

SW Engineering CSC648/848 Summer 2019

TeamSix

Team 6

Austin Tsang

Audrey Wong

Han Huang

Alexander Caley

Gebrezgabiher Mengis

Jesus Valdes

Syed Abidi

Milestone 2

Date: 7/10/19

1. Data Definitions V2

- **Search**

- The site shall give accurate result.
- The search engine will use search-by-keywords algorithm.

- **Filter**

- Users can filter by categories.

- **Registration**

- Clients need to register if there is an issue they want to report.
- During this stage, clients will need to enter their email address, desired password and username to complete registration.

- **Login**

- Email and password are required to sign in.

- **Posting**

- **Comment**

- Comments must consist of letters, symbols, numbers, or images (optional).
- Must not exceed maximum size.

- **Image Upload**

- Image must be relative to the subject.
- Image must be in image format (.jpg, .jpeg, .png, etc.).
- Images must not exceed maximum size.

- **Map Link , Guide and Accessibility** to popular places such as parks, forests, beaches and oceans.

- **Admin**

- Review a report
- Approved or removed an appropriate report
- Update the status if the issue is in progress
- Assign to the crew
- Access and modify all data
- Needs to login/register

- **User**

- **Unregistered User**

- Be able to view posts
- Be able to search
- Does not need to login/register

- **Registered User**

- Be able to report an issue as a picture, in written
- Be able to review
- Be able to search
- Needs to login/register

- **Service User Agreement**

- User terms and conditions
- Service User Agreement must be shown before user starts the registration

- **User Privacy**

- Protect user information
- User has access and modify the profile page

- **Customer Support**

- Client can contact customer support for urgent need.

2. Functional Requirements V2

Function Priority

Priority 1 -must have functions

- **Unregistered User**

- Unregistered user shall be able to view posted environmental problems.
- Unregistered user shall be able to view the status of posted environmental problems.
- Unregistered user shall be able to search environmental problem posts.

- **Registered User**

- Registered user shall be able to do everything that unregistered client could do.
- Registered user shall be able to post environmental problems

- **Admin**

- Admin shall be able to do everything that registered client could do.
- Admin shall be able to delete inappropriate posts.
- Admin shall be able to modify the environmental problem posts.
- Admin shall be able to modify the status of posted environmental problems.

Priority 2 -desired functions

- **Unregistered User**

- Unregistered user shall have the option to register their accounts.

- Unregistered user shall agree on service user agreement to complete registration.
- **Registered User**
 - Registered user must have access to modify their profile.
 - Registered user shall be able to comment on environmental problems.
- **Admin**
 - Admin must have access to modify their profile.
 - Admin shall be able to respond on environmental problems.

Priority 3 -opportunistic functions

- **Registered User**
 - Registered user shall be able to contact the customer support for urgent need.
- **Admin**
 - Admin shall be able to assist registered client based on client's urgent need.

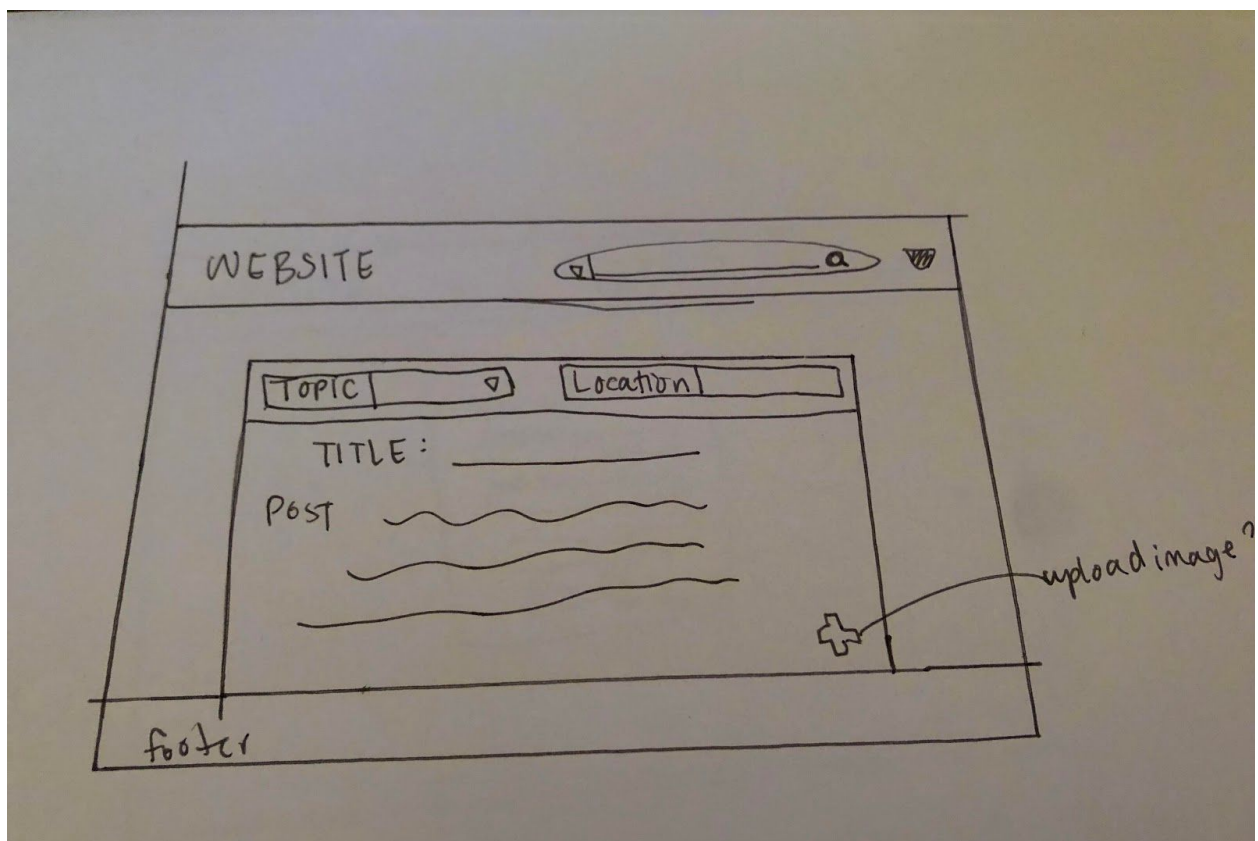
3. Storyboards

Title: Postings

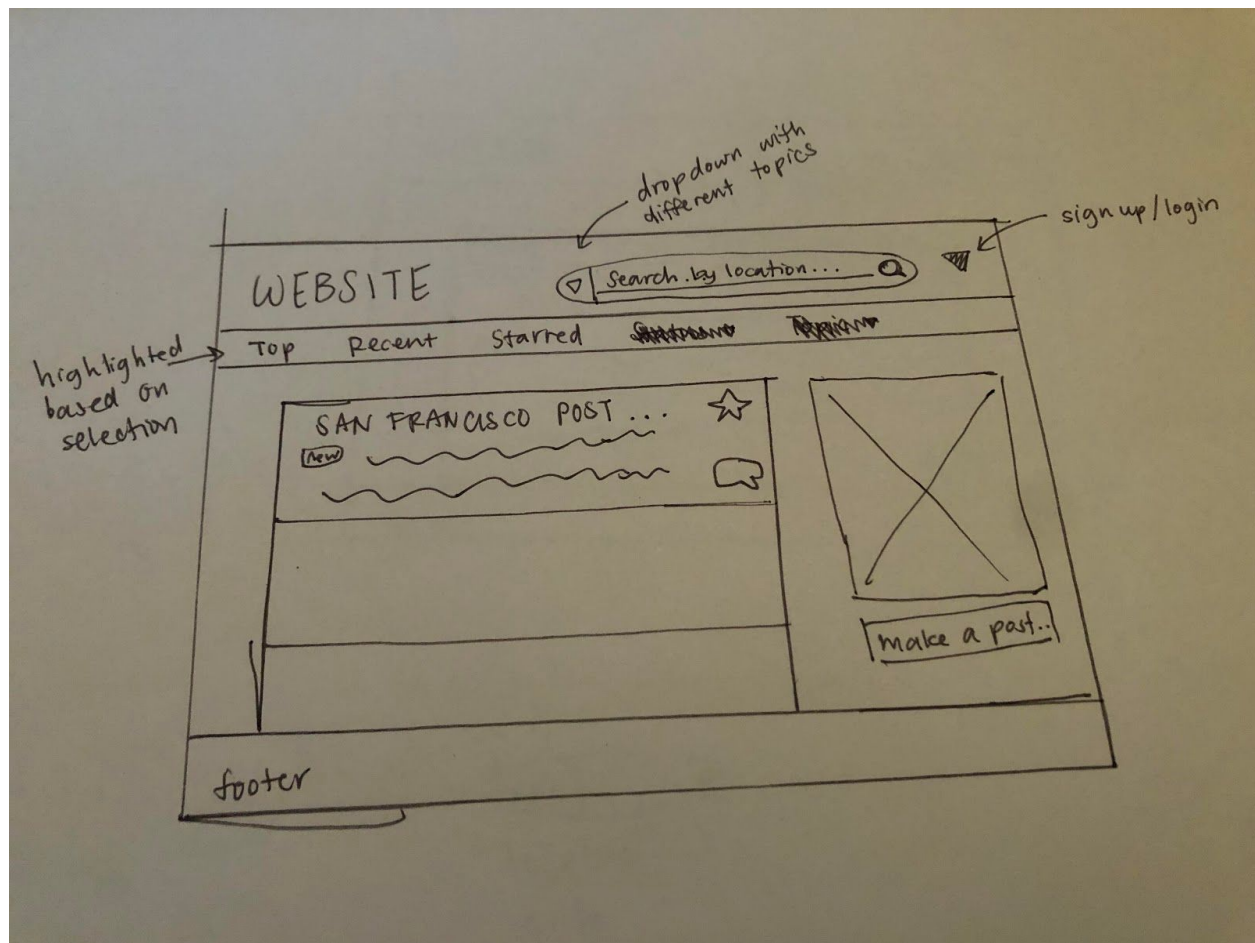
1. John is a very busy dad who loves to take his kids to the park. He likes to make sure his kids are always playing in a safe environment , therefore, he is more than likely to report environmental issues if seen in the community. He works in an office and has strong web skills. He is also very impatient and prefers to do things quickly to fit into his busy schedule. He would prefer to quickly report an issue online in the most fast and efficient way as possible.



- John takes his kids to a park after work and notices a large tree has fallen down and blocked the main path to the park. John goes to our website and searches the park by location, e.g. “San Francisco”, then reviews the list of search results



John finds the specific park and is able to see previous issues with status. He does not see a report on the fallen tree, so he reports the problem by making a post.



John has to sign in before making a post.

Sign up/Login pop-up page

Sign up

First nameLast name

Email Address

Password

Confirm password

☐ You agree to our terms of service

Continue

Already have an account?
Sign in

Logo

Sign in

Email address

Password

☐ Stay signed in

Forgot Password

Log in

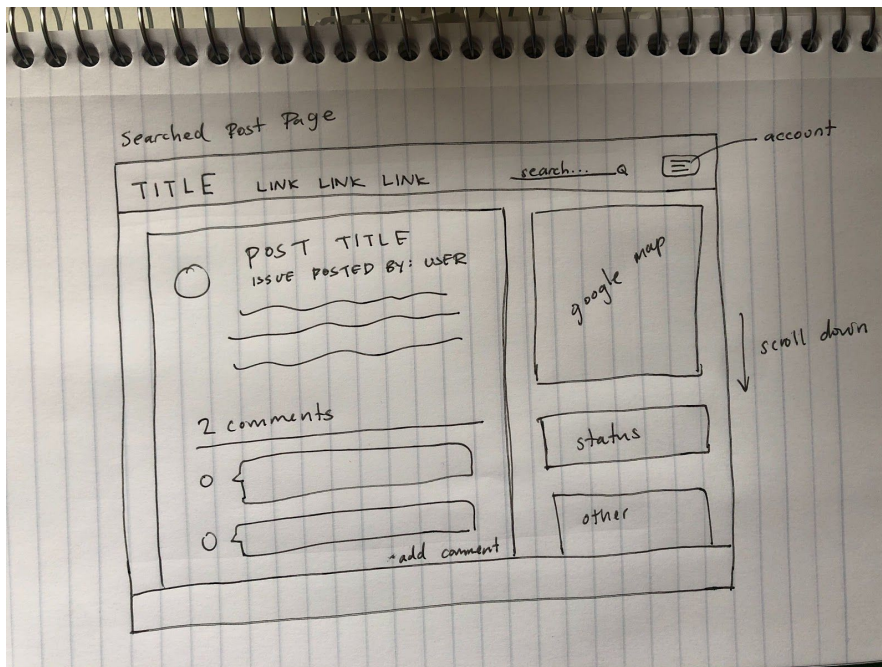
You don't have an account?
Creat an account

Title: Workflow

3. Jerry is a city environmental manager who is often very busy dealing with complaints from people all around the city. While on duty, he frequently take strolls through the different parks around the city to make sure everything is safe and functional. Jerry is in charge of keeping track of what goes on in these parks to make sure the community is able to enjoy these parks just as much as he enjoys them himself. He is not too familiar with WWW apps but has basic WWW skills. He would prefer to get things done quickly.



- Jerry doesn't have time to deal with each complaint individually, but still needs to keep track of what is being work on and what issues have been dealt with. He recently received a complaint about a fallen tree and has called community helpers to remove the tree. He goes onto our website and searches the park by locations, e.g. "San Francisco", then reviews the list of search results. Jerry finds the park and finds the post. He assigns a status: in progress, attached to the post, and the site is updated.



4. High level Architecture, Database Organization

1. **DB organization**

- a. Issues/Ticket/or whatever we are calling the posts
 - i. Unique ID: increment starting at 0; primary key
 - ii. Street: char
 - iii. City: char
 - iv. Zip code: int
 - v. State: char (could just be 2: CA, etc)
 - vi. Description: char
 - vii. Picture: depends on what we decide below
 - viii. Status: char
 - ix. Date Witnessed?: maybe not necessary
 - x. Type: char (maybe dropdown list of items: garbage, air, water pollution, etc)
- b. Users
 - i. Unique ID: increment starting at 0; primary key
 - ii. Username: char
 - iii. Name: char (not really necessary, just in case)
 - iv. City: char
 - v. Zip Code: int (don't really need these 2, depends if we want to make any type of profile for users to see)
 - vi. Password: encrypted

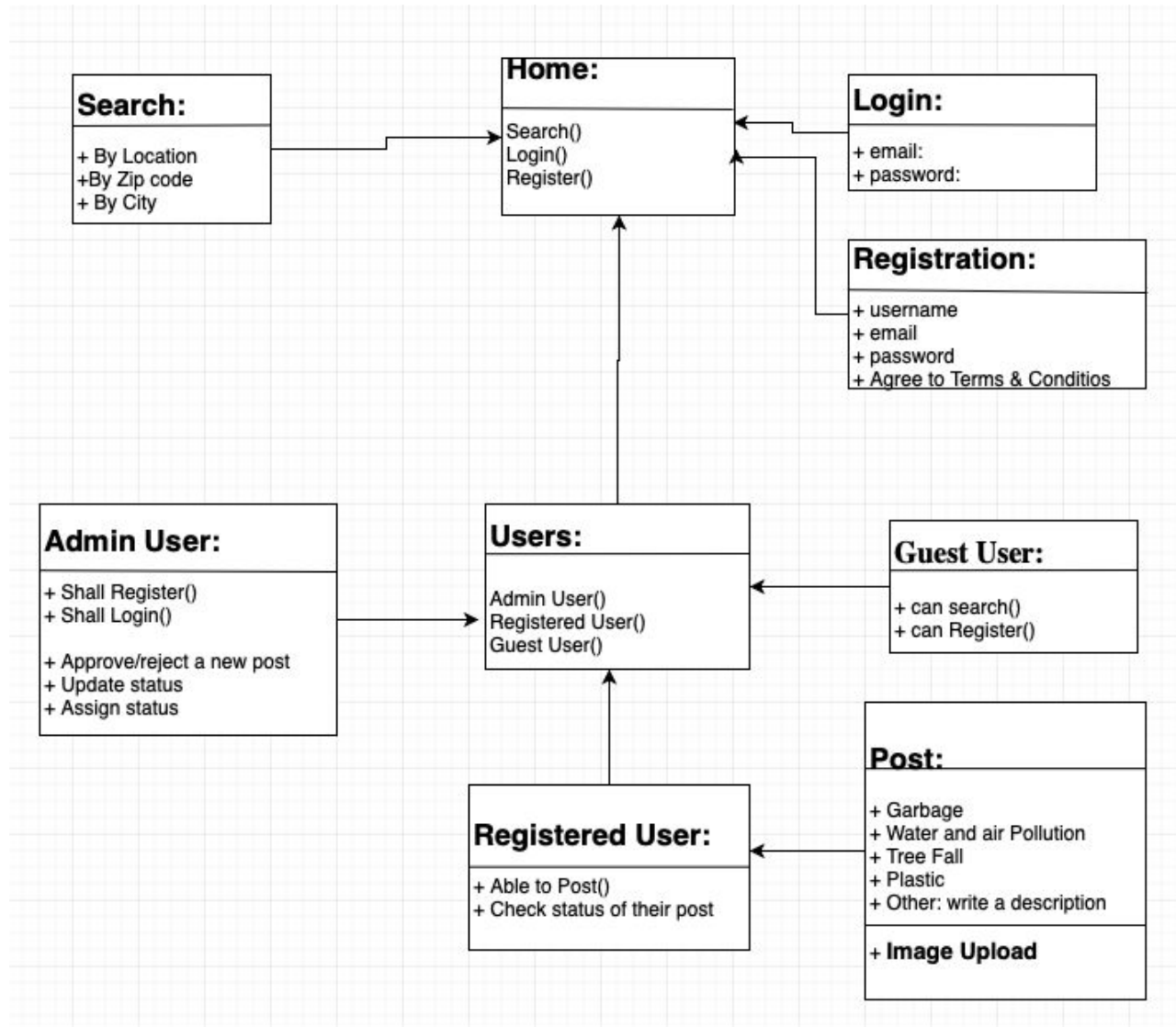
2. **Media Storage:** Due to limited experience with BLOB's it would be easier to keep images in file systems as well as have less of an impact on performance. Only images are needed.

3. **Search/filter:** Can use LIKE with % wildcards like mentioned in class for easiest method. For filters, we can have drop-down menu next to search bar to search for specific zip code, city, state, etc

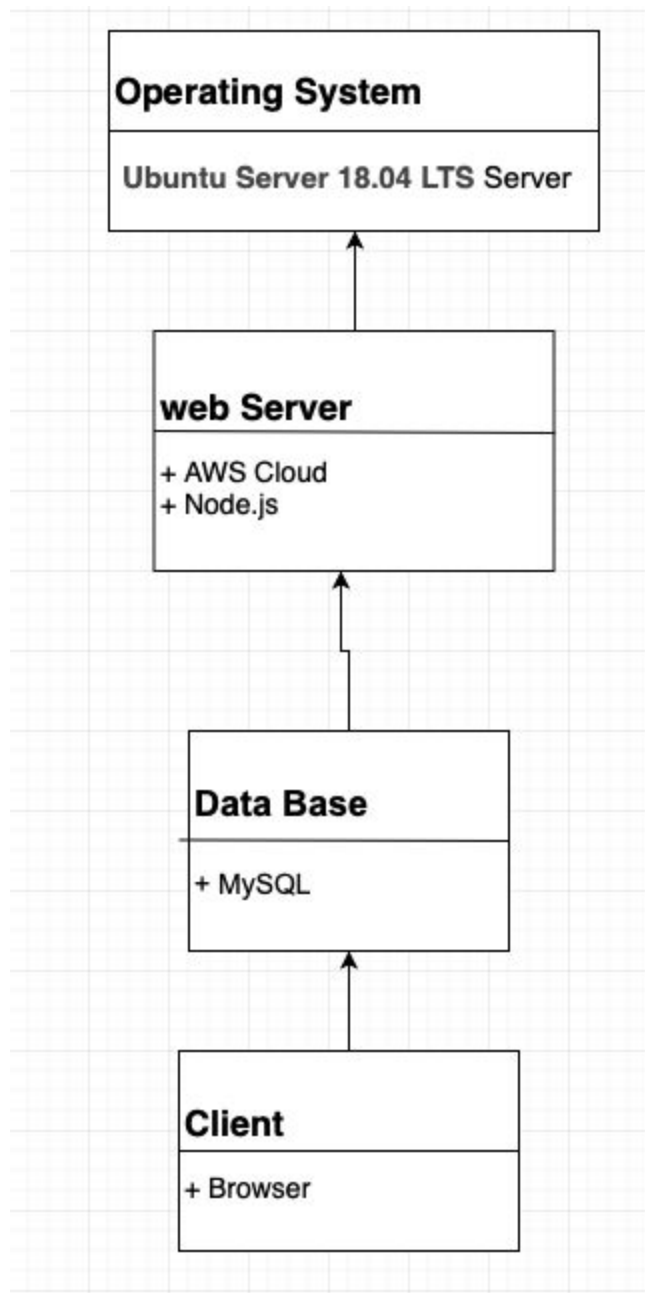
4. **API's:** More to come

5. High Level UML Diagrams

High-level UML class diagrams



UML Component and deployment diagrams



6 .Identify *actual* key risks for your project at this time

In today's world, teamwork is being utilized by companies across the globe. Employers are seeing the value of teamwork, and what can be accomplished when people put the strengths together. These teams consist of people from different cultures with different personalities. Conflict is inevitable when it comes to group dynamics. Conflict resolution is necessary to keep the group functioning efficiently.

Although we have a strong team who are technically well versed across the framework that we're using, we as a team do face some risk factors that need to be kept in mind in order to ensure a successful project delivery.

One of the key factors initially was deciding which framework to use and what language to base our application on. Even though we're using a slightly challenging express framework for our application, we're all confident in our ability to create and deliver it successfully. What assists all of us even more is the fact that we have our team lead who is technically strong and is comfortable with the framework being used; front-end and back-end. With stackoverflow, github repos, and countless resources available online, we all ensure to learn something new, or teach something new to other team members. This ensures a smooth process and assistance given to every team member in case there's a situation where we require assistance.

Our backend team has worked with database elements and are familiar with how to connect to the server and a webpage, however given we're all still learning there might be a

concern as to how proficient we are with SQL Database. In order to tackle this, we do have members in our team who are proficient with SQL, and allow other members to shadow them and learn from them in order to expand their skill set.

Another factor that our team is aware of is the time constraint. Given we're in the summer session, the timeframe goes from four months down to merely ten weeks, which means our deadlines reach sooner. In order to tackle this issue, our team of seven are taking up their responsibilities extremely well, we're delegating our tasks to each member, and once one member finishes up his/her work, they try to assist other team members in our team to ensure we're all on the same page to progress forward.


Many of us are graduating seniors, full time employees, and are taking a high course load – all of this combined together means more pressure on every member of our team to deliver on time and ensure proper time being put into our work. While this may be a key risk, the work being divided between all of us combined with the responsible attitude of our team ensures positive results.

We also need to ensure that we keep an eye on copyright policies to ensure not to violate it whether we or the users claim ownership of posted pictures, videos, etc.

7. Project Management

Starting at M0 we delegated the tasks using the “issues” section on github. Using the “issues” tool, we can assign specific tasks for people to do. Once they complete the task and another team member has peer reviewed it, they can resolve the issue on github. We used this same step for M1 by delegating each section of the document to a single person. This was recorded using “issues” on github. To create the vertical prototype for M2, tasks are being delegated to both the frontend team and backend team using the “issues” section on github. You can label these “issues” based on what the problem is, whether it’s a bug, documentation, needs help, etc. The Frontend portion is being managed by the frontend lead while the backend portion is being managed by the backend lead. The team lead will oversee both frontend and backend development while assisting wherever needed. Branches are separated by frontend, backend and development. Branches handling the frontend will have the origin of the frontend branch and backend branches will have the origin of backend branch. The development branch is to check and make sure that both the frontend and backend branches are working together properly before deploying to production.

This is where team members can view open issues as well as the assignees.

 **CSC-648-SFSU / csc648-su19-Team06** Private

Unwatch ▾ 4

★ Unstar 3

🍴 Fork 0





[Code](#) **Issues 4** [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Filters ▾

🏷️ Labels 9

📅 Milestones 0

New issue

<input type="checkbox"/>	🚨 4 Open ✓ 7 Closed	Author ▾	Labels ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<input type="checkbox"/>	Backend DB design #16 opened 10 days ago by atsang14						
<input type="checkbox"/>	Create Mockup UI #15 opened 10 days ago by atsang14						
<input type="checkbox"/>	Create backend routes #14 opened 10 days ago by atsang14						
<input type="checkbox"/>	Image path changed for aboutus page #13 opened 10 days ago by atsang14						

💡 **ProTip!** Find everything you created by searching [author:atsang14](#).

This is what it looks like after you selected a specific issue. Here you can see all the assignees and comments on the current comments of the issue

Backend DB design #16

[Edit](#)[New issue](#)

atsang14 opened this issue 10 days ago · 0 comments



atsang14 commented 10 days ago



design high level db organization and practice UML



atsang14 assigned alexandercaley, HeyZeusV and qaim110 10 days ago

Assignees



alexandercaley

HeyZeusV

qaim110

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

[Customize](#)



Write

Preview

AA B i “ < > ↻ ☰ ☷ ☰ @ ★ ↶ ↷

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Close issue

Comment

Here is a history of all closed issues. This way you can keep track of previous issues and bring them back if the issue re-appears.

CSC-648-SFSU / csc648-su19-Team06Private

Unwatch4Unstar3Fork0

<> CodeIssues 4Pull

InsightsSettings

Filtersis:issue is:closed

Clear current search query, filter

New issue

CSC-648-SFSU/csc648-su19-Team06 on Jun 26

High level system architecture and technologies used #12

Briefly provide itemized list of all main SW components such as frameworks, APIs, too...

You are assigned and commented

4 Open7 Closed

projectsMilestonesAssigneeSort

High level system architecture and technologies used

#12 by atsang14 was closed 12 days ago

1

Competitive analysis

#11 by atsang14 was closed 11 days ago

list of non functional requirements

#10 by atsang14 was closed 11 days ago

Initial list of funtional requirements

#9 by atsang14 was closed 12 days ago

2

List of main data items

#8 by atsang14 was closed 12 days ago

1

Persona and Use Cases

#7 by atsang14 was closed 12 days ago

Executive Summary

#6 by atsang14 was closed 11 days ago