SW Engineering CSC648/848 Summer 2019

TeamSix

Team 6

Austin Tsang (atsang3@mail.sfsu.edu)
Audrey Wong
Han Huang
Alexander Caley
Gebrezgabiher Mengis
Jesus Valdes
Syed Abidi

Milestone 2

Date: 7/10/19

<u>1. Data Definitions V2</u>

Unregistered User:
- Can access site as a guest
- Be able to view posts
- Be able to search
- Be able to register
- Does not need to login/register

Registered User:
- Can access site as a guest
- Be able to report an issue as a picture, in written
- Be able to review
- Be able to search
- Be able to login
- Needs to login/register

Admin:
- Be able to review a report
- Be able to approved or removed an appropriate report
- Be able to update the status if the issue is in progress
- Be able to assign to the crew
- Be able to access and modify all data
- Needs to login/register

Postings:
- Can be viewed by any user
- Only registered user can post
- Can contain characters, symbols, digits, and/or images

Posting status:
- Only can be updated by admin

Images:
- Can be uploaded under posting section by registered user choice
- Can be in .jpg, .png, .jpeg format

Searching:
- Accessible to anybody
- Can search by keywords, descriptions

Filtering:
- Accessible to anybody
- Can filter by categories

Registration:
- Accessible to anybody

Login:
- Accessible to anybody

Customer Support
- Accessible to registered user

2. Functional Requirements V2

Function Priority

Priority 1 -must have functions
1. **Unregistered User**
   a. Unregistered user shall be able to view posted environmental problems.
   b. Unregistered user shall be able to view the status of posted environmental problems.
   c. Unregistered user shall be able to search environmental problem posts.
   d. Unregistered user shall have the option to register their accounts.
   e. Unregistered user shall agree on service user agreement

2. **Registered User**
   a. Registered user shall be able to do everything that unregistered client could do.
   b. Registered user shall be able to post environmental problems
   c. Registered user shall be able to login to their accounts
   d. Registered user shall be able to contact the customer support for urgent need.

3. **Admin**
   a. Admin shall be able to do everything that registered client could do.
   b. Admin shall be able to delete inappropriate posts.
   c. Admin shall be able to modify the environmental problem posts.
   d. Admin shall be able to modify the status of posted environmental problems.

Priority 2 -desired functions
1. **Registered User**
   a. Registered user must have access to modify their profile.

    b. Registered user shall be able to comment on others' environmental problem.

2. **Admin**
   a. Admin must have access to modify their profile.
   b. Admin shall be able to respond on environmental problems.

Priority 3 -opportunistic functions
1. **Admin**
   a. 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
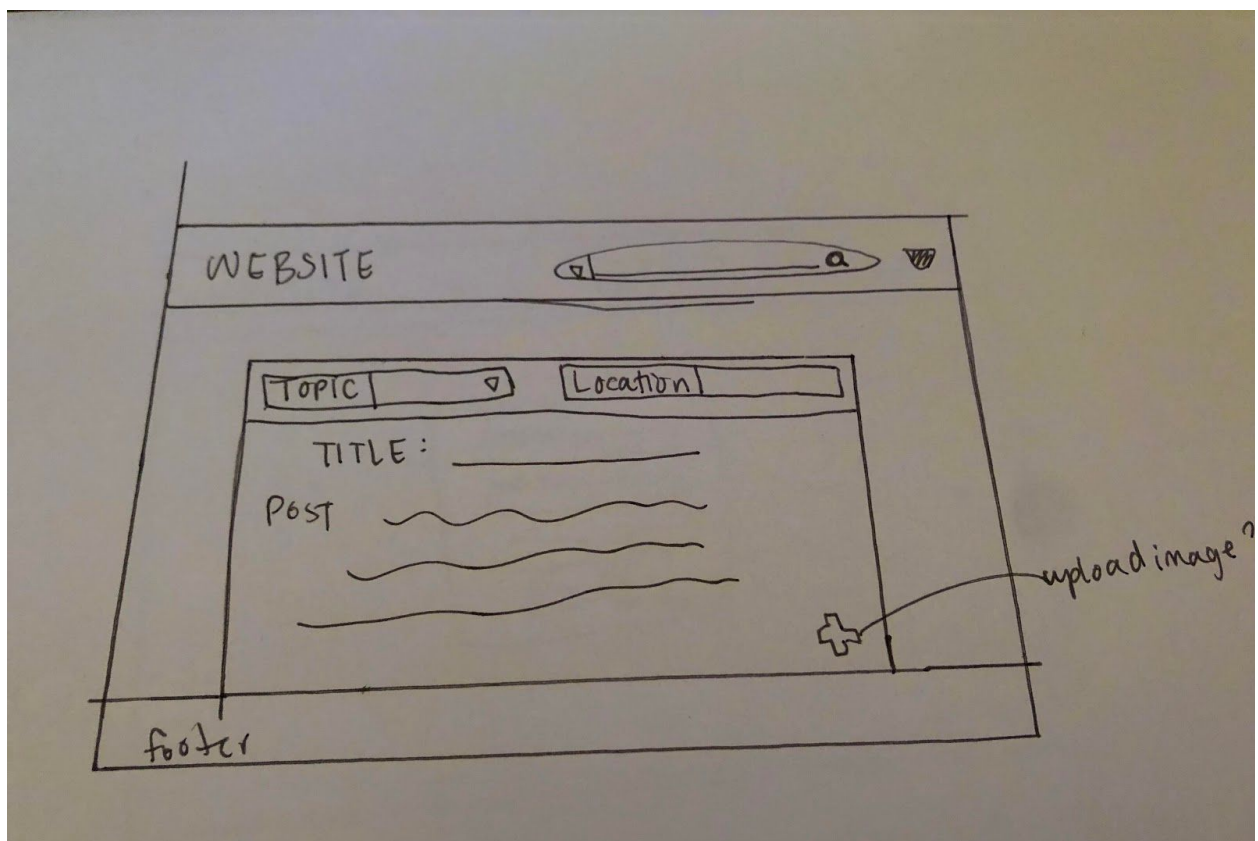
John finds the specific park and is able to see previous issues with status. He does not see a report on the fallen tree, so he reports the problem by making a post.
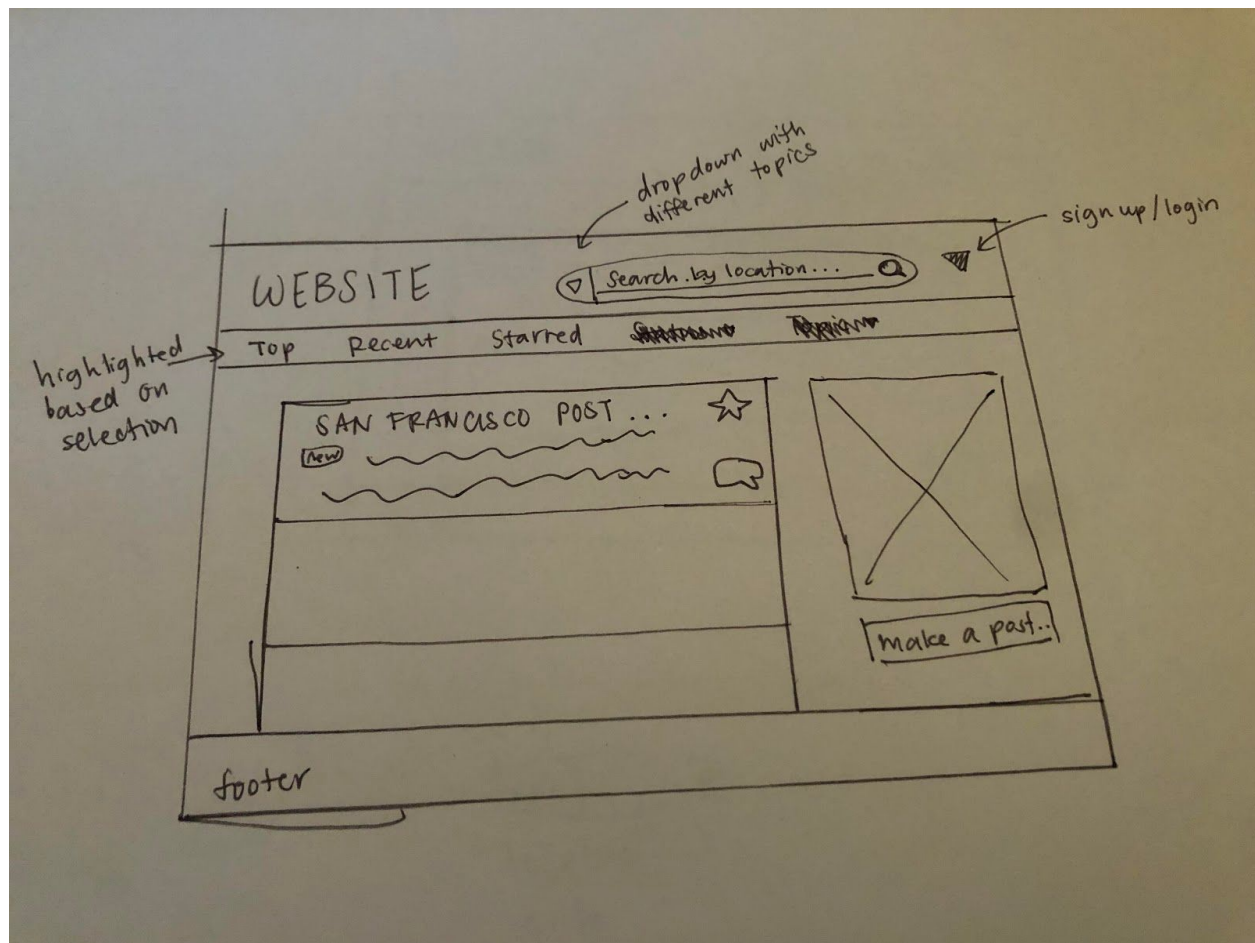
John has to sign in before making a post.

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.
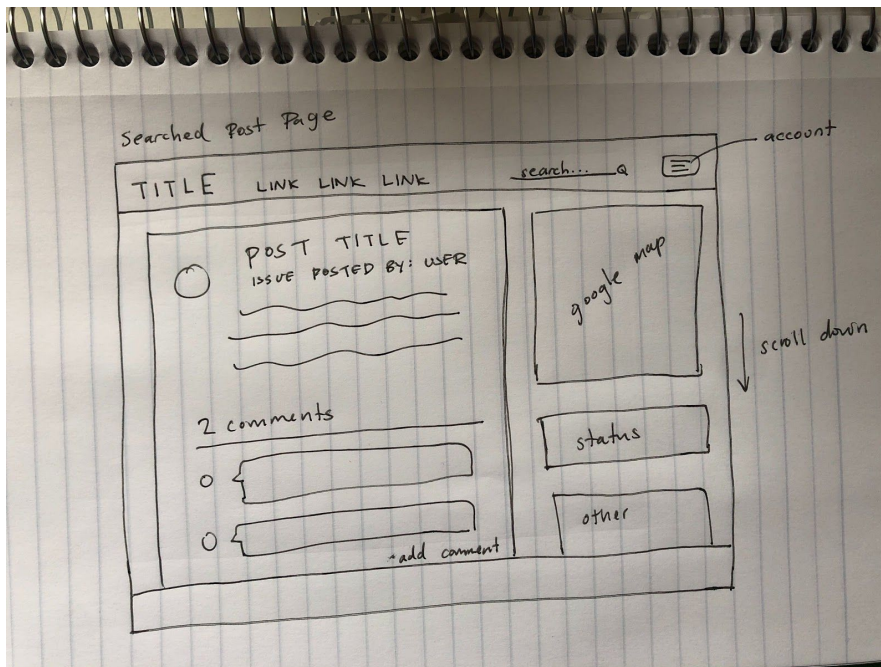
4. High level Architecture, Database Organization

1. **DB organization**
   a. Postings
      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: char (link to picture)
      viii. Status: char (Pending, In Progress, Complete)
      ix. Date Witnessed: char
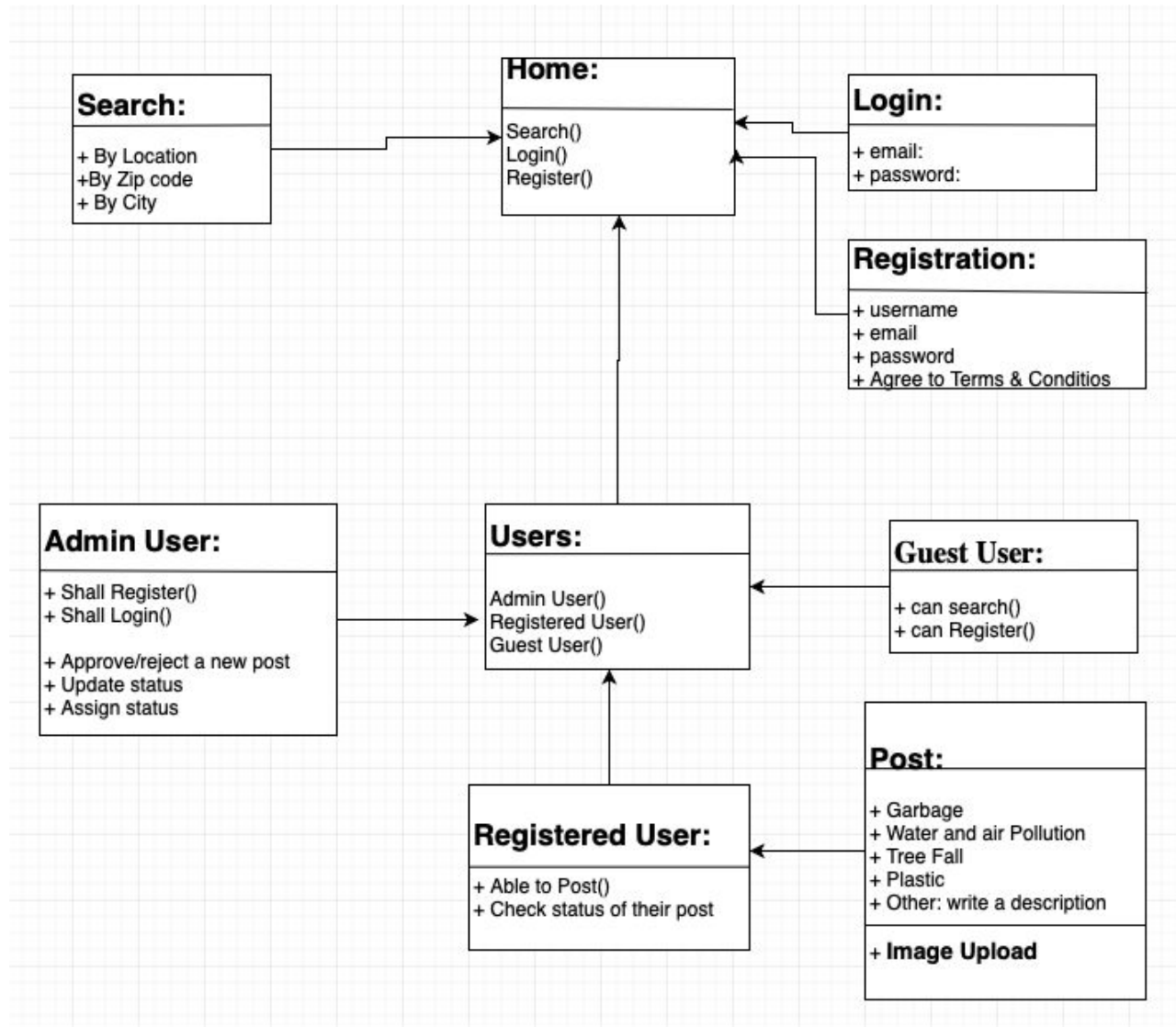   x. Topic: char (dropdown list of items: garbage, air, water pollution, etc)
   b. Users
      i. Unique ID: increment starting at 0; primary key
      ii. Username: char
      iii. Name: char
      iv. City: char
      v. Zip Code: int
      vi. Password: encrypted
2. **Media Storage:** Due to limited 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.
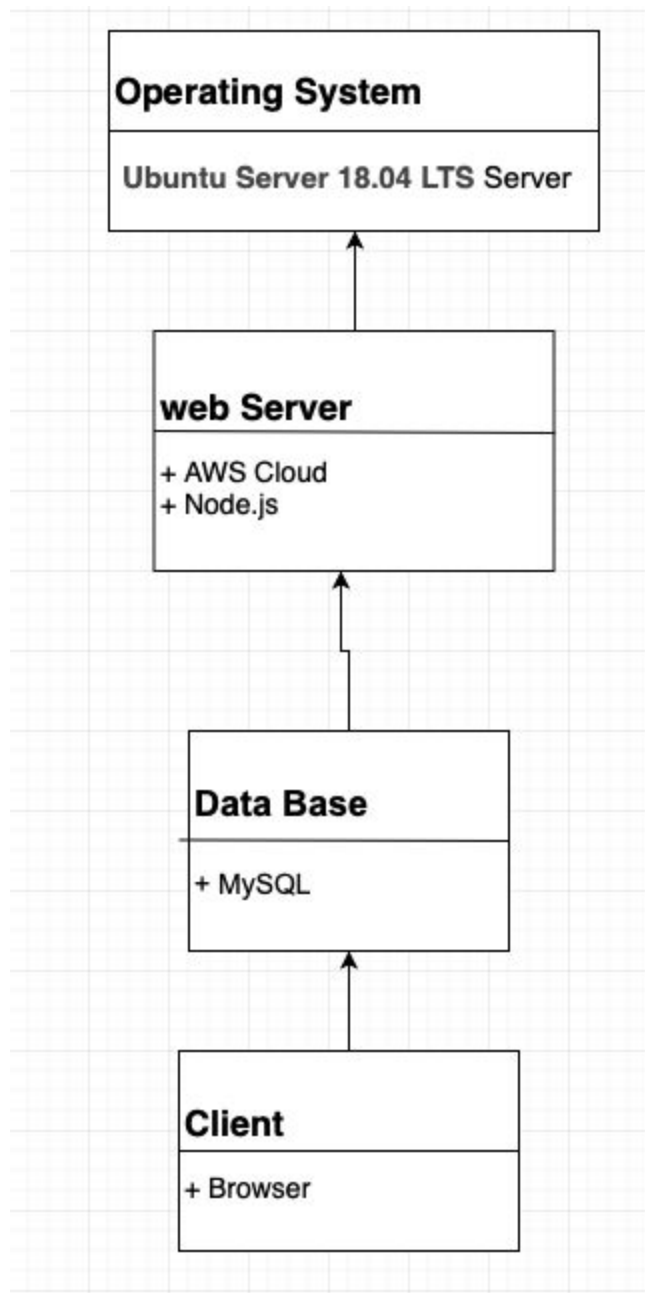3. **Search/filter:** Can use LIKE with % wildcards like mentioned in class for easiest method. For filters, we can have drop-down menu next to search bar to search for specific zip code, city, state, etc

5. High Level UML Diagrams

High-level UML class diagrams

Search:
+ By Location
+By Zip code
+ By City

Home:
Search()
Login()
Register()

Login:
+ email:
+ password:

Registration:
+ username
+ email
+ password
+ Agree to Terms & Conditios

Admin User:
+ Shall Register()
+ Shall Login()

+ Approve/reject a new post
+ Update status
+ Assign status

Users:
Admin User()
Registered User()
Guest User()

Guest User:
+ can search()
+ can Register()

Registered User:
+ Able to Post()
+ Check status of their post

Post:
+ Garbage
+ Water and air Pollution
+ Tree Fall
+ Plastic
+ Other: write a description

+ **Image Upload**

UML Component and deployment diagrams

```
┌─────────────────────────────────────┐
│ Operating System                    │
├─────────────────────────────────────┤
│ Ubuntu Server 18.04 LTS Server      │
└─────────────────────────────────────┘
                  ▲
                  │
┌─────────────────────────────────────┐
│ web Server                          │
├─────────────────────────────────────┤
│ + AWS Cloud                         │
│ + Node.js                           │
└─────────────────────────────────────┘
                  ▲
                  │
┌─────────────────────────────────────┐
│ Data Base                           │
├─────────────────────────────────────┤
│ + MySQL                             │
└─────────────────────────────────────┘
                  ▲
                  │
┌─────────────────────────────────────┐
│ Client                              │
├─────────────────────────────────────┤
│ + Browser                           │
└─────────────────────────────────────┘
```

<u>6 .Identify **_actual_** key risks for your project at this time</u>

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. Although we have a strong team who are technically well versed and have the experience to deliver successfully, we do face some risk factors that need to be addressed 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. What assists all of us is that we have our team lead Austin, who is technically strong and is comfortable with the framework being used.

We're not all proficient with MySQL database, however we all bring various experiences to the table so it's a learning experience for all of us and we've taken this challenge head on in order to learn it.

Another crucial factor that our team is aware of is the time constraint. The time frame that's usually four months is reduced to ten weeks, which means our deadlines reach sooner. In order to tackle this issue, we balance out the work and help each other to ensure we all are 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 tackle copyright policies to ensure not to violate it whether we or the users claim ownership of posted pictures, videos, etc. To do this, we will put in place some legal measures so intellectual information of people online are claimed.

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 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 handing 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.

⊙ Unwatch ▾ 4   ★ Unstar 3   ⑂ Fork 0

<> Code   ⓘ Issues 4   ⑂ Pull requests 0   ▥ Projects 0   ▤ Wiki   🛡 Security   ⊪ Insights   ⚙ Settings

Filters ▾   🔍 is:issue is:open   🏷 Labels 9   ⬚ Milestones 0   New issue

☐   ⓘ **4 Open**   ✓ 7 Closed   Author ▾   Labels ▾   Projects ▾   Milestones ▾   Assignee ▾   Sort ▾

☐   ⓘ **Backend DB design**
      #16 opened 10 days ago by atsang14

☐   ⓘ **Create Mockup UI**
      #15 opened 10 days ago by atsang14

☐   ⓘ **Create backend routes**
      #14 opened 10 days ago by atsang14

☐   ⓘ **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

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.