StartHub Final Documentation

Table of Contents

[Introduction](#)

[Process](#)

[Requirements & Specifications](#)

[Architecture & Design](#)

[Future plans](#)

Introduction

Goals

For the class CS428, students were asked to build upon their experience on some intricacies of managing projects that rely on a non-trivial web application. Students proposed their own projects, determining the platforms and technologies and developing their projects in teams. StartHub is one of the proposed projects.

StartHub Description

StartHub is a web application that provides users with a means to find awesome collaborators for their own startup ideas. StartHub has an interface that allows users to post project ideas that other members can join and to search people by skill, interest, or location. StartHub also allows users to discover, join, or share projects other members are posting. StartHub gives project ideas the opportunity to scale into startups and eventually full fledged companies.

The motivation for StartHub are the two of the major inhibiting factors for creating startups: finding a like-minded team and finding a great idea. The StartHub platform solves both of these factors. By connecting the community and through its features, StartHub helps its members contribute to the projects they are interested in.

StartHub is similar to Kickstarter, a crowdfunding company for initial startup projects. StartHub is different in that it allows individuals to contribute to the project instead of merely funding it.

Process

For the Team process we followed many principles that we learned in CS 427 including: Extreme Programming, Test Driven Development, Refactoring, and User stories. With our team of 6, we split up into pairs of 3 for the Pair programming. We saw a lot of success with this software methodology, our processes and implementation were much more efficient and we decreased our margin of error.

Another factor in our success was user stories. We were able to have very concise requirements when evaluating them from the perspective of clients and users of the web application. In addition, we developed our features by writing tests before actual implementation in order help create much more concise code and specific functions. This enabled our code base to have much higher readability.

For every iteration, each pair had to complete one refactoring within their past use case in order to help improve the functionality of their code. With these software development methodologies not only were we able to complete a complex project, but also learn and showcase a project with high level of code sophistication.

Requirements & Specifications

To be classified as finished, our project needed to meet certain requirements and specifications. These requirements are outlined in the user stories below:

User Stories

There are two main users to the StartHub web application: visitors, users, and project owners.

Visitors

Visitors are the people who first visit the web application. They are the highest level people in the system. They do not have an account setup and have limited features.

- As a visitor to the website, I would like to see the projects worked on StartHub.

Users

Users are the visitors who have an account in the system. By signing up for an account, they can additional features that are not available to visitors.

- As a user, I can create or edit accounts.
- As a user, I can log in or log out.
- As a user, I can see project specifications (skills, interests, location) set by project owner.
- As a user, I can see a list and search projects from other StartHub members.
- As a user, I can up/down vote projects based on my perception of its viability.
- As a user, I can contact the project owner if I am interested in contributing to the project.
- As a user, I can view and create project comments.
- As a user, I can edit the project information.
- As a user, I can upload images to project description.
- As a user, I can see recommended projects based on the skills and interests listed in my account profile.
- As a user, I can change my logo from the edit page.
- As a user, I can filter search results by tag.
- As a user, I can filter projects via location preference.

Project Owners

Project owners are the users who create projects. They have the abilities to edit and delete projects.

- As a project owner, I can message user and view messages sent.
- As a project owner, I can create projects.

- As a project owner, I can tag projects with skills and location.

Use Cases

Create Accounts
The visitor requests to make a new account and the system logs space in the database. The visitor fills out personal information fields. Text is stored in the database and the passwords are stored with auth.

Edit Accounts
The user requests to edit his or her account and the system logs space in the database. The user edits personal information fields. The changes are stored in the database in their respective locations.

View Project Listing
The visitor requests a listing of current projects. The web page will log the request and ask for a list from the database. The list is sorted and presented in order of preference for the visitor. Short descriptions of each project are generated and attached to each listing.

Create Projects
The owner requests to make a new project and the system logs space in the database. The owner writes text, uploads images, and uploads videos for his or her project page. Text is stored in the database and links to the images and videos are stored.

Edit Projects
The owner requests to make edits to an existing project. The owner edits text, uploads images, and uploads videos for his or her project page. Text is stored in the database and links to the images and videos are stored.

Delete Projects
The owner views the edit options for their project. There will be a delete project button on the page. When the owner clicks it a confirmation window will pop up to confirm if the project wants to be deleted. Once the owner confirms, then the project is deleted.

Filter Projects
The visitor requests to filter projects by a string. A search is performed in the database and all projects containing that string will be displayed as a list to the user.

Up/Downvote Projects
The member views a project page and can either vote up or down if they think the project is good or not.

Comment on Projects

The user types comments into the text field and requests to send the comment. The database creates space for the comment in the project and stores it in chronological order with the other comments.

Contact Project Owner

The member views a project page and is interested in working with the project owner on the idea. The member contacts the project owner by clicking on the “Contact project owner” button and filling in his/her contact information on the provided form.

Architecture & Design

Fig. 1 UML Class diagram:

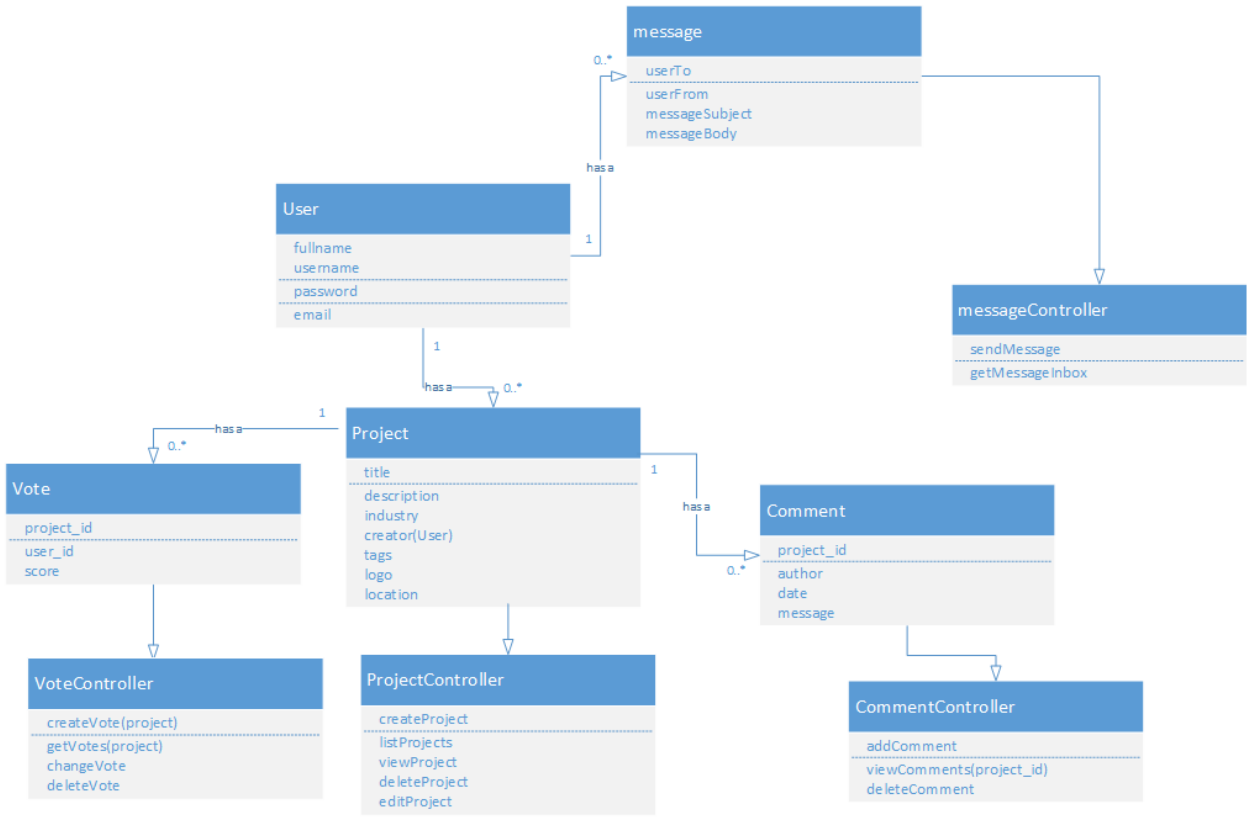
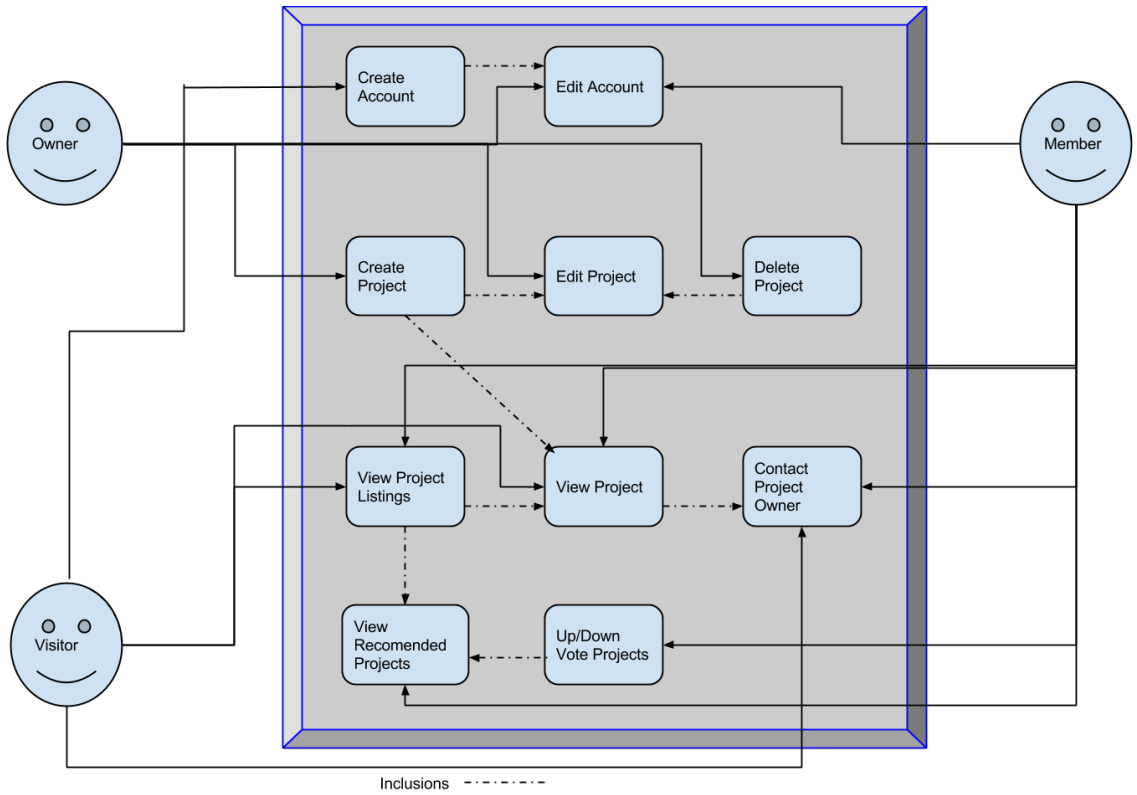


Fig. 2 UML diagram:



TOPDOWN - Implementation from outside to inside

Our system is comprised of client and server just like many other websites or web-apps. The server and client communicate with each other using the REST API standard. The client is implemented using AngularJS, which handles templating of different pages. It also helps us condense functions that otherwise could become large and unwieldy into concise AngularJS services. The server is implemented using Node.js which allows our project to incorporate essential packages. One such package is mongoose.js which we use to access our Mongo database. MongoDB is a non-relational database, which unifies our whole system in JSON format. Our code is organized by the Yeoman generator. Yeoman is used for generating models as well as server and client controllers. We called Yeoman whenever we introduced new use cases such as commenting and messaging. All client side code resides in the "public" folder and all server side code resides in the "app" folder.

On the server side, the database objects are defined in their model files. The controllers define the functions that these models use. The functions and middleware are called through server side routing.

On the client side, the templates are created in the module's view. Client side routes are created to display these views. The functions are created in the module's controllers. You can define a service to make RESTful calls to the server for a higher level approach.

The choice of MEAN stack framework enforces use of RESTful service api standards. Instead of creating different pages as data end points, object names such as /projects or /projects/ids are used as url for routing. So, our website is made to serve JSON objects through the http call of these specified urls. This characteristic of RESTful service required by MEAN stack framework restricts the architecture of our website to the client side rendering, so the server side rendering becomes out of viable options.

Future plans

Ian

- I learned a lot on this project about extreme programming methods, and the challenges involved with working on a large scale development project.
- I also gained valuable knowledge on how to use Node.js and the MEAN stack. I will use these skills in my future web development endeavors.
- I also had a chance to hone my time management skills whilst preparing for iteration meetings.

Dhruv

- I learned the importance of adapting to my team's schedule. Everybody's team schedule is different so I have to sacrifice some of my time for benefit of the team.
- I learned that pair programming lowers the time it takes to implement features and is a fun way to get the work done.
- I also had a learned alot about managerial responsibilities and how to handle relations for the future.

Travis

- While working on this project I was able to incorporate my software engineering skills I learned in the previous semester to meet the requirements of this project.
- I learned how to use Node and Javascript for stack development and have realized that web development is not my strong suit.
- I believe this project was fun to work on but I do not want to continue development.

Adam

- I learned how to work with team members and work around other's time schedules
- I learned about how to work on full stack web development
- I was able to practice core developmental processes while working in a simulated industry environment

Chris

- I learned about the importance of fast response to changing requirement stressed in Extreme Programming.
- I learned how to use Git properly to enhance integration of software development process.
- I believe the web programming concepts and skills that I learned in this project will be very useful in the future.

Matthew

- I learned that refactoring improves code readability and reduces complexity to maintain code. Refactoring is especially important when I work with a team of people who needs to read my code. I am glad it was a requirement to refactor a piece of code after every iteration. The source code gets cleaner and easier to read!
- I learned to communicate early and often with my teammates. This allows me to plan ahead and adapt to necessary changes. I believe this skill will be useful for working in the industry. Everybody has different work schedules and learning to communicate often makes for a more productive and efficient schedule.
- I learned the practice and benefits of Test-Driven Development. It allows for more maintainability and extensibility. Writing tests before code also allows us to make sure the feature we implemented actually reflect what the customer wants.
- I see great potential in StartHub in the future. It doesn't quite live up to its KickStarter competition, but there are still much more opportunities for StartHub to improve on.