# Community Fund

CSC309 Team Project: Phase 3

Team Members: Ou Ye, Daniel Velev, Andrew Zhao

Public URL (Website): http://ec2-52-1-96-34.compute-1.amazonaws.com:8080/

Public URL (Source Code): https://github.com/MapleWorld/CSC309/tree/master/phase3

## Summary of Changes

- Changes made to project schedules and estimates on timelines/phases.
- Update and clarifications on changes in the overall structure of the website.
- Expansions and changes to personal roles; members doing some different or additional tasks than initially planned.
- Had to adjust roles and expectations accordingly after learning Lucas dropped the course.
- Made more specific use cases and scenarios handling the different functionalities of the website.
- Added the datatypes of variables in the database, as well as furthering definition of the main entities.
- Made a formal testing schedule for the weeks after Phase II's deadline

# 1. Feature and Functionality Specification

Community Fund will be a crowd funding website that brings communities and their residents together. It aims to do this by putting power and opportunity for improvement into the hands of those that know the neighborhood best and can provide the most positive change – the residents themselves. Further, Community Fund aims to align those with similar interests worldwide to support projects for which they feel passionately about.

## 1.1 Registration

Anyone with internet access has the ability to register for the site. This means that anybody can begin helping a cause of their choice within minutes. Upon registration, new users are asked for information to be used for authentication, as well as their location and interests. This places them automatically into the communities corresponding to their interests (one community per interest) and their location (one community for the location).

**Scenario**: New user visiting website, wanting to register.

The other method is through clicking “Sign In” on the top right of the menu bar, and then clicking “Sign Up!” from the drop down menu that appears.

The user is taken to a registration page, where they input the information necessary to set up a general account on the Community Fund website. This information includes the user’s name, their email, and password.

**Scenario**: Existing user visiting website, wants to sign in.

The user has two ways to do this:

Through clicking “Account” on the top menu bar, which will redirect the user to the subsequent “Log In” page.

The user can click on the “Sign In” on the top menu bar and from there can input their information to sign in.

## 1.2 Communities and Profiles (Social Networks)

Registration information inputted by the user initially is used to create a profile for every new user. As he or she continues to be active on Community Fund, their profile will gain new additions – reputation score, projects that have been funded or initiated by the user, or communities that they now belong to.

Every community has a page which serves as a social network for the members of the community. Through this page, the user can stay up to date on both current and older projects occurring within the community. For every project started, a post is made on the community's social network, on which users can comment, share, or be linked to the project's page to donate. All members of the community are automatically considered friends.

**Scenario**: New user joining communities.

The ideal scenario has the user, upon registering, entering information such as their location and interests. This in turn will automatically place the users into the communities they belong to. They then will be able to click to go to the social network pages for these communities. Each community will have its own page, and the user can visit these to see information pertaining to that community there. This will include conversations and project postings.

## 1.3 Projects

When a project is created, its information gets displayed onto a project page. Every project has its own page, which features a description of it, a timeline for keeping track of milestones, links to rate or donate to the project and a meter showing the amount remaining to reach the financial goal. This is how users that are looking to fund projects can be exposed to them - data gets fed into social networks for the communities that this project belongs to. This helps keep users informed and targets those who may be the most willing to donate.

Any registered user has the ability to search for or view projects that they feel passionately about and want to donate to. Users view projects on the community page through a toggle showing either the most recently created projects, or the top ones based on a priority system that incorporates initiator rating, project rating, and amount remaining to reach the goal. This will help projects that are in the most need for funding, or which are from the most trusted initiators. This system will encourage a better matching between those that need funding and those that are willing to give it.

**Scenario**: User wants to search for a project to donate to.

The users view the most important projects in many ways: they will be displayed to them on the home page, and they will be visible to them on the community pages for which they are a part of. These are the projects that are most important to the users seeing them.

In the ideal scenario, there is a listing that each user can look at, sorting projects in terms of importance/relevance to the user.

So, the user can view projects in this listing, click on them to view more detailed information such as milestones, and then choose to donate or not – they do this through clicking "Donate", then choose the amount and whether to buy a product or simply donate. Lastly, they will provide a rating for the user, and this constitutes an interaction.

## 1.4 User Roles

Every user is given the power to choose how to use their account and time on the website. A user can choose to view available projects to fund, and contribute towards them. A user can also opt to initiate a project for others to fund.

Upon funding, the user will be told how close the project is to its goal, and how they have contributed to that goal. The user will also be notified about the completion of projects that he or she has funded. The user also has the ability to choose between donating money and paying for a product in return, to be shipped after completion of the project. Both methods of payment will contribute to the financial goal of the project.

To initiate a project, users create a new project page and are asked to input information that will help share it to the public. This information includes a description of the project, the financial goal required to complete it, the communities it is a part of, and prices required to buy the item if that's how the user wishes to contribute. The information is then placed on the project page to be seen by users, who link to these pages through the communities they are associated with.

**Scenario**: User wants to initiate a project.

The user will click on Initiate, and then fill in a form for all of the important aspects needed to start the project, as listed above.

They then have a project open and running, and receive regular updates on milestones and donations given, as well as ratings received. When the user receives a donation, they will then have taken part in an interaction, and rate the corresponding user in return.

**Scenario**: User is notified about the completion of a project that they have provided funding for.

The user receives notification through a thank you message, as well as information about the following steps for this project. If they chose to buy a product, they will be notified regarding the timeline for its creation/delivery to the user. Further, they will be prompted to check out and use the new product/project, and in turn leave another rating/comment.

## 1.5 Reputation

Upon interaction, users can rate one another based on their dealings with each other. Interaction includes funding or communication. Reputation is given in the form of a 1 – 5 star rating. Reputation will assist users in deciding which projects they want to donate most to.

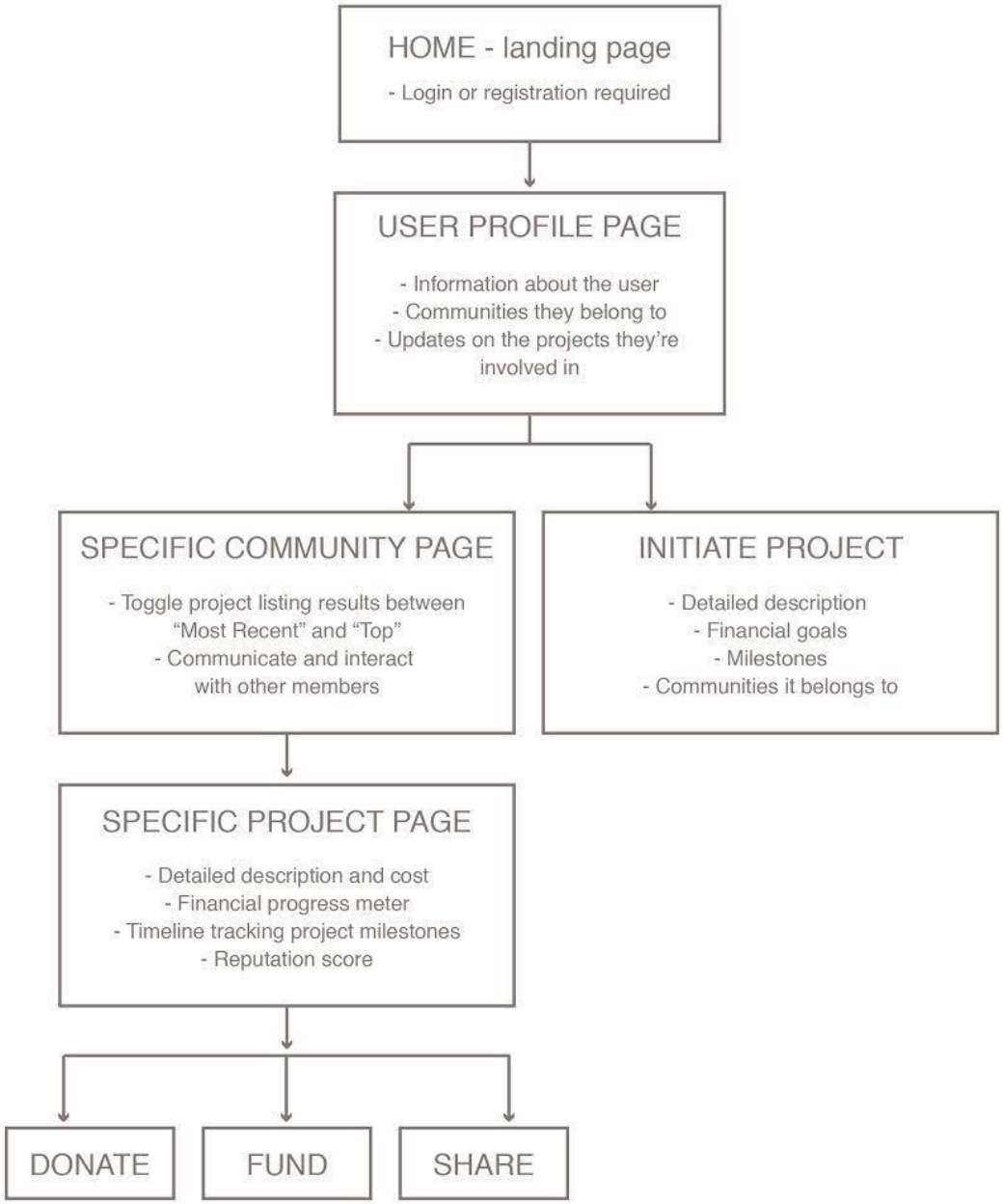
Every user's rating will be placed on their profile page, and that information will also be on the project pages for projects that they have initiated. When projects are completed, users are notified and encouraged to try out the finished product and leave another review.

### 1.6 Potential User Scenario: Joe

Joe visits the new Community Fund website after hearing about a skating rink that is being funded in his community. He is passionate about hockey and loves to play it. Joe creates an account specifying his location and interests, and then visits the social network page for his community. He sees the skating rink project as a post on the recent tab of the social network page, reads some positive comments on the idea, and in doing so feels even more willing to contribute to the cause. Joe clicks the post and is directed to the project page for skating rink, where he sees more detailed specification regarding the project and its progress. He chooses to contribute through donation, $50. He is notified that he has helped the project reach its goal by an additional 3%, and exactly how much is remaining before completion.

Joe then decides to revisit his community's page and sort available projects by "top". He doesn't see another he wishes to contribute to – at least not today – but gets a better idea of what kind of projects are initiated on the site, and even has some potential ideas of his own. Over the coming days and weeks Joe checks back to see progress on the skating rink, and he also enjoys viewing the newest projects to be added.

CommunityFund - User flow chart



## 2. Project Plan

### 2.1 Personal Roles:

- Lucas Pan: dropped the course.
- Drew Zhao will be working on both front and back end, maintaining the connection and code quality between the database and the server. Drew will also work on the preliminary testing of the site, noting areas of interest and areas that need to be discussed in group meetings or taken care of by individual members.
- Ou Ye will be working on back-end and is responsible for the design and creation of the database. Ou will also get the website up and running onto the web server. He will handle the Amazon Web Service that we will use for the web site.
- Danny Velev will be working on the front end UI, design and implementation of the website. He'll combine the efforts of the entire group and make finalizations to the report.

### 2.2 Team Organization:

- The team will communicate and coordinate online using mostly Facebook and Skype for discussion.
- A decision will be made by a proper technique such as paired comparison analysis, analytic hierarchy process or multi-voting.
- There will be a weekly meeting either in person or online for discussion and decision making.
- The weekly update is put together through collaboration and communication, so all members are aware of the current status of the project, and what work is being done.
- The project will be managed using Git for version control.

### 2.3 Project Milestones:

- Phase 1: Design and Development (Feb 5)
    - Design mock-ups of UI based on the required functionality
    - Write the HTML and CSS code for the mock-ups
    - Implement simply front-end functions using JavaScript and JQuery

- Phase 2 Development (March 1)
    - Features server-end functionality for each page using Node.js
    - Modify UI based on the new needs
    - Database Design and Software Architecture
    - Construct Database
    - Integrate test the website

- Phase 3: Launch (March 14)
    - Apply the test plan in section 5
    - Obtain domain and host the web page live
    - Implement simple web page with RESTful design
    - Implement user authentication and session
    - Create prototype MySQL database for testing purpose
    - Establish connection between the database and the web page

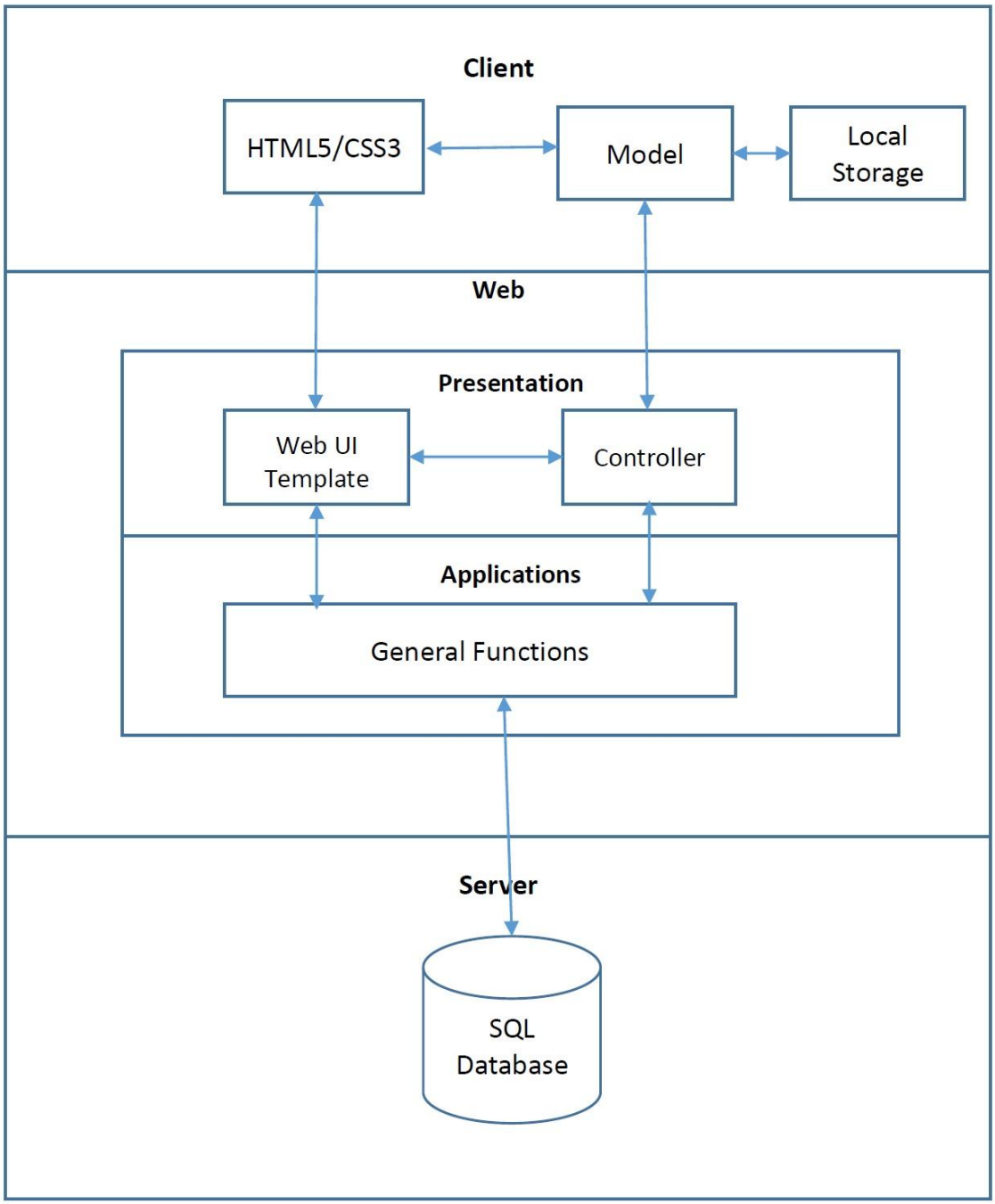
- Phase 4: Maintenance and Code Review (April 1)
    - Implement and complete all features and fully operational
        - Dynamically send data back to front-end and display them
        - Funder, initiator, funding, create project, project(s), and community
    - Add in additional information, such as About, Contact, etc.
    - Design, code review and refactoring
    - Extensive testing and debug

The above Phases represent the general structure and schedule for the implementation of our project.

# 3. Software Architecture and High-level Design

## 3.1 High-level system architecture and interaction with environment

The following figure shows the high-level system architecture of our website which consists of a client layer, a web layer and a server layer.



## 3.2 System decomposition

- Client Layer

  Users will communicate with other users and the website through the client layer (browser). Once a user enters a URL belongs to the website, the server will let the browser to present the designated web page which is to download the HTML/CSS/JavaScript files.

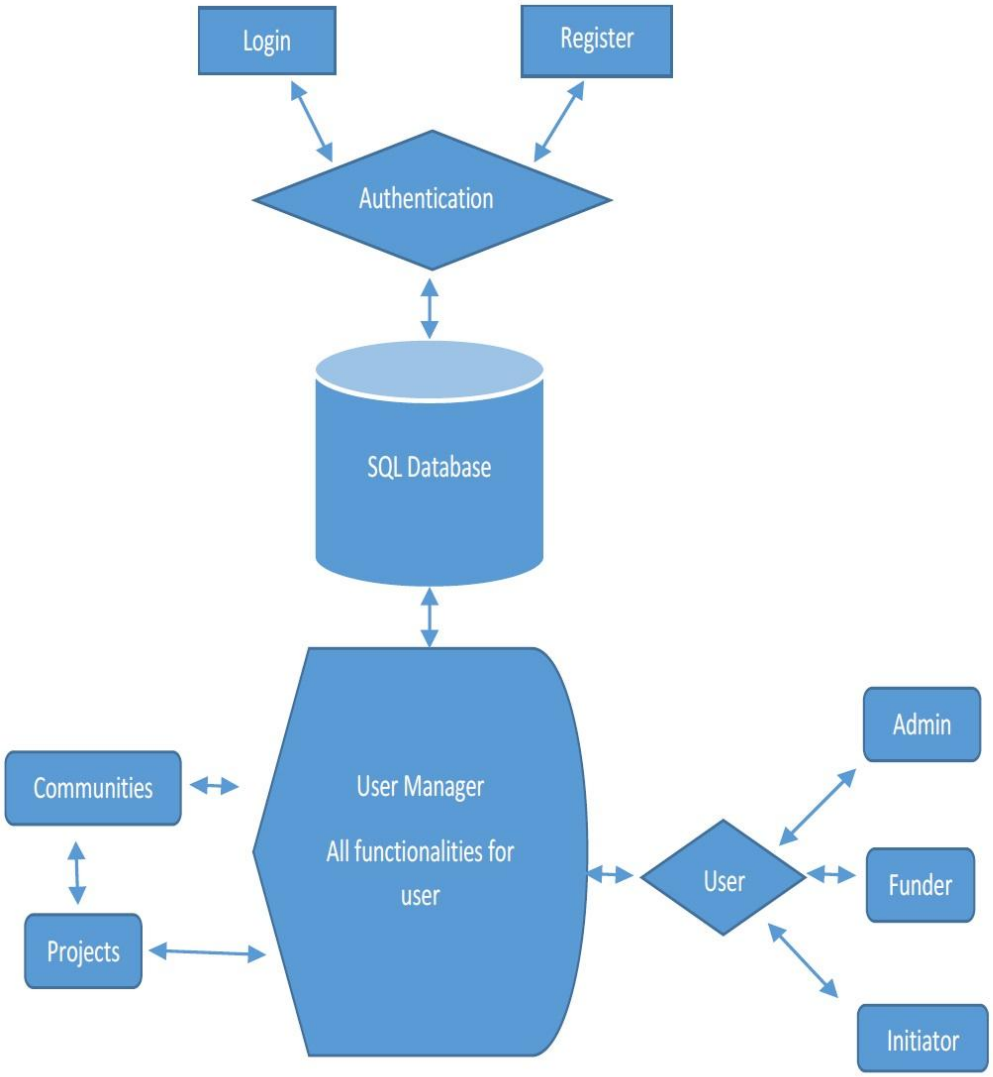
- Web Layer

  This layer works just like a processer in a computer. It builds the connection between the server and client. It has the ability to transfer and process data from one end to another. Other than uploading the web pages to a client, it also features most functionality of the website. For example, it categorizes the projects automatically and saves them to specified places; it processes the users request to find a specific project. All the functions will be implemented in this part of system.

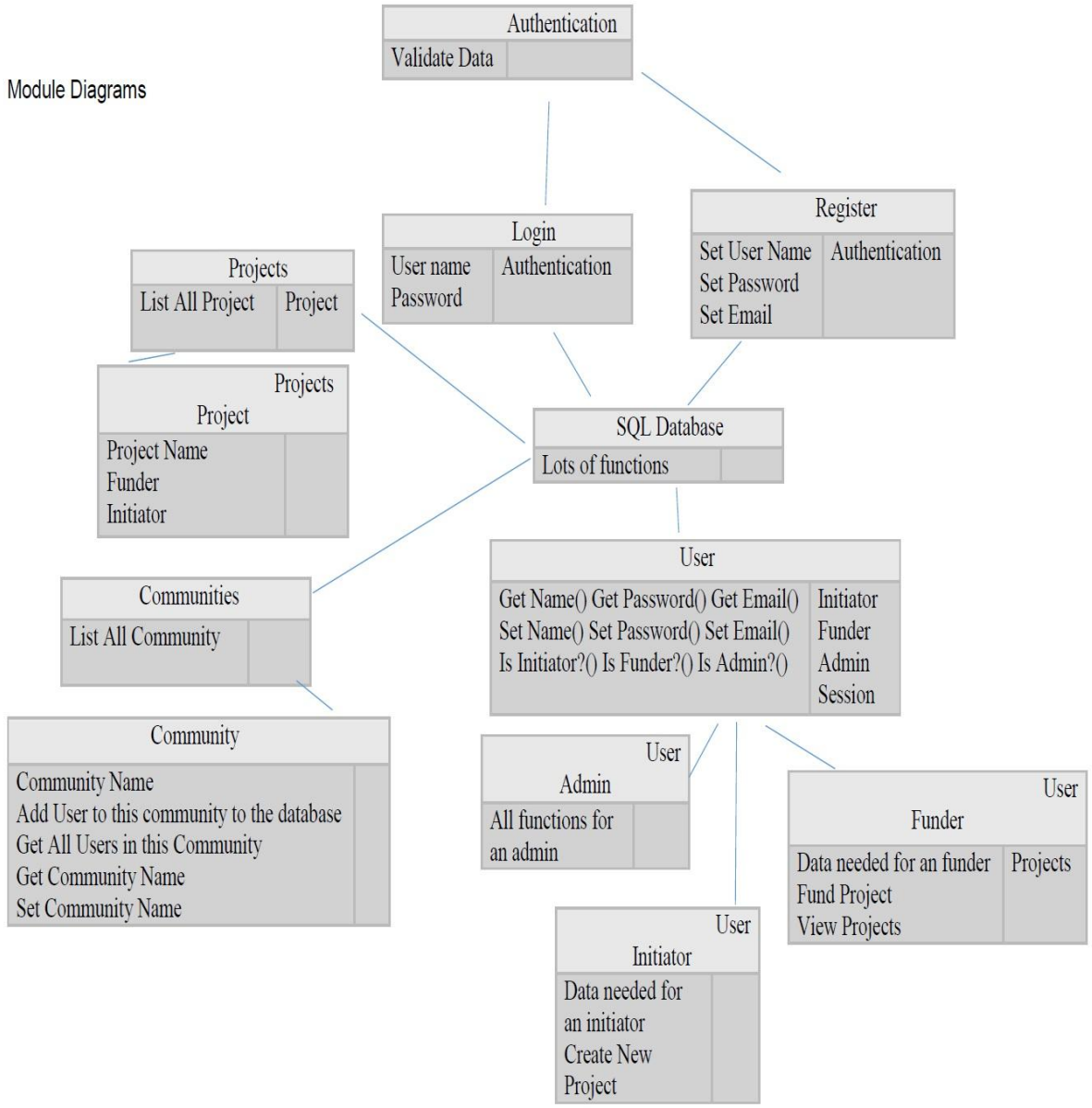
- Server Layer

  This layer can be viewed as the memory of a computer. It can only perform two instructions: Load and Read. It stores all the functions/methods, data and web pages.

The following figure shows how each part of the system is implemented actually in Community Fund:



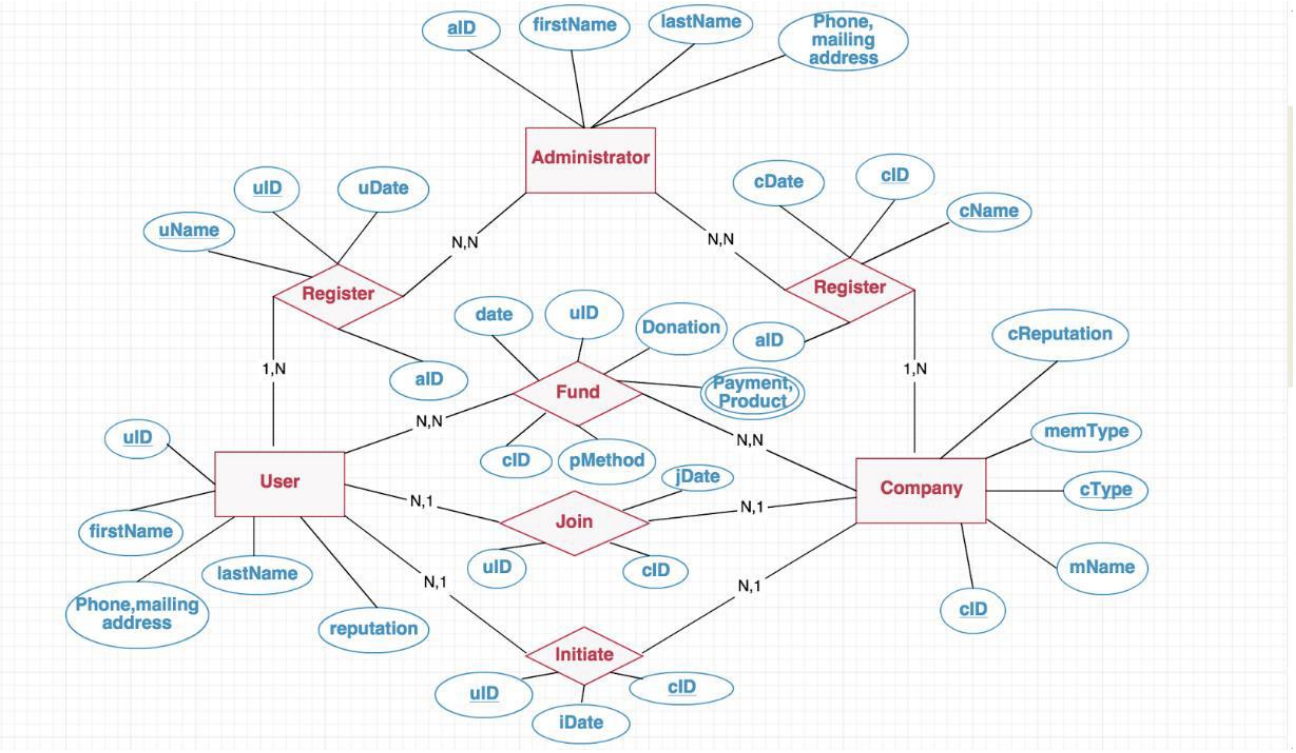
## 3.3 System design



# 4. Information Representation

## 4.1 Data Model

We will be handling our system database through SQL. The diagram below is our ER explaining roughly how we would want the database to be stored:




There will be three main entities.

The first entity would be admin of the server. They are the administrator of the server, which in this case is going to be people in our group. (This group of people is not part of the user entity)

The second entity is our users of the server. They are the people who are going to be using our webpage services whether is a funder or an Initiator, they will both be considered as a User. We do not want to separate the two types of users at the beginning of account registration, We want to avoid constrictions on users to only have an account that can fund, then they suddenly want to make a startup company, they have to make an Initiator account, or vice versa. A User can fund and also initiate a company on the same account, this gives the user more flexibility and create less hassle.

The third entity would be on the group/startup company side. These are startup companies initiated and need to be funded to continue their product/services.

The relationships between each of the entities are:

- User registers from administrator server.
- Company registers from administrator server.
- User can join a company.
- User can fund a company by purchasing their goods and services or just purely donation.
- A user can register a new startup company.

## 4.2 The schema:

Administrator(aID, firstName, lastName, e-mail, Phone, Address)

U-register(uID, uName, uDate)

C-register(cID, cName, cDate)

User(uID, firstName, lastName, reputation, Phone, e-mail, Address)

Company(cID, cType, mName, memType, cReputation)

Fund(uID, cID, Donation, payment, product, pMethod)

Join(uID, cID, jDate)

initiate(uID, cID, iDate)

Join(uID) ⊆ User(uID)

Fund(uID) ⊆ User(uID)

Initiate(uID) ⊆ User(uID)

Join(cID) ⊆ Company(cID)

Fund(cID) ⊆ Company(cID)

Initiate(cID) ⊆ Company(cID)

Company(mName) ⊆ User(uID)

Datatypes:

Administrator(INT, STR, STR, STR, STR, STR)

U-register(INT, STR, STR)

C-register(INT, STR, STR)

User(INT, STR,STR, STR, STR, STR, STR)

Company(INT, STR, INT, STR, STR)

Fund(INT,INT, STR, STR, STR, STR)

Join(INT, INT, STR)

initiate(INT, INT, STR)

- The table Administrator will store the admin's aID, their first name and last name and their contact information.
- The table U-register contains the user id who registered, the name of the user and the date they registered.
- The table C-register contains the company/project ID, the name of the company and the registration date of the company/project.
- The table User contains the User ID their first name and last name reputation, and personal/business contact information.
- The table Company contains the company ID, the company type, members and their project/company reputation.
- The table Fund contains the user ID of the funder, the ID of the company, the amount of donation, the product/payment they made if funder is purchasing a product and the payment method they've used for that transaction, and the date of the transaction.
- The table Join enable User to join a company that they are interested in, contains uID of the joined user and the cID of the company (the invitation sent from).
- The table Initiate allows user to initiate on a new project/company, it contains the user ID who initiates a project, cID for registration purposes and the date the project was initiated on.

## 5. Test Strategy and Test Plan

Generally, testing is completed during production since the cost of fixing a defect in an integrated system could be extremely high. Testing strategies and responsibilities of 4 main areas are described below.

### 5.1 Code Quality (Drew)

For smaller websites with limited budgets, it is important to design with web standards, because it ensures the website's compatibility and minimize the cost on the future maintenance.

- We will use W3C validator to check (X)HTML documents and Cascading Style Sheets(CSS) files. And we will check whether all styles that appear in the (X)HTML are from CSS to ensure the separation between content and presentation.
- All the links will be tested regularly using W3C link Checker.

### 5.2 Accessibility (All)

- Using a screen reader to ensure that all links and "alt" attributes are descriptive for blind users.
- Test the web site on major browsers such as Internet Explorer, Firefox, Safari, Chrome, Opera, etc.
- We did testing on individual browsers included: Internet Explorer, Firefox, Safari, Chrome, and Opera. Made sure our website is compatible with most browsers.
- Using Google Chrome Developer Tools to simulate the web site on other devices including phones, tablets and so on.

### 5.3 Functionality (JavaScript: Drew; Node.js: Danny, Ou; Other: all)

Since in our project most functions will be implemented using JavaScript and Node.js, so here we mainly discuss how to test code written in these two languages. And the main strategy will be used is unit/integration testing.

A unit test is a test on a relatively small piece of code, usually a function or a method of a class. The process includes extracting the code, creating dummy inputs, and verifying the outputs.

We tested on our node.js code and all the JavaScript code we have so far, and they are working functionally in most browsers we tested in section 5.2.

After applying unit tests on the critical parts, they will be assembled together and perform integration testing on them as a whole.

- For JavaScript code, we will first use JSHint to briefly detect errors and potential problems. And perform unit/integration testing using KARMA.
- For Node.js code, the unit test will be performed using Mocha and Supertest library will help perform integration tests involved with servers.

### 5.4 Performance (Ou, Danny)

- We will use the Heroku cloud application platform to load/stress test the website with an add-on called Blitz.

## Test schedule:

**Week of Feb 6-12:** We did html lay out of the website, tested for errors in drop down menus and made sure the code is compatible with most popular browsers.

**Week of Feb 13-19:** Some of JavaScript was implemented this week (register page, login page) and unit testing was done on it. Further compatibility testing on the main page.

**Week of Feb 20-26th:** Part of backend was implemented in the server backend, it was not compatible with CSS we had though.We some deleted CSS on the html pages to fix the compatibility issues with the server.

**Week of Feb 27 - March 5th:** Almost fully implemented in the server end, we are still trying to figure out the compatibility issues with the server and our html webpages.

**Week of March 6th – March 13th:** Deleted log-in and register page because of compatibility issues, and made a drop down menu on the main page for registration and user-login. Different test cases for user log in were tested.

- Invalid password
- Invalid username
- Log-in success
- Duplicate input

Future testing schedule:

Test for user having beensuccessfully added to the database if registered.

Test for project having successfully been added to the database if proposed.

Test for functionality in funder’s funding option.