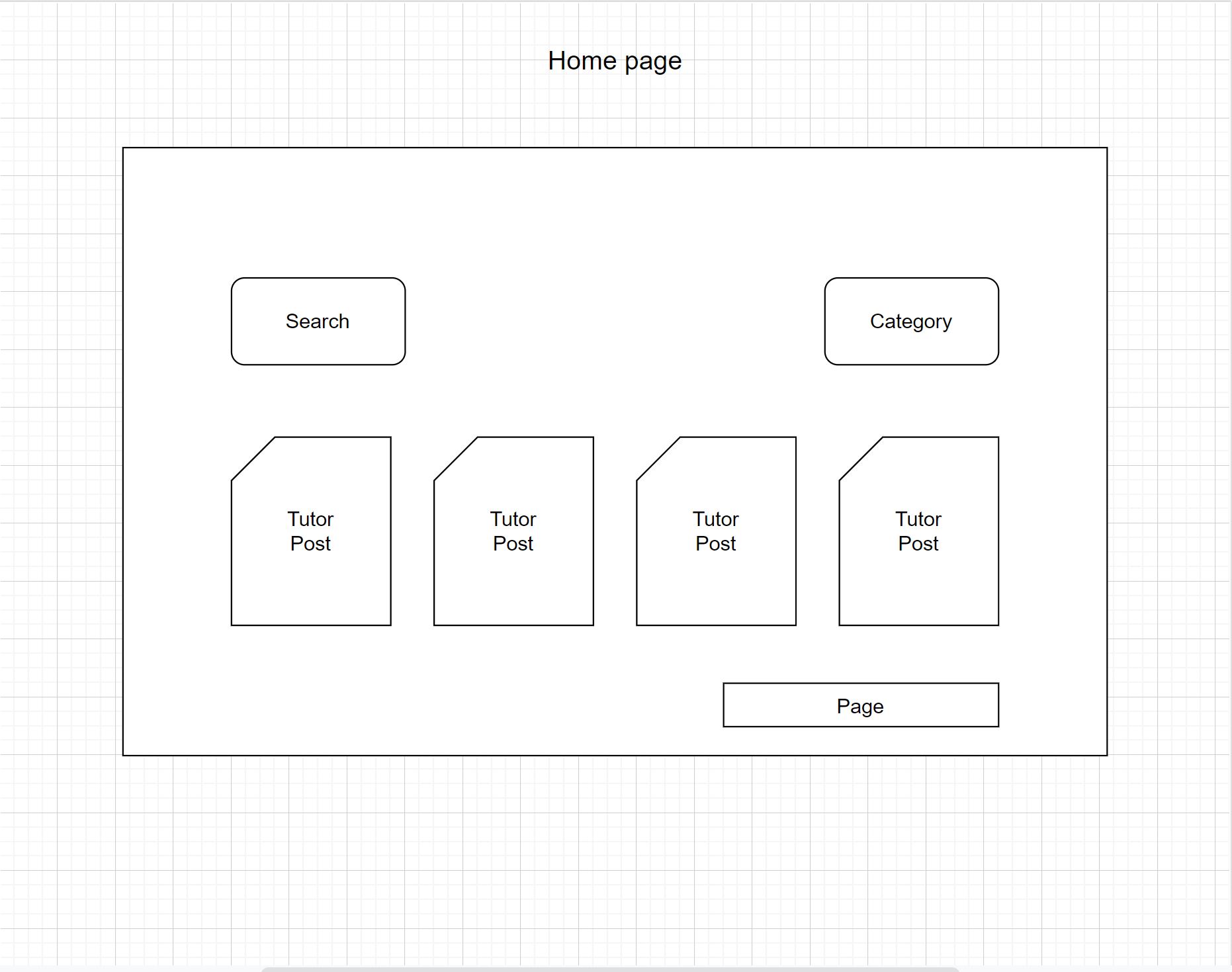
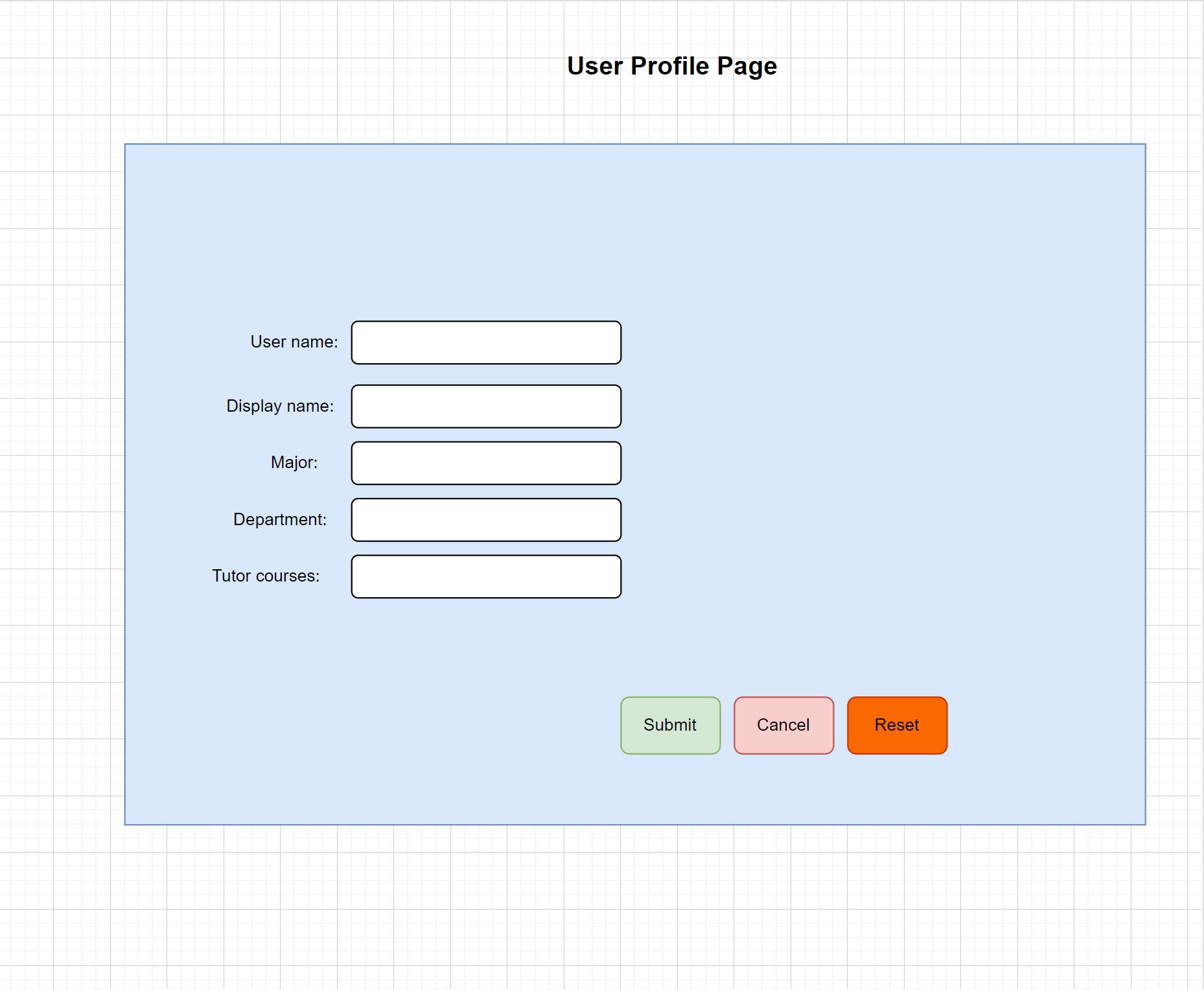
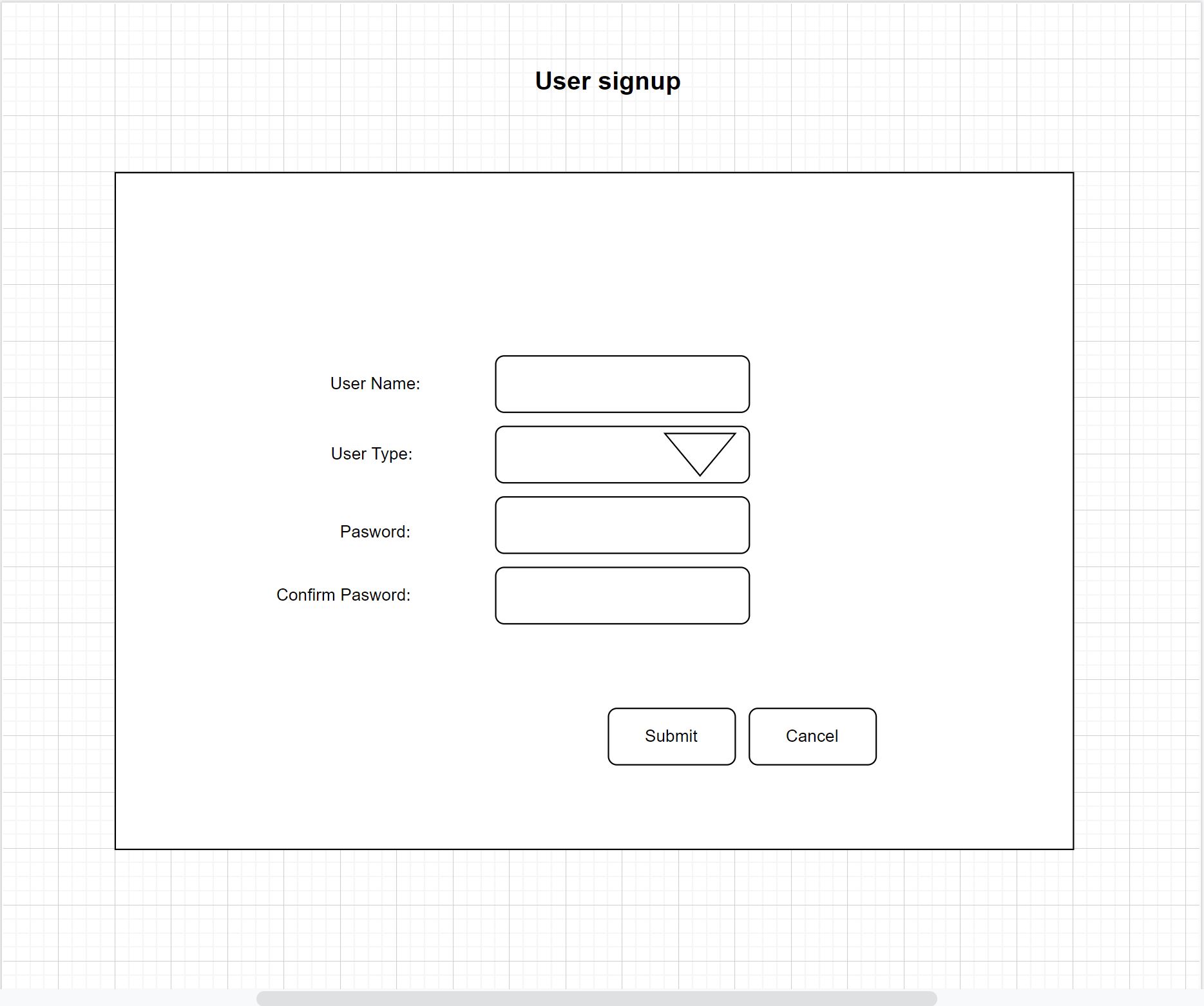
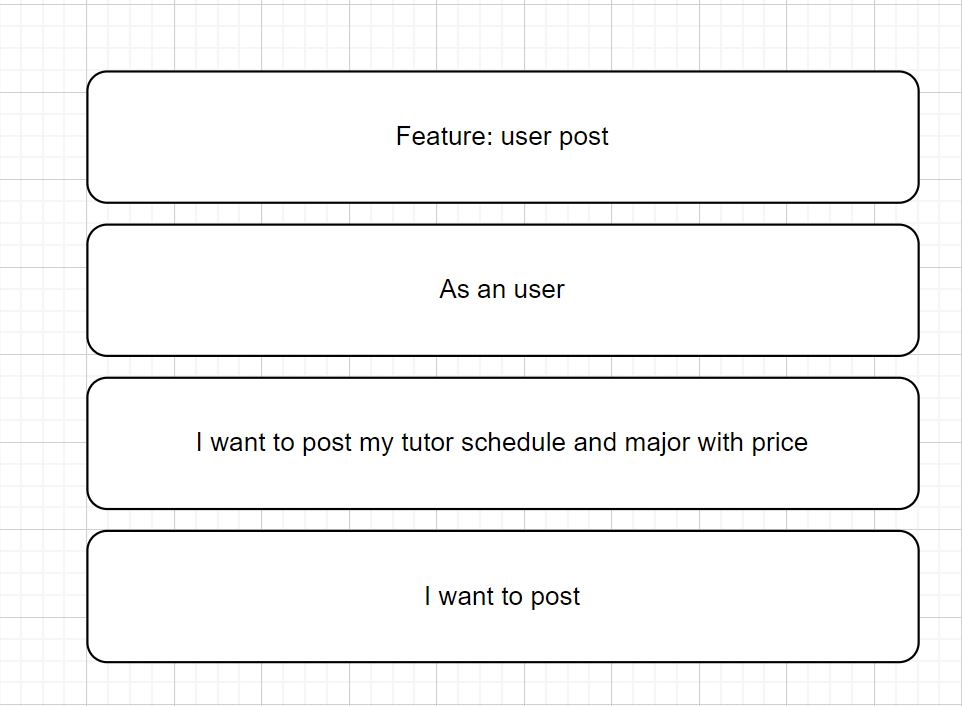
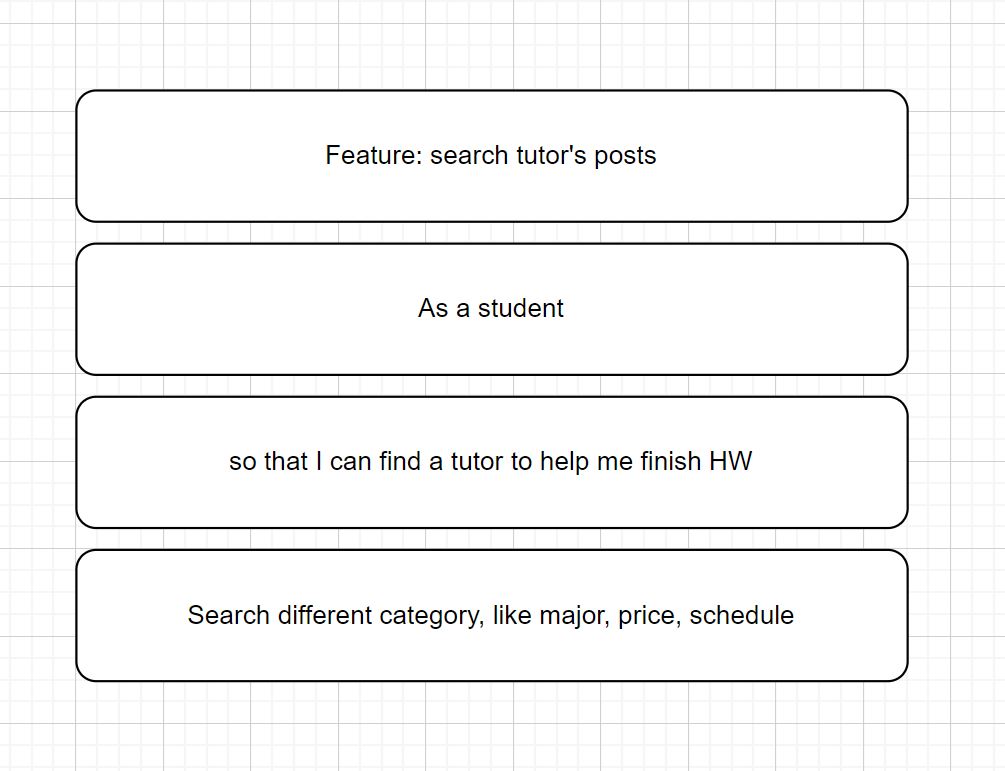
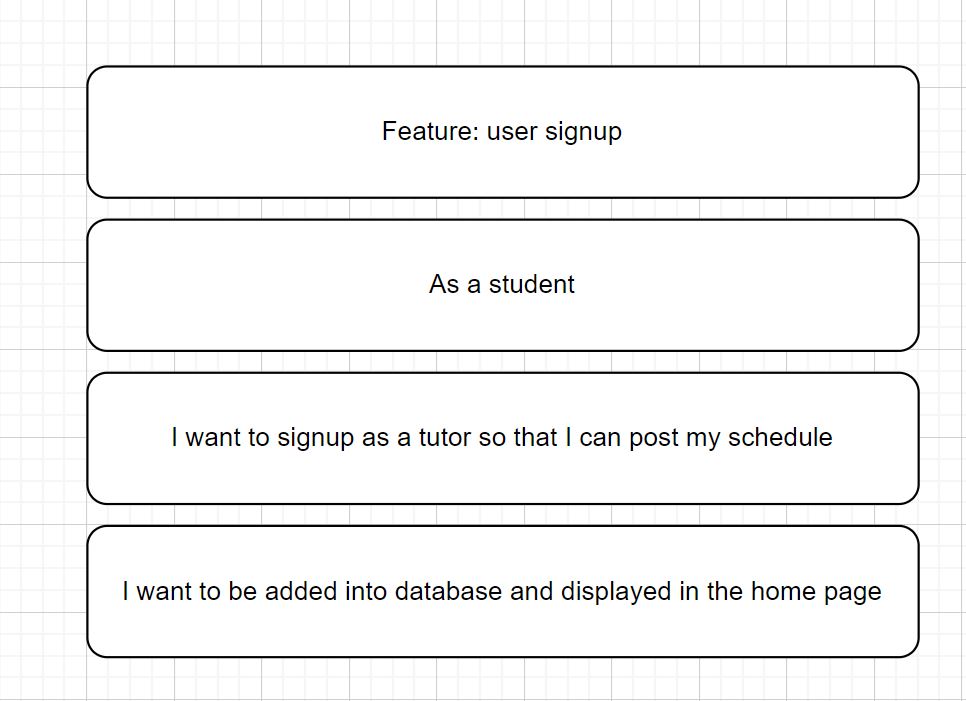
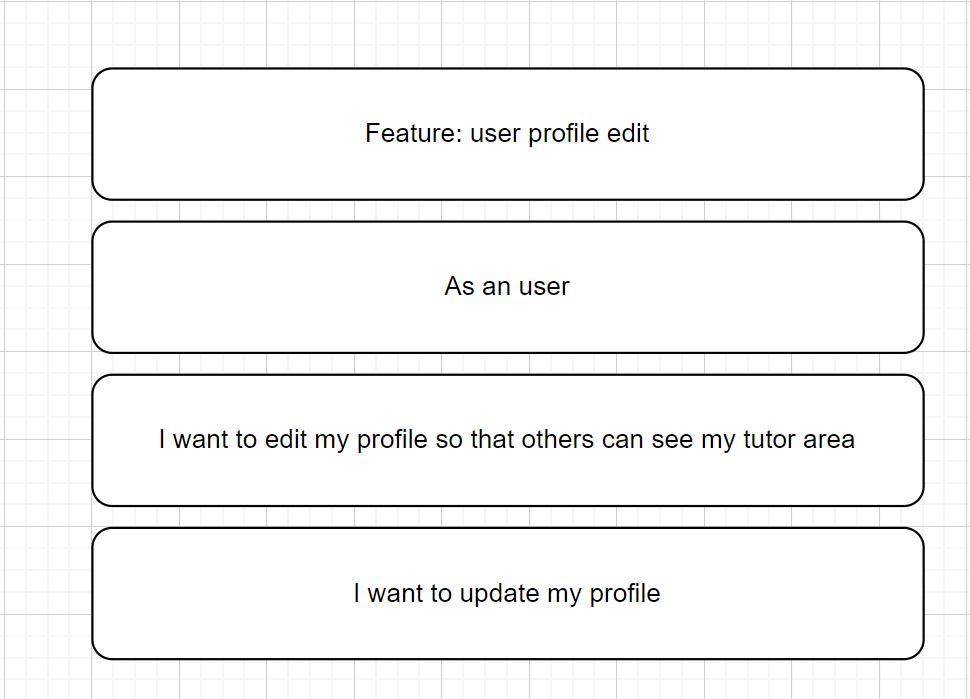
1. Two paragraph summary of the project *as implemented*, including the main customer need and how the application meets it, including who the stakeholders are. This will contrast to what you wrote in Iteration 0.

The main customer need is: to build a web platform for students to find proper tutors. Based on a large demand of tutoring in this university of 55,000 undergraduate students, our customer want to build a web platform for senior student to post tutoring information and for junior students to seek proper teachers in the website. She plans to run this website as a non-profit information sharing platform.

Based on these requirements, we implemented the project as a digital board for those tutors to post their information to match those potential students. Students could register on our website as tutor and post their available tutoring courses. The post includes title, tutor major, available date, self-description and price. Other students could search for tutors or posts that match their interests on our website. In the future, this application can be extended into broader fields like used things sales information, activities information center that anything happens on that physical board.

1. Description of *all* user stories (including revised/refactored stories in the case of legacy projects). For each story, explain how many points you gave it, explain the implementation status, including those that did not get implemented. Discuss changes to each story as they went. Show lo-fi UI mockups/storyboards you created and then the corresponding screen shots, as needed to explain to stories.



1. For legacy projects, include a discussion of the process for understanding the existing code, and what refactoring/modification was performed on the code, in addition to the user stores listed above.

We don’t have legacy code

1. List who held each team role, e.g. Scrum Master, Product Owner. Describe any changes in roles during the project.

Team role:

* 1. Anmin Fang: Product Owner & full stack developer
  2. Dawei Xiang: Scrum Master &Back-end developer
  3. Yiming Li: front-end developer

Lei Zheng left our team halfway through.

1. For each scrum iteration, summarize what was accomplished and points completed.

Iteration 0:

We met the client on 02/27/2022, and discussed the main function and requirement for the platform.

Iteration 1:

We met the client on 3/10/2022 and set up a few user stories to implement first. We showed our design for user model and post model. The client only required signup with username and password, and other detailed information would be filled when signing in. The client also said that post table should have columns as title, major, price, schedule and self-description.

user signup (4): to allow users to signup and create their profiles. We accomplished several tasks in this iteration, as database table design, backend model design and map to database and frontend implementation.

user profile edit (4): to allow users edit their profiles. We created a profile model, controller and view in rails.

We used rspec to develope test cases for user model and controller.

Iteration 2:

We met the client on 3/17/2022 and updated our progress. We showed the client the user signup and profile edit function. We discussed the next function of editing posts.

public post: to allow users to public their post. we added post’s model, view and controller.

Iteration 3:

We met the client on 4/8 2022 and updated our progress. We showed the client the post edit function. Client wanted us to add a search function at homepage and we discussed some detail about the search function.

add login in function: to allow users login in with their passcode.

Display posts: to show all the posts that tutor created.

We used rspec to develop test cases for post model and controller.

Iteration 4:

We met the client on 4/21/2022 and updated our progress. We showed the client the post searching function, and client wanted us to show the recent posts on the homepage.

search post: to allow users to search some topics in post. Thus, they could find the tutor they are interested in. We designed search controller, search function in model.

1. List of customer meeting dates, and description of what happened at the meetings, e.g. what software/stories did you demo.

We formally meet with our customer five times:

|  |  |
| --- | --- |
| date | Discuss content |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Explain your BDD/TDD process, and any benefits/problems from it.

We developed our platform following BDD combined with TDD. At each iteration, we developed a few user process and user needs based on the user stories we finished at the beginning. Then we wrote test cases for these user stories. We first implemented our code to pass the test. Later we refactored our code to make it cleaner and more efficient.

BDD helped us capture what we should do in each iteration. Following the general target, we wrote our test cases to guide the detailed code implementation in TDD. Developing in TDD alone forces us to change the code every time we add new test implementation, while if in BDD alone, it’s hard to check special test case and we don’t have specific target. So we adopt BDD + TDD.

1. Discuss your configuration management approach. Did you need to do any spikes? How many branches and releases did you have?
2. Discuss any issues you had in the production release process to Heroku.

We had several issues:

1. The development database is different from the production database. So we configure database again in Heroku.
2. Describe any issues you had using AWS Cloud9 and GitHub and other tools.
3. Describe the other tools/GEMs you used, such as CodeClimate, or SimpleCov, and their benefits.

We used SimpleCov to analyze our code. SimpleCov automatically test our project and generate test results reports, thus we could easily know the coverage across our test suites.

1. Make sure all code (including Cucumber and RSpec!) is pushed to your public GitHub repo.
2. Make a separate section discussing your repo contents and the process, scripts, etc., you use to deploy your code. Make very sure that everything you need to deploy your code is in the repo. We have had problems with legacy projects missing libraries. We will verify that everything is in the repo.
3. Links to your Pivotal Tracker, public GitHub repo, and Heroku deployment, as appropriate. Make sure these are up-to-date.
4. Links to your poster video and demo video.

iteration 4 report: Find tutor

github: <https://github.com/anminfang-tamu/tamu-find-tutor>

heroku: <https://tamu-find-tutor.herokuapp.com/> use <https://tamu-find-tutor.herokuapp.com/users> to see current users and use ​​ [https://tamu-find-tutor.herokuapp.com/](https://tamu-find-tutor.herokuapp.com/users)users/new to add user

pivotal tracker: <https://www.pivotaltracker.com/n/projects/2556191>

1. customer meeting

Date: 4/21 Wed

Duration: 10 mins

Discussion content: We just discussed recent progress. We told client that we have finished the research function.

1. User stories implemented

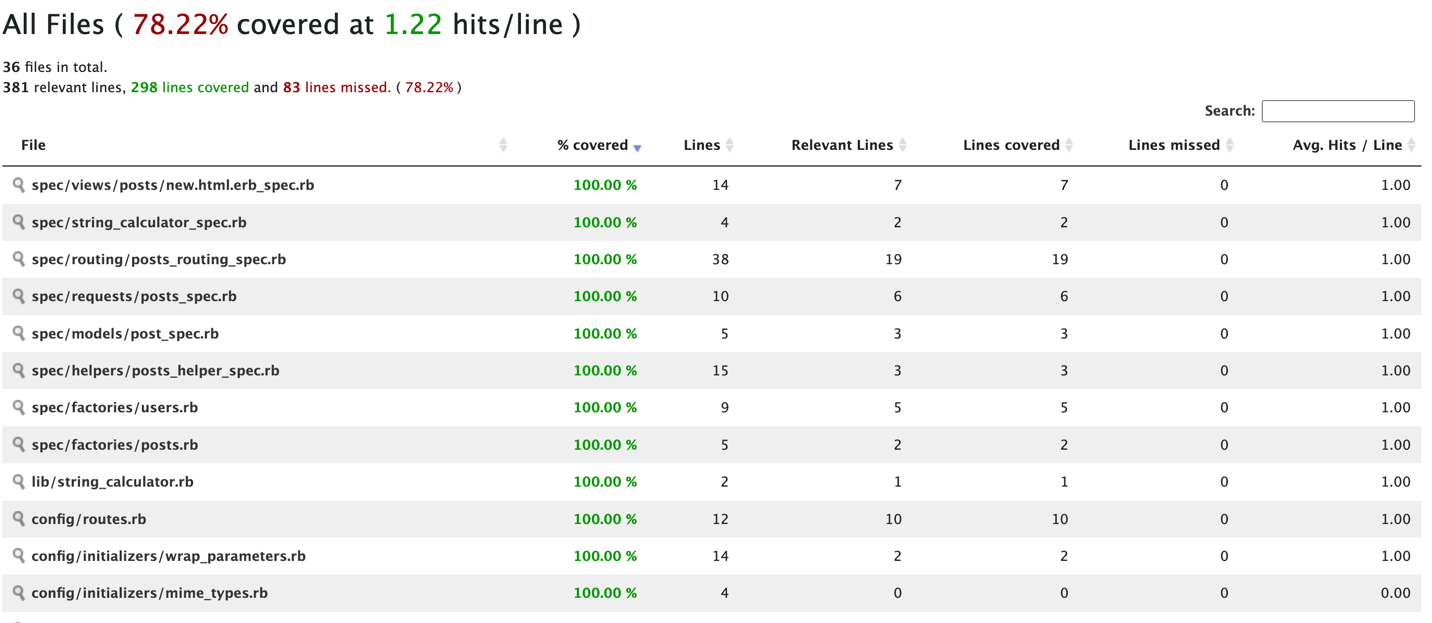
Last time, we left two user stories of adding a search function. we have implemented two user stories in this iteration:

2.1 search post: to allow users to search some topics in post. Thus they could find the postes they are interested in. It was divided into several tasks, as search controller design, search function in model and redirect in top\_bar.

2.2 search post: to allow users to search some specific tutor. Almost the same as search post. Thus, they could find the tutor they are interested in. We implemented by adding a search method in user model.

1. Rspec test

We used rspec to develope some test cases for our model, controller. Factories, helper, request and view. The simplecov coverage is as follow(the result can also be seen in document/spring2022/simplecov.html:



Because codeclimate doesn’t approve our application, we don’t have the codeclimate result.

1. relational database diagram

We created two models: user and post. Post has user\_id as foreign key. The relationship between user and post is as:

