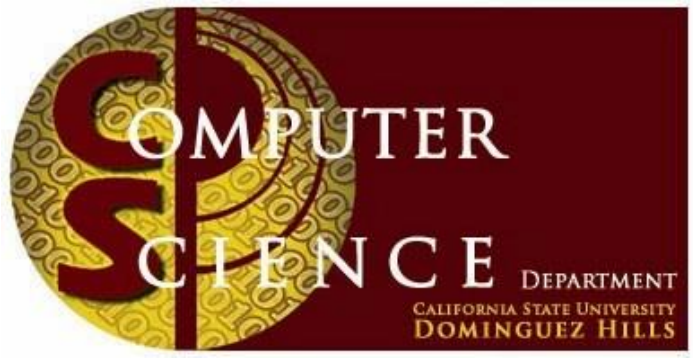


Spring 2018

Senior Project Report

Computer Science Department

California State University, **Dominguez Hills**

abstract definition source: https://users.ece.cmu.edu/~koopman/essays/abstract.html
Problem statement: What problem are you trying to solve?

- Answering the same question multiple times

Motivation: Why do we care about the problem and the results?

- Save people time and energy answering the same question

Approach: How did you go about solving or making progress on the problem?

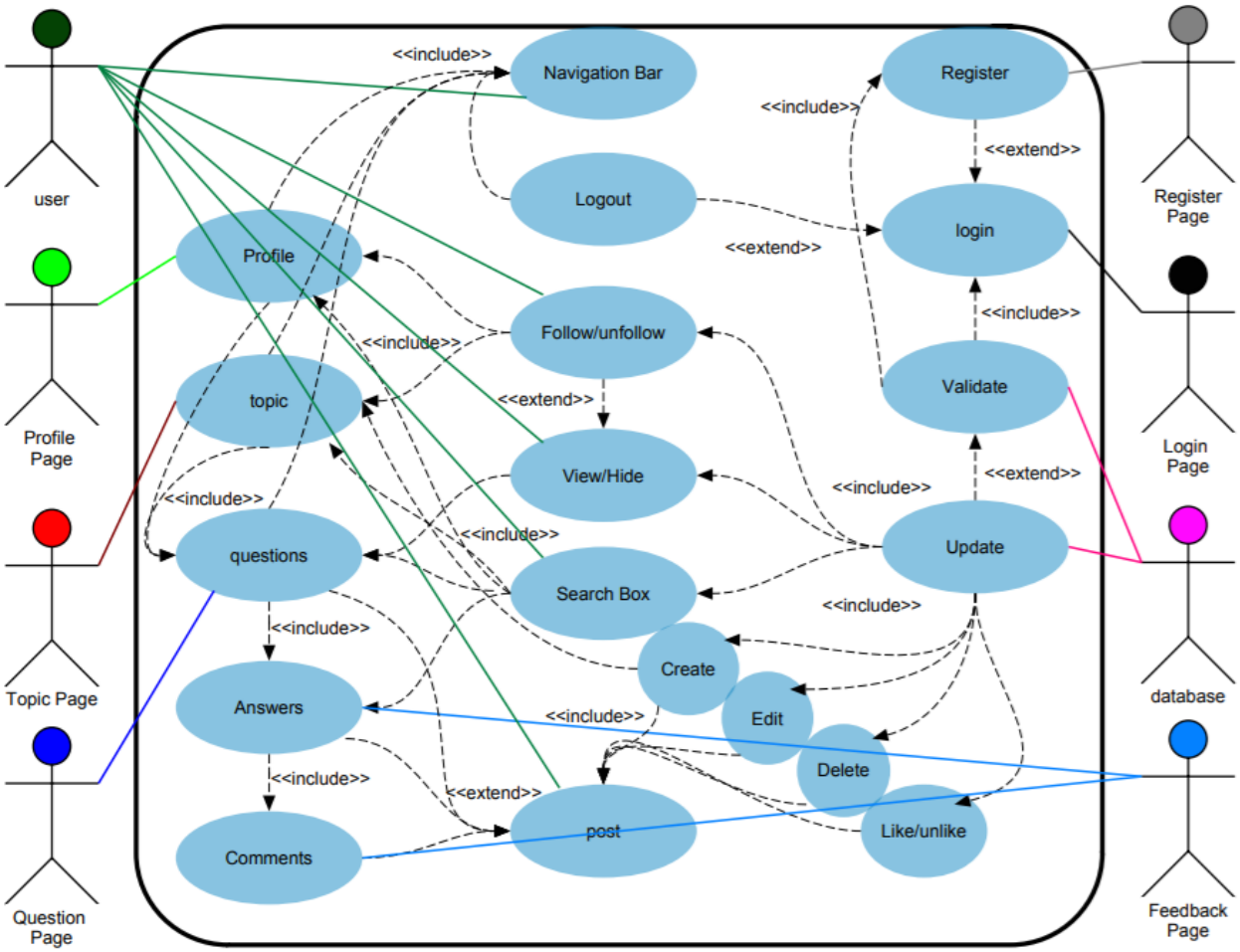
- I made a website called FAQexchange where people post frequently ask questions and answers.

Results: What's the answer?

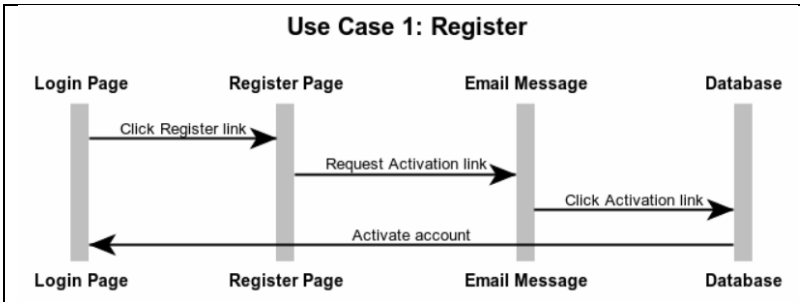
- I can answer question on FAQexchange,
- if someone ask me a question I already answered, I tell them to look at my FAQexchange.

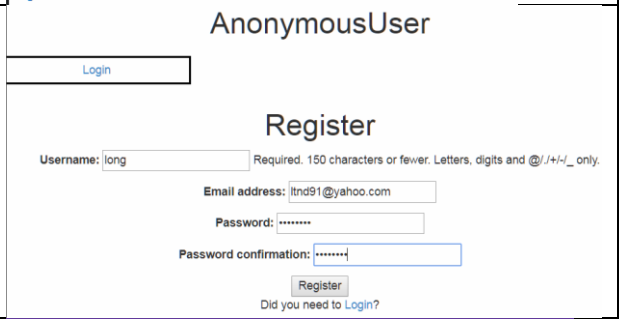
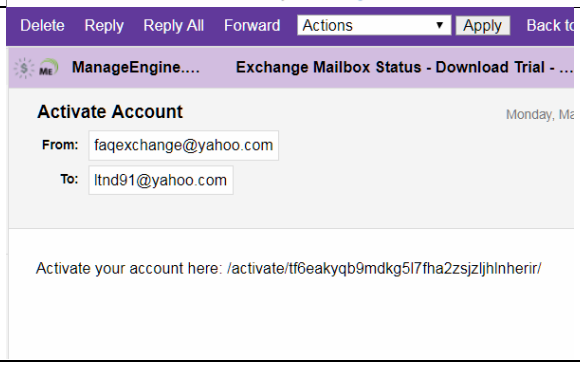
Conclusions: What are the implications of your answer?

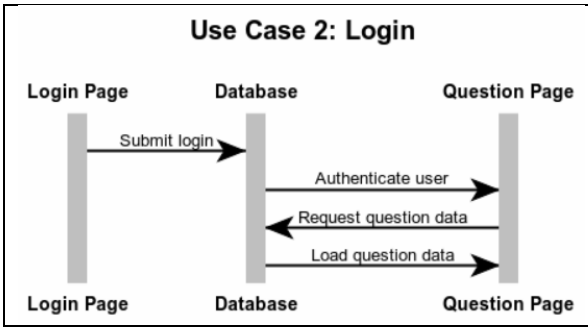
- People will look at a person FAQexchange before asking questions

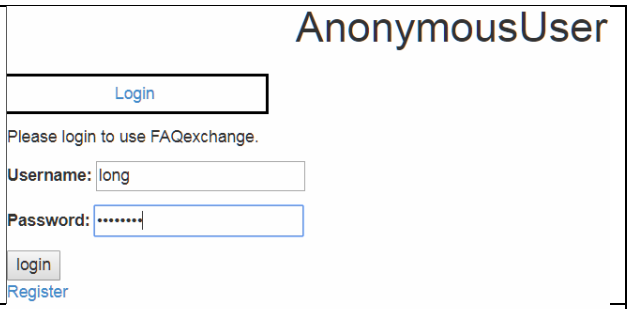
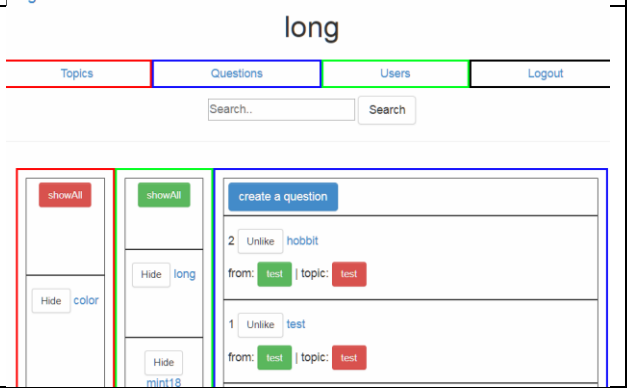


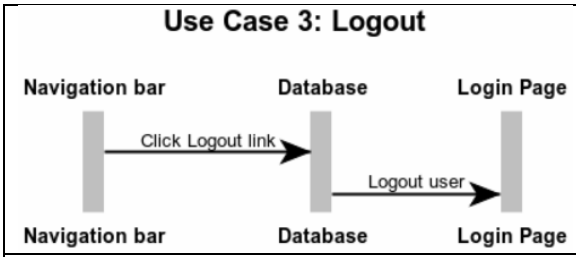
Actors	Use Cases
• User • Profile Page • Topic Page • Question Page • Feedback Page • Database • Login Page • Register Page	• New User Registration • User Login • User Logout • Follow/unfollow Profile • Follow/unfollow Topic • View/Hide Profile • View/Hide Topic • Search Box Filter 3 types of post: question, answer, comment • Create a post • Edit a post • Like a post • Delete a post

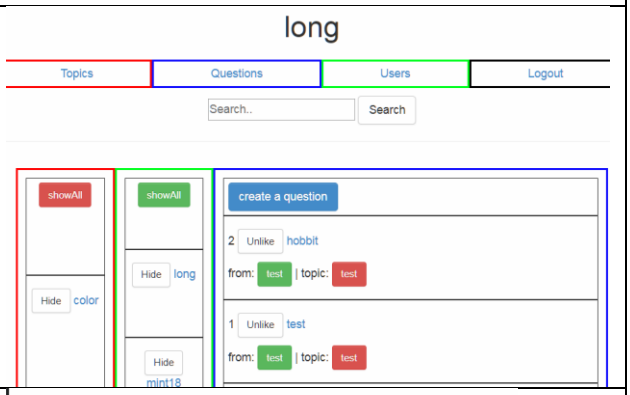
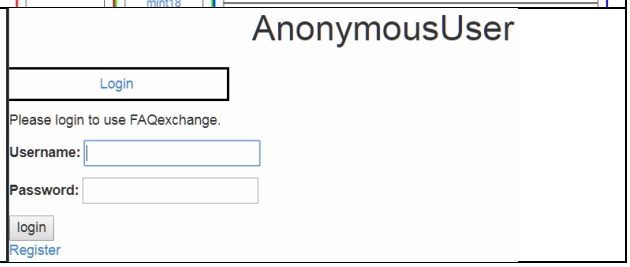


Click Register link	
Enter valid registration information Click register to request activation link	
Follow activation link to activate account	

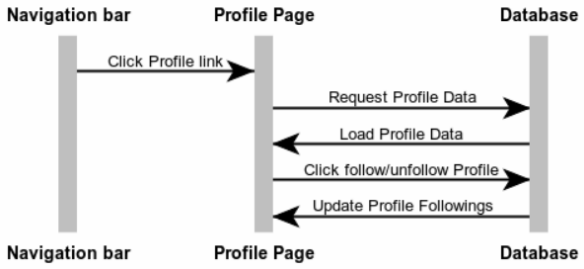


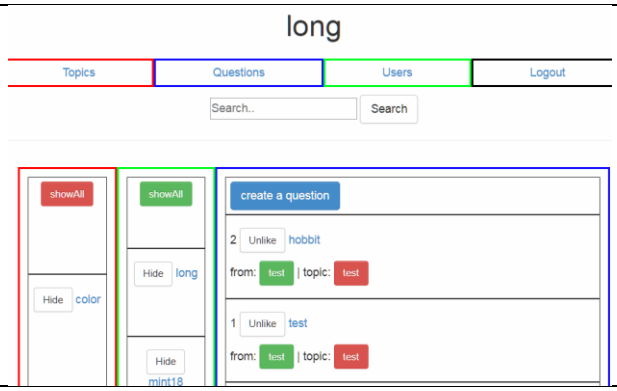
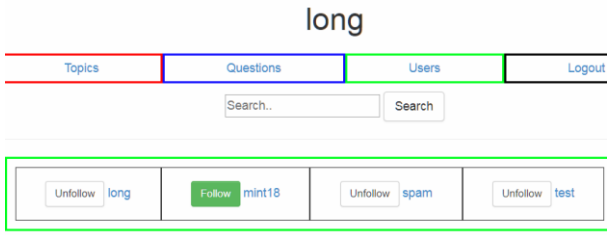
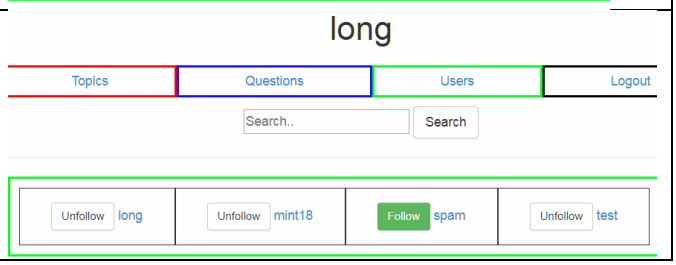
Submit Login	
Authenticate user (appears on top of page) Request question data Load question data	



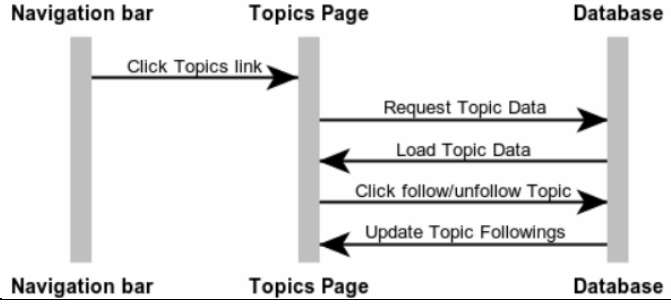
Click Logout link	
Logout user (AnonymousUser means no one is logged in)	

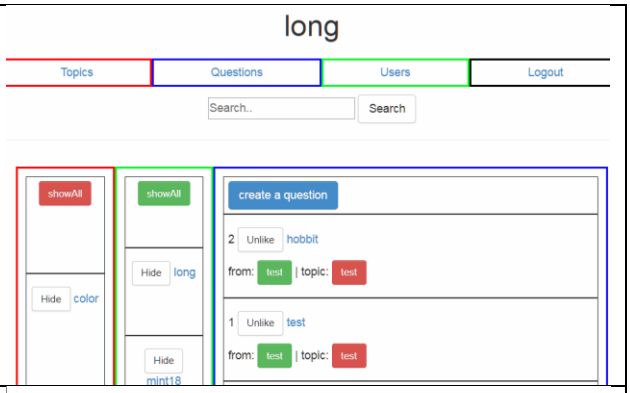
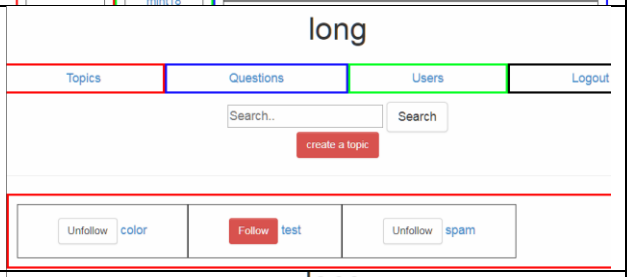
Use Case 4: Following Profile

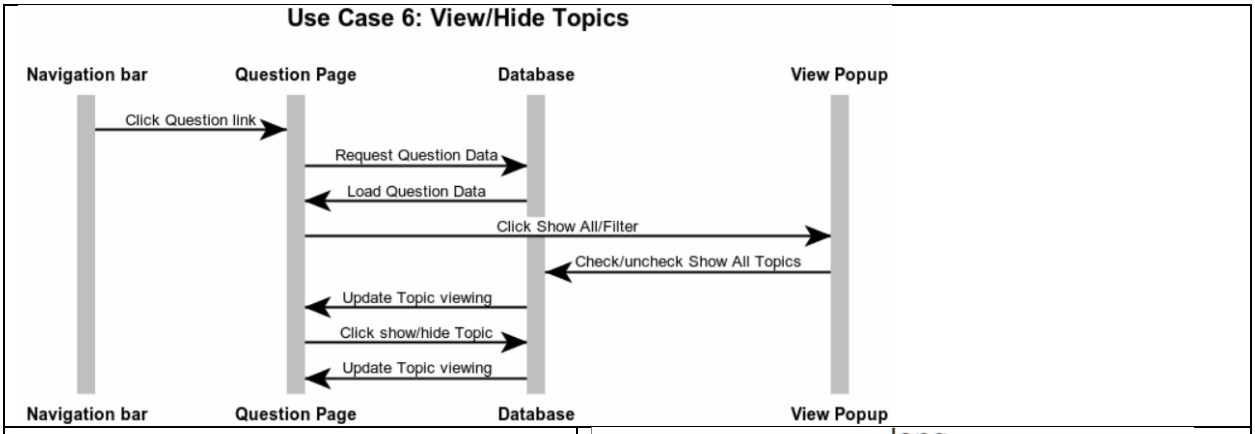


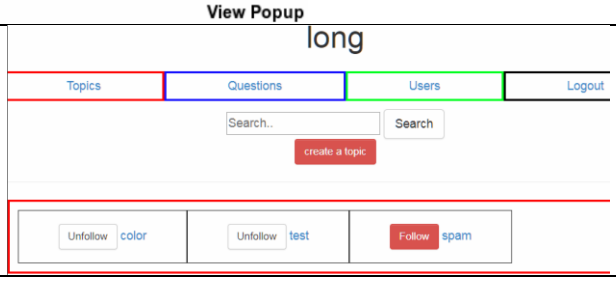
Click Profile(users) link	
Request profile Data Load Profile Data Click follow/unfollow Profile • Unfollow spam • Follow mint18	
Show Update of profile followings	

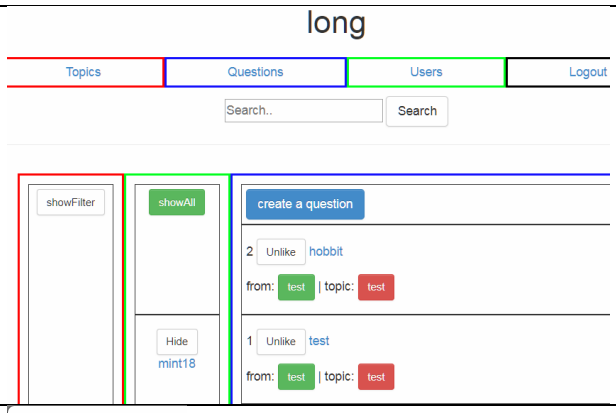
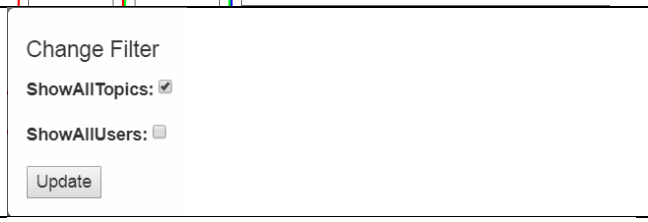
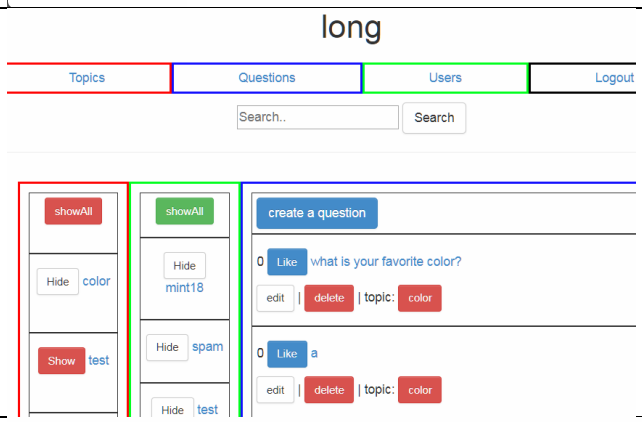
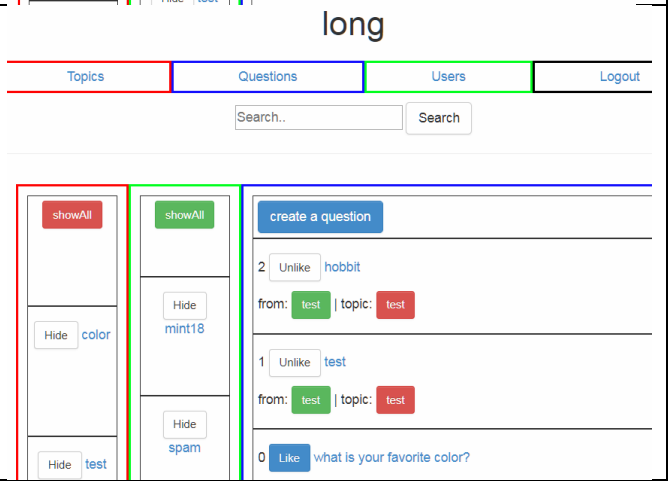
Use Case 5: Following Topics

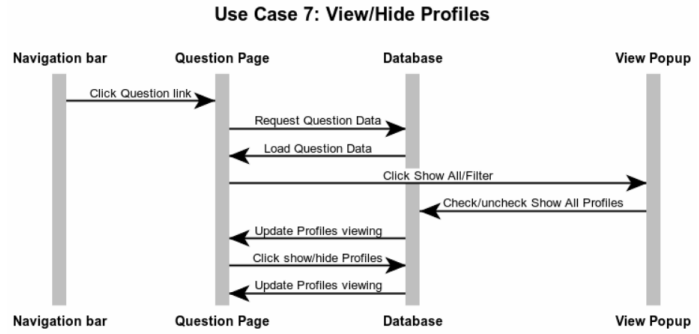


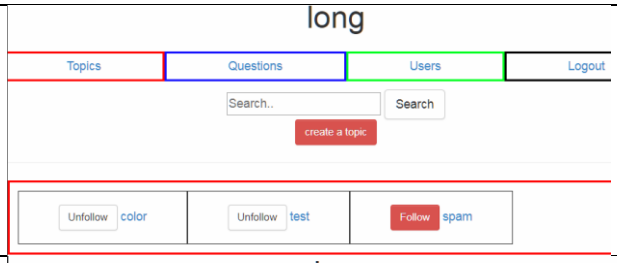
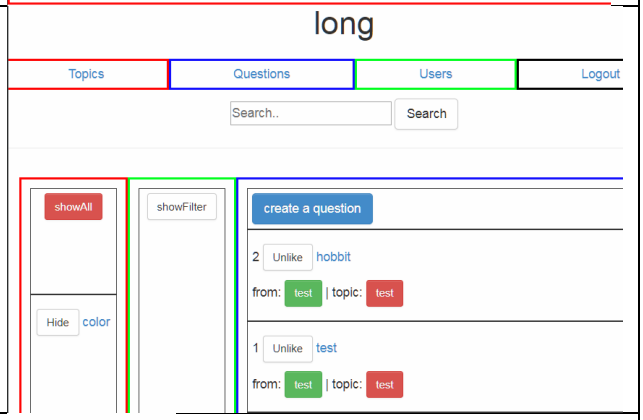
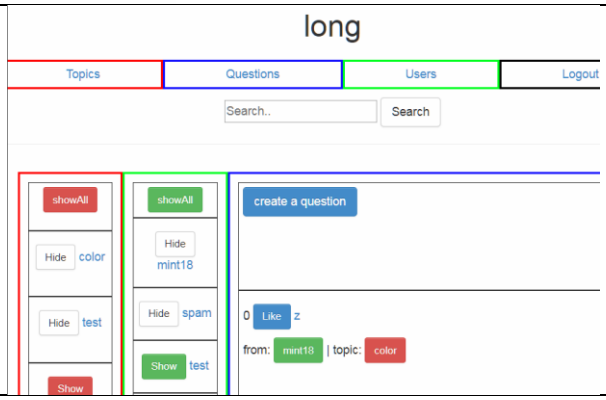
Click Topics link	
Request topic Data Load topic Data Click follow/unfollow topic • Unfollow spam • Follow test	
Show Update of topic followings	

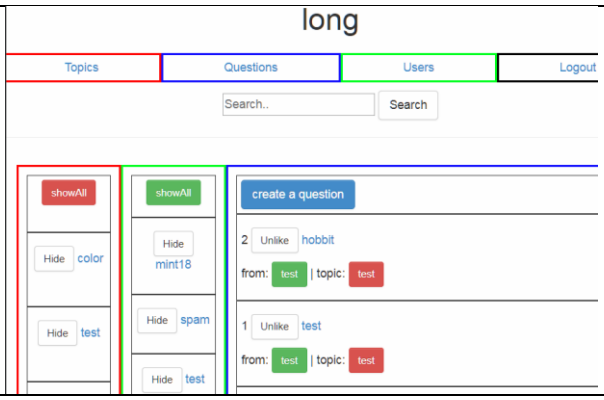


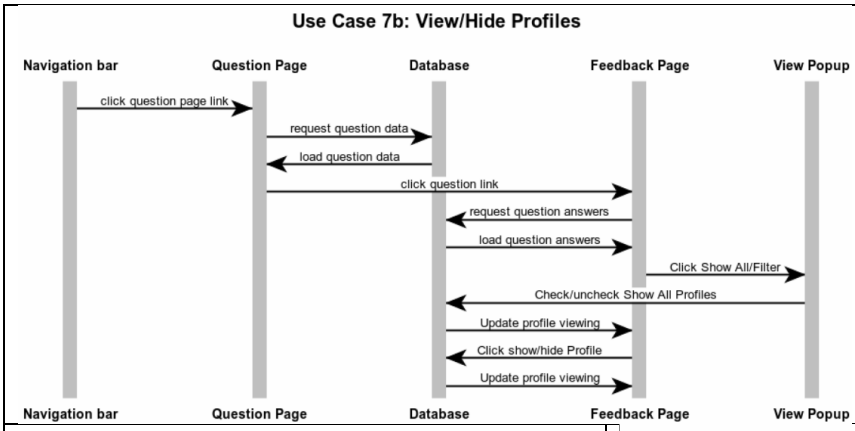
Click Questions link	

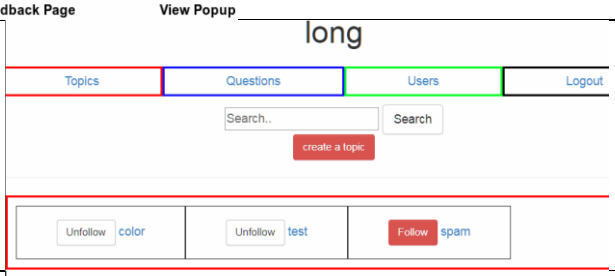
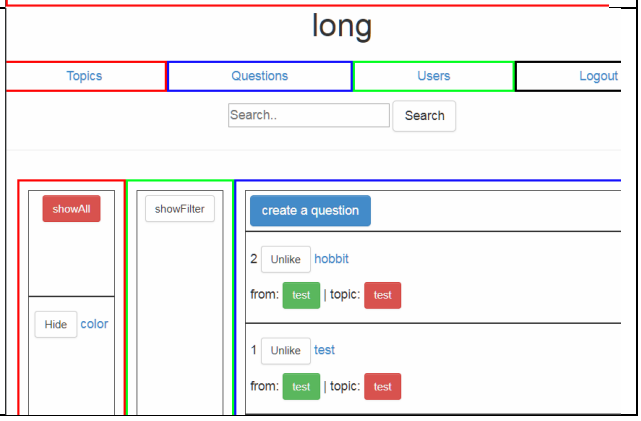
Request Question Data Load Question Data Click showFilter	
Uncheck show all topics click update	
Show Update topic viewing Click show/hide topic • Click show test	
Show Update topic viewing	

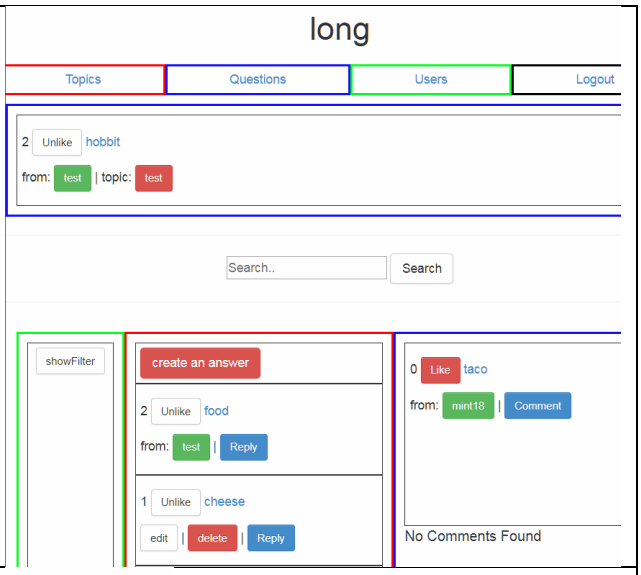
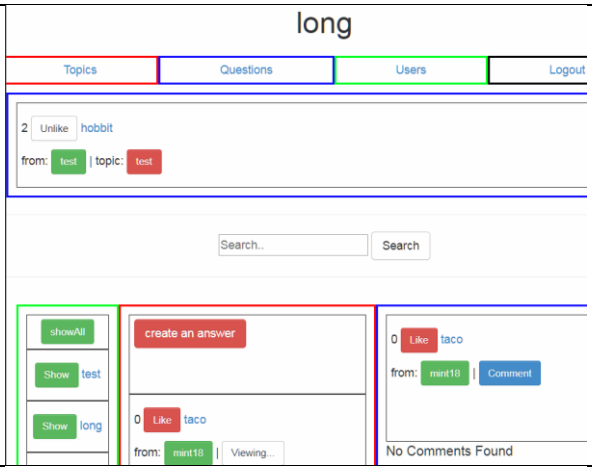
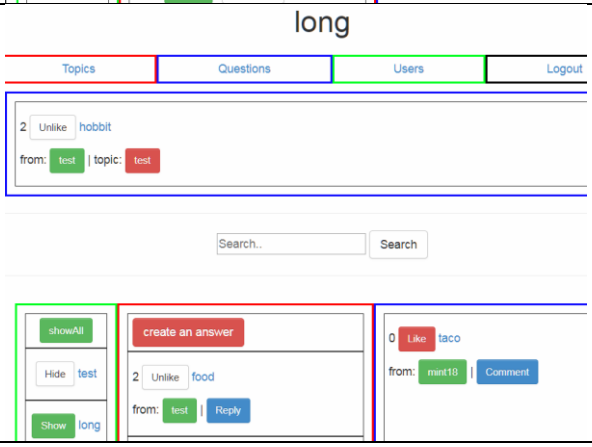


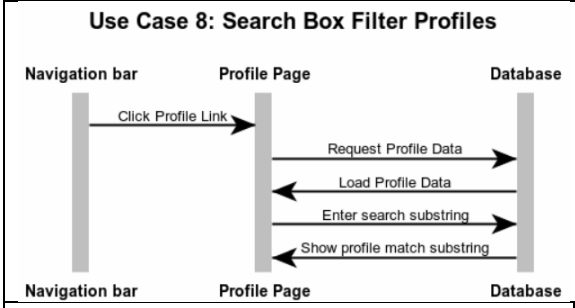
Click Question link	
Request Question Data Load Question Data Click showFilter	
Uncheck show all users Click update	
Show Update profile viewing Click show/hide profile • Click show test	

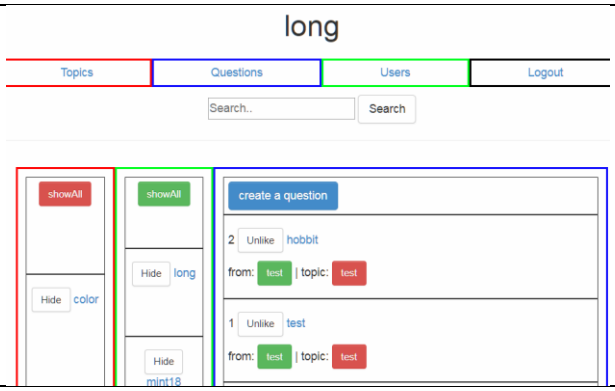
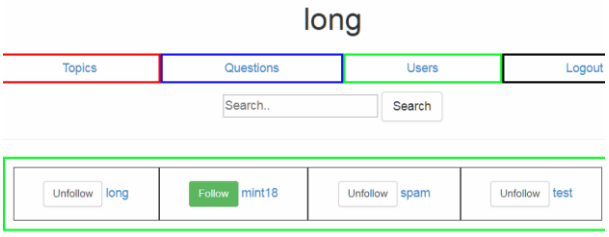
Show Update profile viewing	

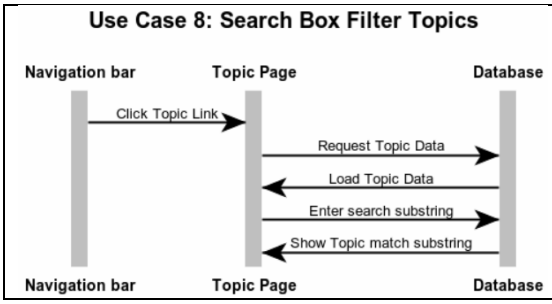


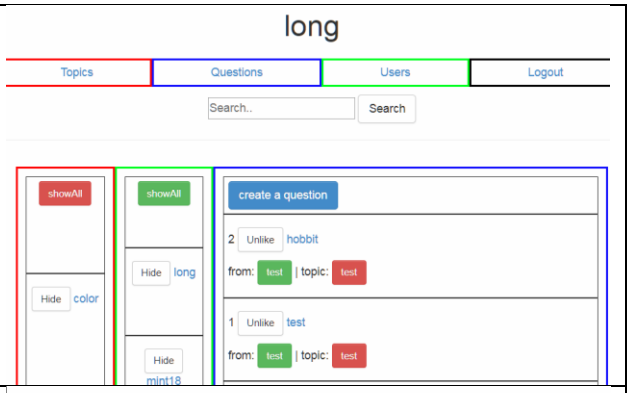
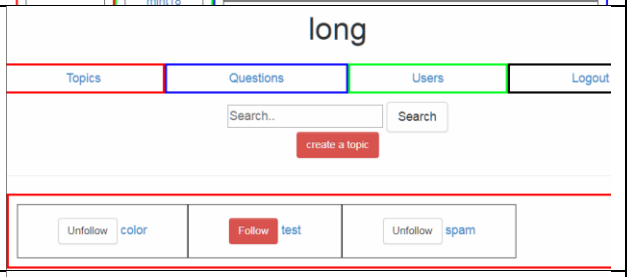
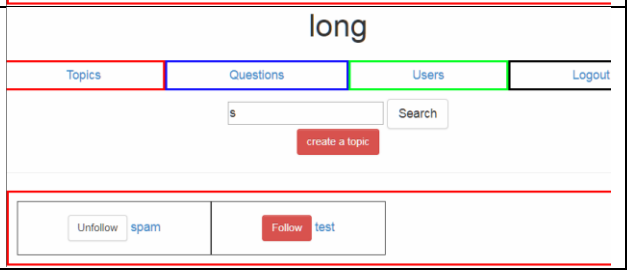
Click Question page link	
Request Question Data Load Question Data Click question link • Click hobbit	

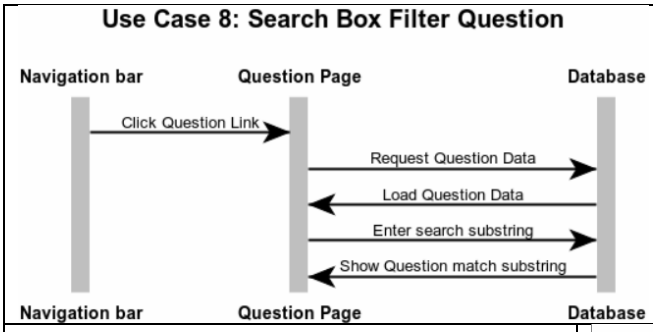
request question answers load question answers click showFilter	
Uncheck show all users Click update	
Update profile viewing Click show/hide profile • Click show test	
Update profile viewing	

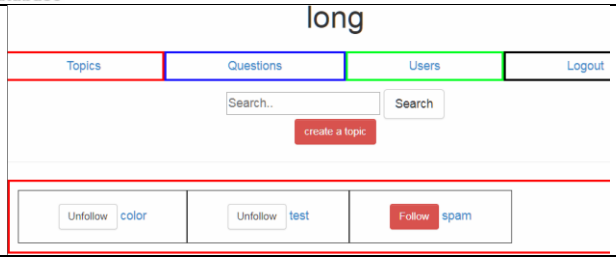


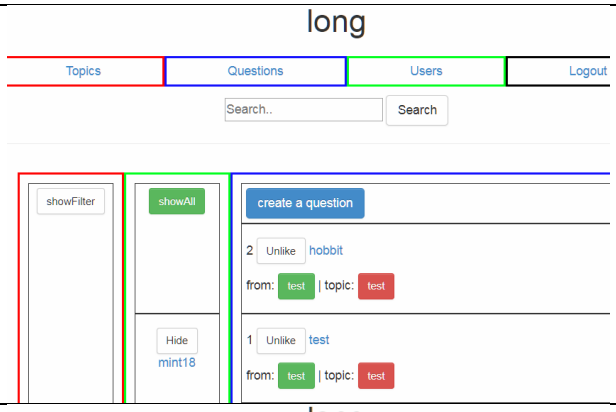
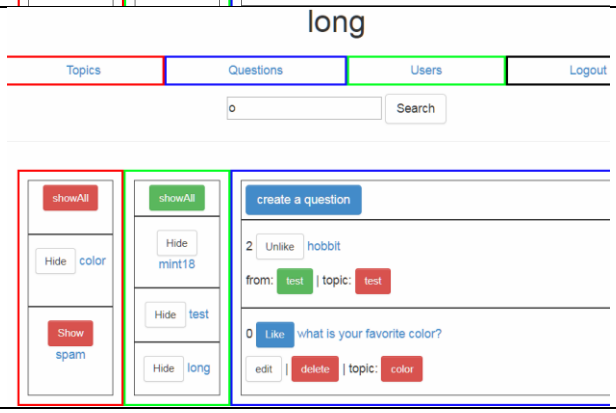
Click Profile(users) link	
Request profile Data Load Profile Data Enter search substring • n	
Show profile match substring	

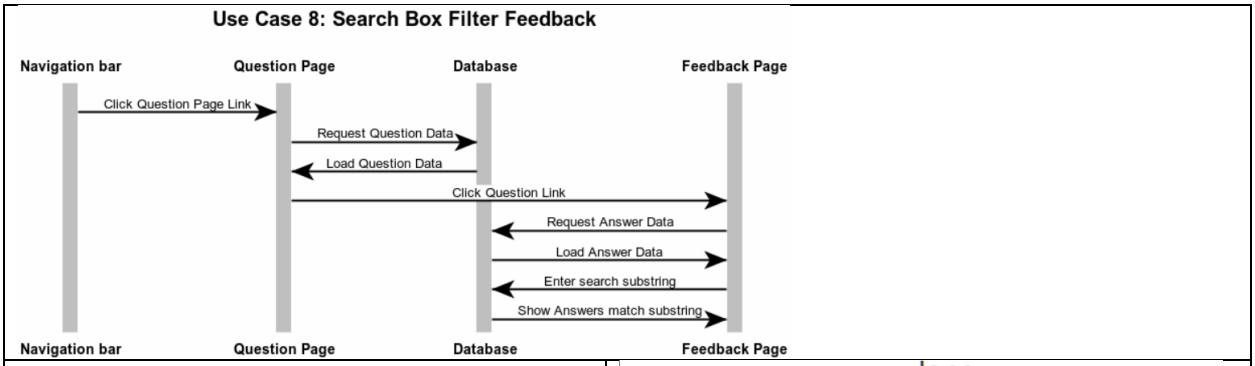


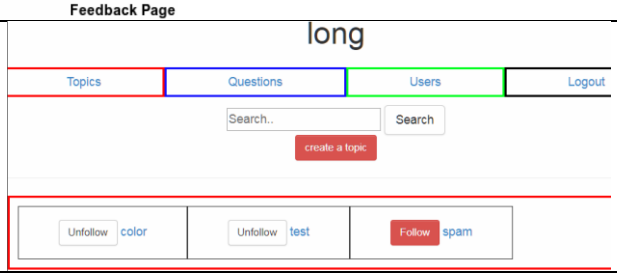
Click Topics link	
Request topic Data Load topic Data Enter search substring • s	
Show Topic match substring	

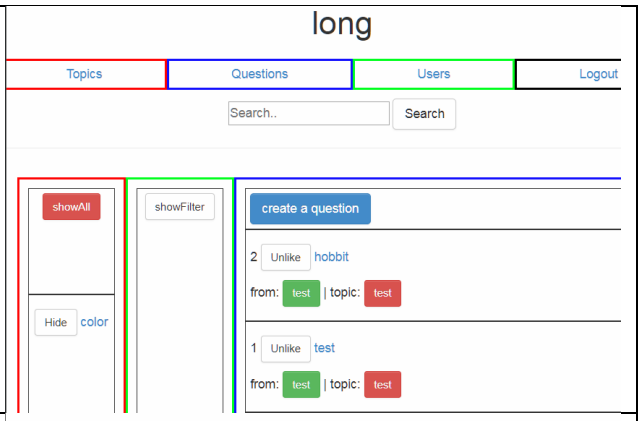
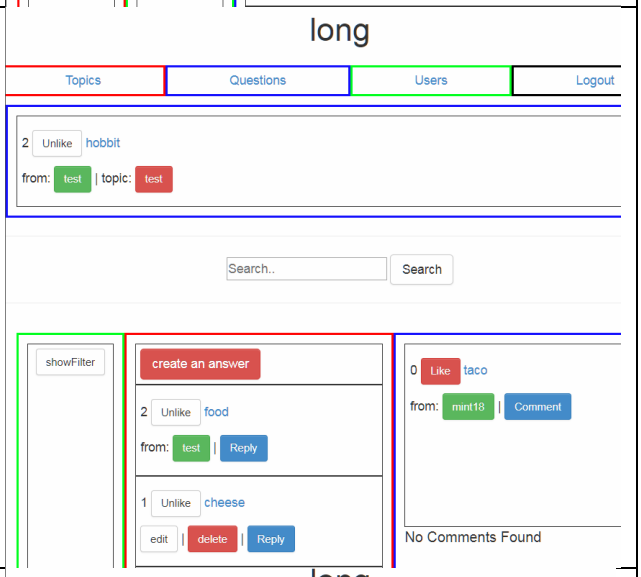
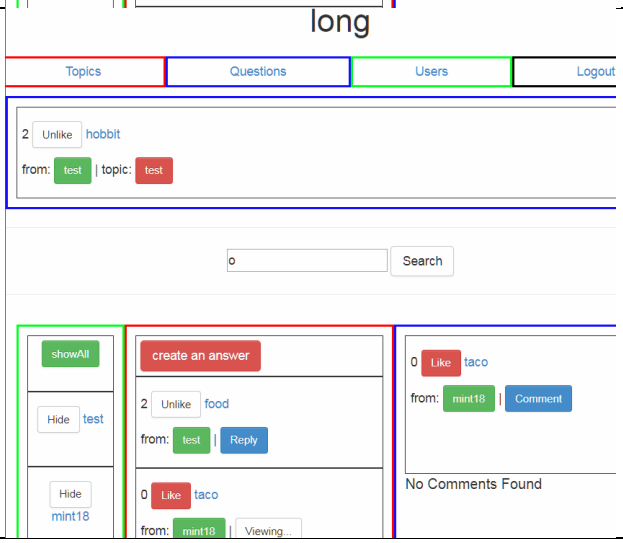


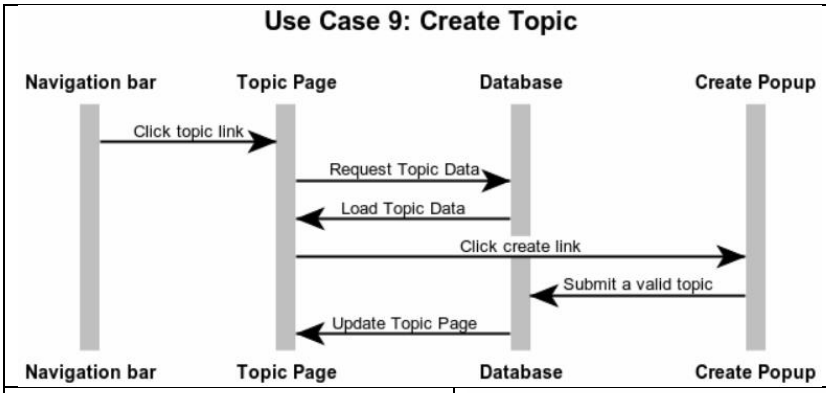
Click Questions link	

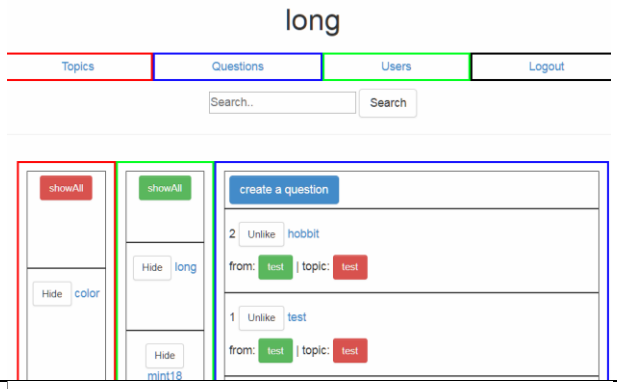
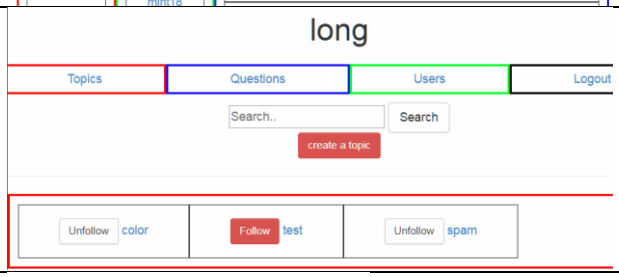
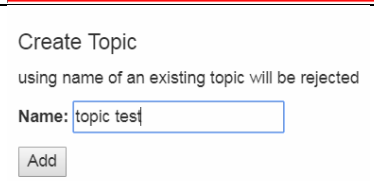
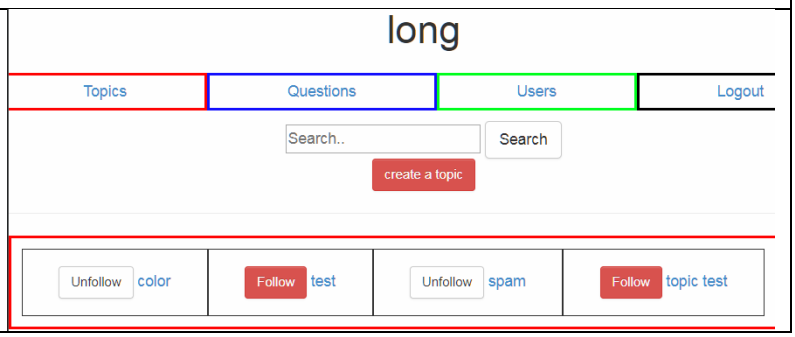
Request Question Data Load Question Data Enter search substring • o	
Show question match substring	

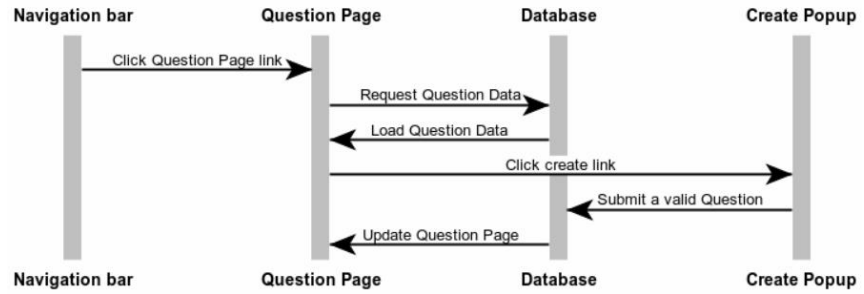
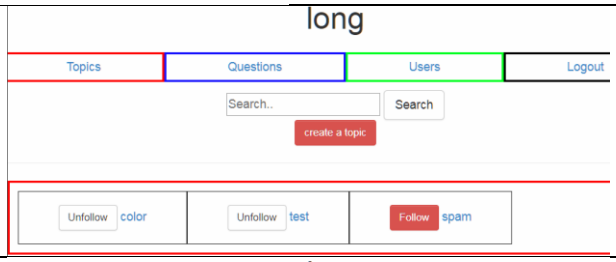
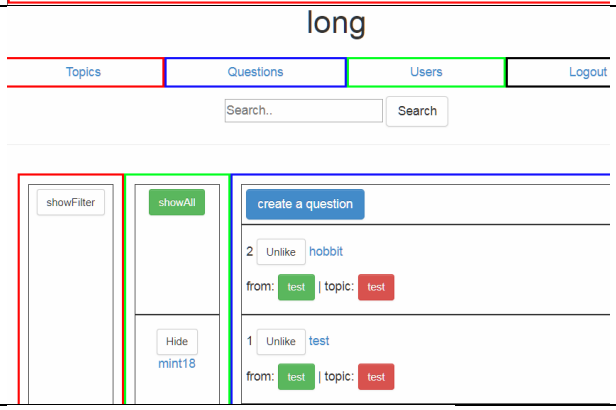
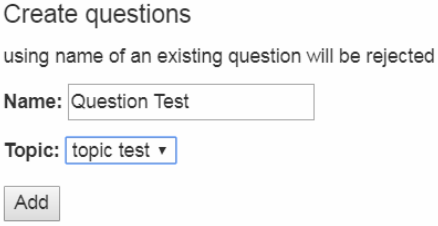


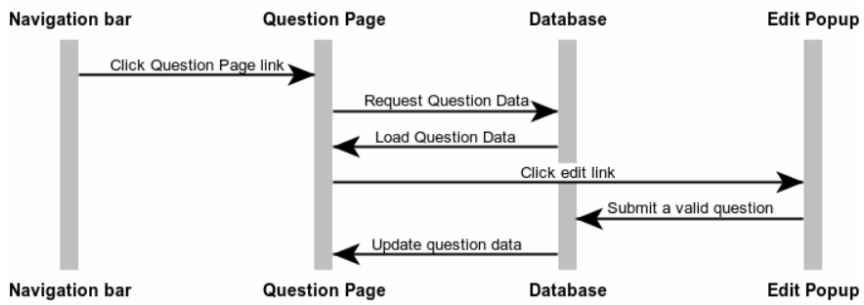
Click Question page link	

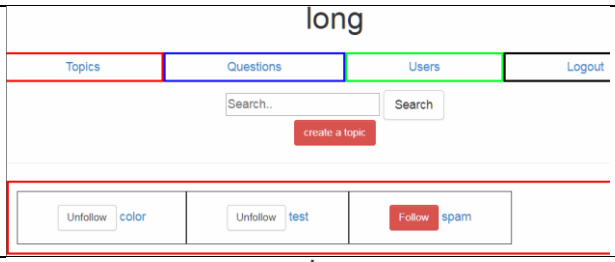
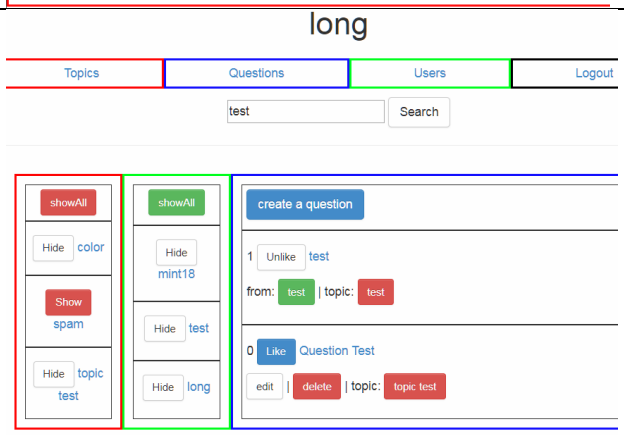
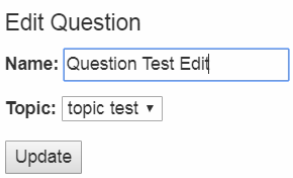
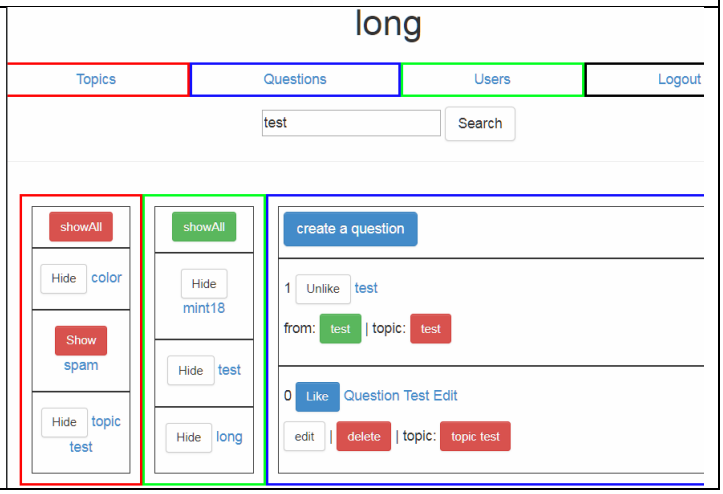
Request Question Data Load Question Data Click question link • Click hobbit	
request answer data load answer data Enter search substring • o	
Show answers match substring	

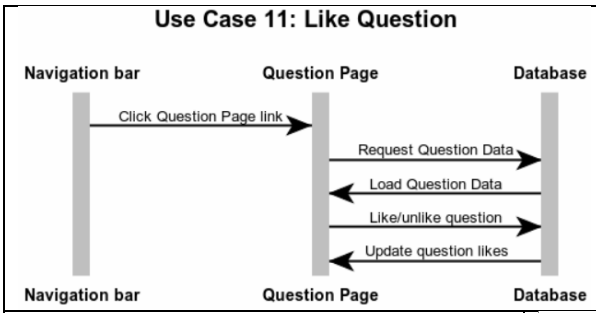


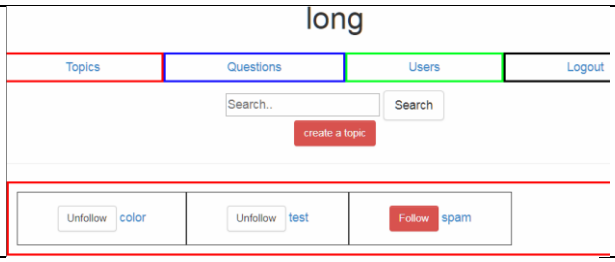
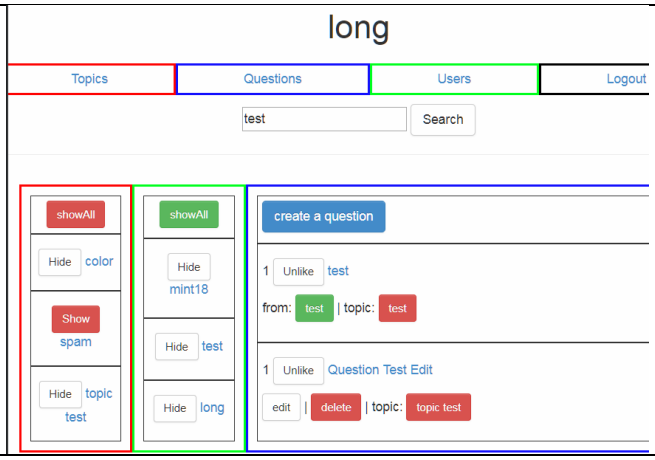
Click Topics link	
Request topic Data Load topic Data Click create a topic	
Enter topic test Click add	
Show Updated topic page	

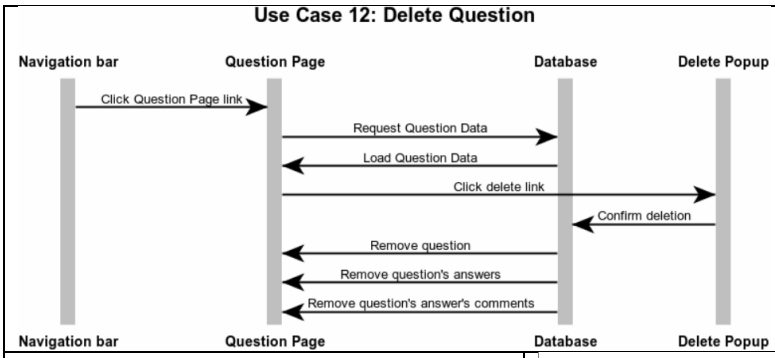
<table>
<tr><td colspan="2">Use Case 9: Create Question
 
</td></tr>
<tr><td>Click Questions Page link</td><td> 
</td></tr>
<tr><td>Request Question Data
Load Question Data
Click create a question</td><td> 
</td></tr>
<tr><td>Enter Question Test
Select Topic Test
Click add</td><td> 
</td></tr>
<tr><td>Show Updated Question Page</td><td></td></tr>
</table>

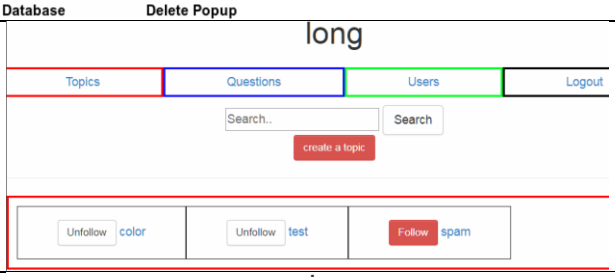
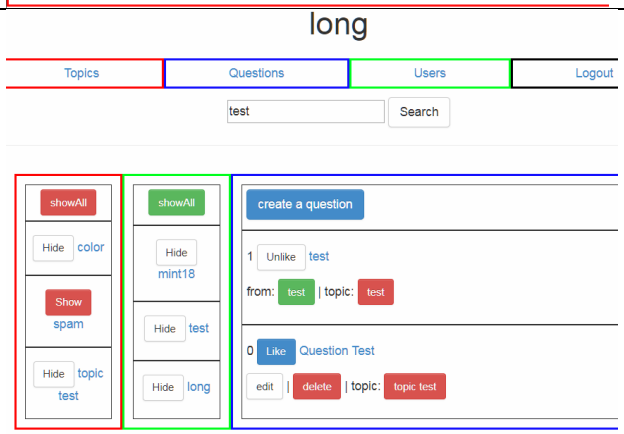
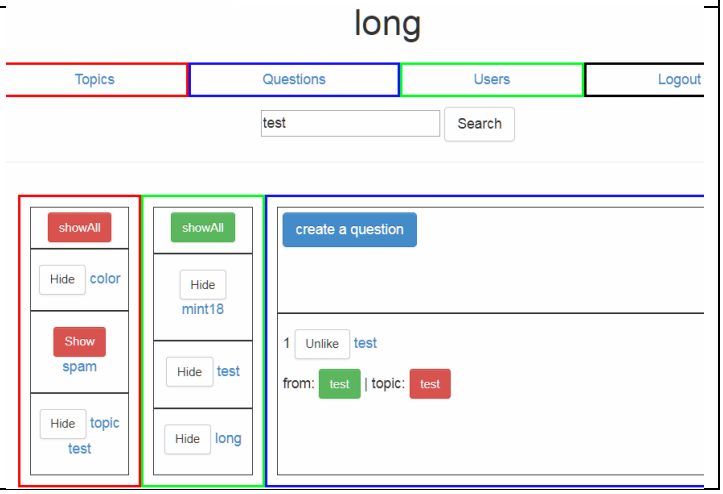
<table>
<tr><td>Use Case 10: Edit Question
 
</td></tr>
</table>

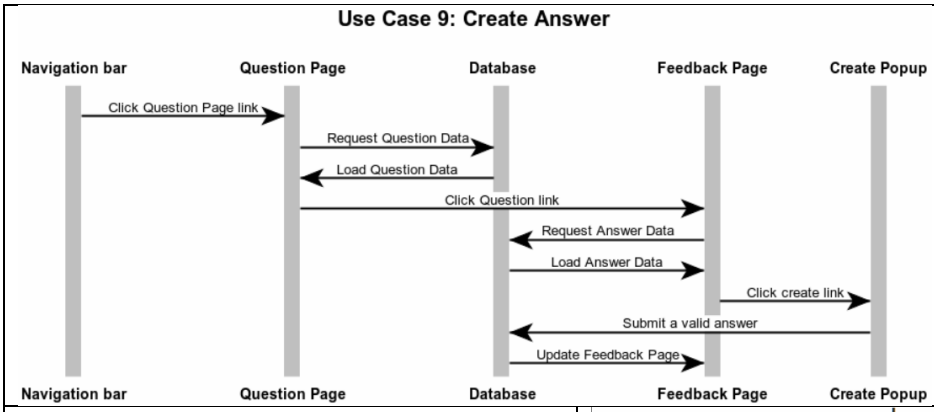
Click Questions Page link	
Request Question Data Load Question Data Click edit Question Test	
Enter Question Test Edit Select Topic Test Click add	
Show Updated Question Page	

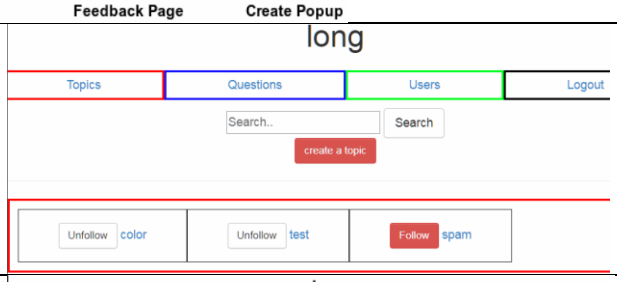
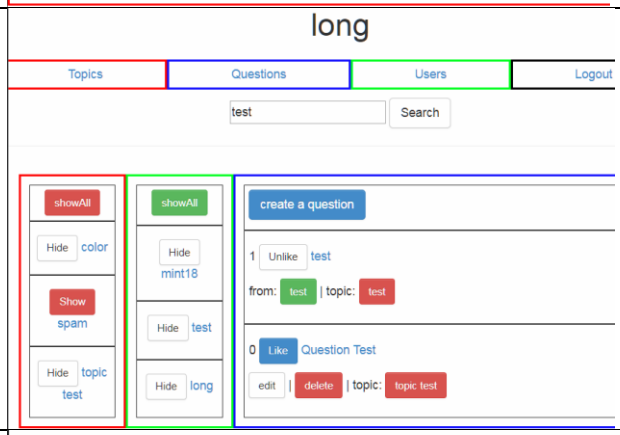
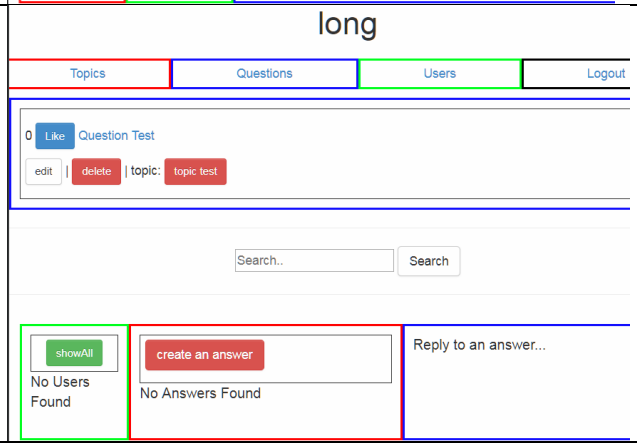


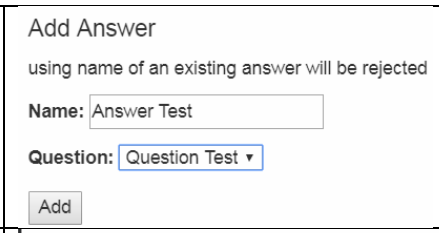
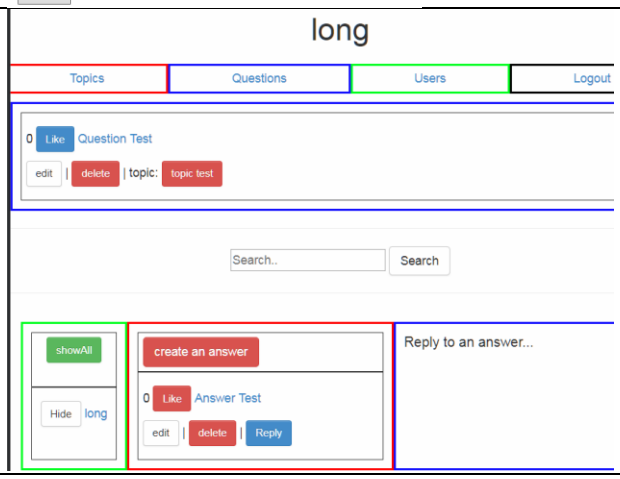
Click Questions Page link	
Request Question Data Load Question Data Click like Question Test Edit	
Show update question Test edit likes	

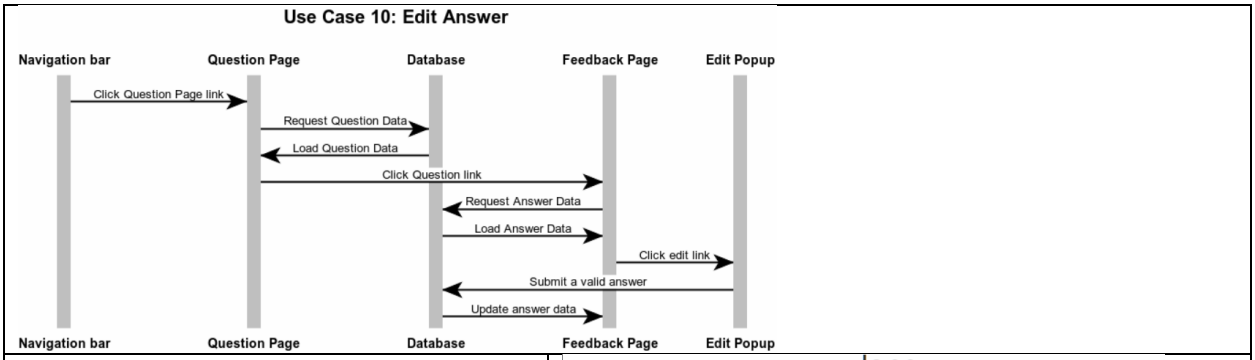


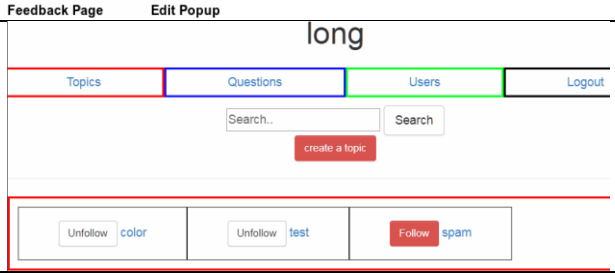
Click Questions Page link	
Request Question Data Load Question Data Click delete Question Test	
Click delete	
Show deleted Question Test Also removes associated answers and comments	

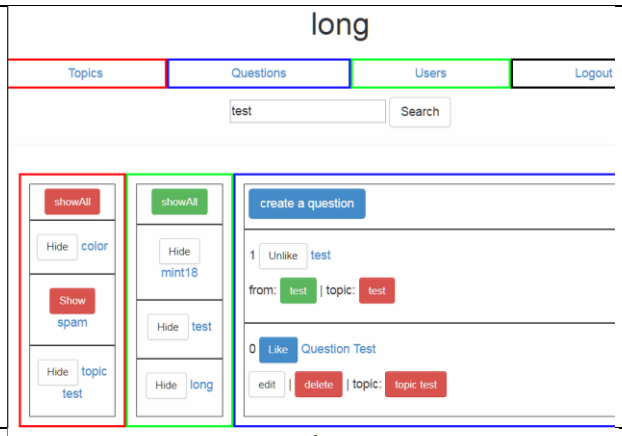
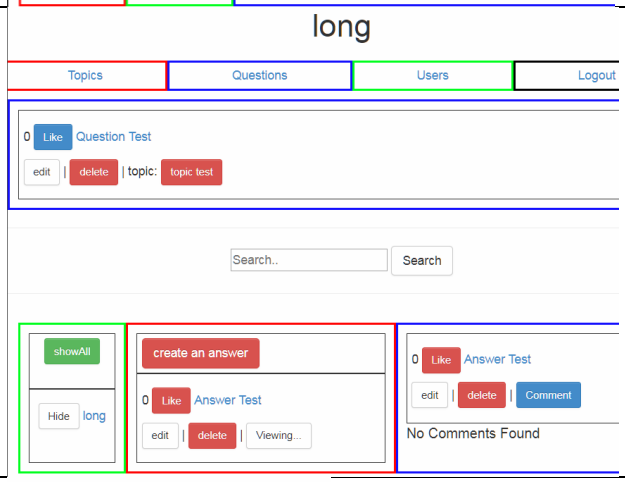
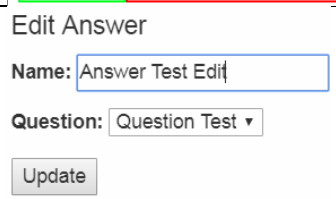
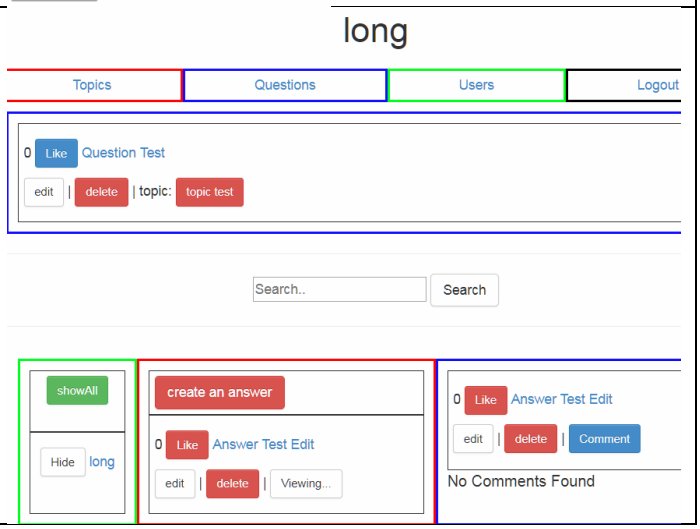


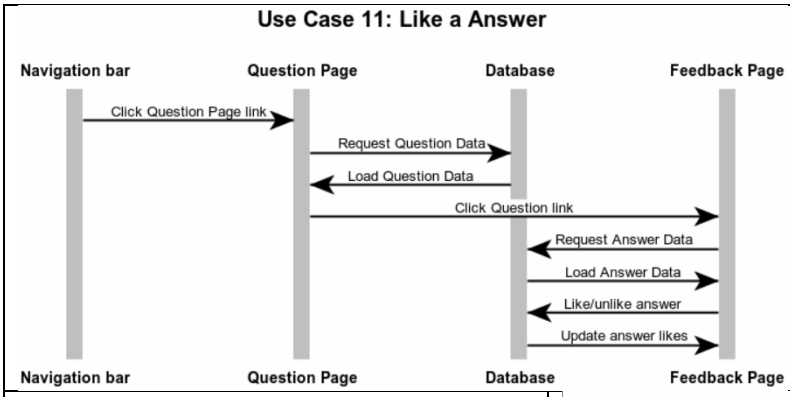
Click Questions Page link	
Request Question Data Load Question Data Click Question link • Question Test	
Request Answer Data Load Answer Data Click create an answer	

Enter Answer Test Select Question Test Click add	
Show Updated Feedback Page	

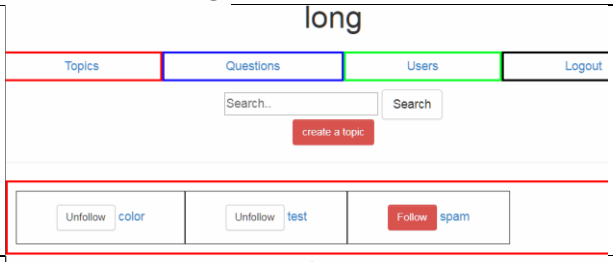


Click Questions Page link	

Request Question Data Load Question Data Click Question link • Question Test	
Request Answer Data Load Answer Data Click edit answer test	
Enter Answer Test Edit Select Question Test Click add	
Show Updated Feedback Page	

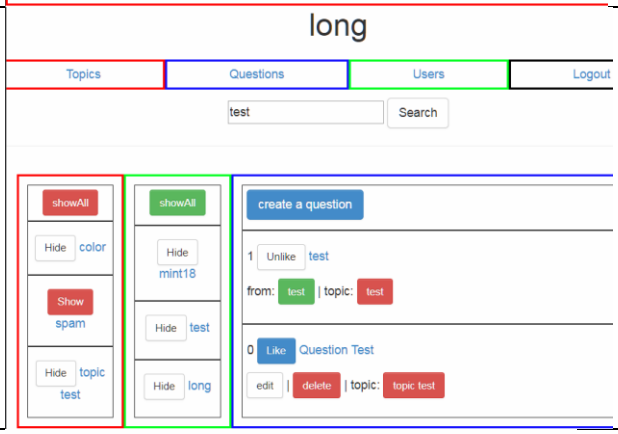


Click Questions Page link

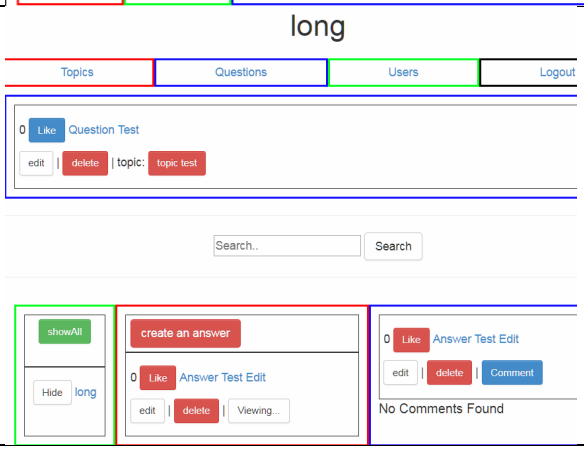


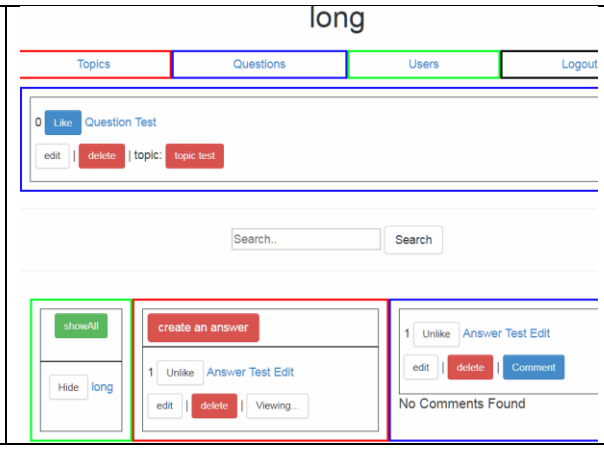
Request Question Data
Load Question Data
Click Question link

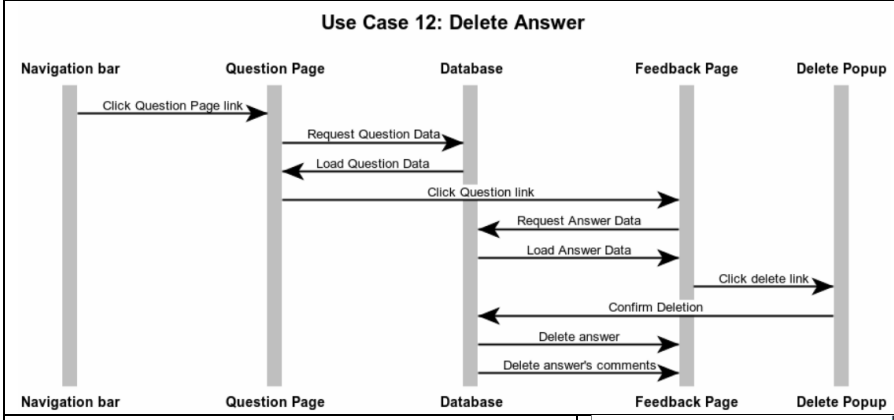
- Question Test

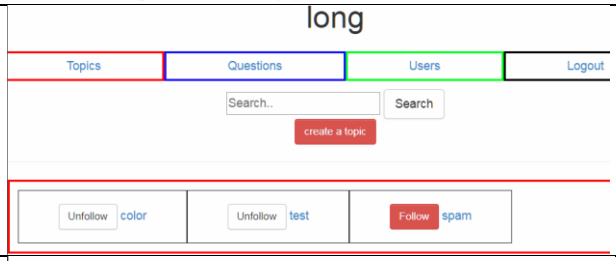
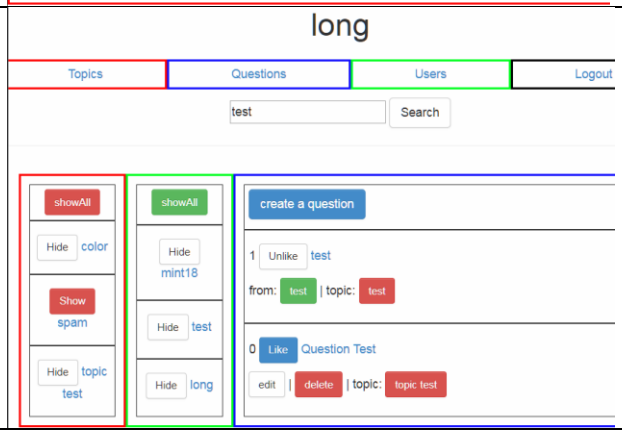


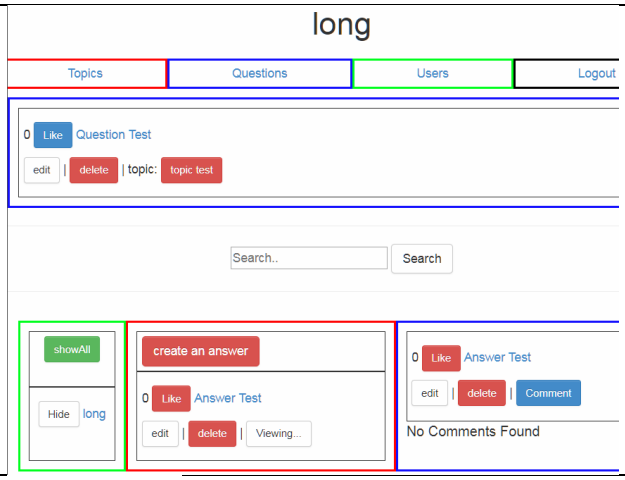
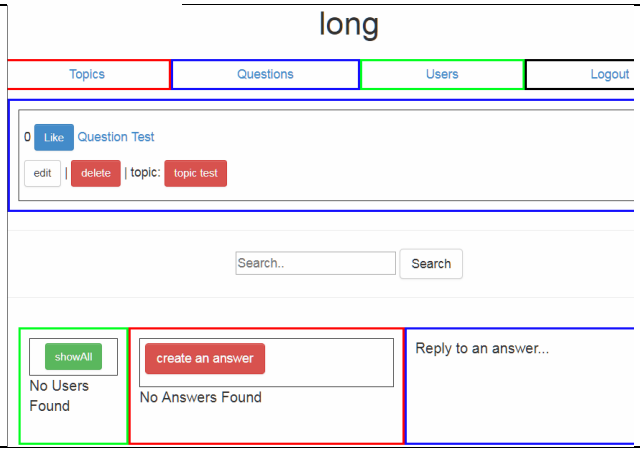
Request Answer Data
Load Answer Data
Click like answer test edit

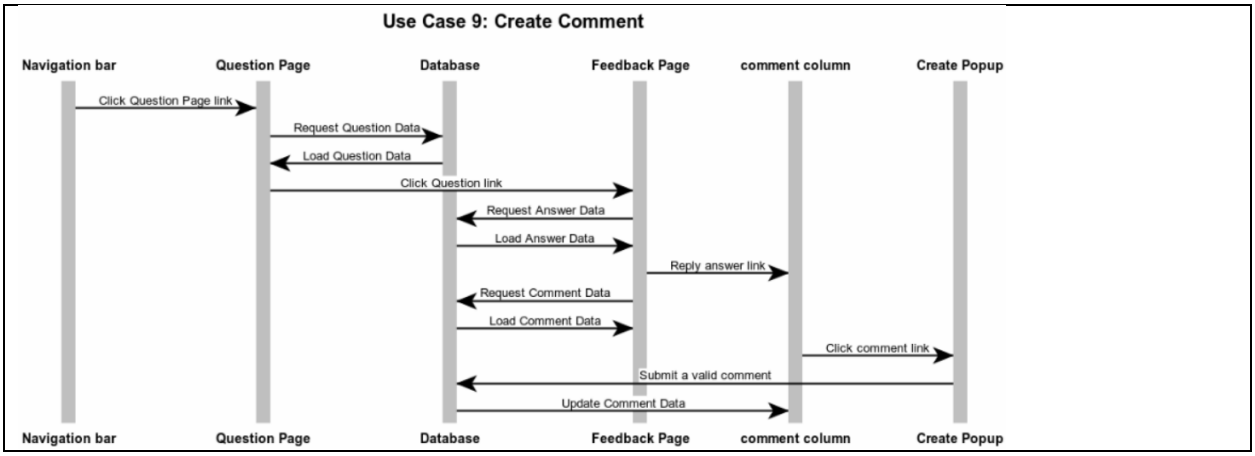


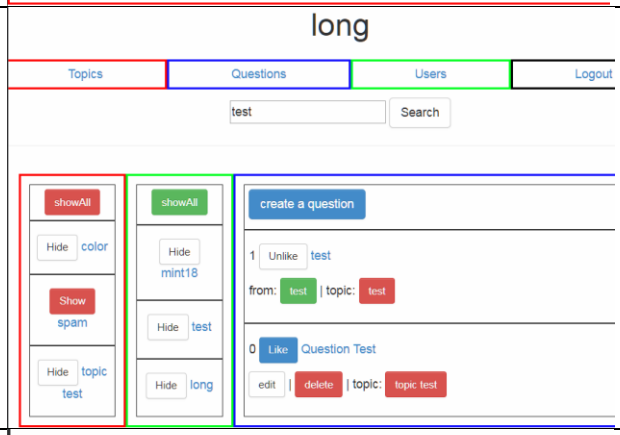
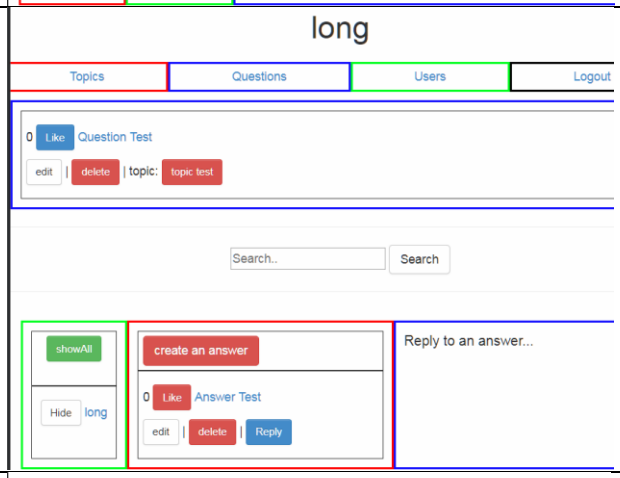
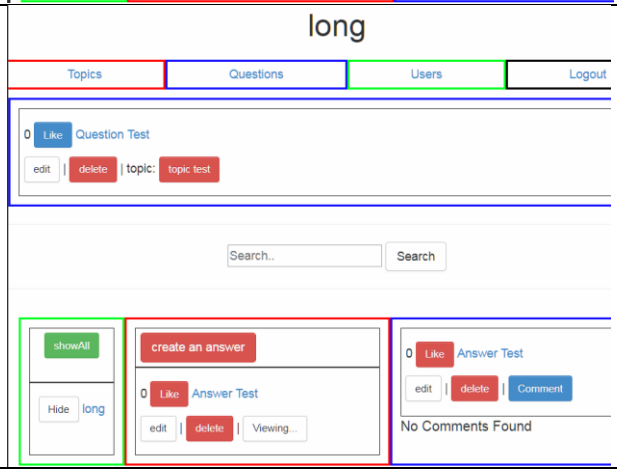
Show Updated answer test edit likes	

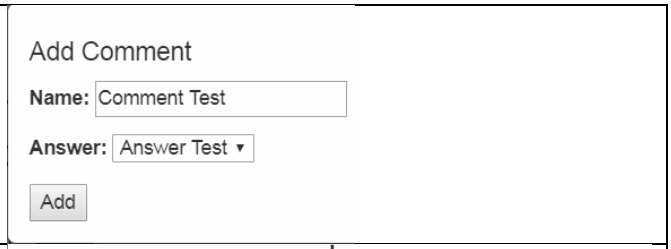
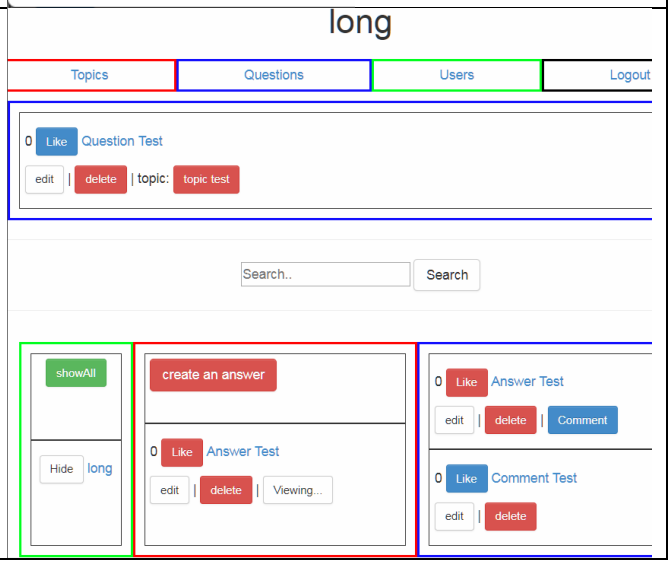


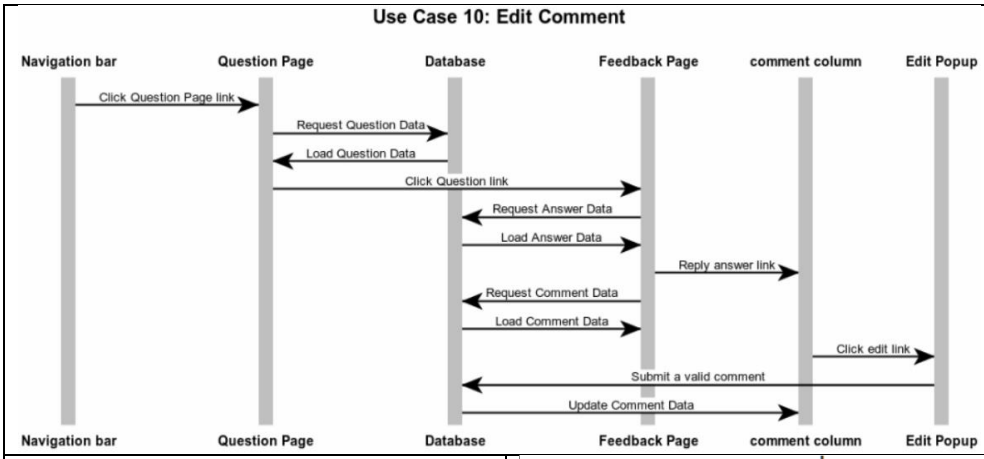
Click Questions Page link	
Request Question Data Load Question Data Click Question link • Question Test	

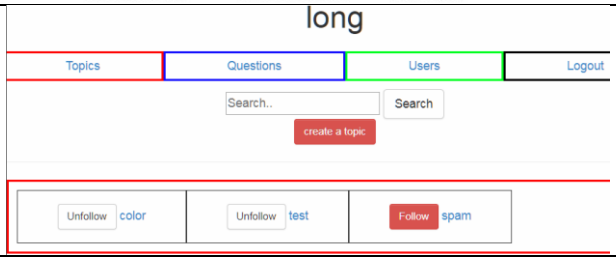
Request Answer Data Load Answer Data Click delete answer test	
Click delete	
Show updated deleted answer test Note: when you delete an answer you also delete comments that belong to that answer	

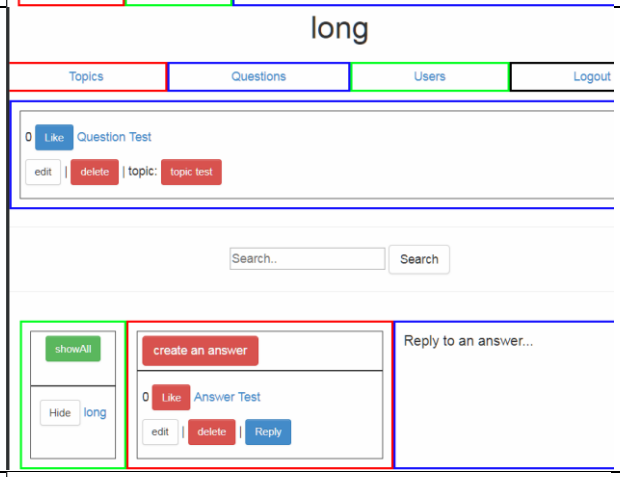
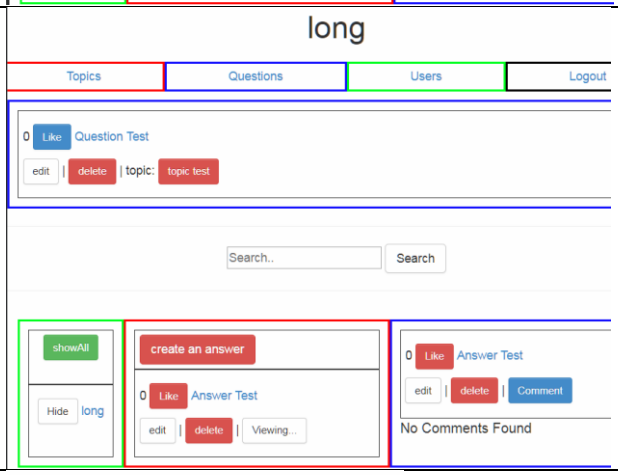
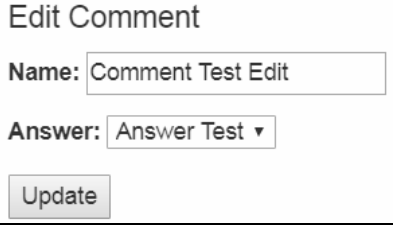


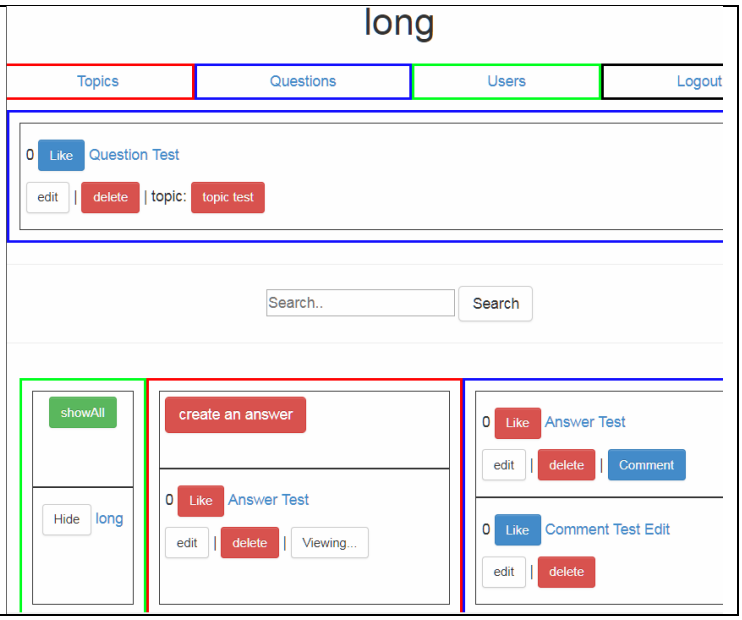
Click Questions Page link	
Request Question Data Load Question Data Click Question link • Question Test	
Request Answer Data Load Answer Data Click reply to answer Test	
Show comment column Click comment	

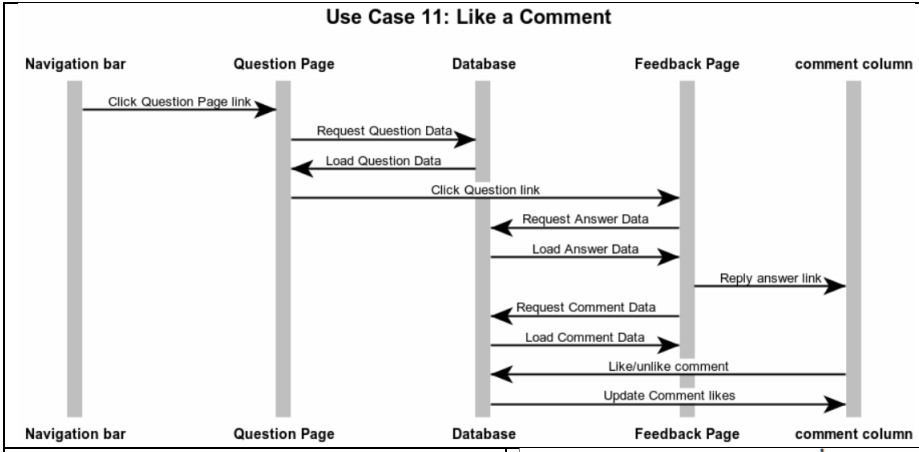
Enter comment test Select Answer Test Click add	
Show Updated Comment Data	

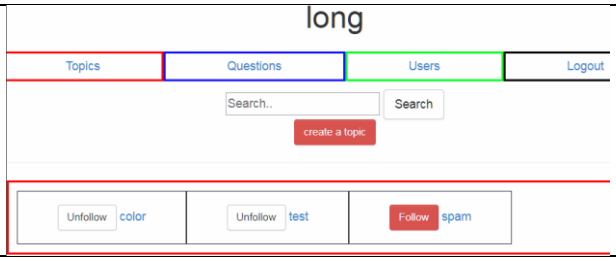


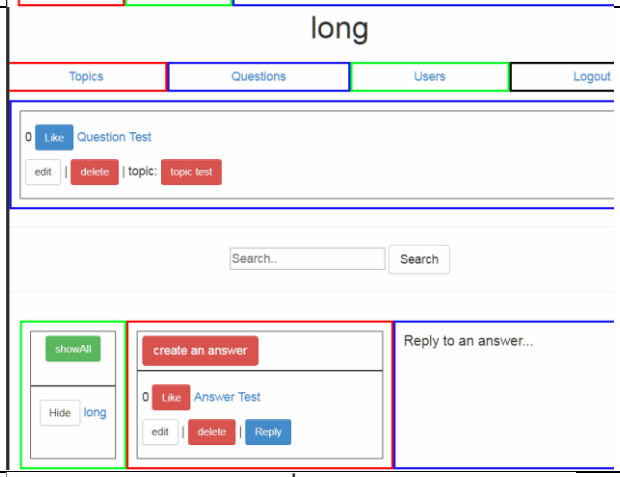
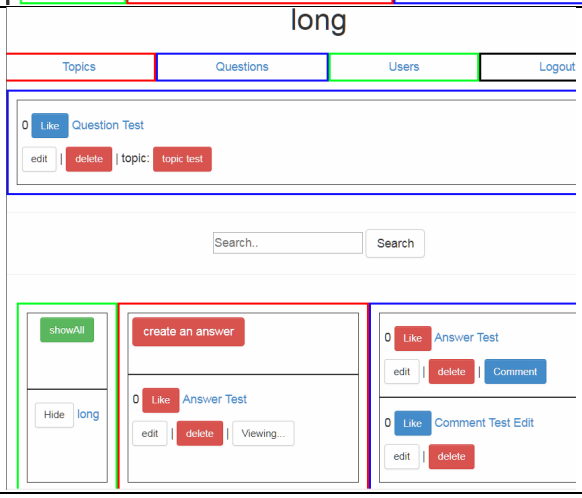
Click Questions Page link	

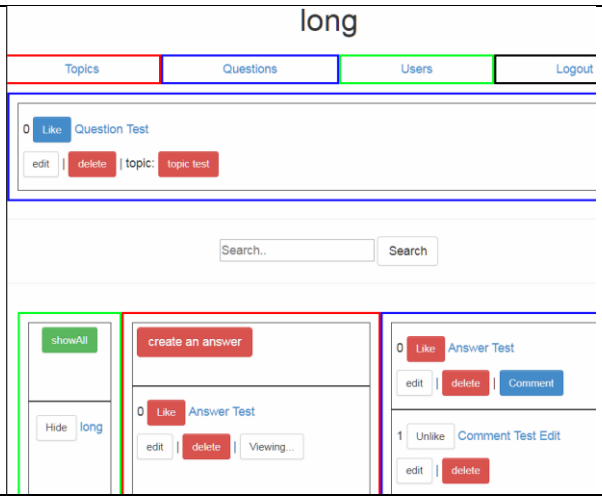
Request Question Data Load Question Data Click Question link • Question Test	
Request Answer Data Load Answer Data Click reply to answer Test	
Show comment column Click Edit	
Enter Comment Test Edit Select Answer Test Click add	

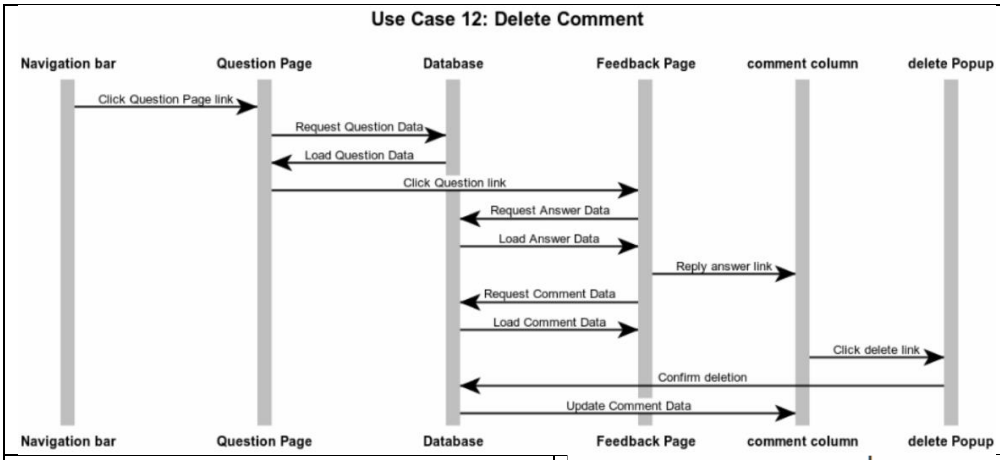
Show Updated Comment Data	

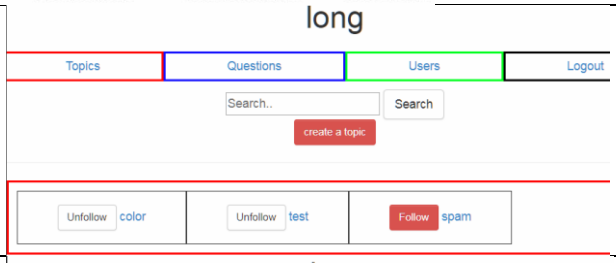
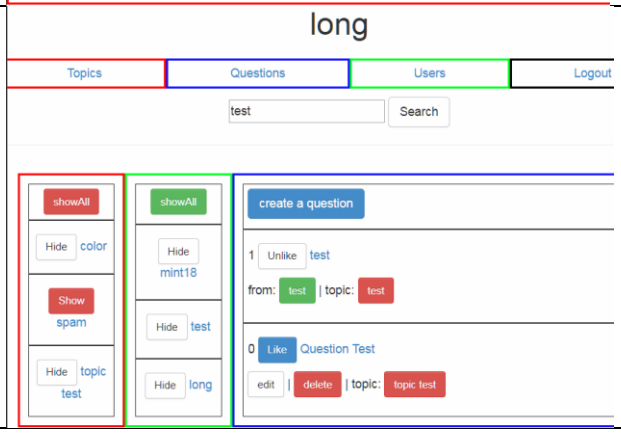


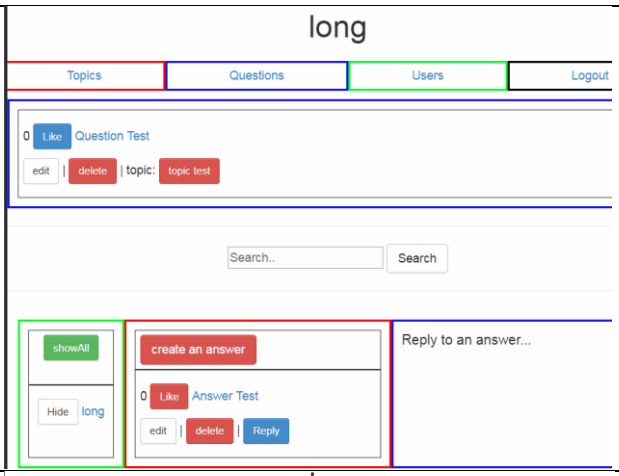
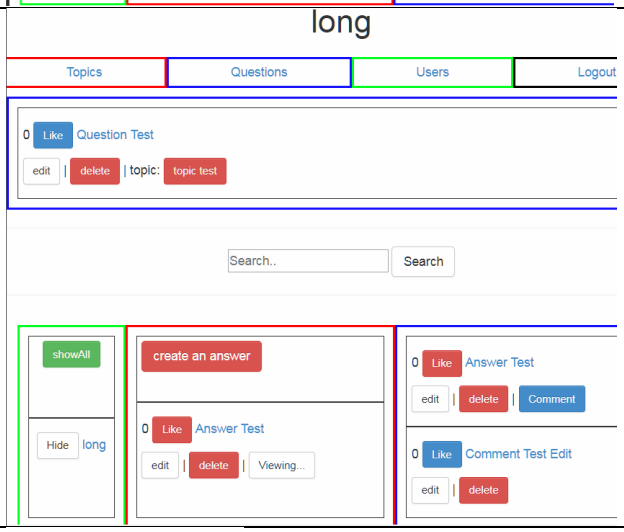
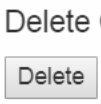
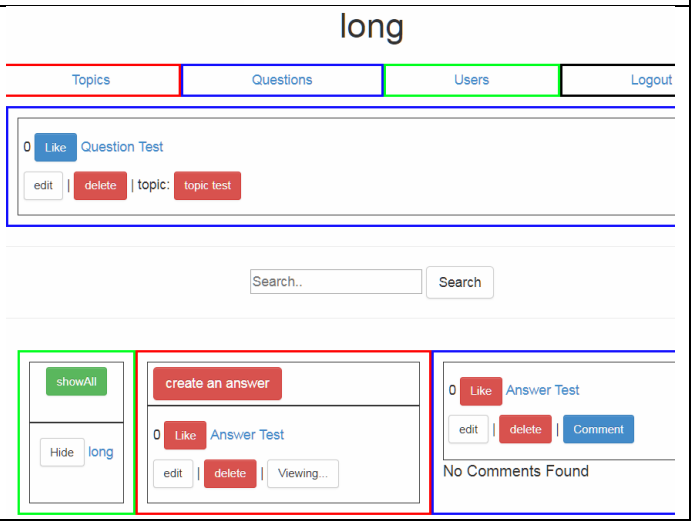
Click Questions Page link	

Request Question Data Load Question Data Click Question link • Question Test	
Request Answer Data Load Answer Data Click reply to answer Test	
Show comment column Click Like Comment Test Edit	

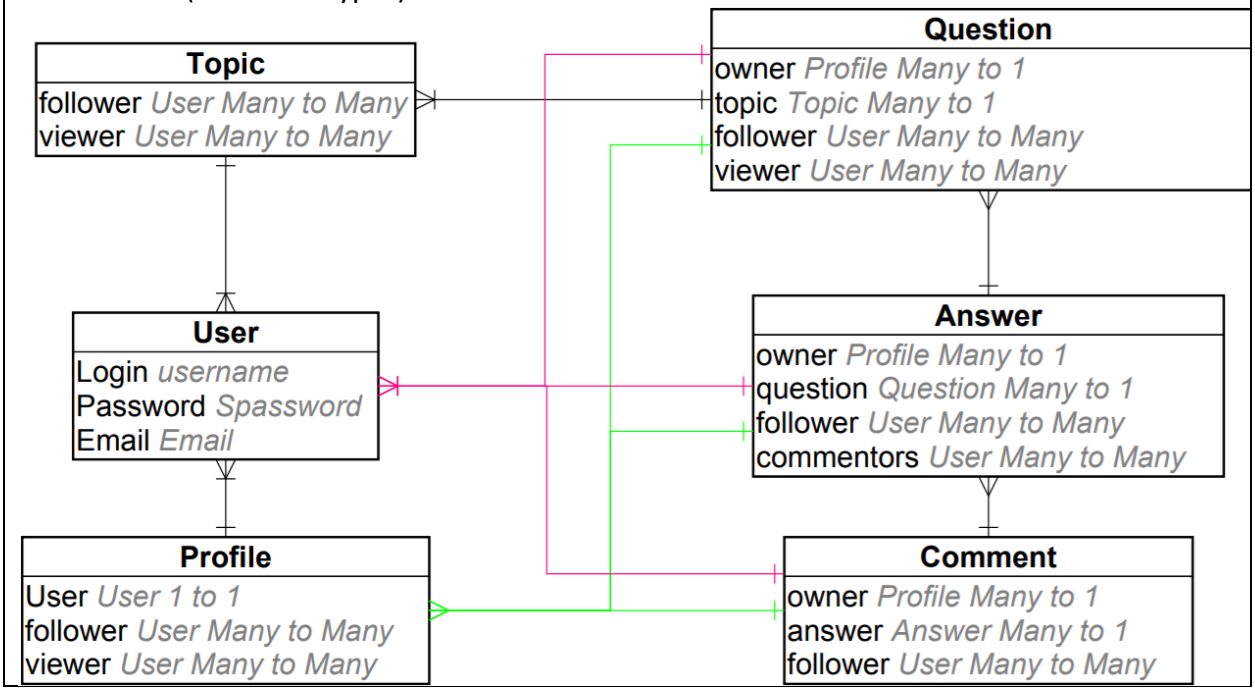
Show Updated Comment Likes	



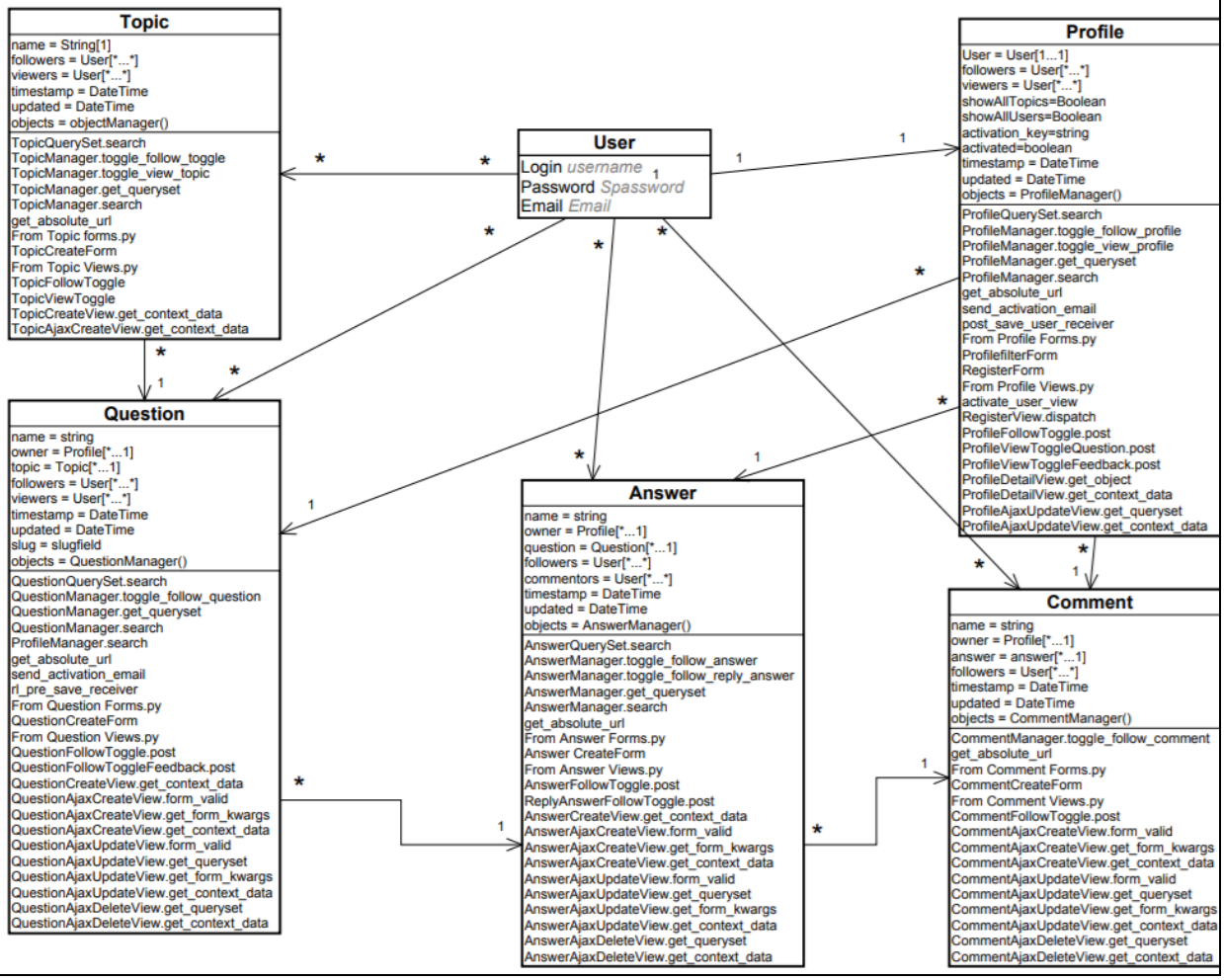
Click Questions Page link	
Request Question Data Load Question Data Click Question link • Question Test	

Request Answer Data Load Answer Data Click reply to answer Test	
Show comment column Click Delete Comment Test edit	
Click Delete	
Show Deleted Comment	

Complete ER diagram for the DB design such as adding cardinalities for the relations, and add table details (fields and types).



Complete the class diagram, and add class details, including fields and methods/functions.



Conclusion

- Project works, but is not good enough for public use…

Future work

- Refactor project using react.js to improve ajax performance
- Be able to add pictures to questions and answers
- Be able to add, edit, delete post within the field instead of a popup

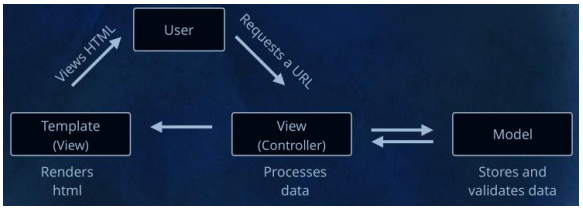
Appendix of source codes

- Web development has too many modules
- listing them out sequential will make it difficult to trace modules and logic
- I will instead include my readme that explains the source code as a substitute.

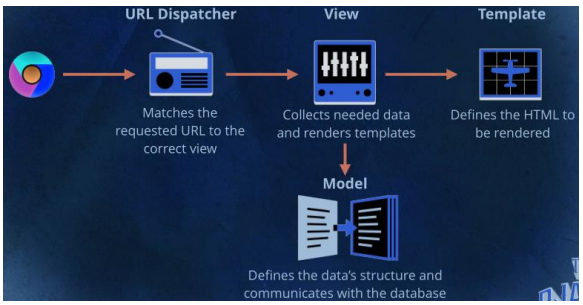
This readme explains how Django web frameworks and an example implementation.

MVC, model view controller is a software design pattern. The model is a database and defines the class objects. The view is the page layout and items, what the user sees. The controller has functions that take in input from users and manipulates the model and view.

I had to learn Django web framework. Django name their controller “view”, which mislead people thinking the view is suppose to be a view, but it’s actually the controller. The view is named template, and the model is named model.

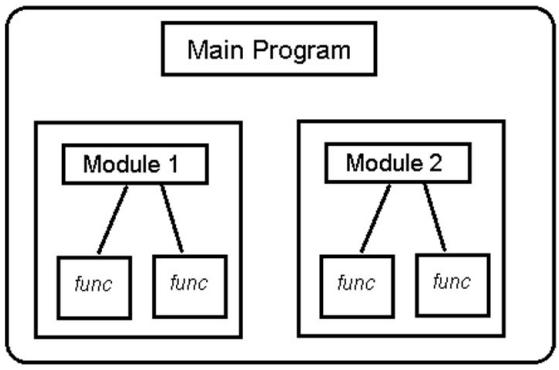


When a user inserts inputs or makes a request a URL, it goes to the url dispatcher to find the view that matches the url, then the view manipulates the model, add, delete, edit data and then sends a response URL to the dispatcher which is sent to the view then is rendered by template.

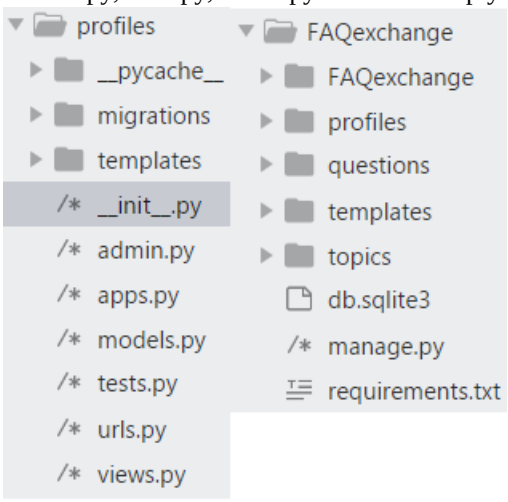


A Website can have different types of pages and features. It is easier to separate the different features than to have all the different features is one file. Django project supports different apps. We make an app for every different feature of the project.

This is called modularity. Which allows us to call the module in the main program, instead of writing the same code multiple times.

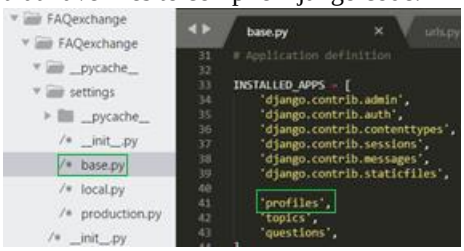


When I make a new app.
I got to the command line.
I go to the root directory of my Django project
I run command: python manage.py startapp appname
This creates the app folder in the root directory.
For example, in my root, I have 3 apps, profiles, questions, topics.
The profile is the app I just created.
by default it comes with __init__.py (we need this here so python compiler can ignore resolving names that may cause compilation errors) https://stackoverflow.com/questions/448271/what-is-init-py-for it also comes with admin.py, apps.py, models.py, tests.py, views.py which are empty.

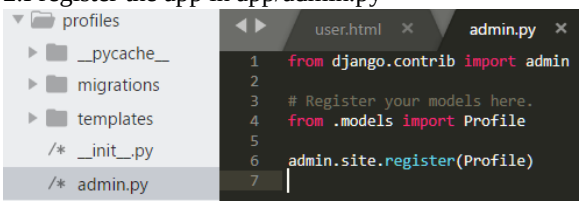


1. I add the app to my FAQexchange/settings/base.py
note: __pycache__ is auto-generated by Django to

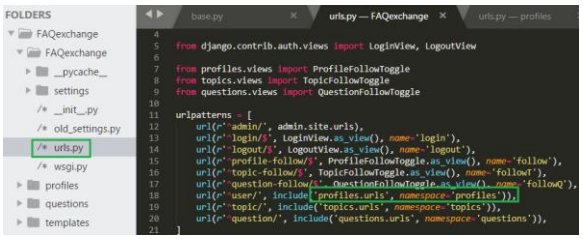
that have files to compile Django code.



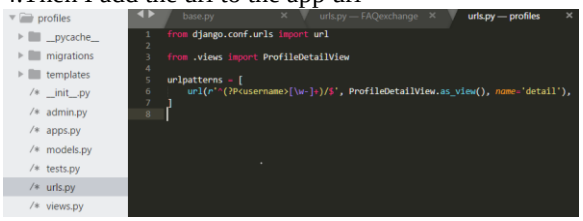
2.I register the app in app/admin.py



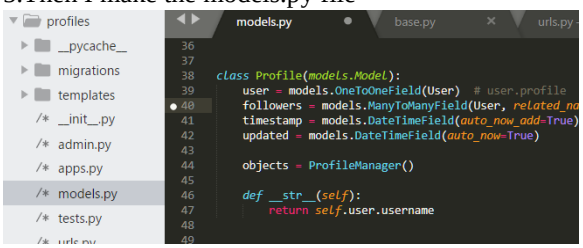
3.Next I add the urls to the root url



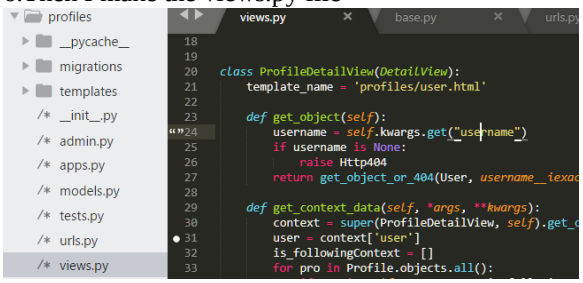
4.Then I add the url to the app url



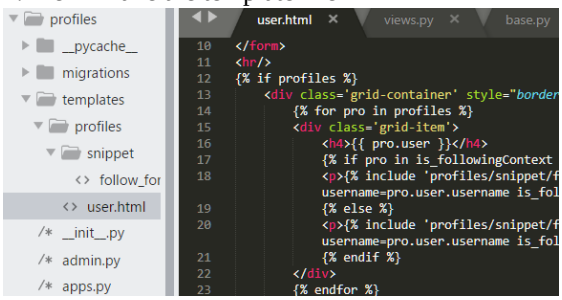
5.Then I make the models.py file



6.Then I make the views.py file



7.Then I make the template file

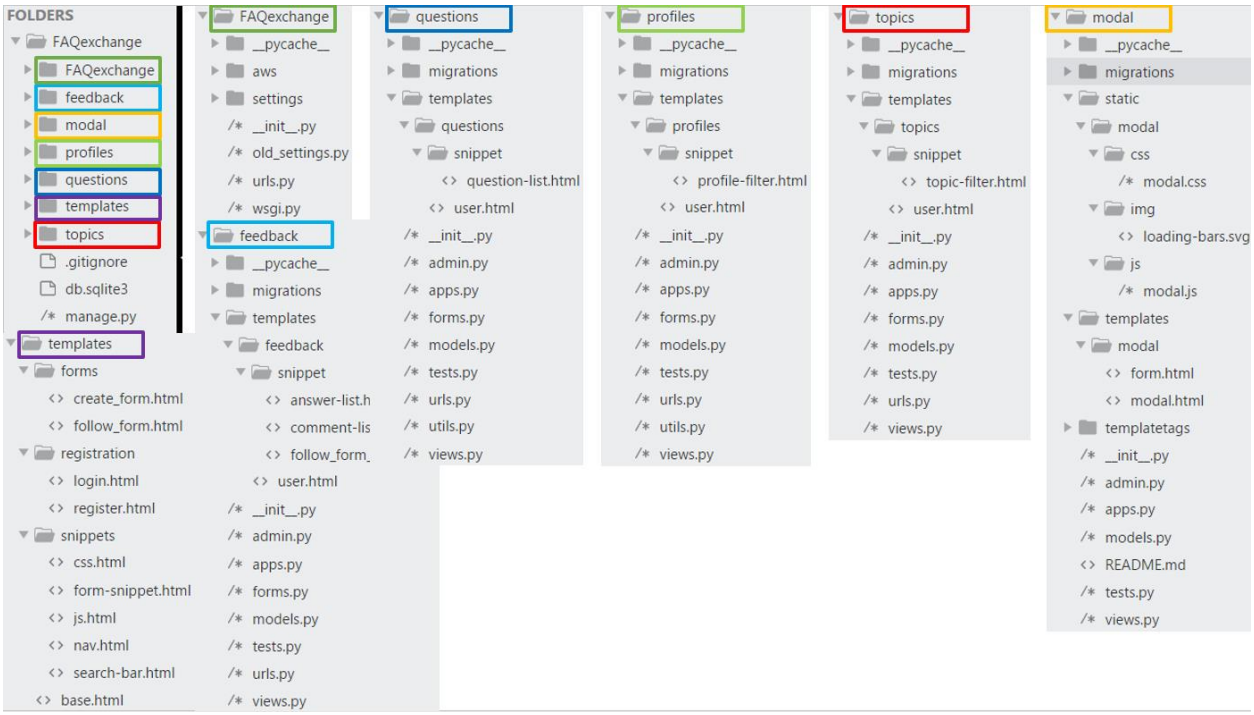


8.Then I go back to Django project directory update the database with new model via:
python manage.py makemigrations
python manage.py migrate

My project has a total of 5 apps

- Profiles
- Topics
- Questions
- Feedback
- Modal (API for djangoAJAX)

All 5 apps have the same django MVC data flow as the profile app example.



This is the full directory tree of my project and web apps.