Use Cases:

Designer:

Use Case: Register Designer:
- ● Participating Actor: Designer
- ● Entry Condition
  - ○ None
- ● Exit Criteria
  - ○ Designer created
- ● Flow of Events
  1. User requests to create designer
  2. CrowdSourceApp creates designer, refreshes display

Use Case: Create Project:
- ● Participating Actor: Designer
- ● Entry Condition
  - ○ Designer is registered
- ● Exit Criteria
  - ○ Project is created
- ● Flow of Events
  1. Designer requests to create project
  2. CrowdSourceApp creates project, refreshes display

Use Case: List Projects:
- ● Participating Actor: Designer
- ● Entry Condition
  - ○ Project(s) have been created by the designer
- ● Exit Criteria
  - ○ List of projects are shown
- ● Flow of Events
  1. Designer requests to list projects
  2. CrowdSourceApp displays a lists of projects created by the designer, refreshes display

Use Case: Delete Project:
- ● Participating Actor: Designer
- ● Entry Condition
    - ○ Project has been created by designer
    - ○ Project has not been launched
- ● Exit Criteria
    - ○ Requested project has been deleted
- ● Flow of Events
    1. Designer requests to delete project
    2. CrowdSourceApp deletes the project, refreshes display

Use Case: Create Pledge:
- ● Participating Actor: Designer
- ● Entry Condition
    - ○ Project exists
- ● Exit Criteria
    - ○ Pledge created
- ● Flow of Events
    1. Designer requests to create pledge
    2. CrowdSourceApp creates the pledge, refreshes the display.

Use Case: Delete Pledge:

- ● Participating Actor: Designer
- ● Entry Condition
    - ○ Pledge exists
    - ○ Project is not active
- ● Exit Criteria
    - ○ Pledge is deleted
- ● Flow of Events
    1. Designer requests to delete pledge
    2. CrowdSourceApp deletes the pledge, refreshes the display.

Use Case: Launch Project:
- Participating Actor: Designer
- Entry Condition
    - Requested project exists
    - Project is not yet active
- Exit Criteria
    - Project has been launched
- Flow of Events
    1. Designer requests to launch project
    2. CrowdSourceApp launches project, refreshes display.

Use Case: Review Project Activity:
- Participating Actor: Designer
- Entry Condition
    - Project exists
- Exit Criteria
    - Project activity displayed
- Flow of Events
    1. Designer requests to review activity(ies) of a specific project
    2. CrowdSourceApp display an activity history regarding the project, refreshes display

Use Case: Satisfy Deadline:
- Participating Actor: None
- Entry Condition
    - Deadline has passed
    - Funding goal is met
- Exit Criteria
    - Deadline is declared as satisfied
- Flow of Events
    1. System timer detects deadline has passed and the funding goal was met
    2. CrowdSourceApp declares the deadline as satisfied, refreshes display

Use Case: Failed project:
- ● Participating Actor: None
- ● Entry Condition
    - ○ Deadline has passed
    - ○ Funding goal not met
- ● Exit Criteria
    - ○ Pledges are canceled
    - ○ Funds returned to supporters
- ● Flow of Events
    1. System timer detects deadline passed and the funding goal was not met.
    2. CrowdSourceApp removes project from search list, returns funds to reporters, refreshes display

Supporter:

Use Case: Register supporter:
- ● Participating Actor: Supporter
- ● Entry Condition
    - ○ None
- ● Exit Criteria
    - ○ A supporter created with a fixed budget and a unique email address
- ● Flow of Events
    1. Supporter requests to create supporter
    2. CrowdSourceApp creates supporter based on a unique email address filled, and refreshes display showing a successful message

Use Case: Search Projects:
- ● Participating Actor: Supporter
- ● Entry Condition
    - ○ Project(s) have been created
- ● Exit Criteria
    - ○ A list of project(s) are shown, sorted by search criteria
- ● Flow of Events
    1. Supporter requests to review projects by project type
    2. CrowdSourceApp displays a list of projects that are still active, sorted by search criteria, and refreshes the display.

Use Case: Review supporter activity:
- Participating Actor: Supporter
- Entry Condition
  - Supporter has done some activity in the app
- Exit Criteria
  - A list of activity history done by the supporter shown
- Flow of Events
  1. Supporter requests to review supporter activity
  2. CrowdSourceApp display a list of activity history done by supporter, refreshes display

Use Case: Add funds:
- Participating Actor: Supporter
- Entry Condition
  - Supporter account exists
  - Project is created
- Exit Criteria
  - A fixed amount of $ Funds added to supporter's account balance
- Flow of Events
  1. Supporter requests to increase funds by a fixed amount of $
  2. CrowdSourceApp adds a fixed amount of $ of funds to the account balance of the supporter

Use Case: View project:
- Participating Actor: Supporter
- Entry Condition
  - Project is created and currently active
- Exit Criteria
  - Project information displayed
- Flow of Events
  1. Supporter request to view an active project
  2. CrowdSourceApp display the project information, refreshes display

Use Case: View pledge:
- Participating Actor: Supporter
- Entry Condition
  - Pledge is created in a specific active project
- Exit Criteria
  - Pledge information displayed
- Flow of Events
  1. Supporter request to view a pledge from a specific active project
  2. CrowdSourceApp display pledge information, refreshes display

Use Case: Claim pledge:
- Participating Actor: Supporter
- Entry Condition
  - Pledge is created in a specific active project
  - Supporter can afford pledge
  - Supporter maximum has not been reached on the pledge
- Exit Criteria
  - The pledge has been claimed.
- Flow of Events
  1. Supporter request to claim pledge from specific active project
  2. CrowdSourceApp adds the supporter to that pledge under its project, reduce the amount of money described by that pledge in the supporter's balance, and refreshes display

Use Case: Direct support:
- Participating Actor: Supporter
- Entry Condition
  - Project is active
  - Supporter can afford a fixed amount of money claimed by supporter
- Exit Criteria
  - A fixed amount of money has been transferred from supporter's fund balance to requested project
- Flow of Events
  1. Supporter requests to support a fixed amount of money from a specific active project.
  2. CrowdSourceApp reduces the fixed amount of money from supporter's balance and adds direct support to the project, and refreshes the display showing a successful message
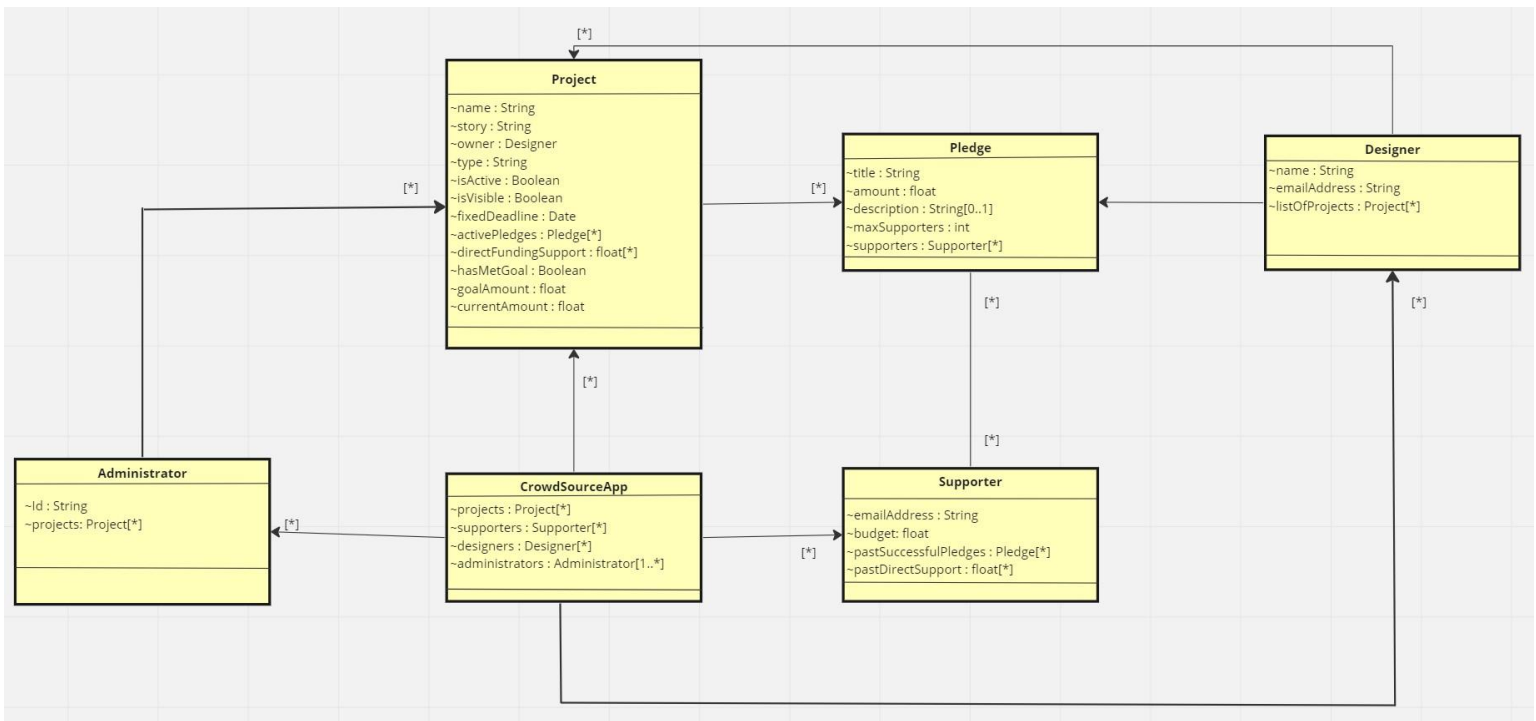
Administrator:

Use Case: List projects:

- Participating Actor: Admin
- Entry Condition
    - Project(s) exist
- Exit Criteria
    - List of projects displayed
- Flow of Events
    1. Admin requests list of projects
    2. CrowdSourceApp lists all projects with details, refreshes display

Use Case: Delete project:
- Participating Actor: Admin
- Entry Condition
    - Project has been created
- Exit Criteria
    - Project is deleted
- Flow of Events
    1. Admin requests to delete a project
    2. CrowdSourceApp deletes project, returns funds to reporters, refreshes display showing a successful message

Class Diagram:

Find a better resolution in the zip file this document came with

Project:

- name : String : single attribute
- story : String : single attribute
- owner : Designer : single attribute (Object)
- type : String : single attribute
- isActive : Boolean : single attribute
- isVisible : Boolean : single attribute
- fixedDeadline : Date : single attribute (Object)
- active Pledges : Pledge[*] : array
- directFundingSupport: float[*] : associative array
- hasMetGoal : Boolean : single attribute
- goalAmount : float : single attribute
- currentAmount : float : single attribute

Pledge:

- title: String: single attribute
- amount: float: single attribute
- description: String: single attribute
- maxSupporter: single attribute
- supporters: Supporter[*]: array

Supporter:

- emailAddress: String: single attribute
- Budget: float: single attribute
- pastSuccessfulPledges: Pledges[*]: array
- pastDirectSupport: float[*]: array

Designer:

- name : String : single attribute
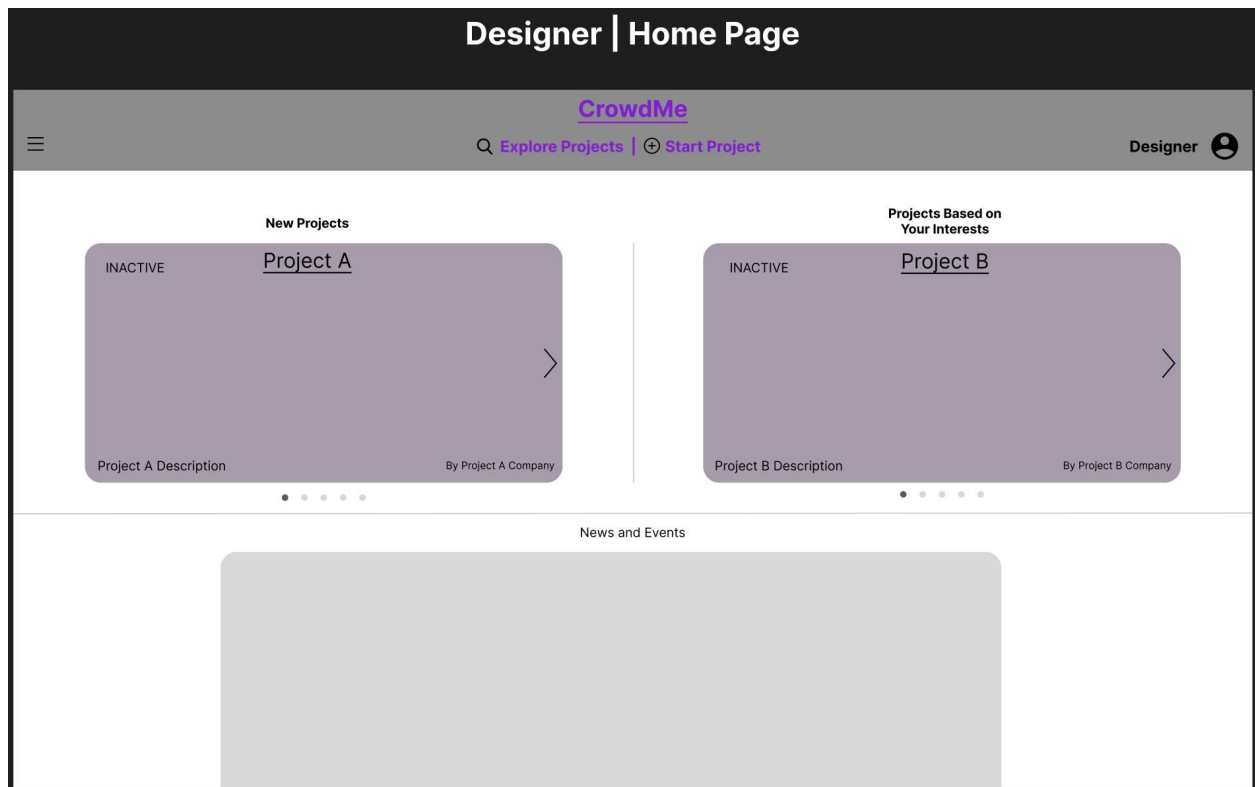- emailAddress : String : single attribute
- listOfProjects : Project[*] : array

Administrator:

- name : String : single attribute
- projects : Project[*] : array

CrowdSourceApp:

- projects : Project[*] : array
- supporters : Supporter[*] : array
- designers : Designer[*] : array
- aministrators : administrator[*] : array

Storyboards:

Find a better resolution in the zip file this document came with along with all 33 other storyboards for this deliverable. There is one for each use case plus extras for context.