Project Report:

Crowdsourcing of biodata curation website

This is the final report for the project of course CSCE 606 Software Engineering. The goal of this project is to design a website which serves as a basic framework for crowdsourcing of biodata curation. In Prof. Yu's requirements, the website should support that any interested individual is able to register on the website as a data curator and website should record the curation results of these registered data curators.

In the website we implemented, users can register with their Google+ or Facebook account, and answer questions which administrators assigned to them with reference data sets. Administrator can manage users, data sets and questions with features provided such as Promote/Demote vice admin, Search data set and propose question, Manage users in groups and Get the statics of users' answers.

Website: https://disease.herokuapp.com/
Provided by: Team Flyer
Team member: Xiaopei Xu (Product Owner), Wenrui Zhang(Scrum Master), Jianyu Zuo, Lang Feng, Bowen Li, Yinan Zhu
Customer: Prof. Peng Yu Website: https://yubiolab.wordpress.com/

Project Resources: https://github.com/xvxiaopei/CSCE606-project
Customer Interview: https://youtu.be/YVqrbsdxi5M
https://youtu.be/1UA2nmF-1dI
Demo: https://youtu.be/SOZwwola0a0
PivotalTracker: https://www.pivotaltracker.com/n/projects/1887551

Resources for Development and Deployment:

Github: https://github.com/
Heroku: https://www.heroku.com/
Cloud9: https://c9.io/
PivotalTracker: http://www.pivotaltracker.com

Legacy Report:

Basic user system includes two levels of users and a password encryption process.

Overall layout of the website is header, footer and body.

Users will see profile page after logging in. They can choose to answer the questions that are pre-stored inside the database. The questions are only about the connection of a disease and a dataset. The answers answered once cannot be edited later.

The administrator will see main admin page from the profile page after logging in. From the admin page, he can see all the diseases(questions) and all the submissions. The questions cannot be modified or added. And there's only 1 admin.

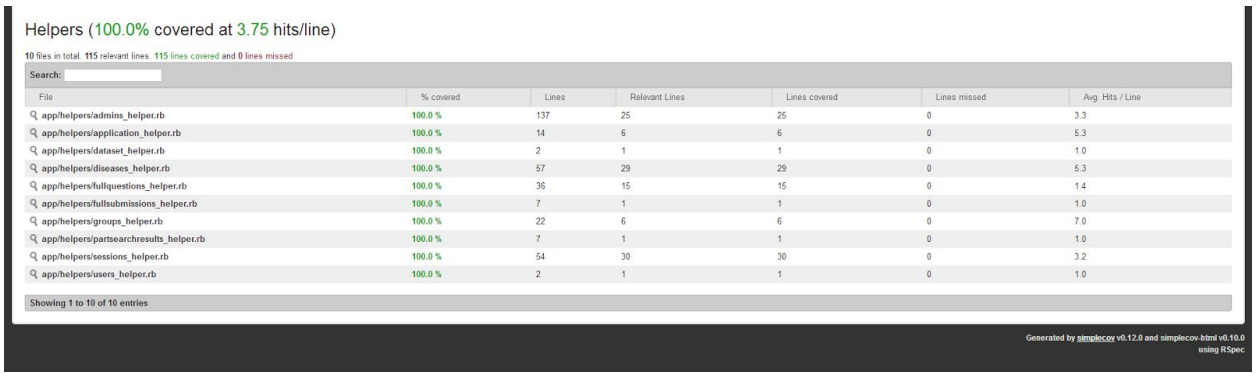
Legacy Refactoring:

The legacy codes use User has many submissions and has many diseases through submissions associations. We tested their codes on user system part and decided to adopt that part as our login system. In other words, we maintained their users model and the users/admins controller, and discard all the other parts since our goal is to develop an interactive way of adding/submitting/showing questions. The disease model of the legacy is hard to expand and ill-structured. Thus, we redesigned the database structure of the submissions and the questions to make them more reliable and easy to adapt the customer's demands. We accept the idea of User has many Fullsubmissions and has many Fullquestions through Fullsubmissions associations. Since our customer want to have a group system inside the user system. We accept the idea of user has and belongs to many groups association. Finally, we added some basic data analysis on the questions and submissions of the users to our system.

TDD+BDD Development:

We allocate user stories to group members on every weekly meeting. The 0-3 iterations' developments are based on TDD. We have our model tests, controller tests and feature tests. We tested all the models and controllers except the fullquestions_controller, which is completed during 4th iteration. This is partly because of time limit. However, we still use the interactive debugger to test this part function. So, we're sure of the user stories implemented on our website are well-tested and fully functional. Some of the controllers are legacy so they may influence the whole test coverage, for there's no test in the legacy codes. The TDD development process shows us an efficient way to transform users' need into the codes we should have. From the red failures to the green passes, we could develop the functions that are exactly what the users need. We believe this is the main advantage of the process of TDD and the platform Ruby On Rails, too.

Test Coverage(Simplecov):



Helpers (100.0% covered at 3.75 hits/line)

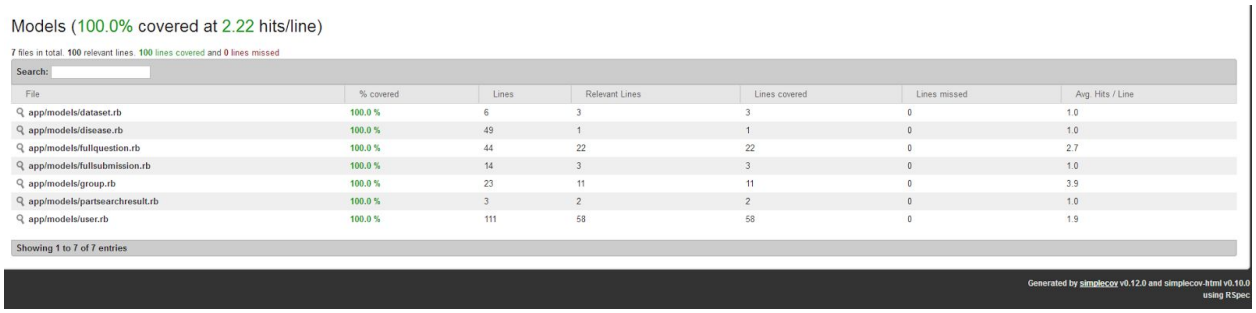
10 files in total. 115 relevant lines. 115 lines covered and 0 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/helpers/admins_helper.rb	100.0 %	137	25	25	0	3.3
app/helpers/application_helper.rb	100.0 %	14	6	6	0	5.3
app/helpers/dataset_helper.rb	100.0 %	2	1	1	0	1.0
app/helpers/diseases_helper.rb	100.0 %	57	29	29	0	5.3
app/helpers/fullquestions_helper.rb	100.0 %	36	15	15	0	1.4
app/helpers/fullsubmissions_helper.rb	100.0 %	7	1	1	0	1.0
app/helpers/groups_helper.rb	100.0 %	22	6	6	0	7.0
app/helpers/partsearchresults_helper.rb	100.0 %	7	1	1	0	1.0
app/helpers/sessions_helper.rb	100.0 %	54	30	30	0	3.2
app/helpers/users_helper.rb	100.0 %	2	1	1	0	1.0

Showing 1 to 10 of 10 entries

Generated by simplecov v0.12.0 and simplecov-html v0.10.0 using RSpec



Models (100.0% covered at 2.22 hits/line)

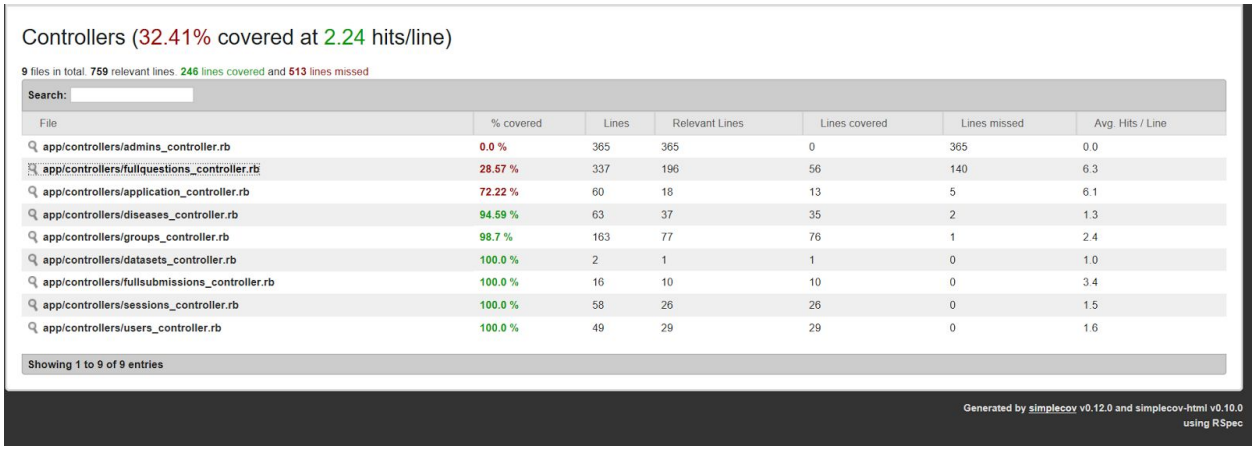
7 files in total. 100 relevant lines. 100 lines covered and 0 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/models/dataset.rb	100.0 %	6	3	3	0	1.0
app/models/disease.rb	100.0 %	49	1	1	0	1.0
app/models/fullquestion.rb	100.0 %	44	22	22	0	2.7
app/models/fullsubmission.rb	100.0 %	14	3	3	0	1.0
app/models/group.rb	100.0 %	23	11	11	0	3.9
app/models/partsearchresult.rb	100.0 %	3	2	2	0	1.0
app/models/user.rb	100.0 %	111	58	58	0	1.9

Showing 1 to 7 of 7 entries

Generated by simplecov v0.12.0 and simplecov-html v0.10.0 using RSpec



Controllers (32.41% covered at 2.24 hits/line)

9 files in total. 759 relevant lines. 246 lines covered and 513 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/admins_controller.rb	0.0 %	365	365	0	365	0.0
app/controllers/fullquestions_controller.rb	28.57 %	337	196	56	140	6.3
app/controllers/application_controller.rb	72.22 %	60	18	13	5	6.1
app/controllers/diseases_controller.rb	94.59 %	63	37	35	2	1.3
app/controllers/groups_controller.rb	98.7 %	163	77	76	1	2.4
app/controllers/datasets_controller.rb	100.0 %	2	1	1	0	1.0
app/controllers/fullsubmissions_controller.rb	100.0 %	16	10	10	0	3.4
app/controllers/sessions_controller.rb	100.0 %	58	26	26	0	1.5
app/controllers/users_controller.rb	100.0 %	49	29	29	0	1.6

Showing 1 to 9 of 9 entries

Generated by simplecov v0.12.0 and simplecov-html v0.10.0 using RSpec

(admins_controller is legacy; fullquestions_controller test needs more time)

Overall test coverage(counting legacy part): 47.23%

```
......................................................................" "
......

Finished in 1 minute 58.39 seconds (files took 4.78 seconds to load)
89 examples, 0 failures

Coverage report generated for RSpec to /home/ubuntu/workspace/DiseaseDatasetCuration/coverage. 461 / 976 LOC (47.23%) covered.
hawkzuo:~/workspace/DiseaseDatasetCuration (group_rspecs) $
```

Iteration Schedule

Iteration 0	Time period: Tue 09/27/2016 ~ Mon 10/10/2016	
Task Description	Accomplished by	Points
Write summary for iteration 0	Wenrui Zhang	1
Hold an interview with customers and create user stories and user interface, and produce a video for interview	Yinan Zhu,Bowen Li	5
Learn and setup github, Cloud9 and PivotalTracker	Lang Feng, Xiaopei Xu	3
Read legacy code and write a document	Jianyu Zuo	2

Iteration 1	Time period: Tue 10/11/2016 ~ Mon 10/24/2016	
Task Description	Accomplished by	Points
FB account login	Xiaopei Xu	3
Google+ account login	Wenrui Zhang	3
Modify and improve login page of legacy code	Wenrui Zhang	1
Learn test and write test code for login page	Lang Feng	2
Write report for iteration 1	Jianyu Zuo	1
Git legacy code and put website online	Xiaopei Xu	1

Iteration 2	Time period: Tue 10/25/2016 ~ Mon 11/07/2016	
Task Description	Accomplished by	Points
Create a tutorial page for our website	Jianyu Zuo	1
Create a feature to search datasets from ArrayExpress	Wenrui Zhang, Yinan Zhu	3
Create a feature to create new questions with datasets	Bowen Li, Lang Feng	3
Write test for features accomplished in this iteration	Jianyu Zuo	2
Write report for iteration 2	Wenrui Zhang	1

Iteration 3	Time period: Tue 11/08/2016 ~ Mon 11/21/2016	
Task Description	Accomplished by	Points
Design and implement hierarchy in users' levels	Jianyu Zuo, Xiaopei Xu	3
Implement admins' operations in groups	Jianyu Zuo, Xiaopei Xu	3
Implement a feature of answering questions	Wenrui Zhang,Lang Feng	3
Write report for iteration 3	Wenrui Zhang	1

Iteration 4	Time period: Tue 11/08/2016 ~ Mon 11/21/2016	
Task Description	Accomplished by	Points
Write test code for all features we accomplished	Jianyu Zuo	3
Produce videos for demo and user interview.	Yinan Zhu,Bowen Li	2
Design and create Poster.	Xiaopei Xu, Yinan Zhu	2
Implement a feature to show statics of users' submissions	Wenrui Zhang,Lang Feng	3
Write report for iteration 4	Wenrui Zhang	1

Meeting with Customers

Fri 10/07/2016 4:30pm-6:00pm CBGSE (101 Gateway, Suite A, College Station)
This is our first meeting with customer. We asked about the goal of this project and discussed the improvement we can do during this semester(as our project is a legacy project). We created some user stories and lo-fi UI mockups(please see details in our project resources in github). We also produced a video of customer interview.

Fri 10/21/2016 4:30pm-6:00pm CBGSE (101 Gateway, Suite A, College Station)
In this meeting we did a demo to show the new login and register system to our customer. Then we discuss with the customer about the Dataset Search and Question Propose systems. We also created several new user stories about users' level system.

Fri 11/04/2016 4:30pm-6:00pm CBGSE (101 Gateway, Suite A, College Station)
In this meeting we show the interface of Dataset Search and Question Propose systems and tutorial page. Then we get some advices to improve our system. And we also introduce our design of users' hierarchy of our website.

Fri 12/02/2016 5:00pm-7:30pm CBGSE (101 Gateway, Suite A, College Station)
In this meeting we deploy our website online and did a demo with both poster and our website. We show our customer all features we accomplished this semester, include new login and registration system, users' hierarchy and group operations, dataset search and question propose systems. Customer gave more user stories and some advice for improvement. We also produce a interview video.

User Stories

In this project, we implement 11 features. All features implemented has been deployed in our website online.
As the time is limited, there are some features still need to be implemented. We also record these user stories and have data and information in our project resources(can be found in our github repository), which should be helpful for other team in the future.

Feature 1:	Registration using Google+ and Facebook accounts	6 points
As a user who has Facebook and Google+ accounts So that I can register conveniently I want to register by these accounts		
Description:	We used some gems provided by facebook and google to help users login with their facebook or google account. In this way we can get some information of users(Email, name, etc.) and store them in our registration system.	

Feature 2:	Tutorial page	1 point
As a person without biodata curation training So that I can learn how to use the website I want to have access to training materials rendered in a user-friendly fashion		
Description:	We add a page to show the tutorial material which we get from the customer. Users can also download this material in PDF format.	

Feature 3:	Contact button	1 point
As a user with some problem of this website So that I can contact the owner of this webpage I want to contact the owner of this website when I meet some problem		
Description:	We add a “Contact” button at the bottom of every page. By click this button, user can email website owner.	

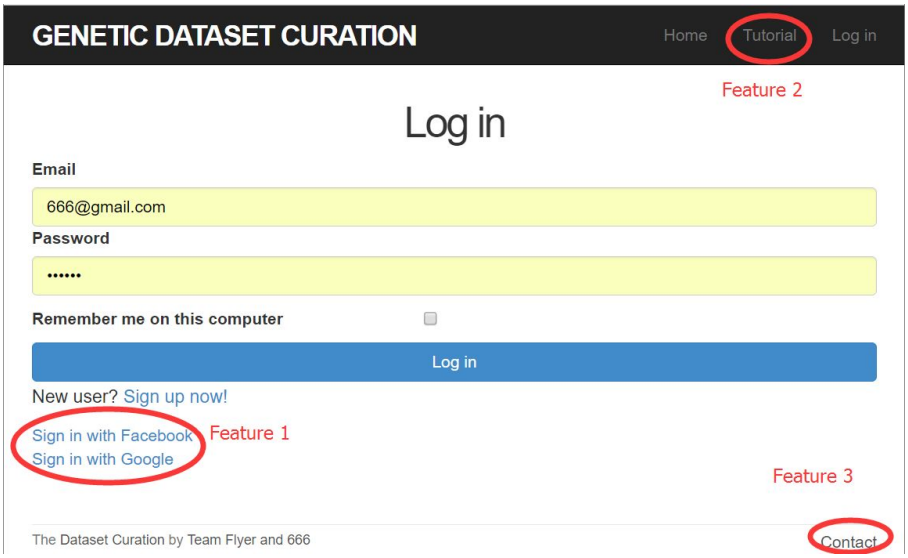


Figure 1. Feature 1, 2, 3

Feature 4:	Promote and demote a group administrator	3 point

As the administrator So that I can manage a user's level I want to promote a user to a group administrator or demote a group administrator to a user	
Description:	Administrator can see a function button in admin page. By click this button, adminstrator can get access to Promote/Demote page and get a list of all users. Then administrator can click promote/demote button to manage the user's level.

Feature 5:	Change/Reassign a group administrator to a group	3 point
As an administrator So that I can designate any user as a group administrator I want to change/reassign a group administrator		
Description:	Main administrator can click "Reassign Group Admins" button to change the job of each group admin, or use "Choose Group Admin from Group" to promote a group user from the group to become group admin of this group directly.	

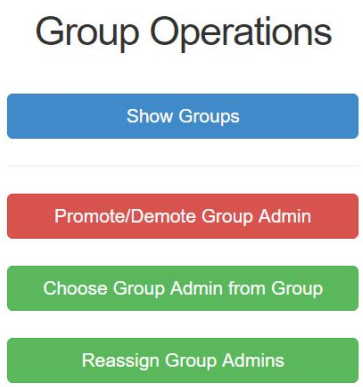


Figure 2. Feature 4,5,6

Feature 6:	Groups operation	3 point
As an administrator/group administrator So that I can manage users to groups, edit the group description or create/delete a group I want to manage groups		
Description:	Administrator can see "Show Groups" button in admin page. By click this button, adminstrator can get access to Groups page and get a list of groups he can manage.Then administrator can manage users, edit group, delete group or create group.	

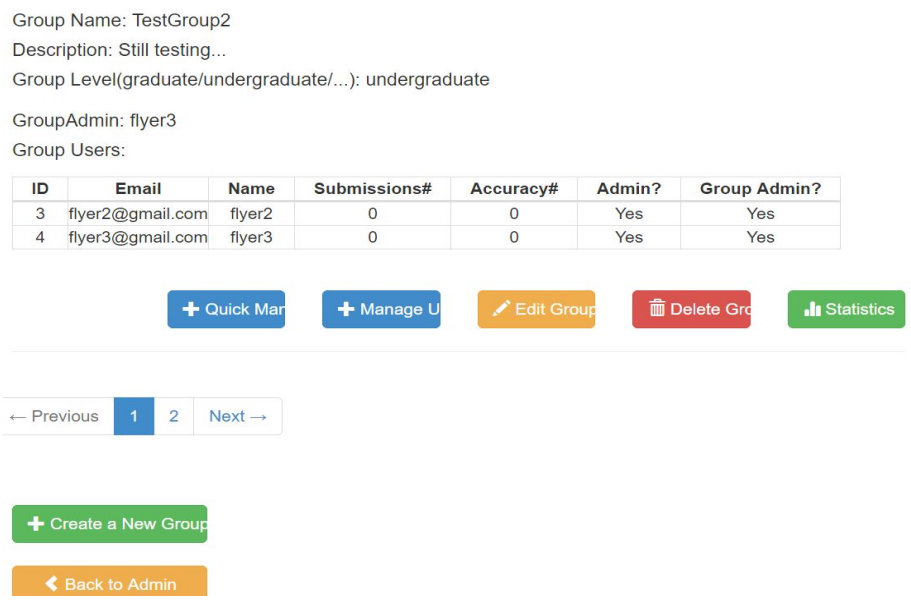


Figure 3. Feature 6

Feature 7:	Search datasets from public websites	2 point
As an administrator So that I can get a large number of datasets from public websites I want to search keywords in ArrayExpress or GEO		
Description:	Admin or group admin search dataset with a keyword from ArrayExpress, a biological website. Then, the results will be showed in a form. Admin can select the datasets he need. Finally, admin can submit the data sets and direct to adding question or back.	

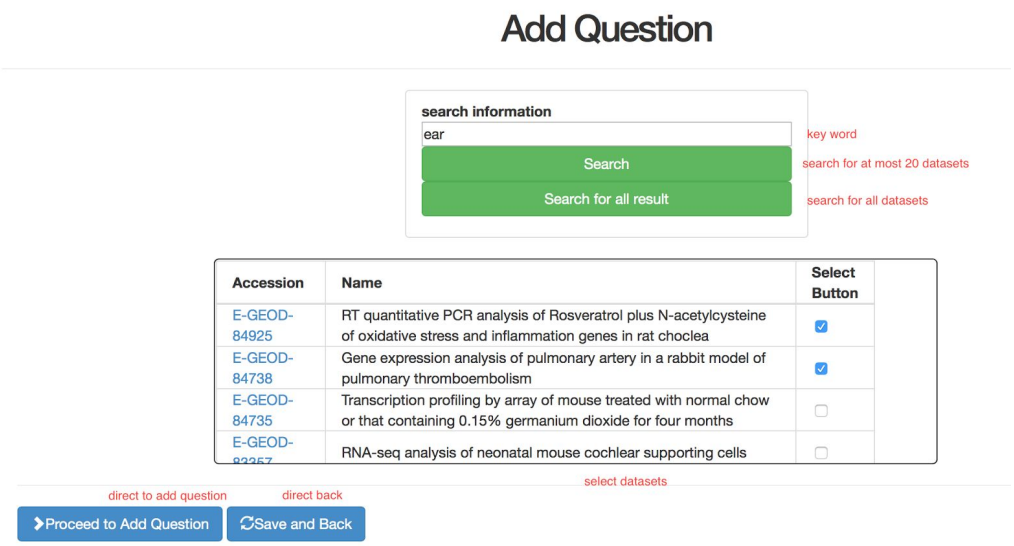


Figure 4. Feature 7

Feature 8:	Select datasets and add questions	2 point
As an administrator So that I can create questions regarding specific disease for users to answer I want to select datasets before I create questions		
Description:	Admin can add one or more questions according to the datasets he want to assign. He can write the question in text and choose to give or not give the standard answer. By clicking add another question, he can continue add questions to the same bunch of datasets.	

Feature 9:	Select groups to propose questions	1 point
As an administrator/group administrator So that I can submit questions to my groups I want to select groups before I propose questions		
Description:	After selecting datasets, admin can choose the groups he wish to assign these datasets as well as following questions. By clicking checkboxes, he can modify his selection.	

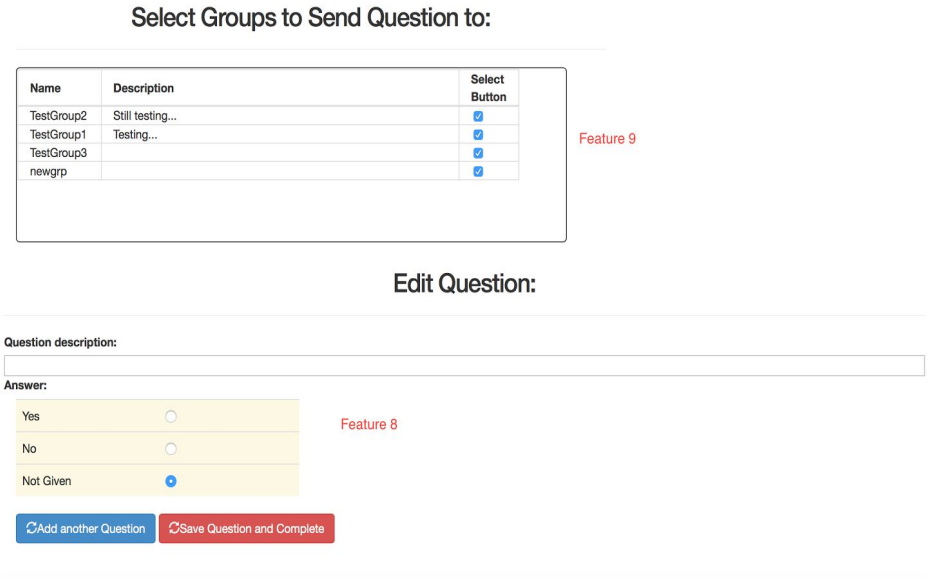


Figure 5. Feature 8,9

Feature 10:	Submit and modify answers to questions	3 point
As a user So that I can submit my question answers I want to answer the questions I received		
Description:	As a user, I can see all the questions assigned to me in the homepage, and click “start” to answer those questions. For the questions I have made choices, they will display in the bottom of each dataset, and I can see the choice I made last time. I can choose “yes” or “no” for each question, and add the reason for it.	

Feature 11:	Statistics of answers	3 point
As an administrator So that I can have an intuitionistic knowledge of the curation results I want to calculate the accuracy of each question, each user and draw graphs		
Description:	As an administrator, I can check the status of all the questions in every dataset by click “Dataset Statistic” in Admin page, where I will know how many submissions of every question and how many users answered it correctly. I can also check the statistics of users by click “All Users” in the homepage, where I can know the number of submission, correct submission and accuracy of every user. Finally I can also check the group statistics by click “Statistics” in Group page, where I can know the accuracy distribution of the users in every group.	

Feature 12:	Free form questions	N/A
As a administrator/group administrator So that I can propose a question in flexible form I want to propose questions in some other form instead of choice of yes/no		
Description:	Not implemented yet.	

Feature 13:	Dynamic and interactive training function	N/A
As a user So that I can learn how to use this page easily I want to learn how to use this website easily by following a interactive tutorial process		
Description:	Not implemented yet.	

New Gems Added:

1.omniauth-facebook: This gem is used to allow user to login with their facebook account, and receive information about their name and email from facebook server.

2.omniauth-google-oauth2: It is the same as the previous one. It allows user to login via their google account

3.json: It can receive the search results from an external website and parse the data into hash.

Acknowledgement:

We would like to extend out sincere thanks to Dr. Hank Walker, TAs and graders for helping us throughout the project. Special thanks to our client Dr. Peng Yu for being supportive. We also express gratitude towards the Department of Computer Science, Texas A&M for all the resources available to us and for this opportunity.