CSC309 Phase 2 Description

Shihan Ai (g3aishih)
Timothy Lai (c2laitim)

2.1 Feature and Functionality Specification [25%]

***Root** (First page the user sees)*

- ★ Users can log in or sign up
- ★ Links to ***Sign up***
- ★ Links to ***Log in***

Sign up

- ★ Allows users to enter information and create an account
- ★ Links to **Set up profile**

Log in

- ★ Allows users to enter information and log in
- ★ Links to **Lost password**
- ★ Links to **Main page**

Lost password

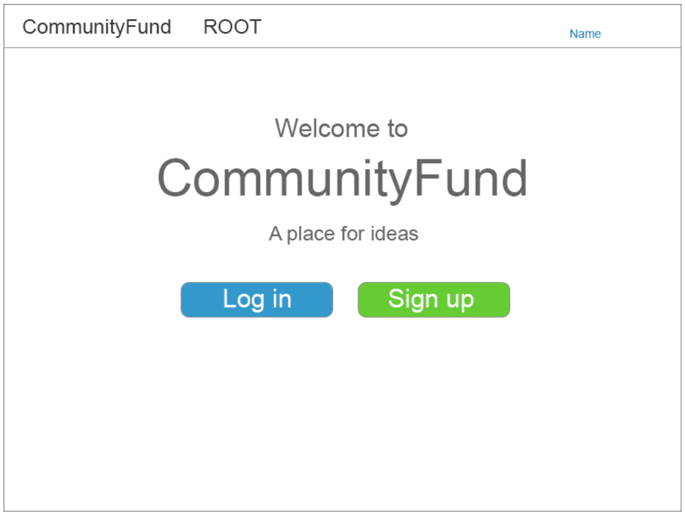
- ★ Allows users to enter an email and retrieve their password
- ★ Links to **Root**

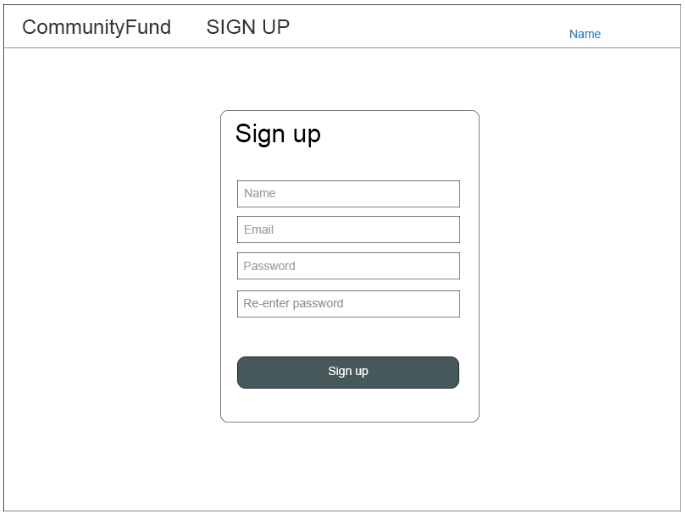
Set up profile

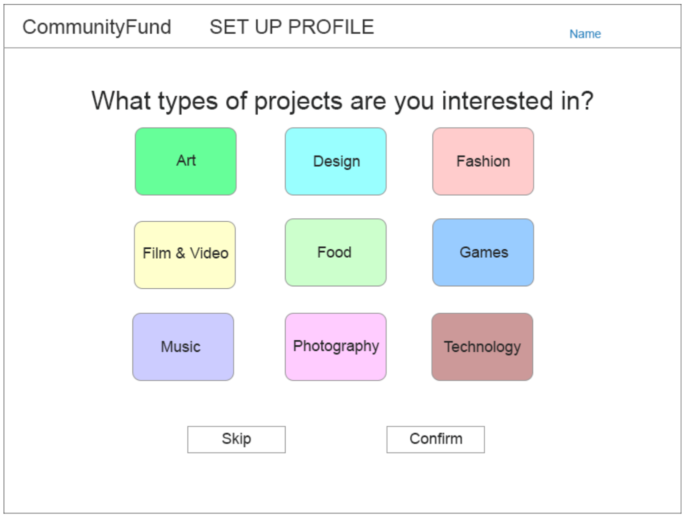
- ★ Users enters information regarding the communities that they would like to be apart of
- ★ Select as many categories as needed, then confirm
- ★ Links to **Main Page**

Main pageDisplay up to three of the most popular projects in the user's community/interests and their attributes

- ○ Attributes of a project include
    - ■ Project ID
    - ■ Community/Tags
    - ■ Initiator
    - ■ # of funders
    - ■ Money raised
    - ■ Description of project
    - ■ Fund goal
    - ■ Date created
- ★ Displays up to three of the most funded projects that the user funds
- ★ Displays up to three of the user's initiated projects
- ★ Able to browse all projects (instead of just three). Links to **Project List**
- ★ Clicking on a project links to its **Project Page**







What types of projects are you interested in?

- ★ Allows users to create a new Project, links to **Create Project**
- ★ Drop Down menu appears by clicking the user's name
  - ○ Dropdown menu can
    - ■ View profile, links to **User Profile**
    - ■ Account settings, links to **Account Settings**
    - ■ View admin page, links to **Admin page**
    - ■ Log out, links to **Root**

Account Settings

- ★ Change password
- ★ Links back to **Main Page**

User Profile

- ★ View/Change name
- ★ View/Change project interests
- ★ View/Change location interests
- ★ View reputation (like/dislike)

Project List

- ★ Displays a table with project info, table is populated appropriately (eg, if a user wants to view all of the projects in a community, the table will only contain projects in that certain community)
- ★ Clicking on the project name will link to the **Project Page** of that project

Project Page

- ★ View information about a specific project
- ★ Allows users to donate to the project
- ★ Allows users to leave a rating on the project initiator
- ★ Initiators cannot donate or rate their own project

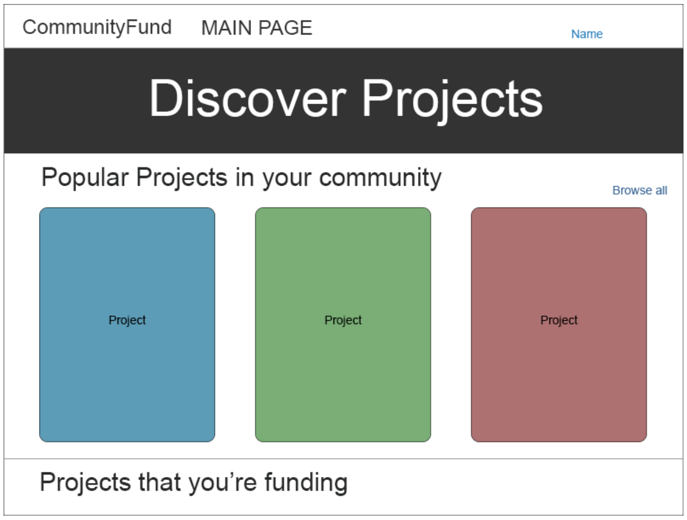
Create Project

- ★ Users can initiate their own projects
- ★ Users can enter all the required information needed to start a new project

Admin Page

- ★ Only authorized accounts can see this page
- ★ View total number of projects funded
- ★ View average days to reach a fund goal
- ★ View total amount pledged
- ★ View other useful analytics

2.2 Project Plan [13%]



CommunityFund MAIN PAGE

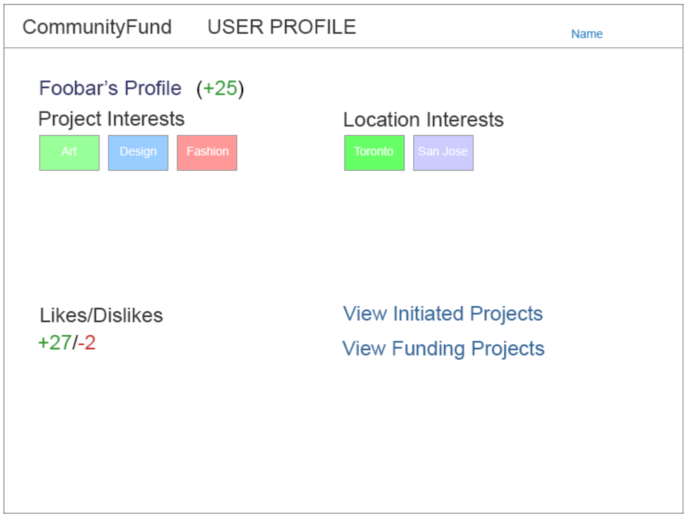
Name

Discover Projects

Popular Projects in your community

Browse all

Projects that you're funding



CommunityFund USER PROFILE

Name

Foobar's Profile (+25)

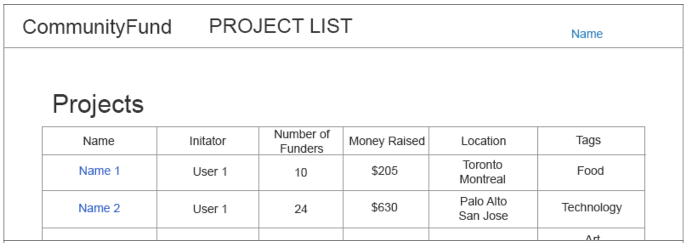
Project Interests

Location Interests

Likes/Dislikes
+27/-2

View Initiated Projects

View Funding Projects

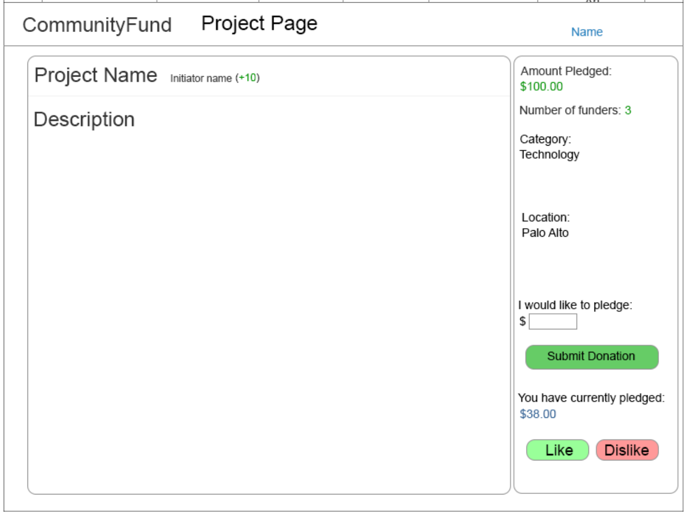


CommunityFund PROJECT LIST

Name

Projects

Name	Initator	Number of Funders	Money Raised	Location	Tags
Name 1	User 1	10	$205	Toronto Montreal	Food
Name 2	User 1	24	$630	Palo Alto San Jose	Technology
					Art



CommunityFund Project Page

Name

Project Name Initiator name (+10)

Description

Amount Pledged:
$100.00

Number of funders: 3

Category:
Technology

Location:
Palo Alto

I would like to pledge:
$

You have currently pledged:
$38.00

Personal Roles and Team Organization

Each member should have a chance to work with each language. The work will be divided to give each member a chance at practicing the languages learned in class. However, if a group member excels at a certain field he/she will most likely work on that field more than others. One member will be responsible for the TA weekly status reports. Each member will be responsible for developing their own pages and testing their own pages as well as another group member's pages.

Real-time communication will be done using Facebook by creating a group conversation. In addition, we will be using github to coordinate and update our work. github will allow our group to track our progress and stay on task. Issue warning will enable members to see what needs to be completed and where the group is having problems. We hope to have a group leader or someone who will take the responsibility of distributing the workload evenly amongst group members and guiding the group throughout the process. Most of the design and functions will be decided by the end Phase 2, but our group will be open to new ideas. In order to resolve problems, we will using a democratic approach by having a vote on conflicting ideas. If we are stuck with a tie, we will seek advice from our TAs.

Project Milestones

We will first prioritize html code and creating the general layout of our web pages. Following this, aesthetics will applied to our webpage via css bootstrap. Functionalities, user interactions and browser control will be implemented after. Using a kanban board we can organize the tasks that are required to be accomplished for each milestone. By the end of each week, the kanban board should have it's weekly requirements completed.

Weekly group discussions will be held to discuss what we have accomplished in the last week and what is to be completed for the following week.

Week 1 - Intro page, sign in, sign up
Week 2 - User profile page, project creation
Week 4 - Main page, project pages
Week 6 - Reputation System, Administrative view system

2.3 Software Architecture and High-level Design [25%]

We plan on using Node.js, React.js and SQL (probably mysql) to implement our project.
Node.js - Implement functionalities
React.js - Generate front end of web page
mySQL - Store and retrieve data
The three will integrate together to form our product
We will store data in SQL tables, namely the following tables:
Users(userID, email, password, name, accountPrivileges, numLikes, numDislikes)
Users_Projects(userID, projectID, initiator, amountDonated) *intiator column contains true/false, if it's true that means userID is the initiator of projectID, else, userID is funding projectID
Users_Interests(userID, interest)
Users_Locations(userID, location)
Projects(projectID, description, dateCreated, fundGoal, currentFunding, numFunders)
Projects_Interests(projectID, interest)
Projects_Locations(projectID, location)

A brief example of a sample database is as follows

Users

userID	email	password	name	accountPrivileges	numLikes	numDislikes
1	a@gmail.com	apass	userA	admin	10	2
2	b@gmail.com	bpass	userB	user	5	1

Users_Projects

usersID	projectID	initiator	amountDonated
1	1	True	NULL
1	2	False	20.46
2	1	False	10.35
2	2	True	NULL

Users_Interests

userID	interest
1	Art
1	Technology

2	Fashion

User_Locations

userID	locations
1	Toronto
2	Palo Alto

Projects

projectID	description	dateCreated	fundGoal	currentFunding	numFunders
1	descriptionA	02/05/15	2000.00	10.35	1
2	descriptionB	01/04/15	750.00	20.46	1

Projects_Interests

userID	interest
1	Art
1	Technology
2	Fashion

Projects_Locations

userID	locations
1	Toronto
2	Palo Alto

We will need modules for:

Creating and managing user accounts and data
Creating and managing projects

Methods include:

create new account (insert account info into RDBMS with SQL)
log in
change password
retrieve password
create project object, data stored in a separate sql table
set attributes of project object
retrieve data from RDBMS

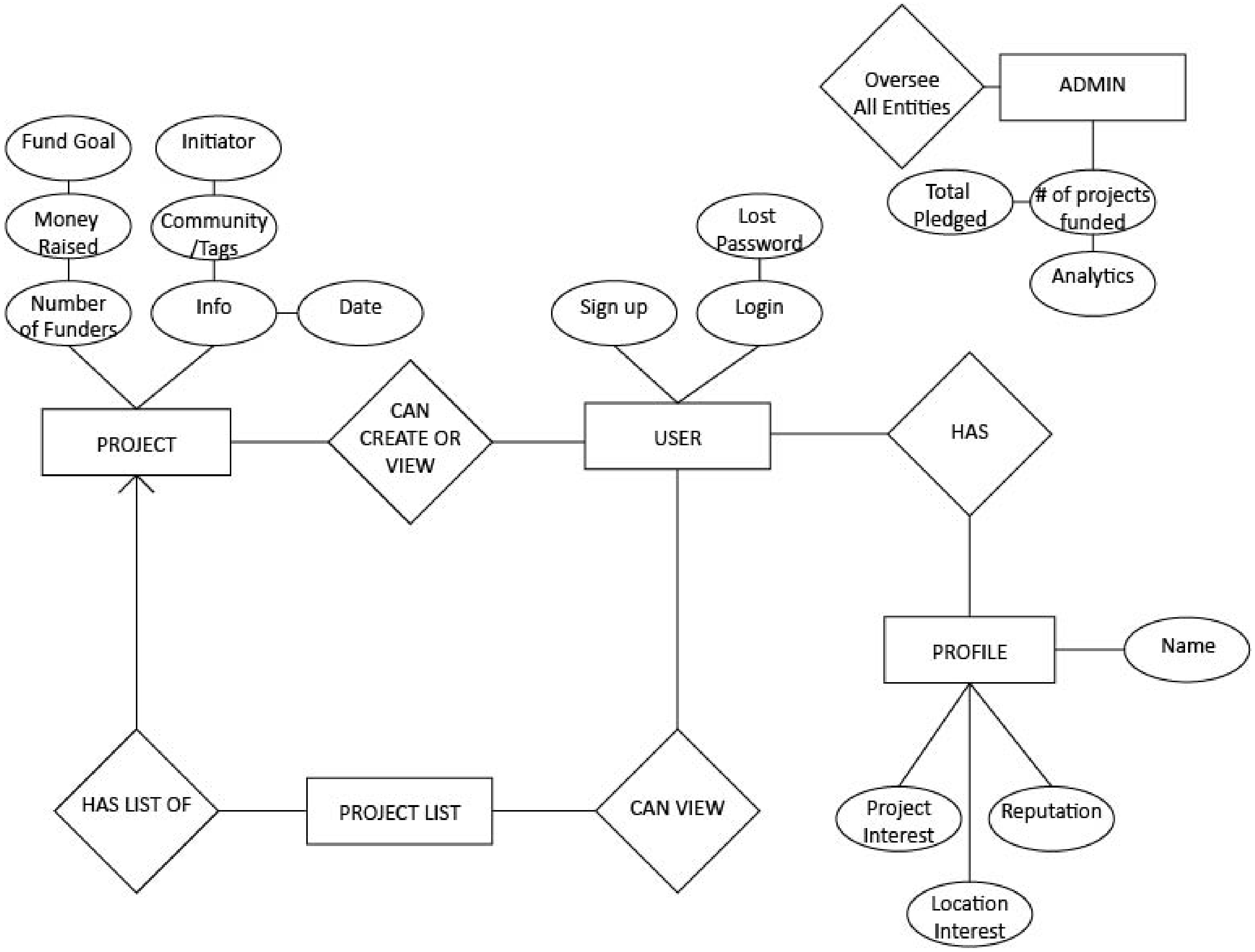
populate web page using react.js

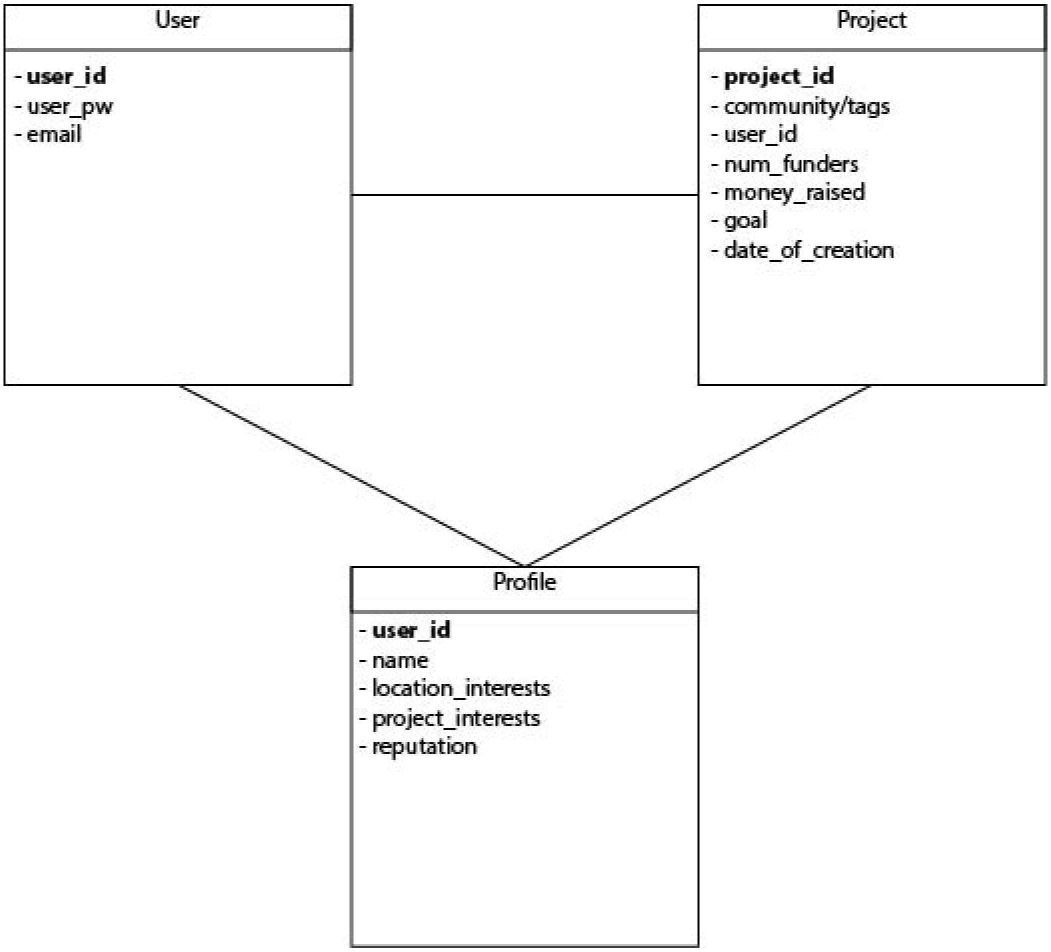
FUNCTIONALITY / METHODS / PACKAGES

Root
    Send POST/GET requests, links to other web pages
Sign up
    Checks that the email isn't already in the database
    Checks constraints on password (must contain special character, min 8 characters)
    Checks that the re-typed password matches the original password
    Insert data into database
Log in
    Check that the user entered the correct username/password
Lost password
    Using nodemailer.js, send email to user containing password
Set up profile
    Retrieve options that user selected as interests/locations
    Insert data into Users table with SQL
Main page
    Retrieve data from Users table
    Retrieve data from Projects table
    Populate web page with data
Account settings
    Users can change password. Updates User table with new password
User profile
    Change name, change project/location interests
    Update, retrieve data from tables
Project List
    Figure out the type of request the user has made (eg, show projects only in Toronto)
    Retrieve data from tables and populate web page
Project Page
    Calculate percentage funded
    Submit donation, update SQL tables function
    Update SQL tables when leaving a like, dislike for a user
    Retrieve data from tables and populate web page
Create Project
    Parse data entered by user and store it in database
Admin Page
    Calculate all the aggregate analytics required to be displayed (eg. total amount of users)

2.4 Information Representation [25%]

Data, information will be stored in an SQL database





2.5 Test Strategy and Test Plans [12%]

Testing will be done frequently to prevent the accumulation of bugs and errors.

Functionality testing will be done to check the links of the webpage, incorrect inputs, verify workflow of system, and data integrity.

Link Check

- check all internal links
- check any links jumping on the same pages
- check external/outgoing links from pages from a certain domain
- check for broken links.

Form Check

- check validations on fields
- check default values of fields
- check incorrect input to the fields in the form

Usability testing will verify the user-friendliness of the application which includes testing the navigation, content checking and user intuition. The website should be simple to navigate and it should be clear to get what the user needs. Check for spelling errors and check that text/images are clearly visible.

Interface testing will check the interface and data flow is fluent and consistent from different systems. Primarily check that the interaction between servers are fine, and handle errors properly whether that be catching and displaying the error or doing nothing.

Compatibility testing may be performed to test for browser compatibility, operating system compatibility, and device compatibility. We must check if the UI and functionality is consistent amongst different browsers, operating systems, and devices

Performance testing will be done to check the performance through several conditions.

Web load testing

- Test performance with many users accessing the same page

Stress testing

- Test performance with excess stress on website

Security testing will be done to test the web application's security with unauthorized access and data theft.

- Test internal pages access
- Test invalid usernames and passwords
- Web directories and/or files are to be unaccessible directly