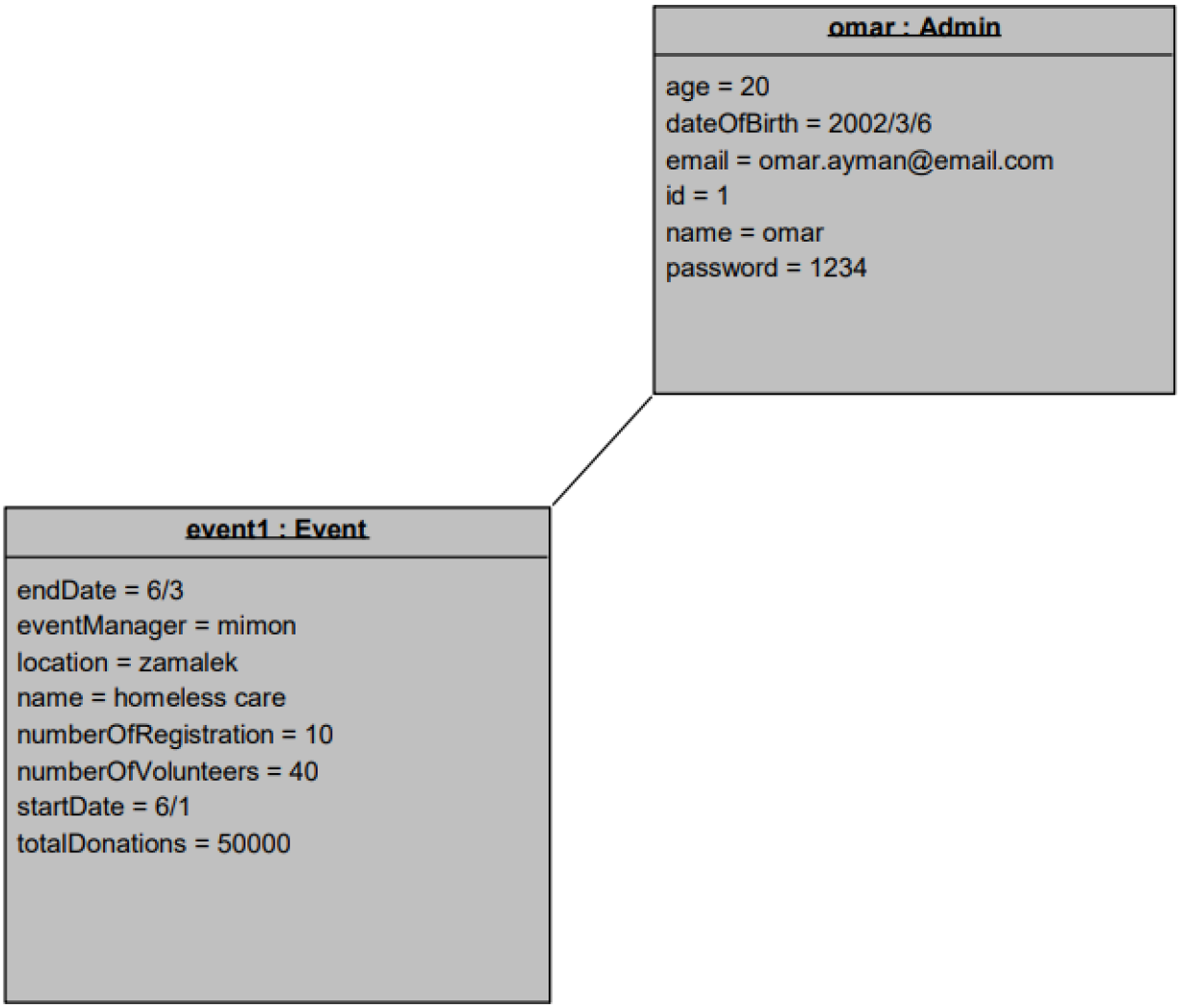
WorkInMarket()
Postcondition: volunteer has attended a market.



Pre-condition and Post-condition for set event total donation: -

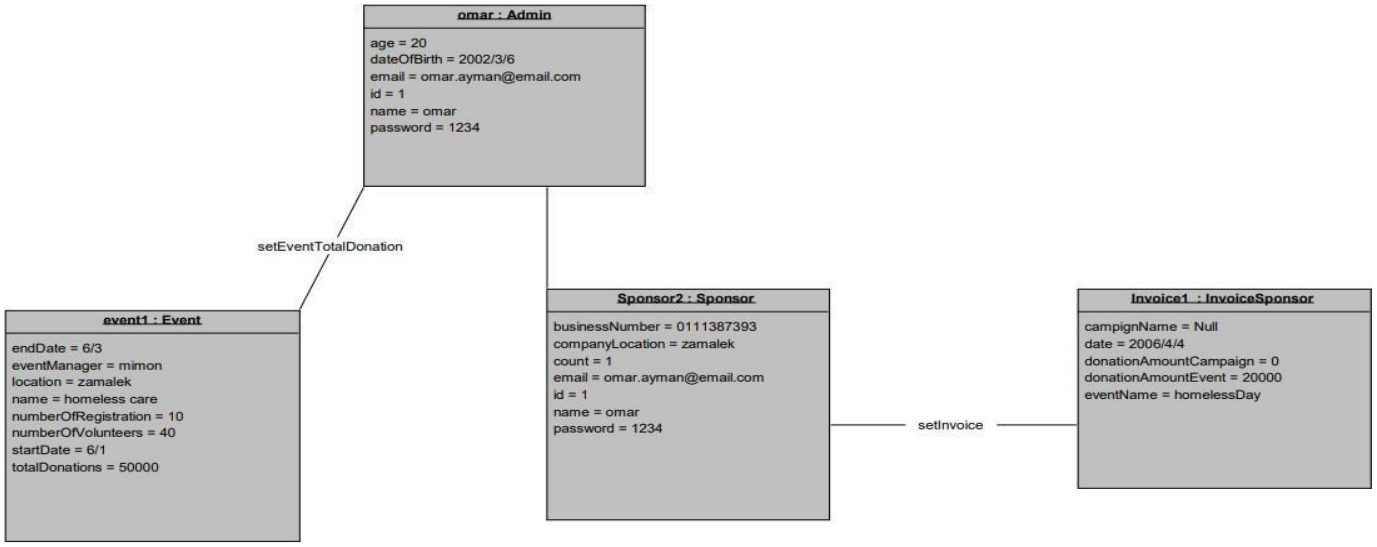
Precondition

setEventTotalDonation ()
precondition: the admin makes an event and specifies all the attributes for it such as the location and date
postcondition: the event has been sponsored and the invoice has been made for it, now after using setEventTotalDonation () the admin can know how many donations have been made for each event

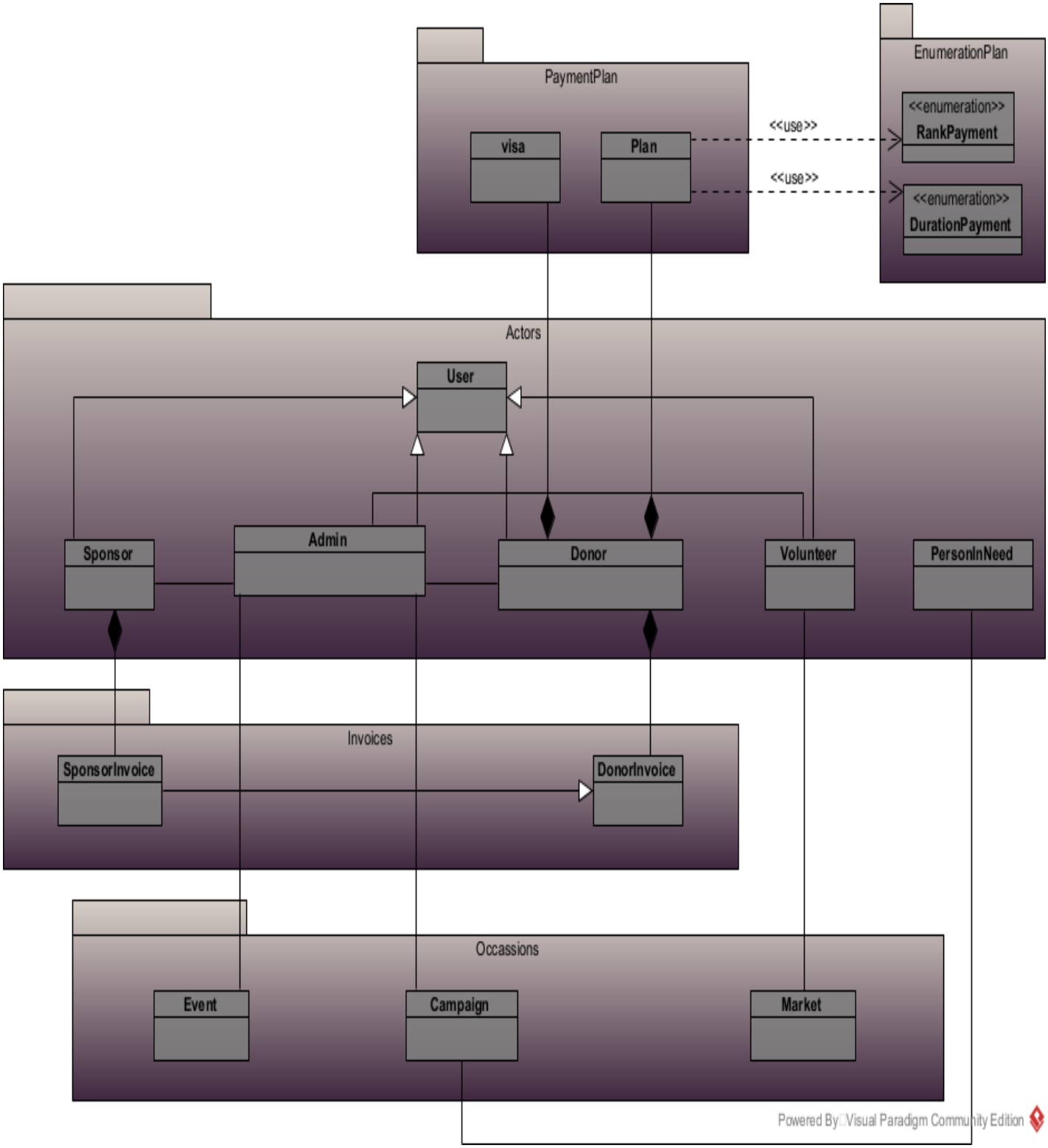


Postcondition

setEventTotalDonation ()
postcondition: the event has been sponsored and the invoice has been made for it, now after using setEventTotalDonation () the admin can know how many donations have been made for each event

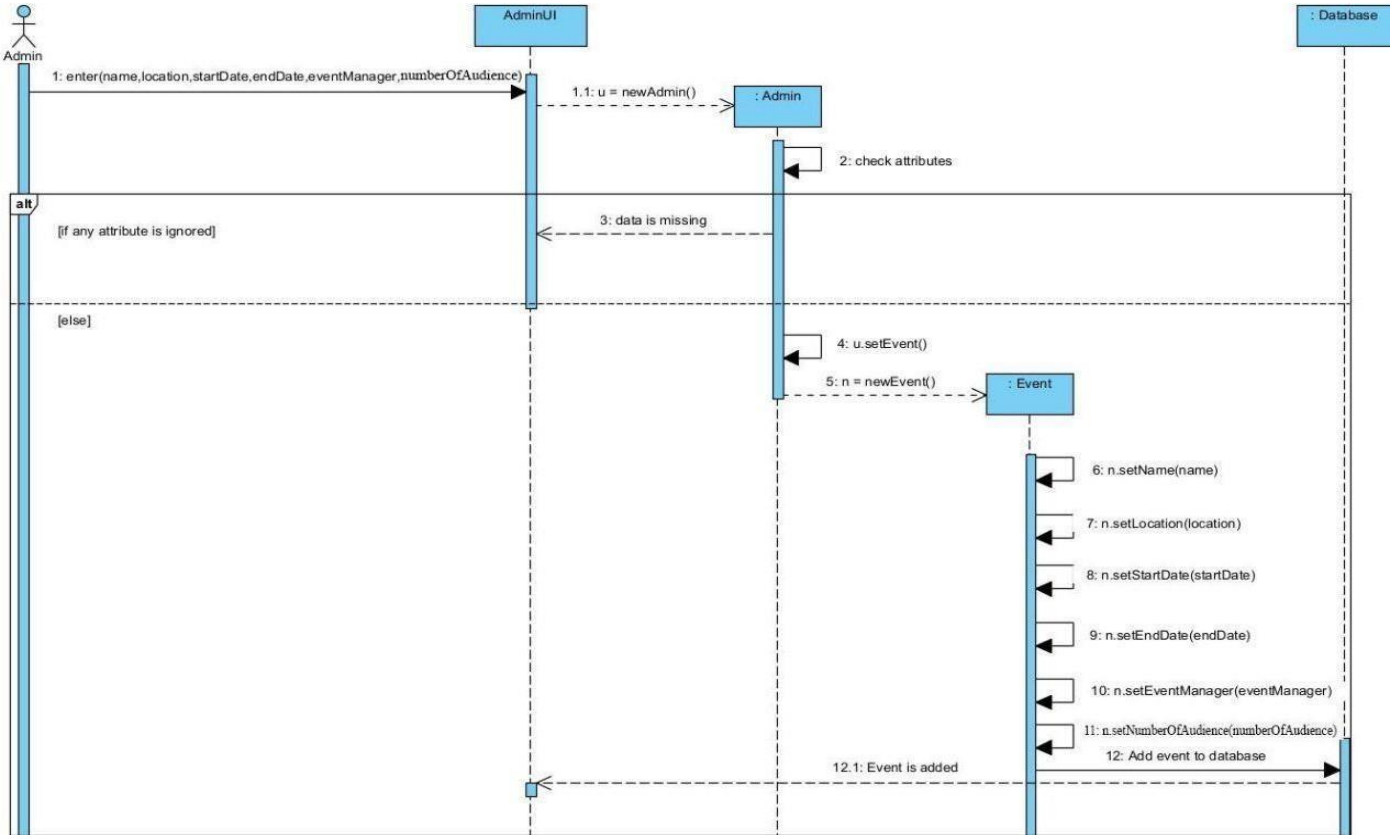


Package diagram

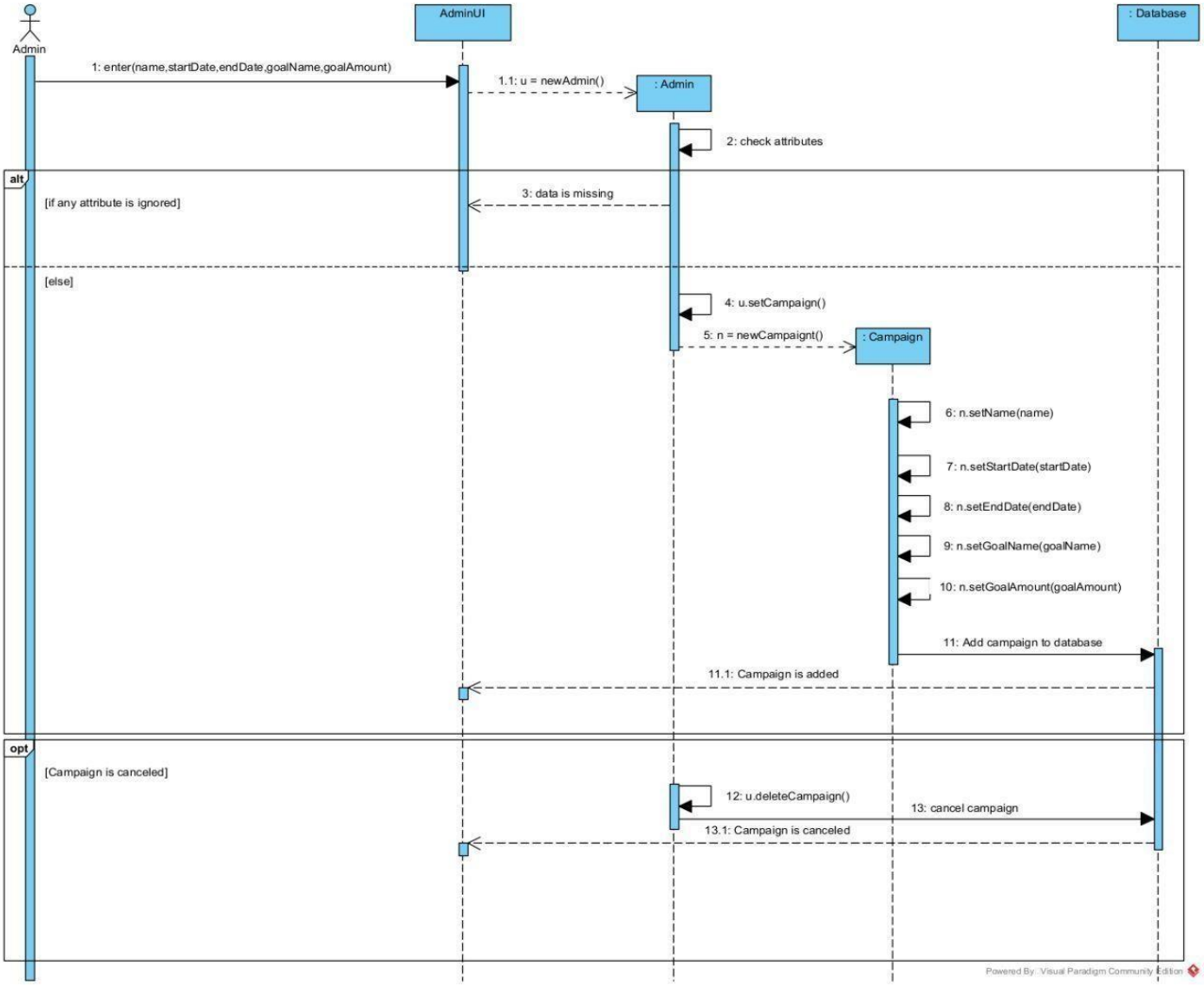


Sequence diagram

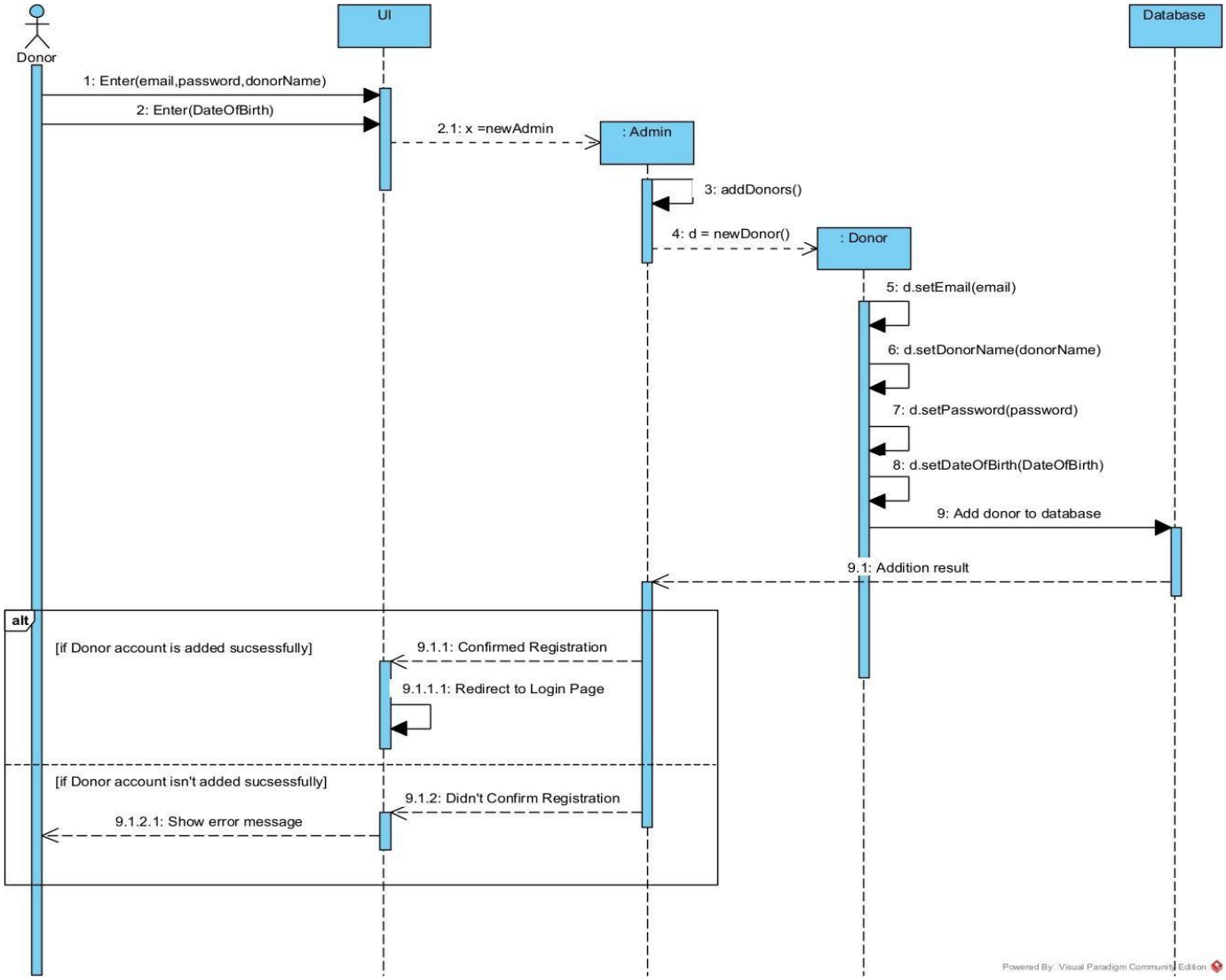
Add event diagram.



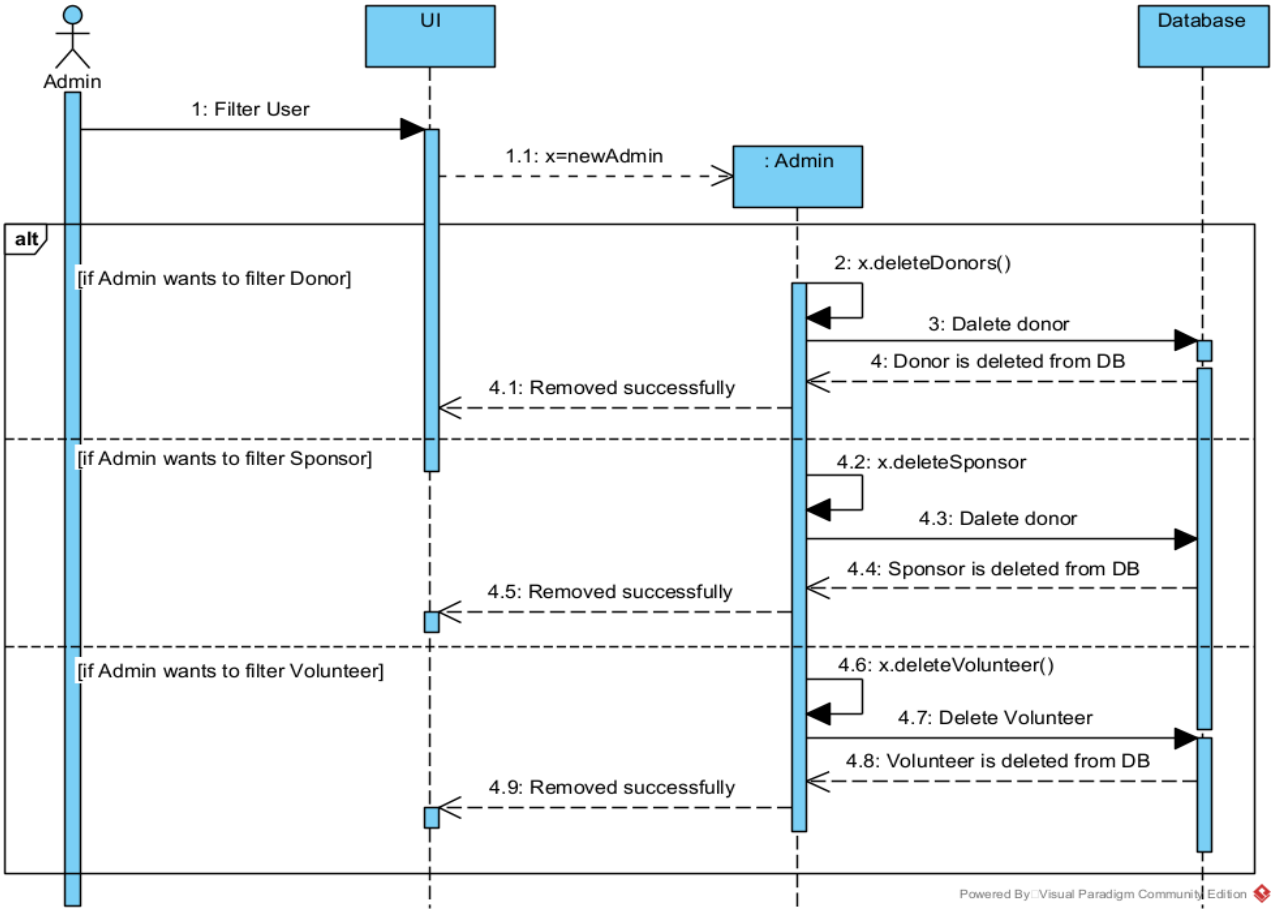
Add campaign diagram.



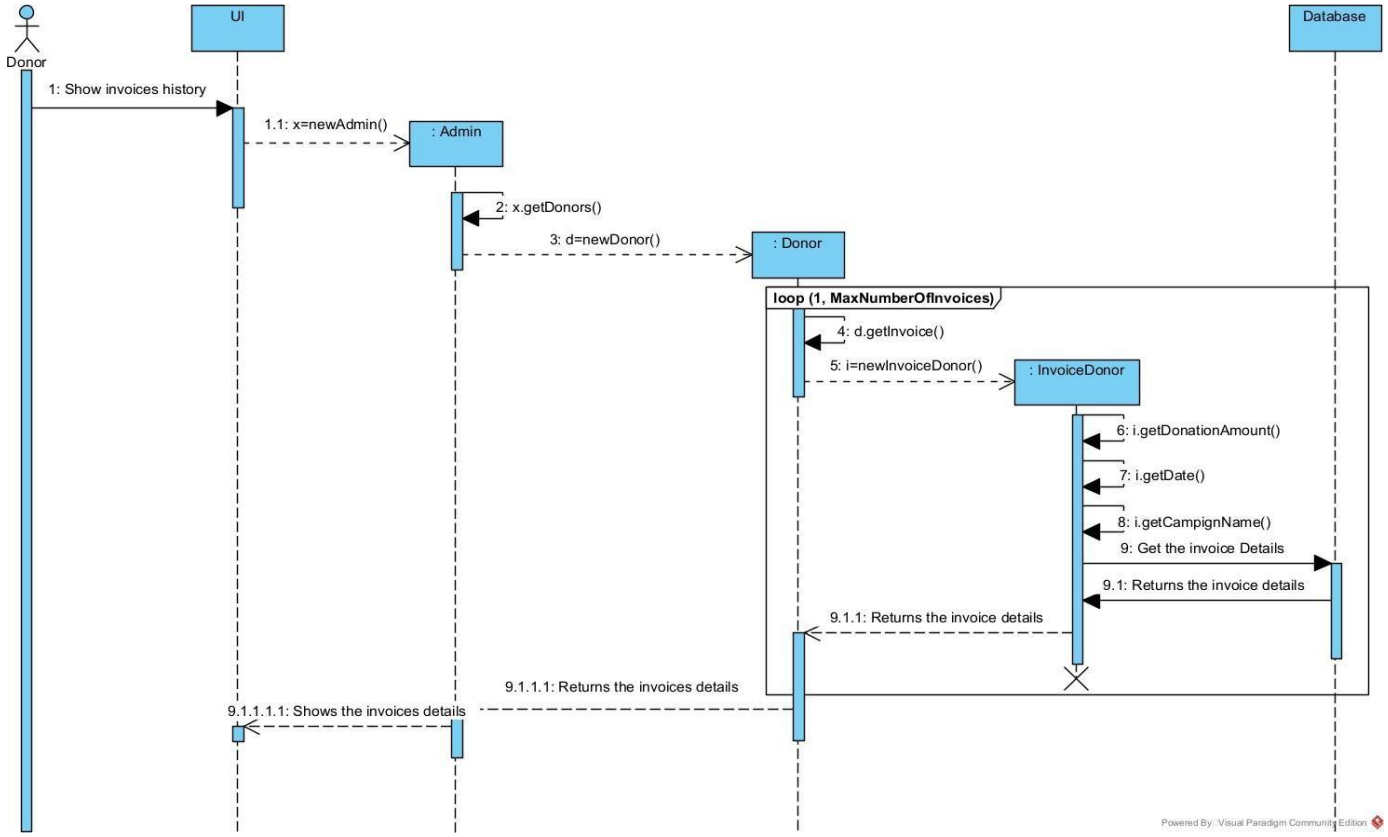
Donor sign up diagram



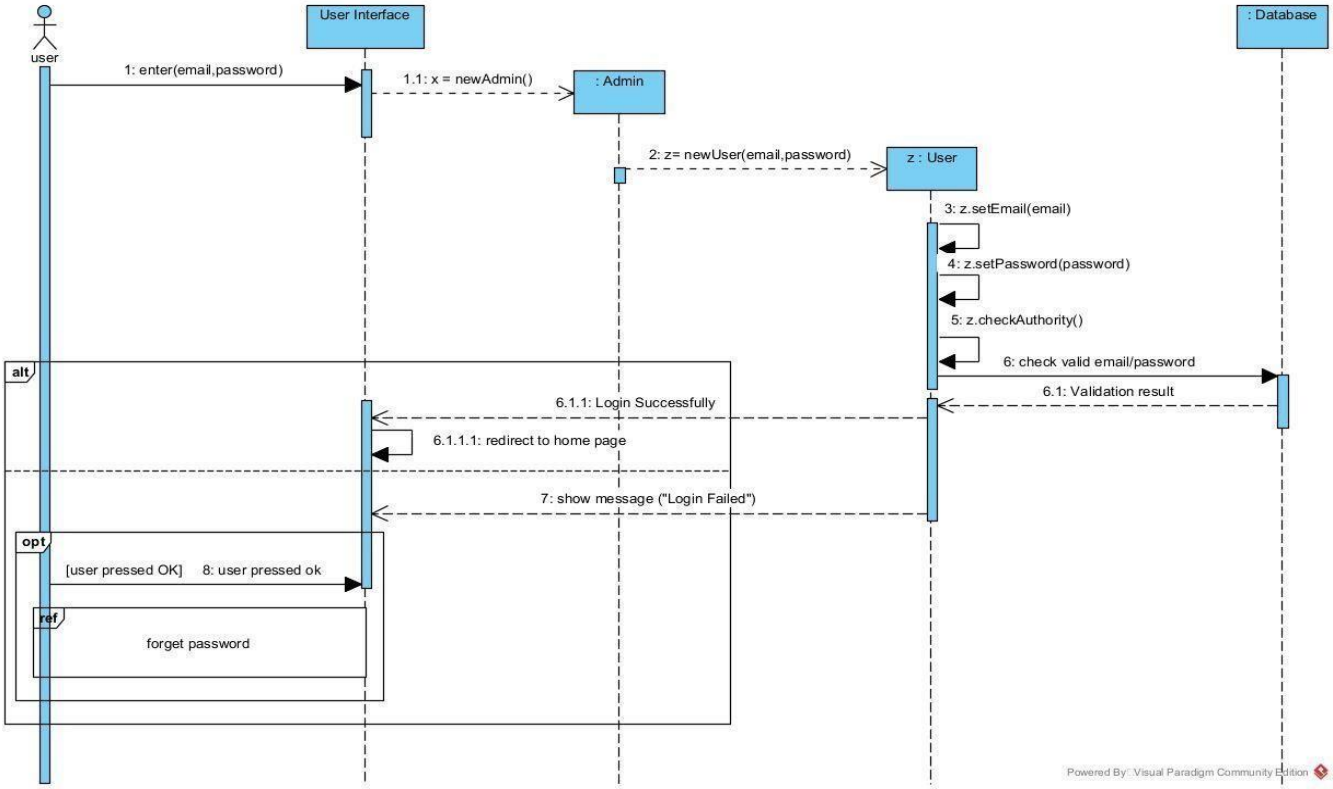
Filtering users by admin diagram.



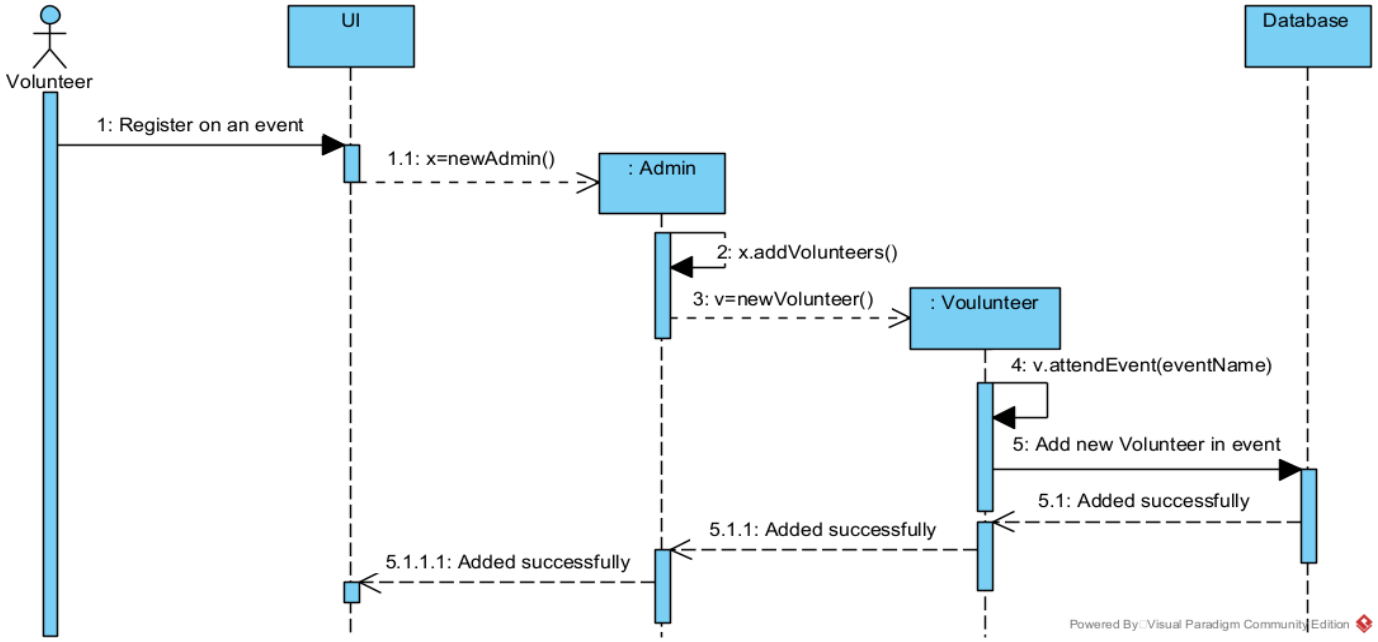
Invoices history diagram



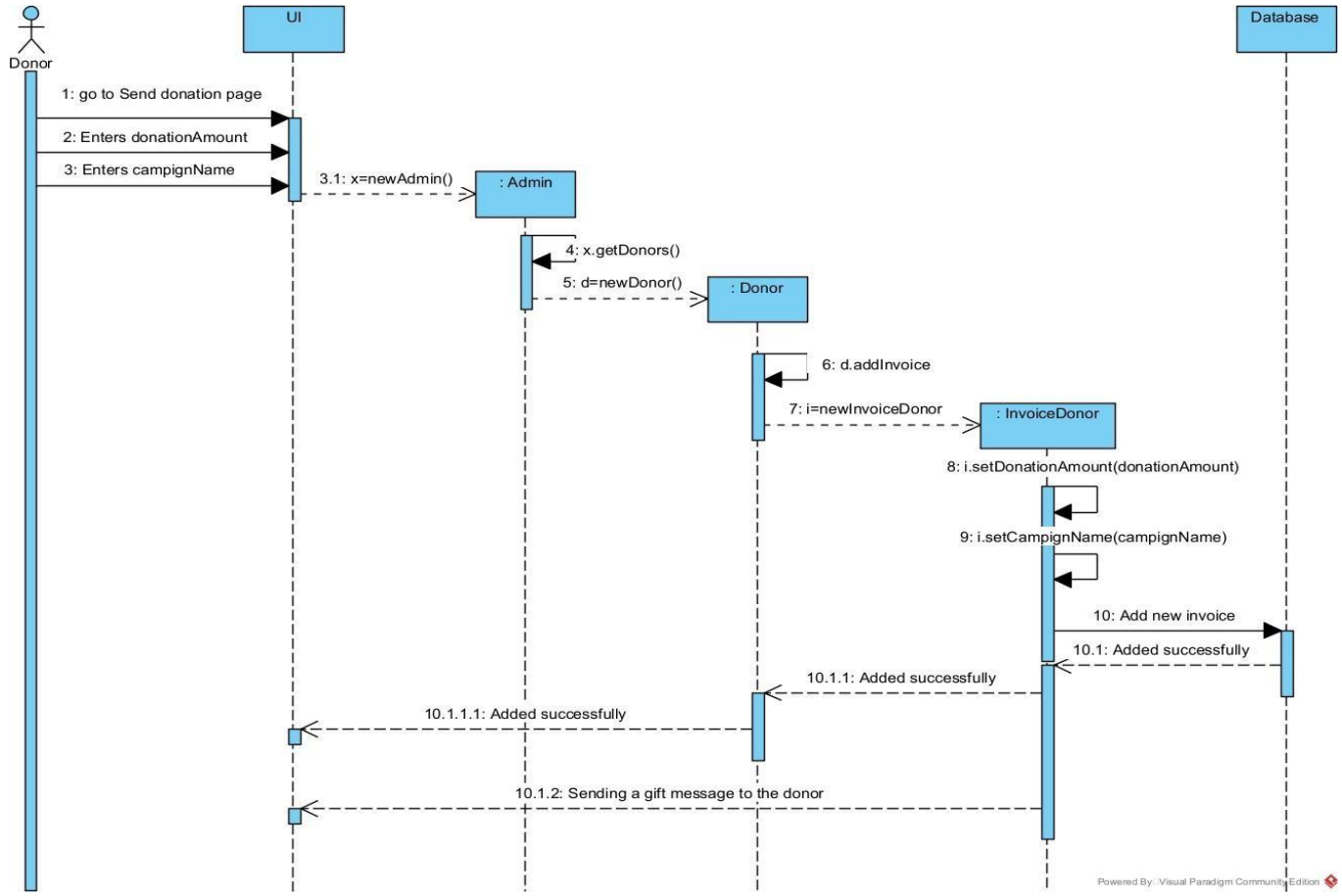
Login diagram.



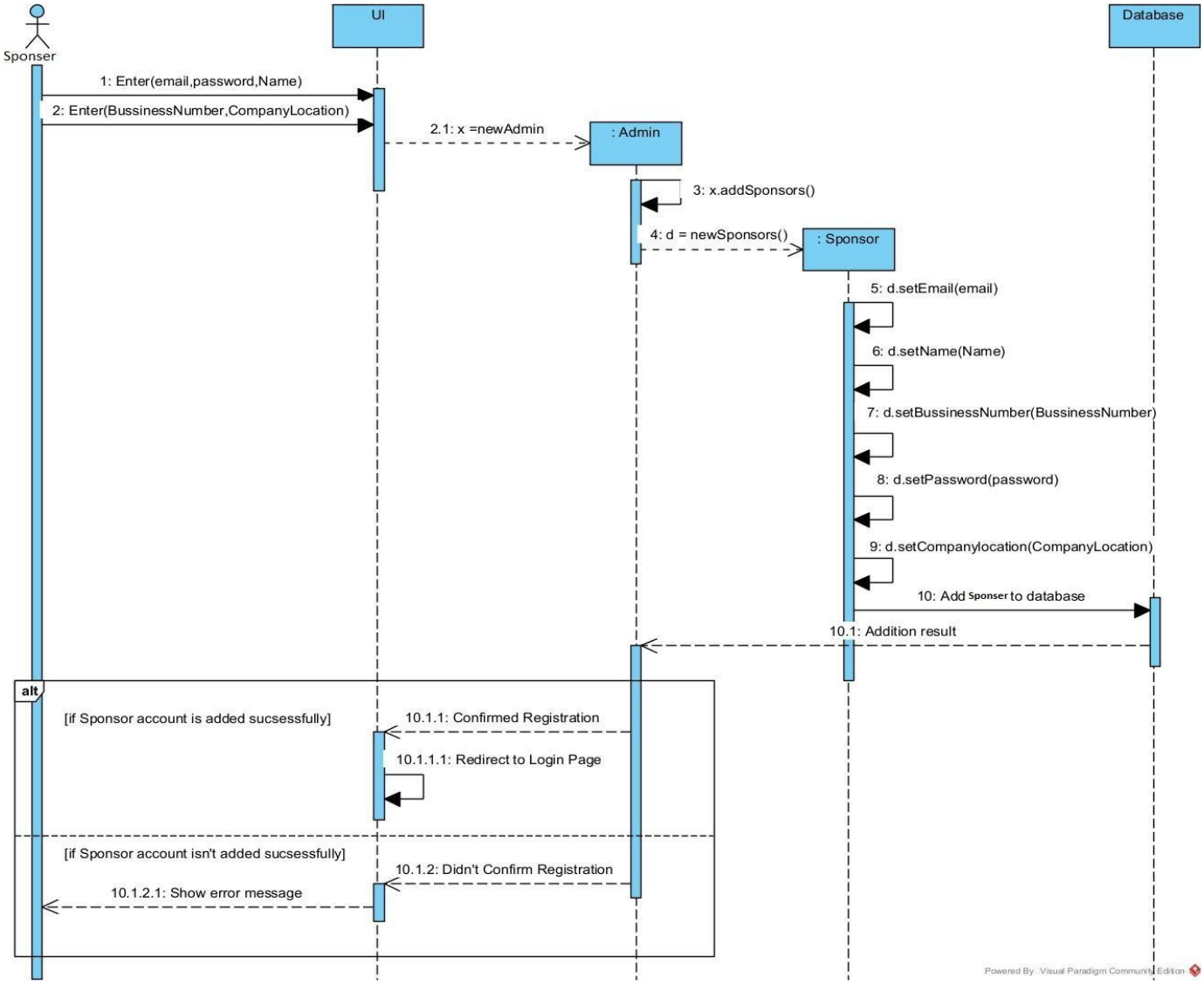
Volunteer register on an event diagram



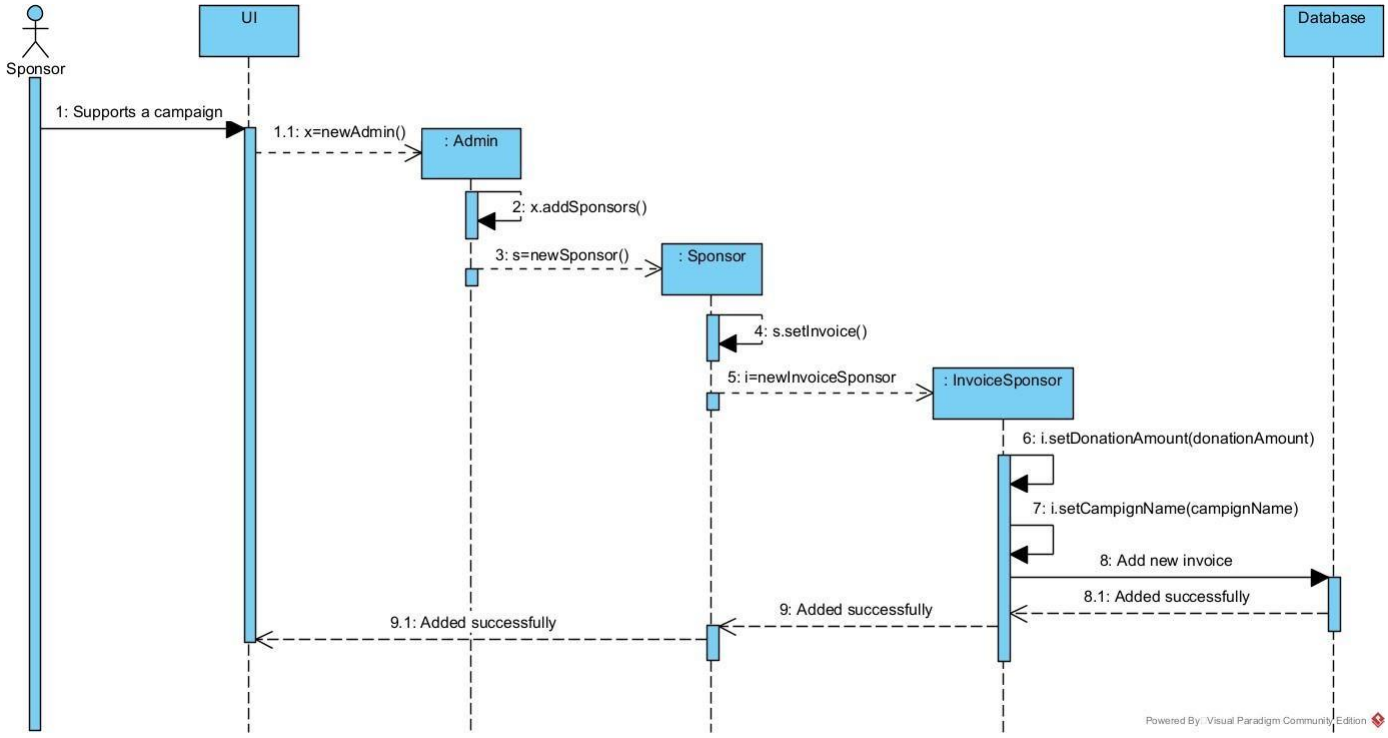
Sending donations diagram.



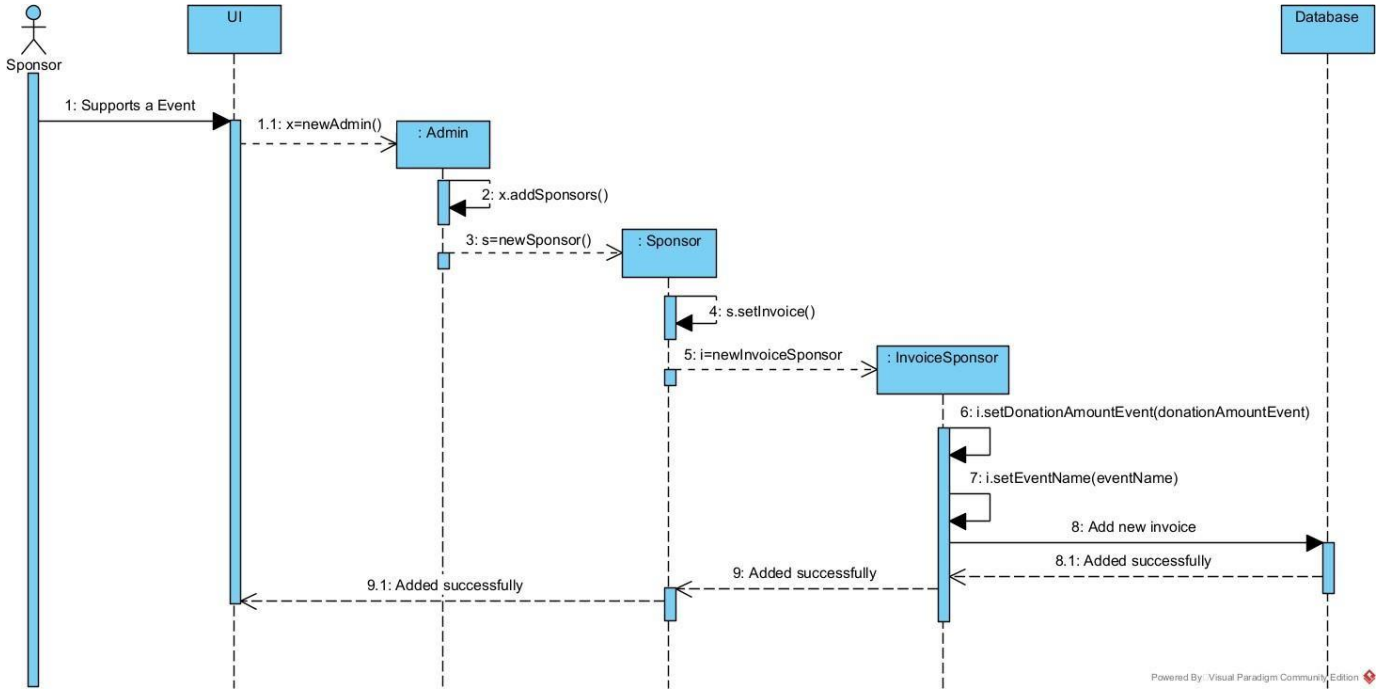
Sponsor sign-up.



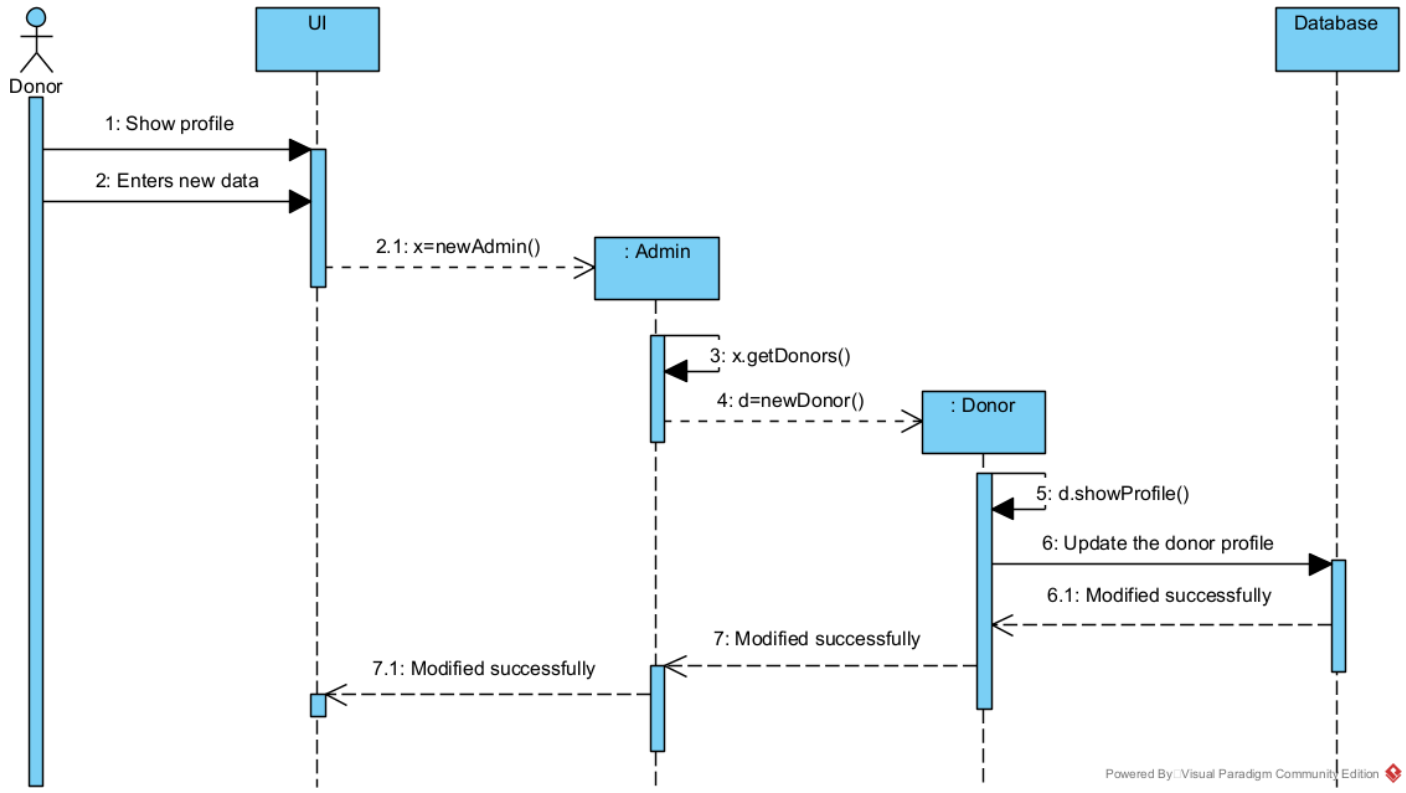
Support campaign



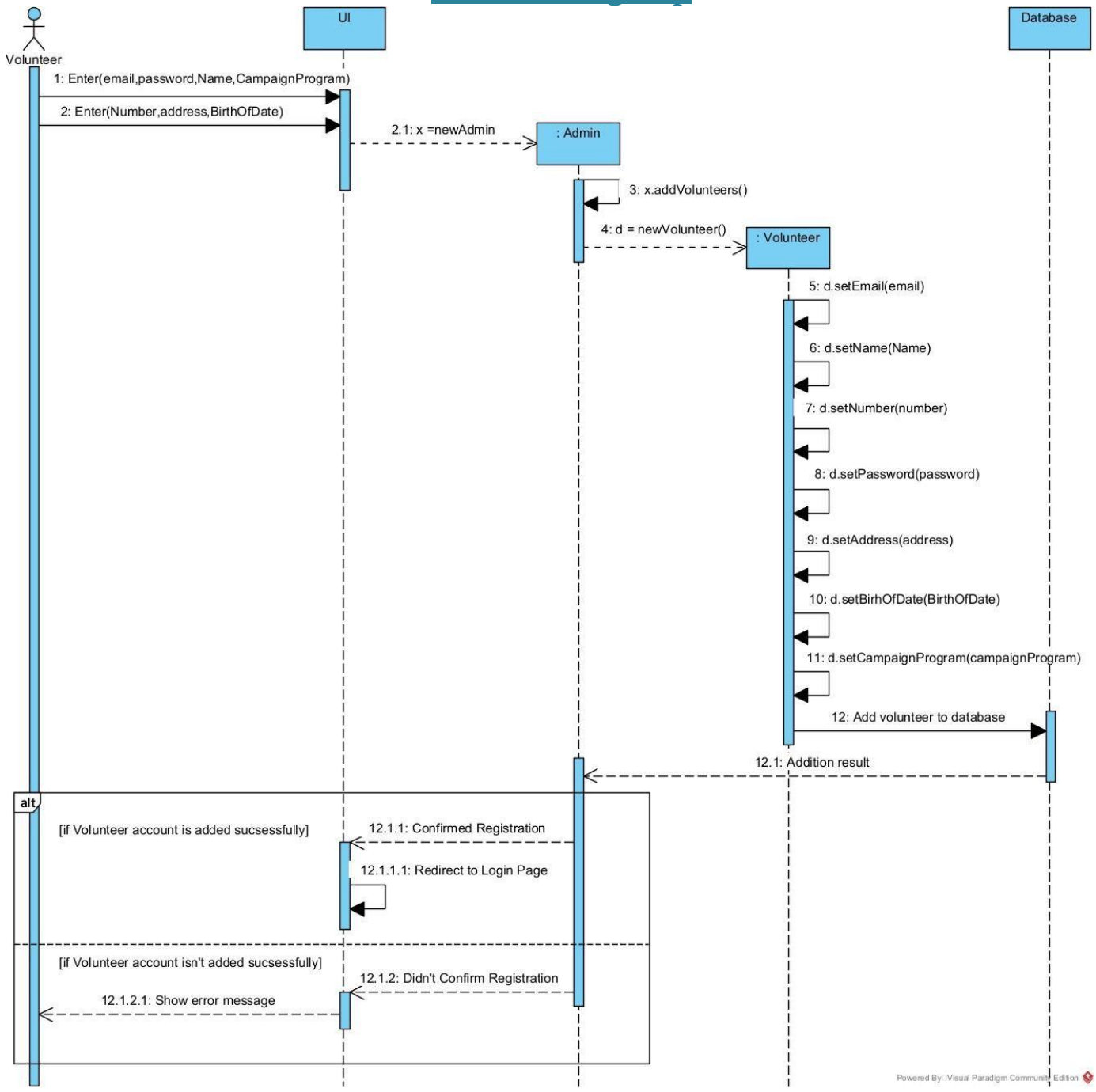
Support event diagram



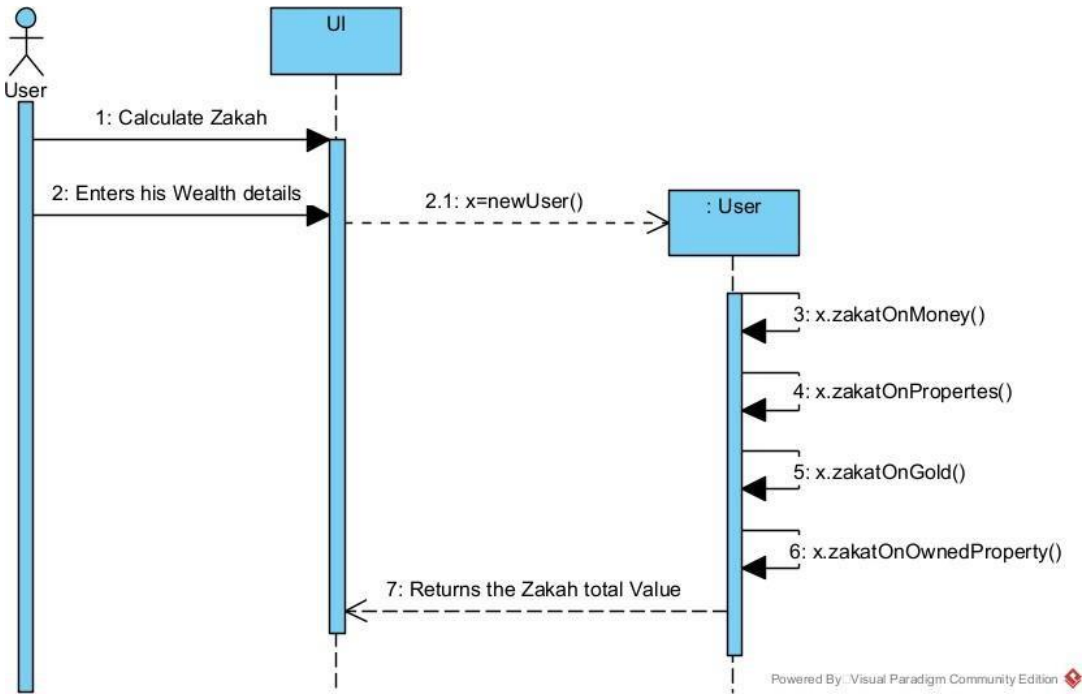
Update profile diagram



Volunteer sign-up

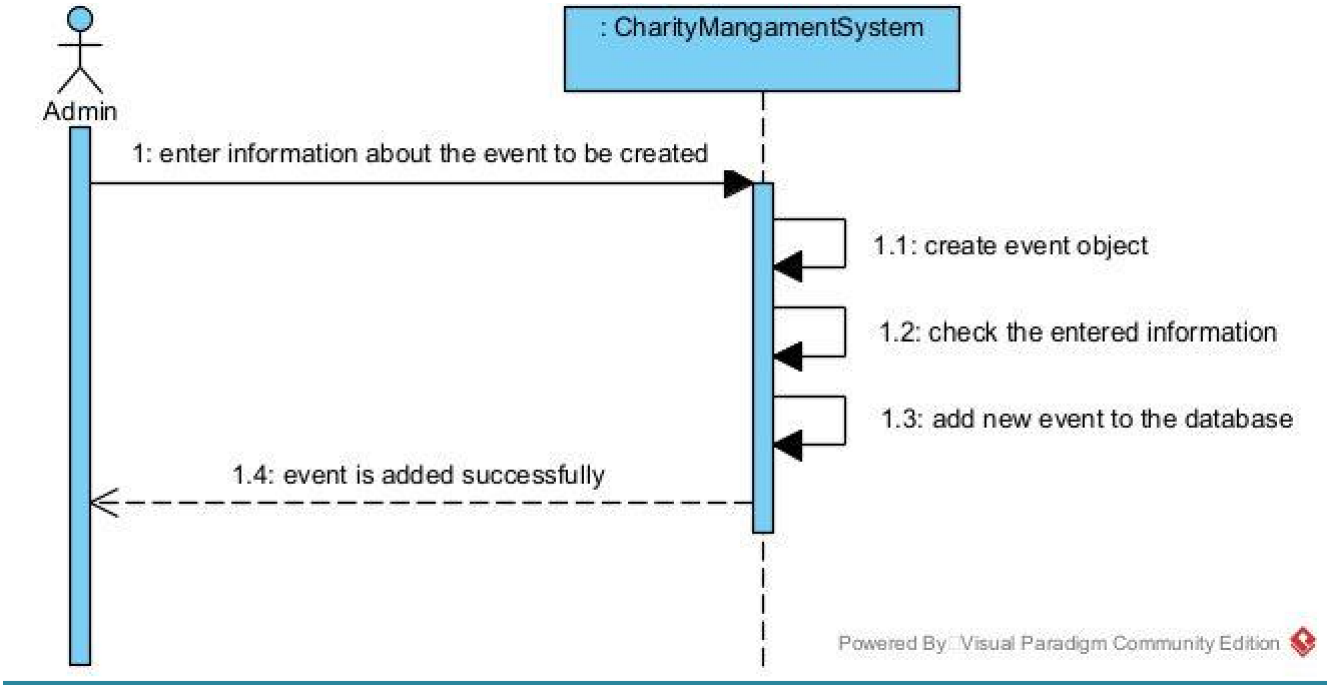


Zakat calculator diagram

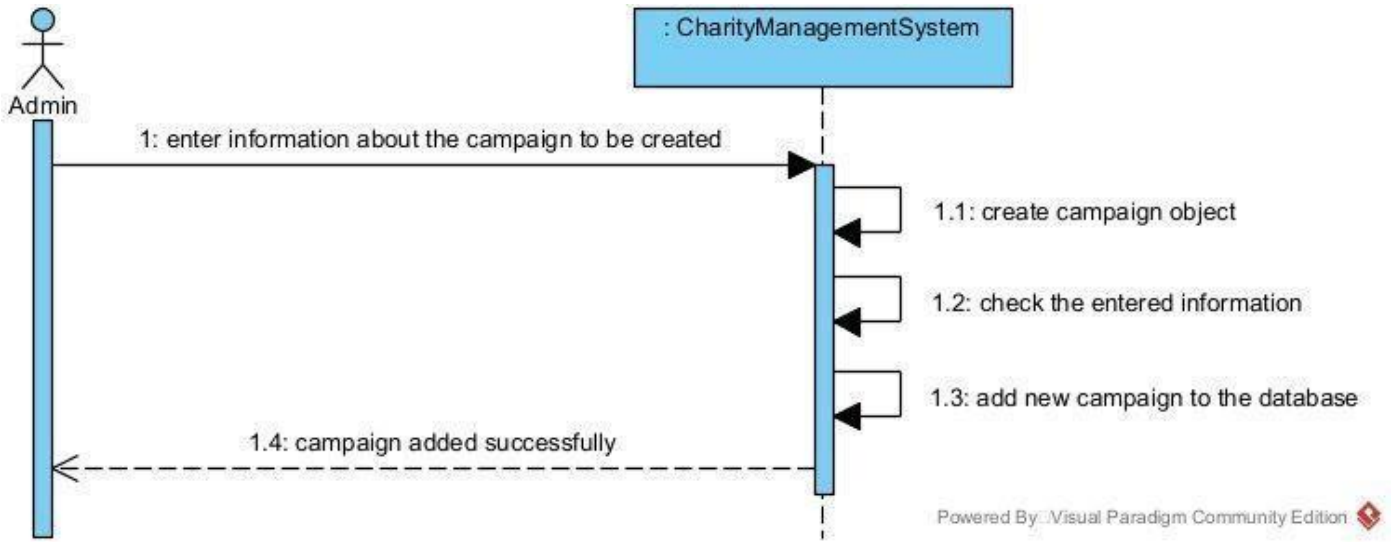


System Sequence diagram

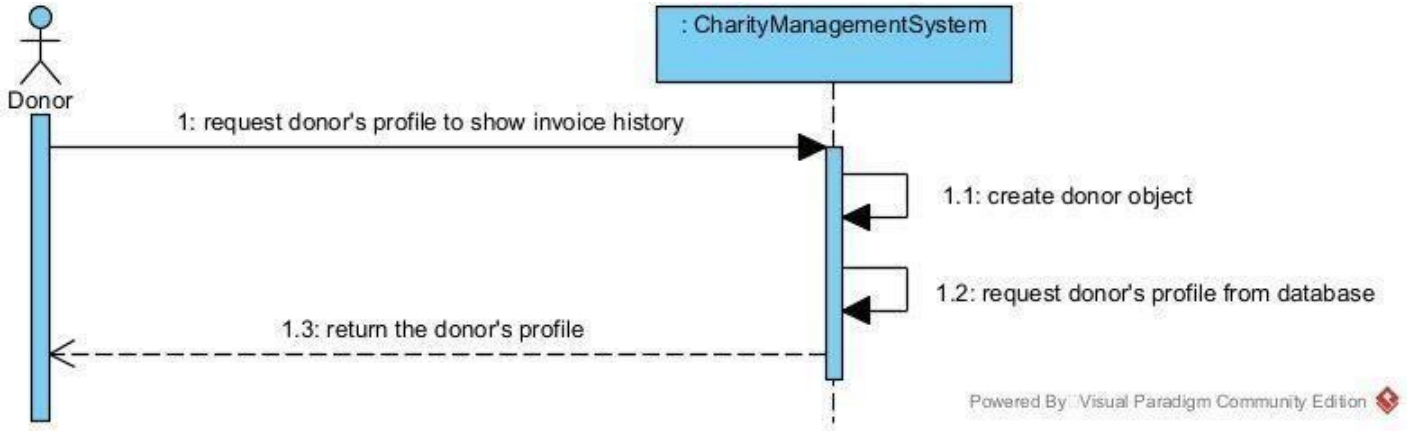
Add event diagram



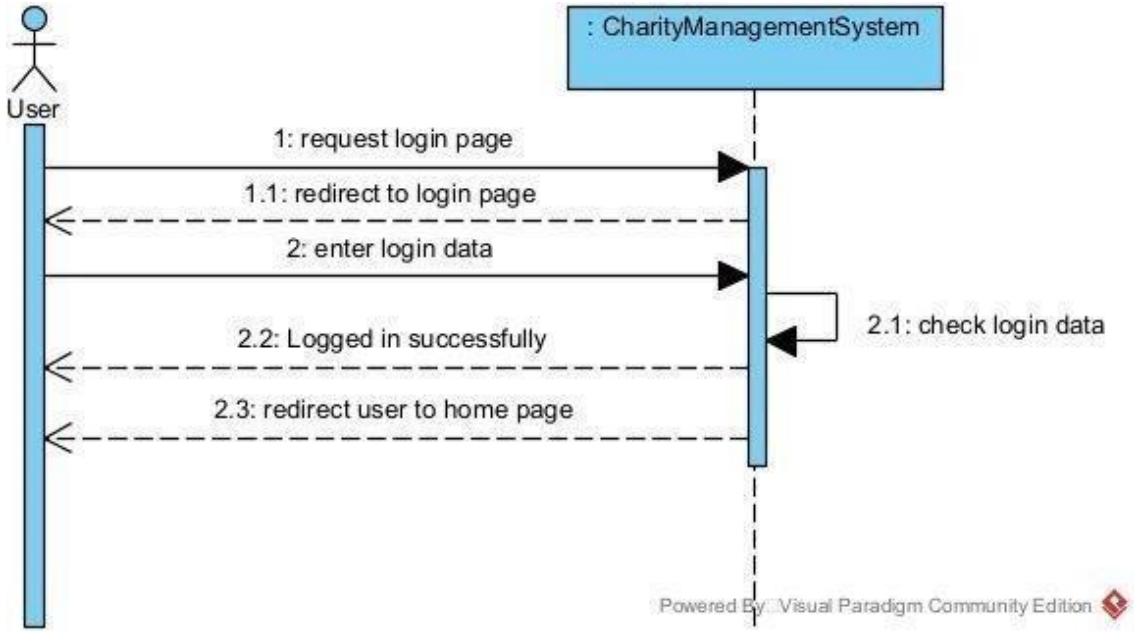
Add campaign diagram



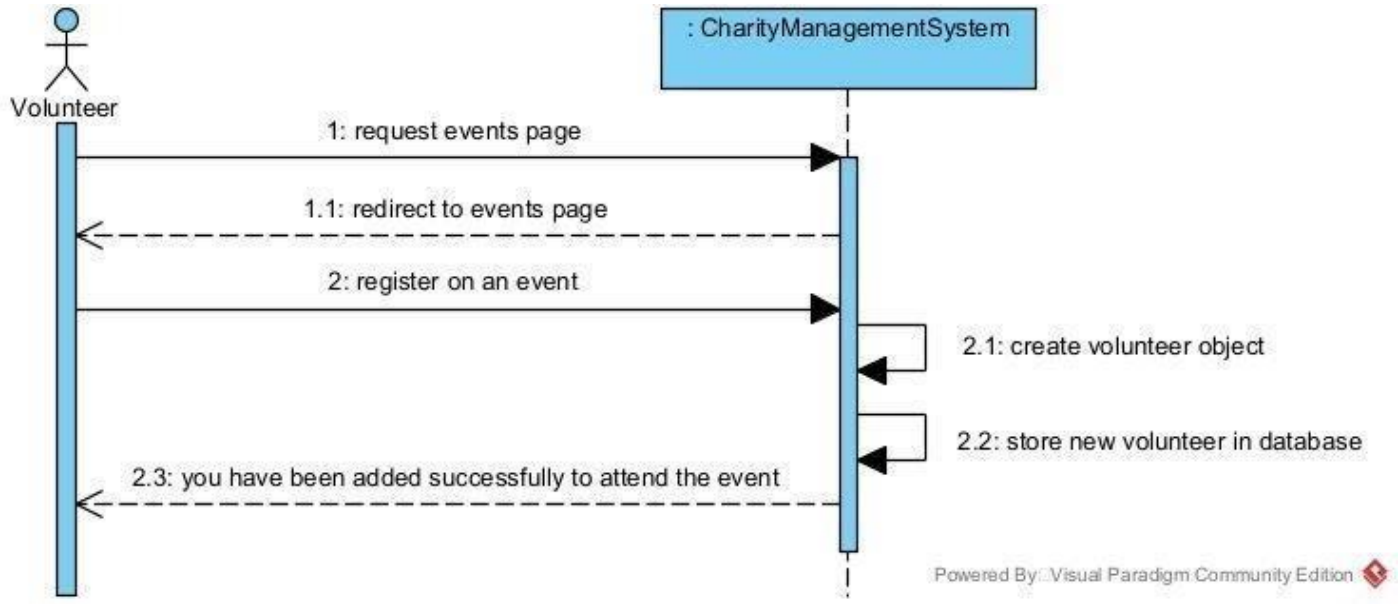
Invoice history diagram



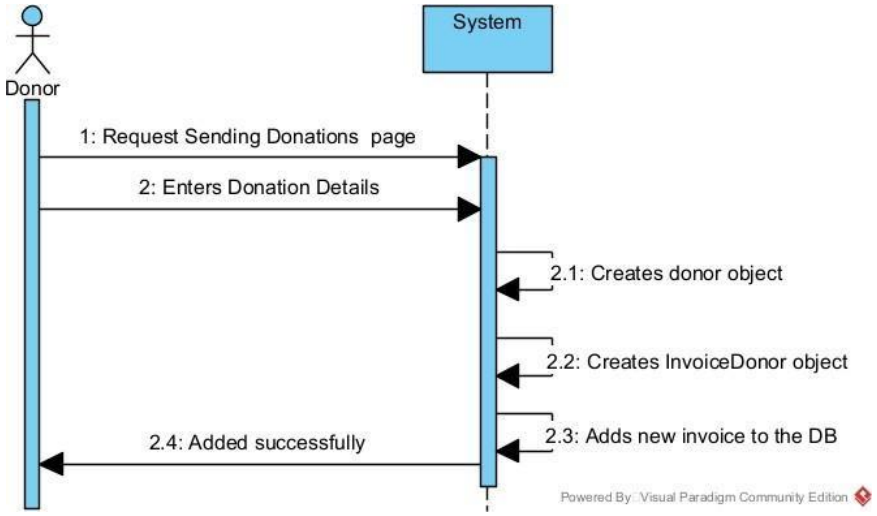
Login diagram.



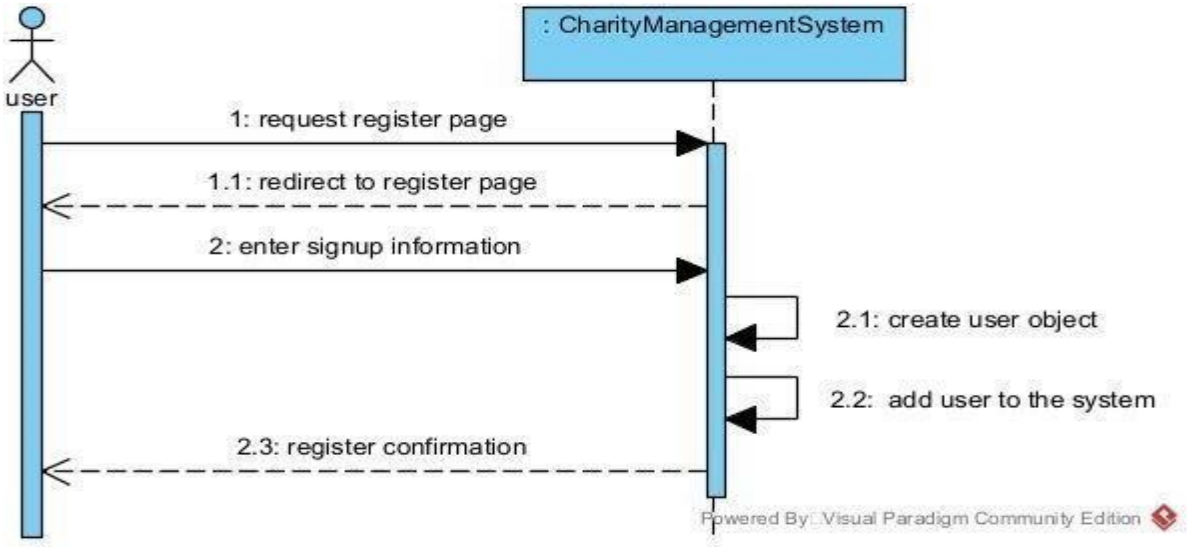
Register on an event diagram



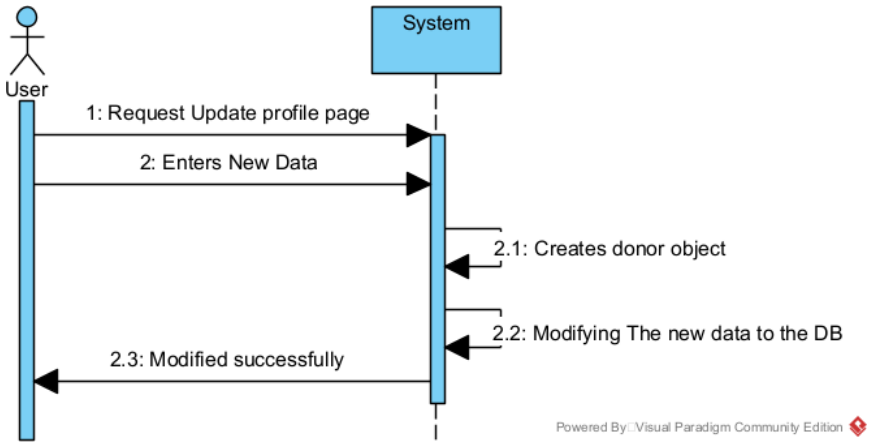
Send donations diagram.



Signup diagram.

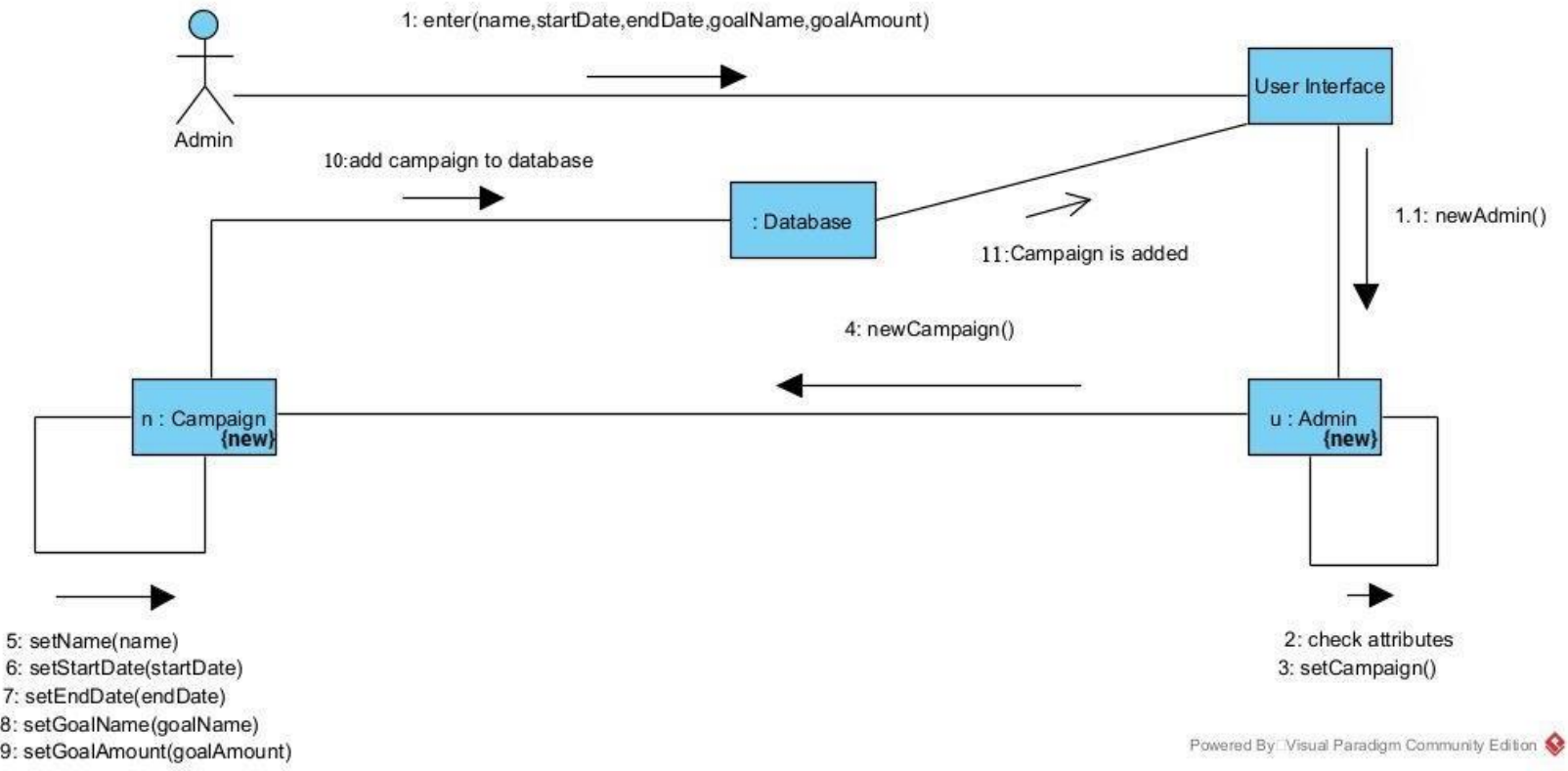


Update profile diagram

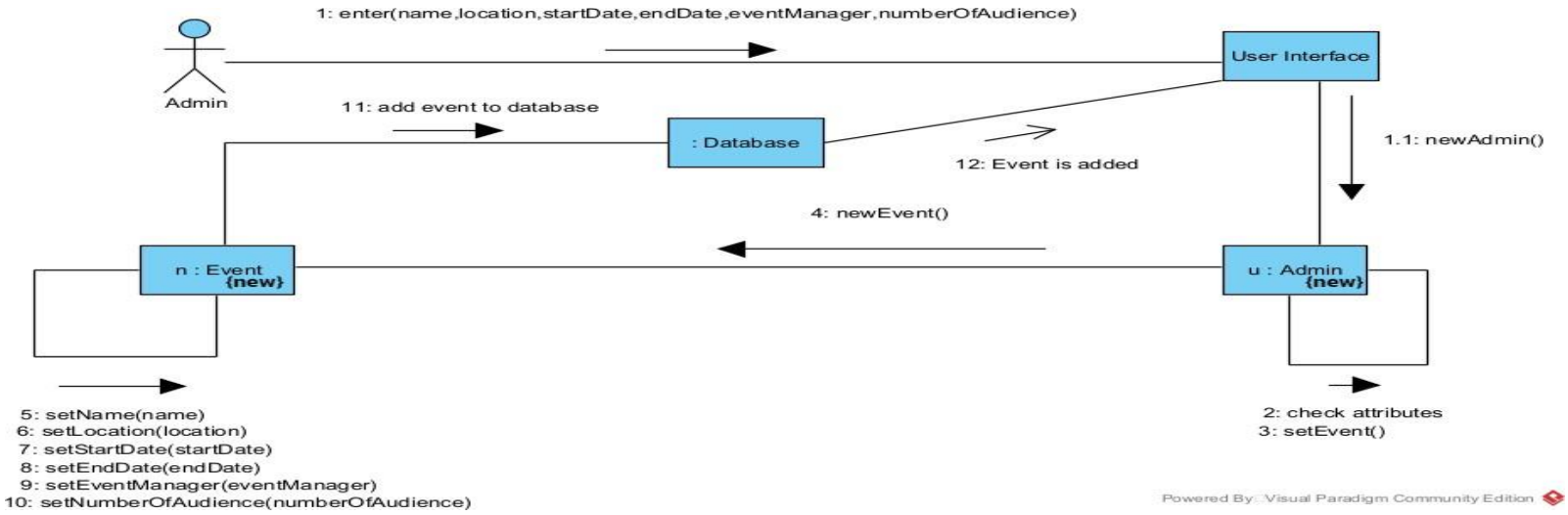


Collaboration/Communication diagram

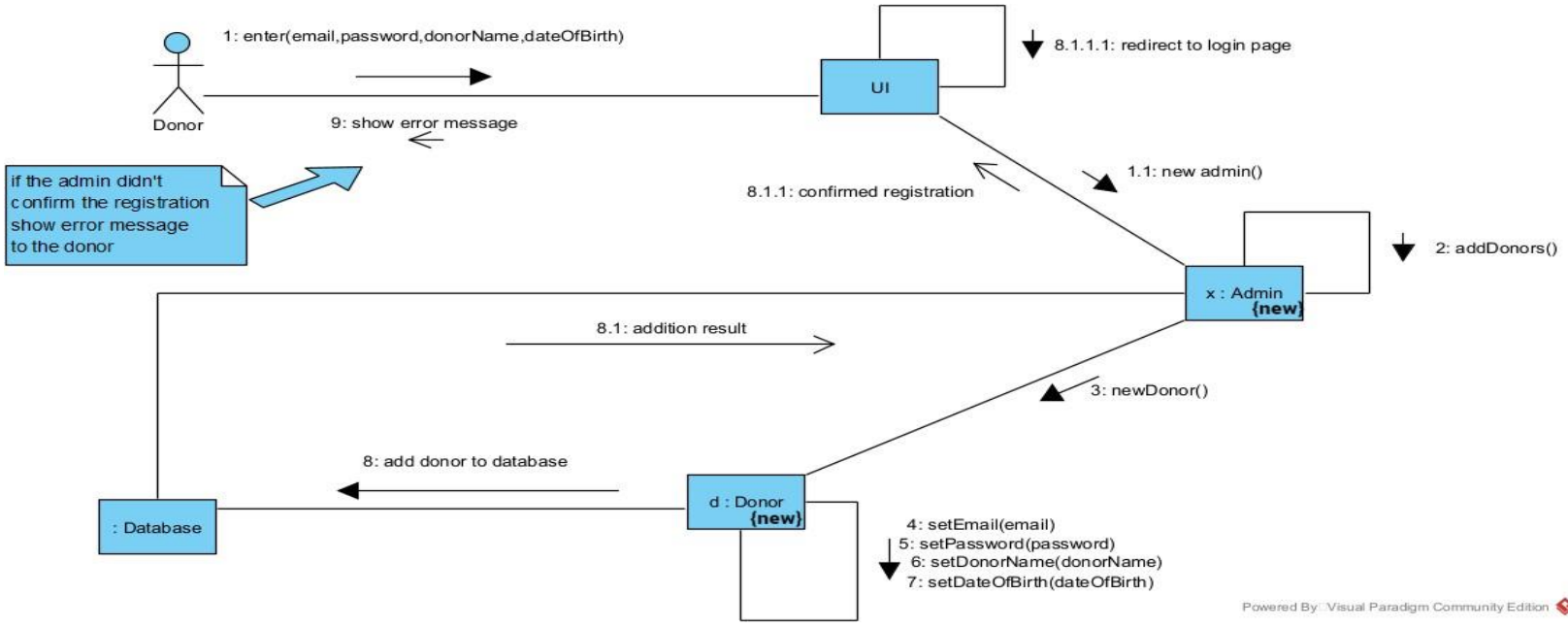
Add Campaign diagram



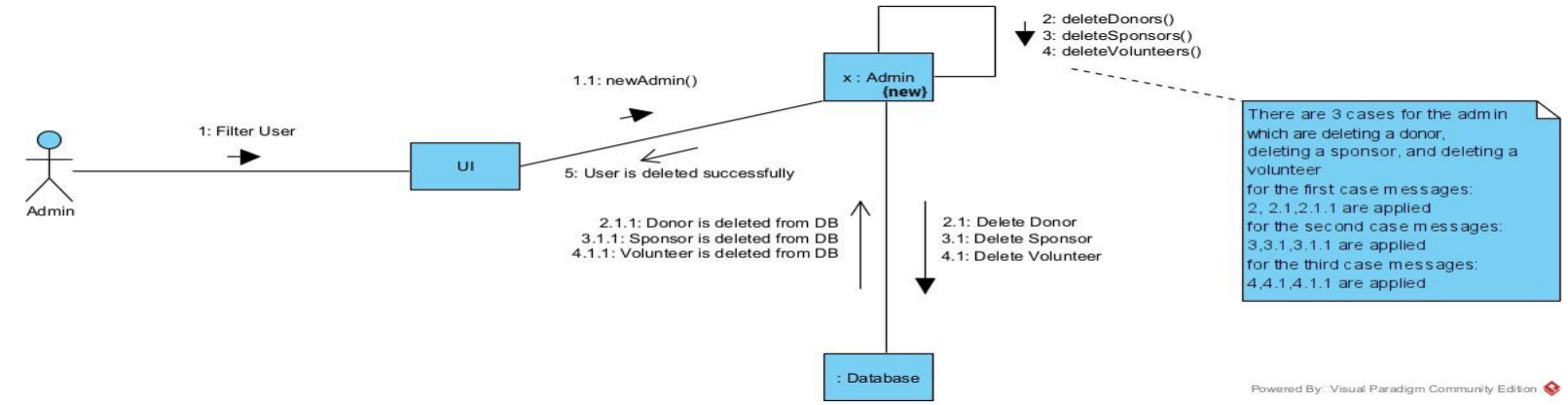
Add event diagram



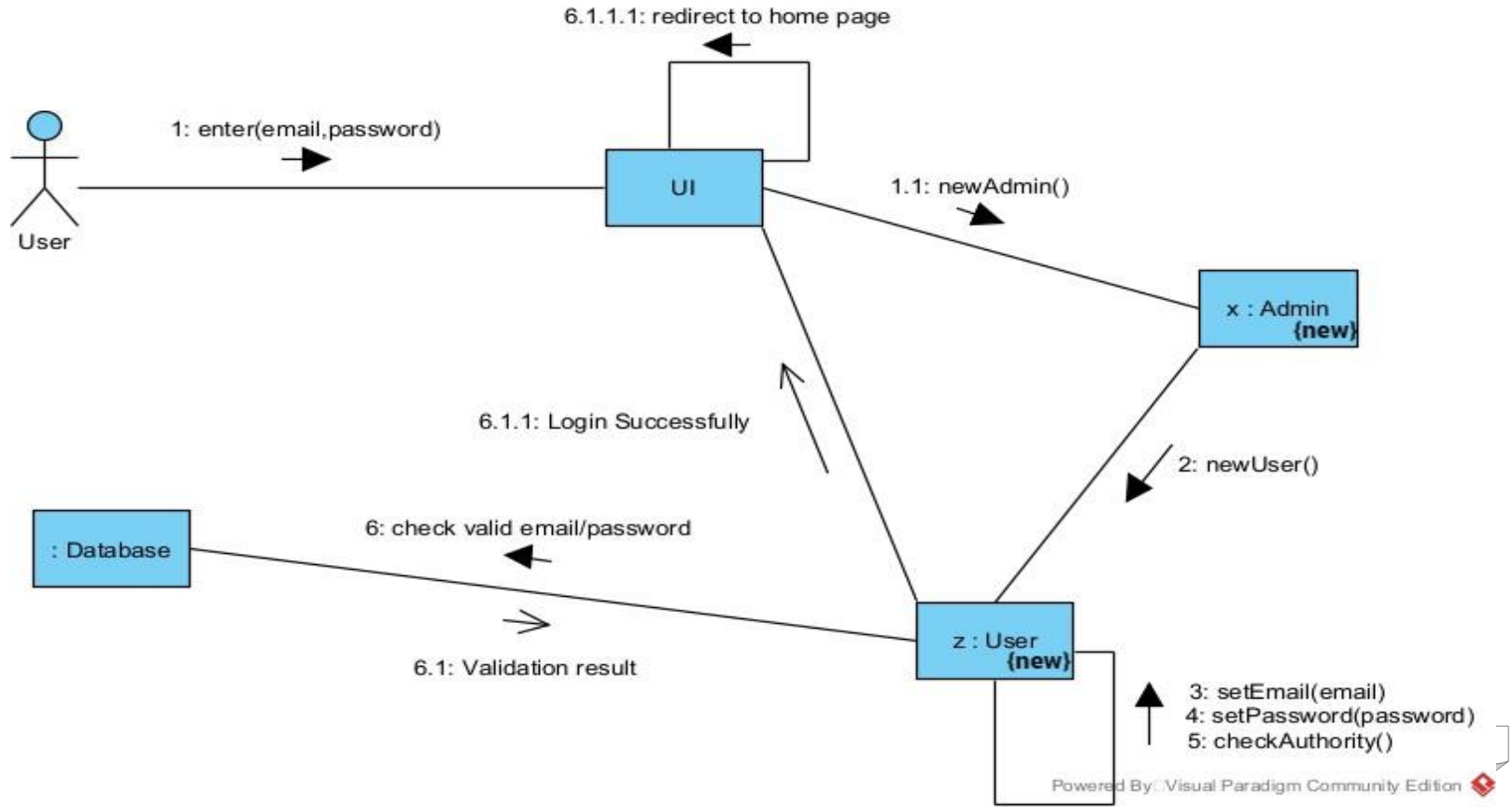
Donor Signup



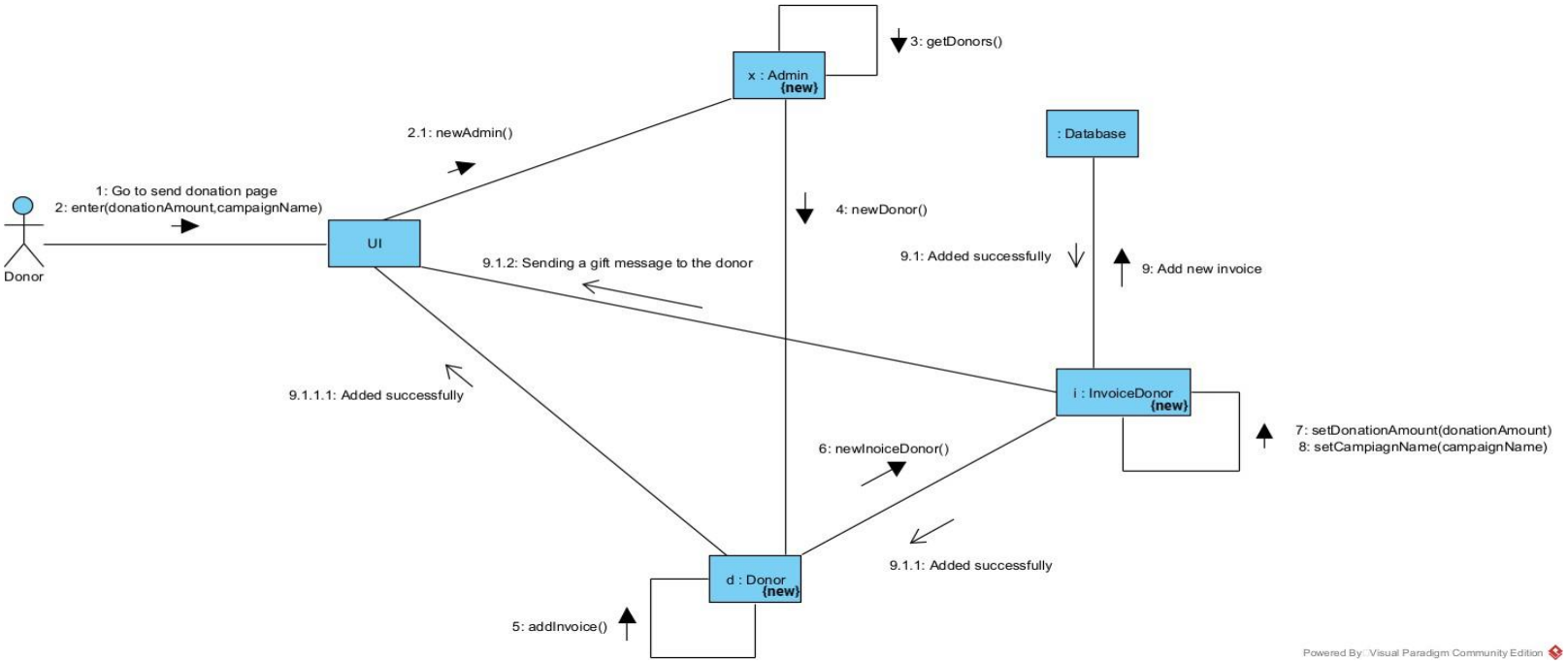
Admin Login



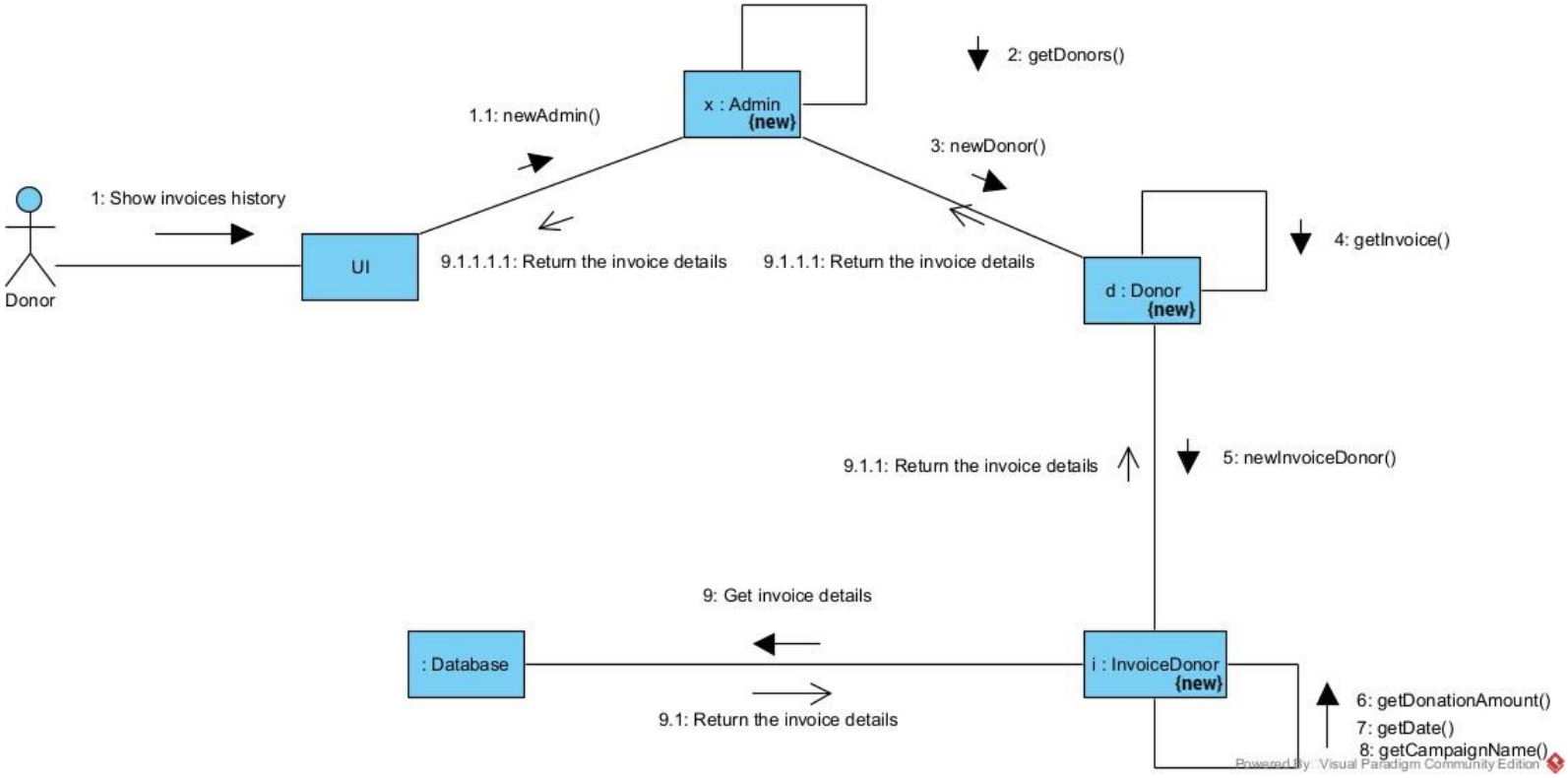
Filtering Users



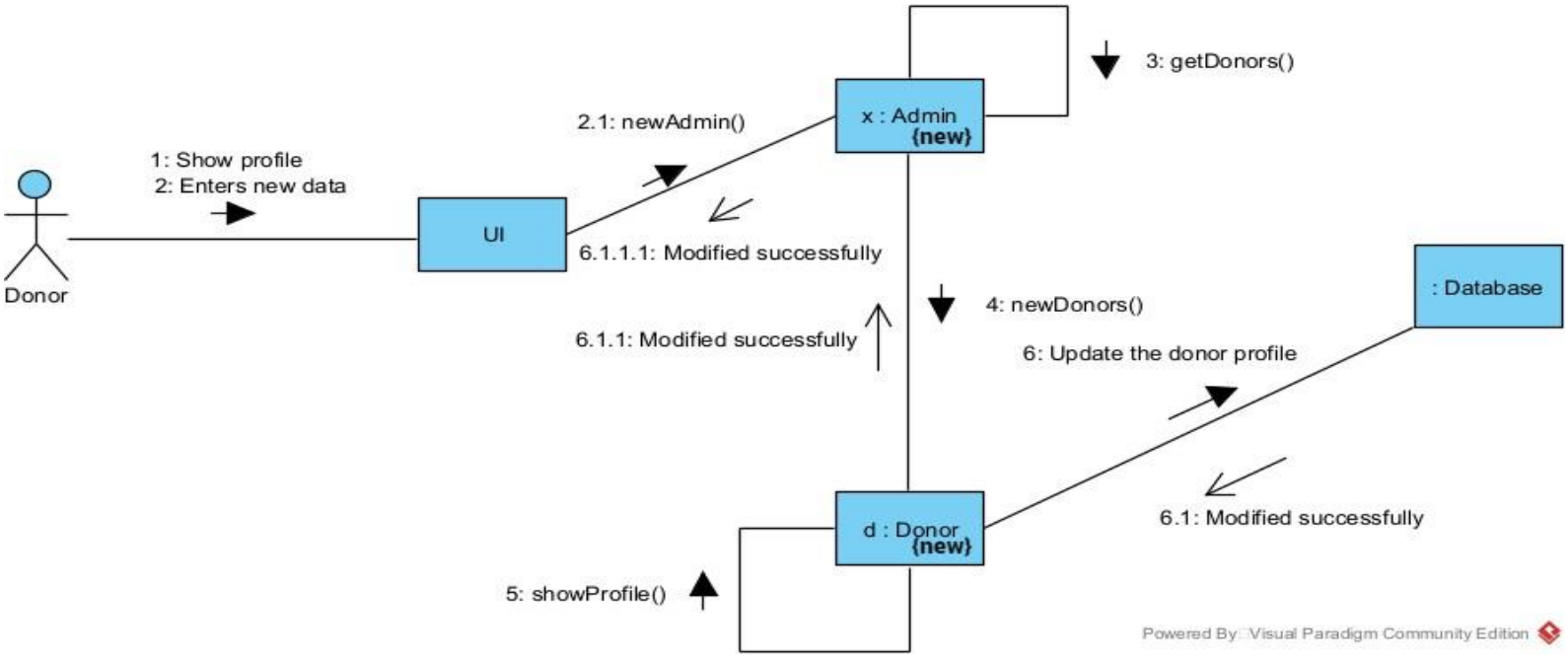
Sending donations by user



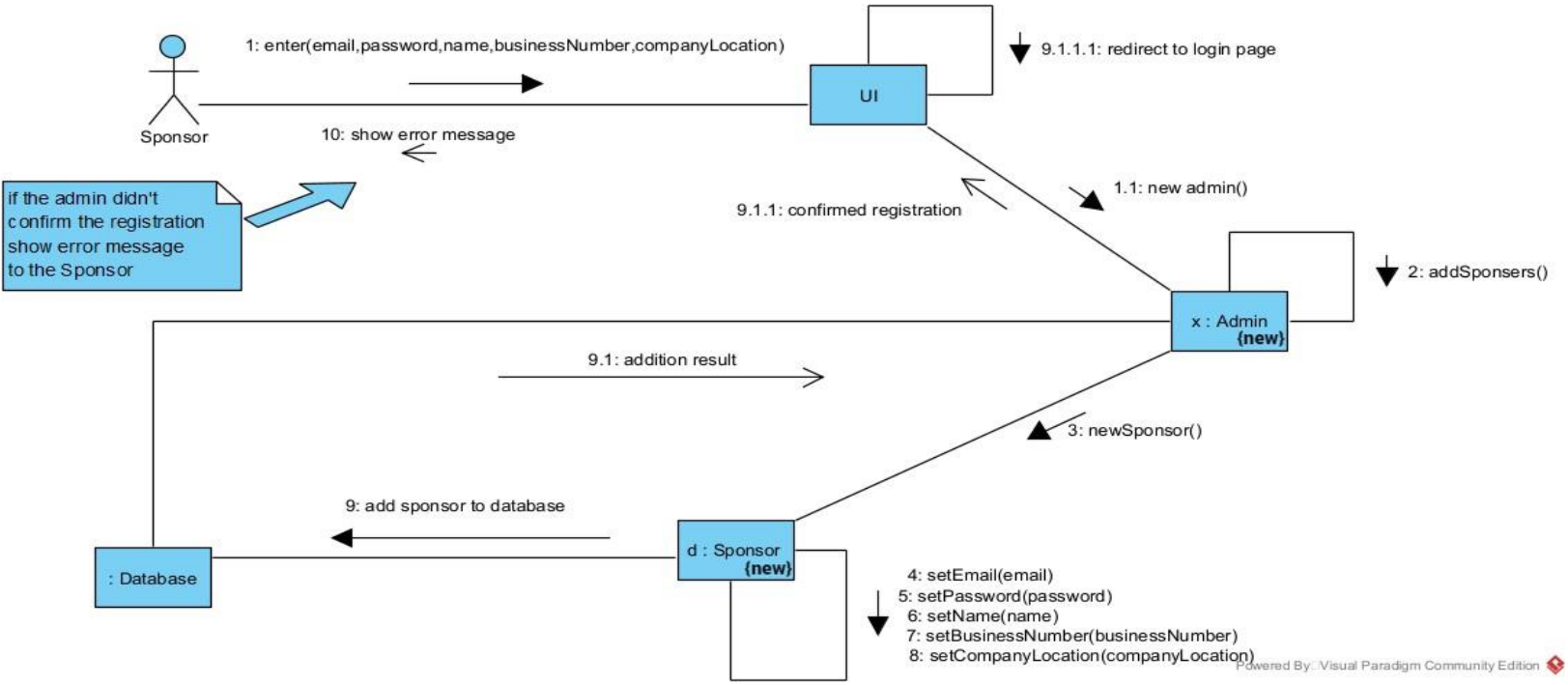
Show invoices history



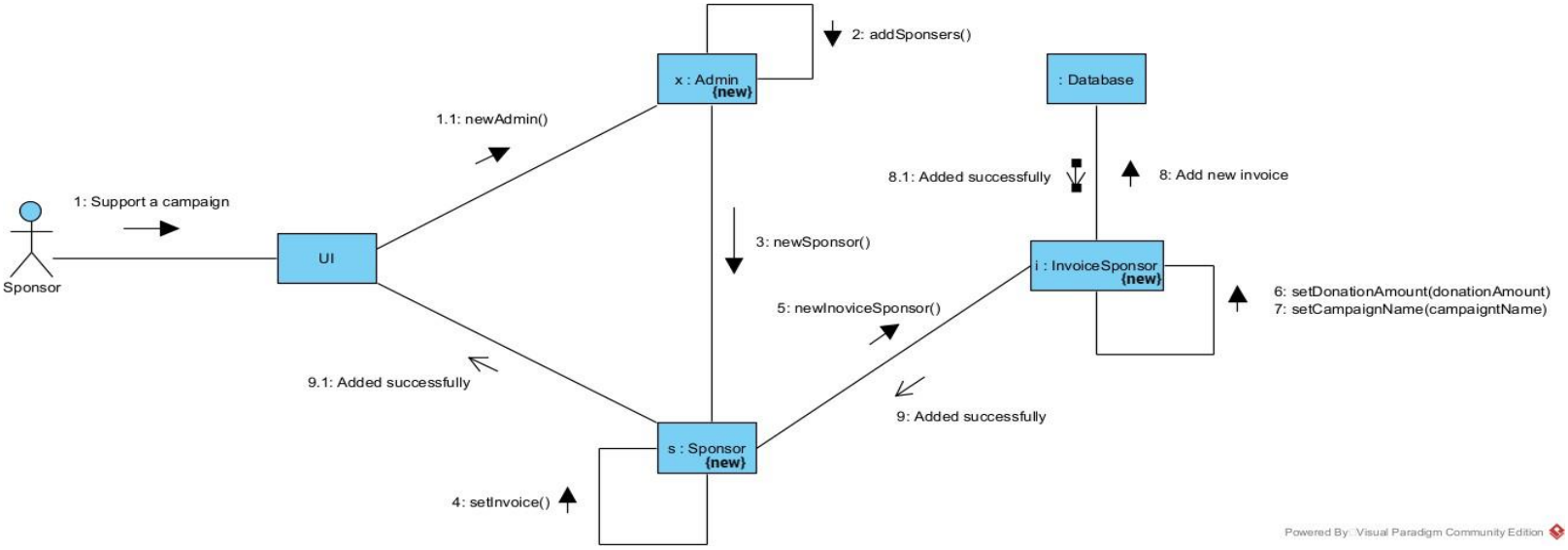
User displays his profile



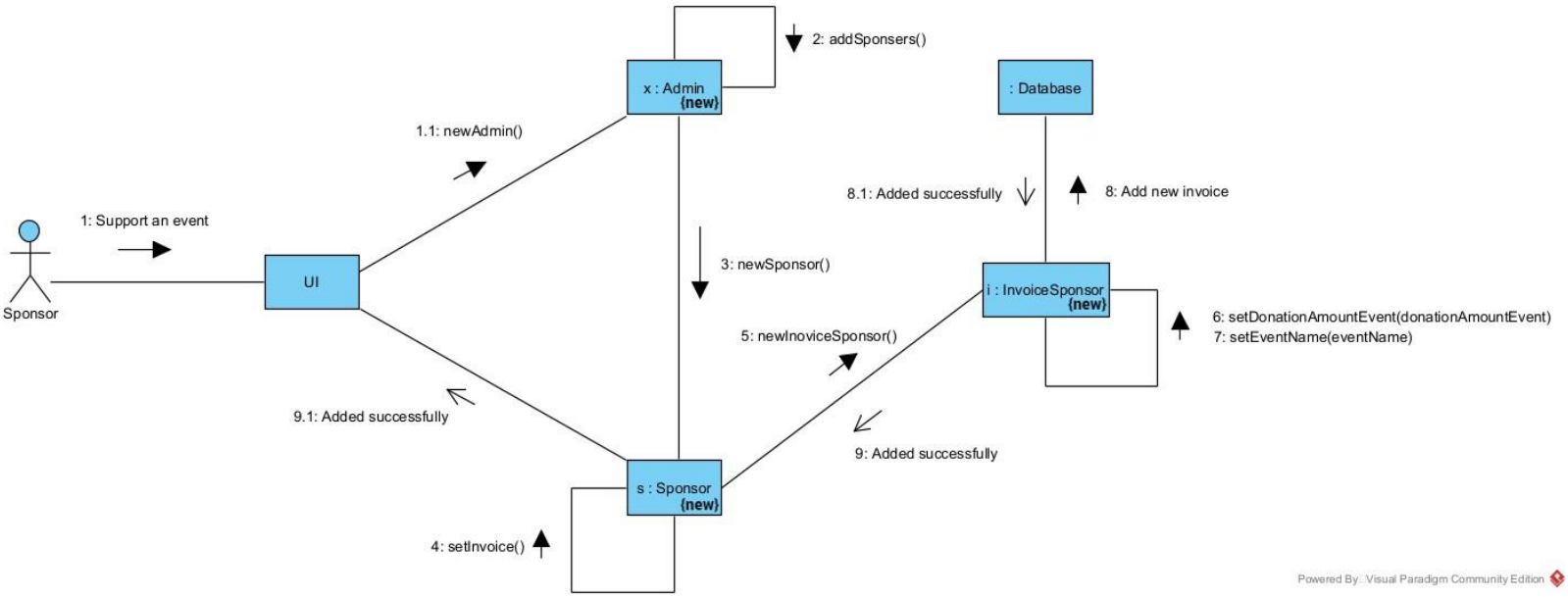
Sponsor Signup



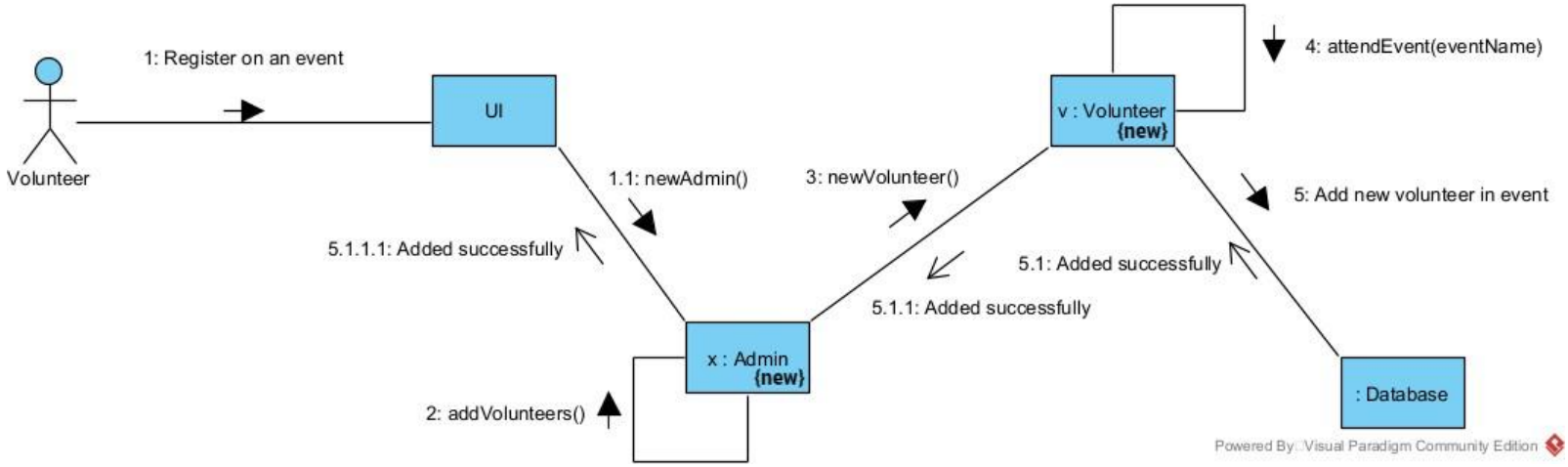
Sponsor supports the campaign



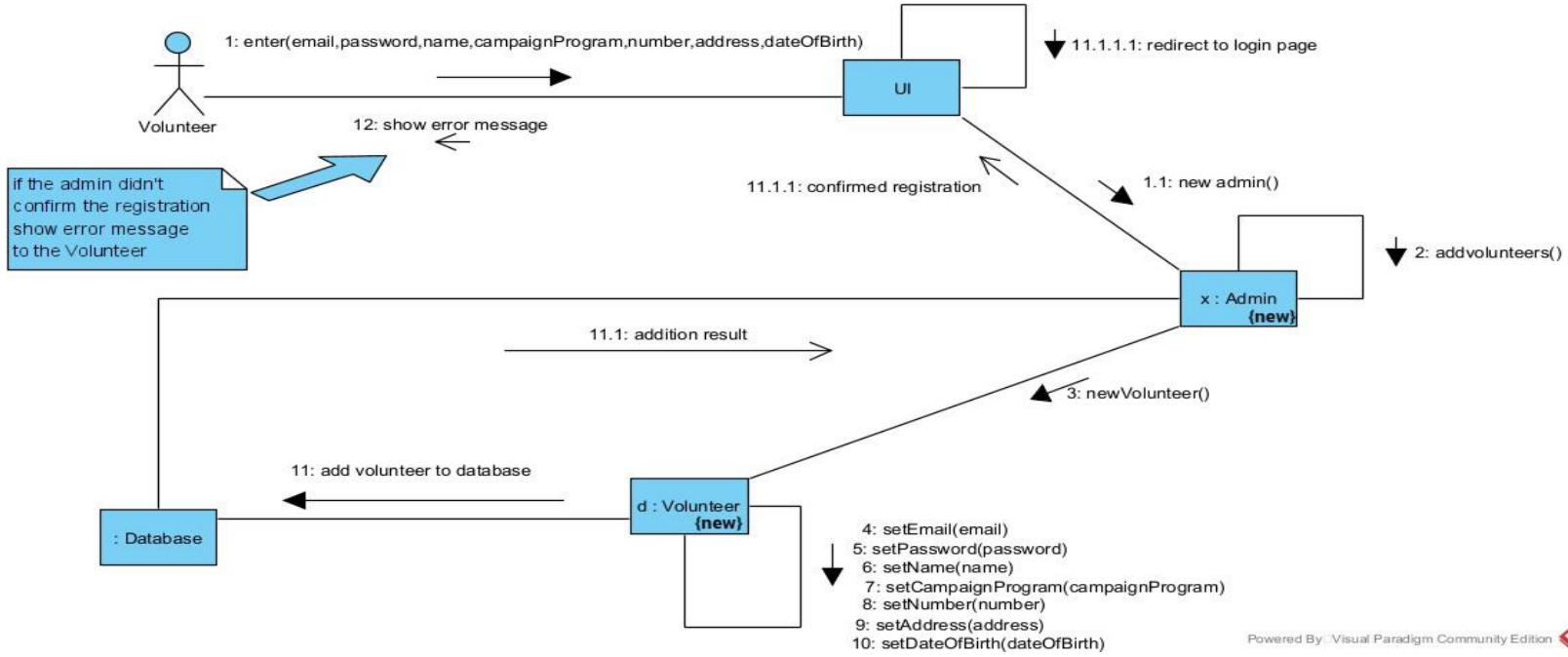
Sponsor supports the event



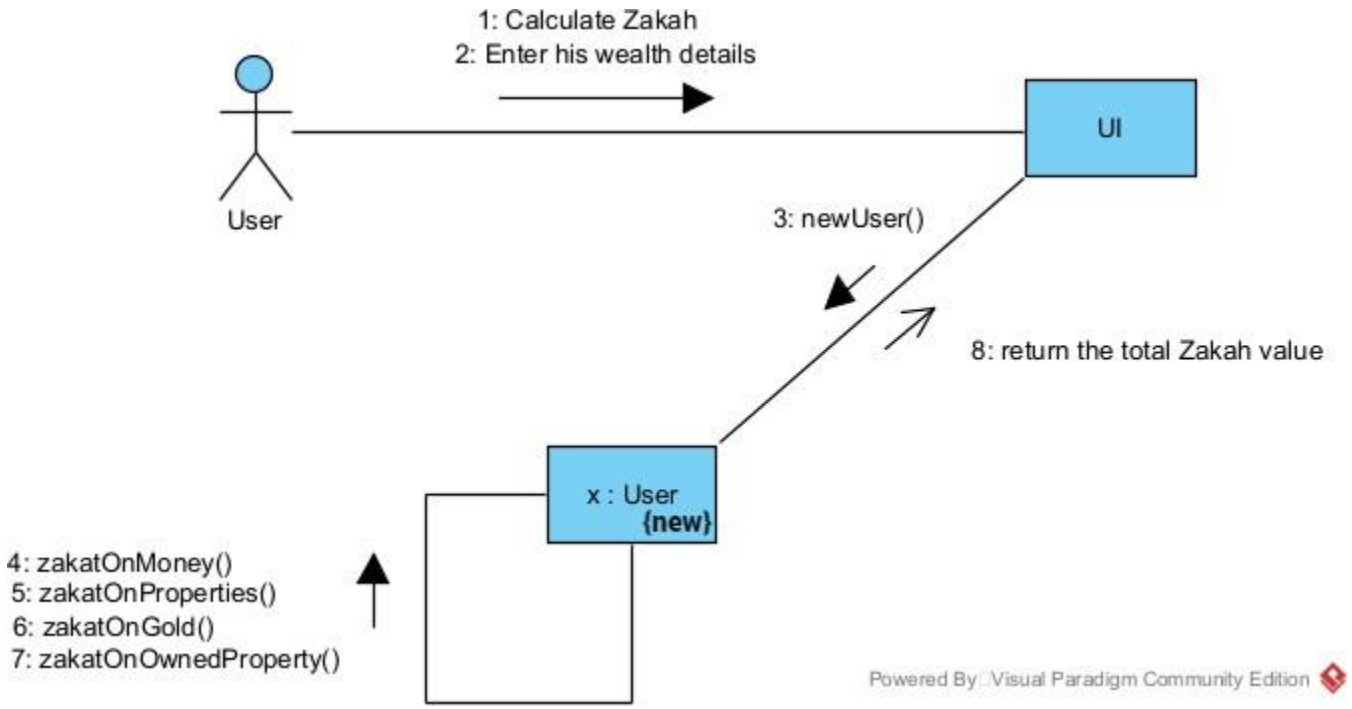
Volunteer registration on the system



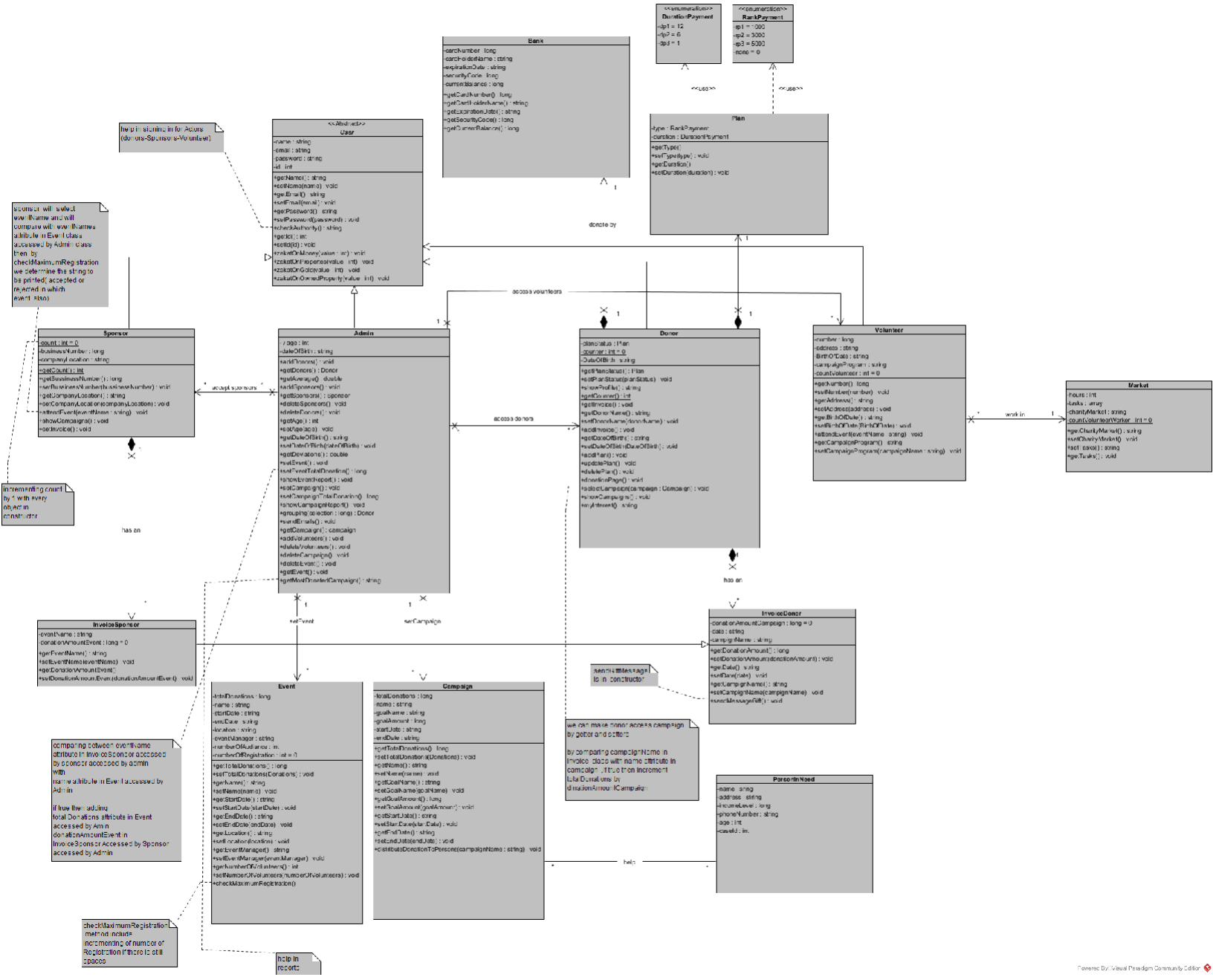
Volunteer Signup on the system



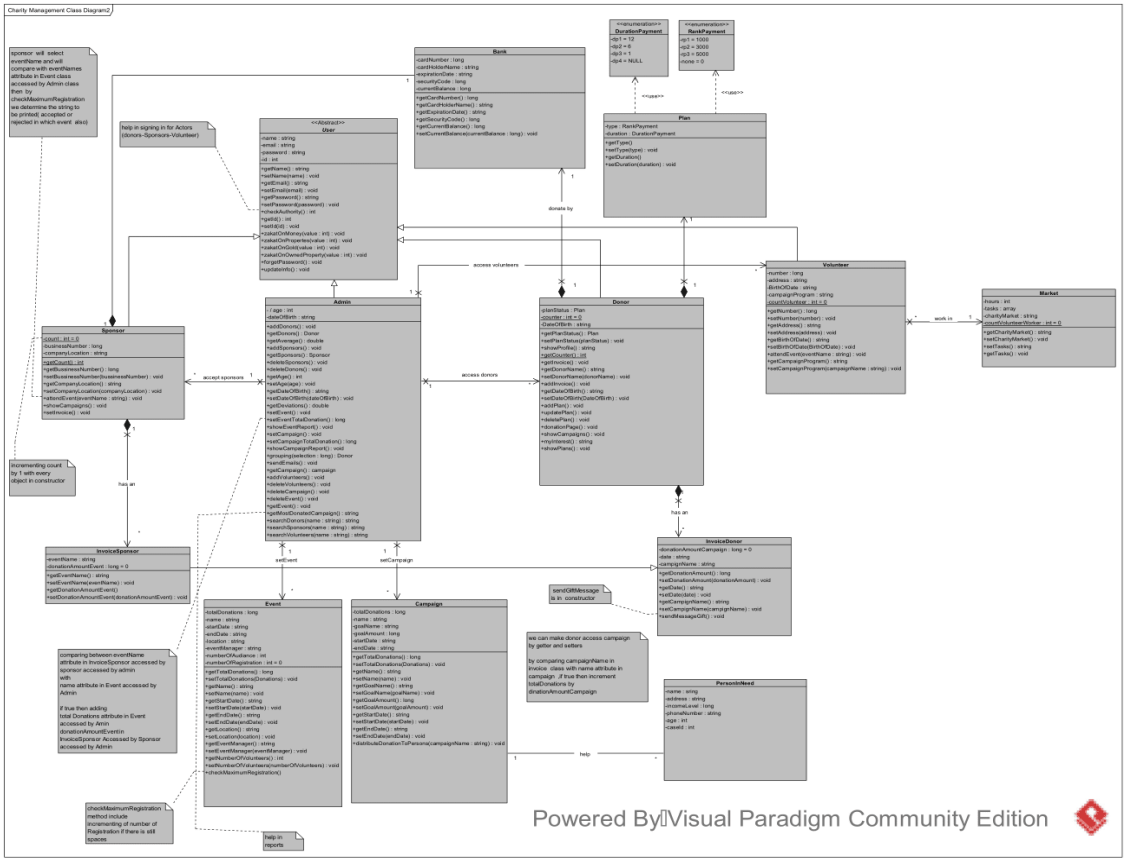
Zakat Calculator



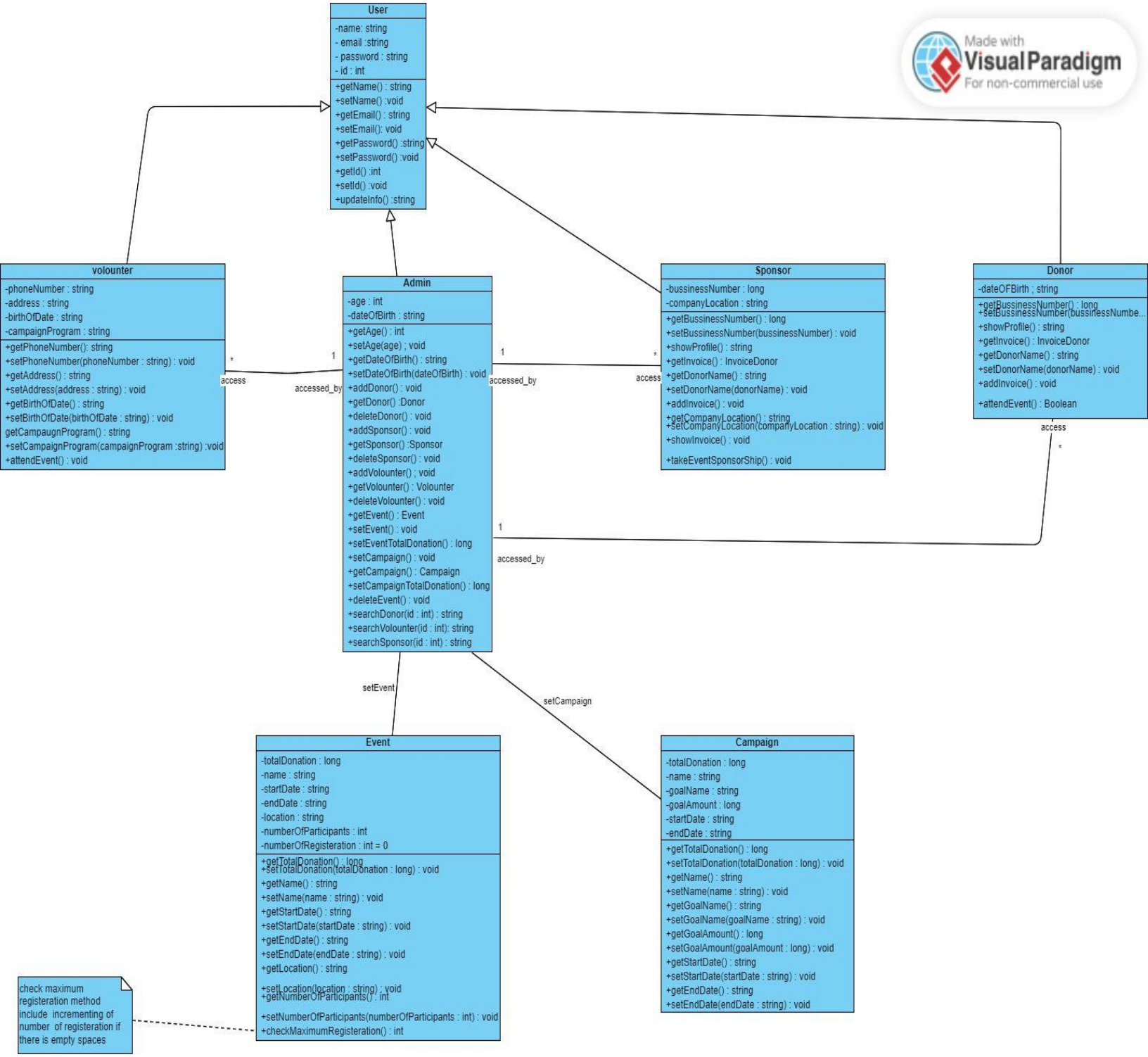
Class Diagram (Intermediate Version)



Class Diagram (Final version)



Final Class Diagram



Ocl

Context event
Inv MaximimNumberOfParticipants:
Self. numberOfParticipants<=100

Context event
Inv TotalDonation:
Self->select(totalDonation>=10,000)

Context event
Inv startDatebeforeEndDate:
Self. startDate<endDate

Context Admin
Invariant: self.setEvent->size()<=10

Context Admin
Inv validEventLoc:
Self.set event->select(location->notEmpty())

Context Admin
Inv validCampGoalAmount:
Self.set campaign. goalAmount->size()>0

Context Event
Pre: self. numberOfParticipants->size()>0 and self. numberOfParticipants->size()<=100
Post:
If self. numberOfParticipants->size()<100 then
self. numberOfParticipants++

```
    else
        self. numberOfParticipants=100
```

Context Admin
Invariant:self.accessDonor.getInvoice->select(d |d.totalAmount >100)

Context Admin
Invariant: self.accessVolounter->reject(d | d.campaignProgram->isEmpty())

Context Admin
Invariant:self.accessSponsor
.getId->includes(sponsorId)