

Requirements Document

The Borne Collective: NKUNet

[Change Discussion](#)

[Iteration 2](#)

[Use Case Model](#)

[Context Diagram](#)

[Use Cases](#)

[UC1: View Class](#)

[UC2: Ask Question](#)

[UC3: Answer Question](#)

[UC4: View Thread](#)

[UC5: Manage Thread](#)

[UC6: Manage Users](#)

[UC7: Manage Schedule](#)

[UC8: Authenticate User](#)

[UC9: Vote](#)

[UC10: Manage Class](#)

[UC11: View User](#)

[Domain Model](#)

[System Sequence Diagrams](#)

[View Class](#)

[Ask Question](#)

[Answer Question](#)

[View Thread](#)

[Manage Thread](#)

[Manage Users](#)

[Manage Schedule](#)

[Authenticate User](#)

[Vote](#)

[Manage Class](#)

[View User](#)

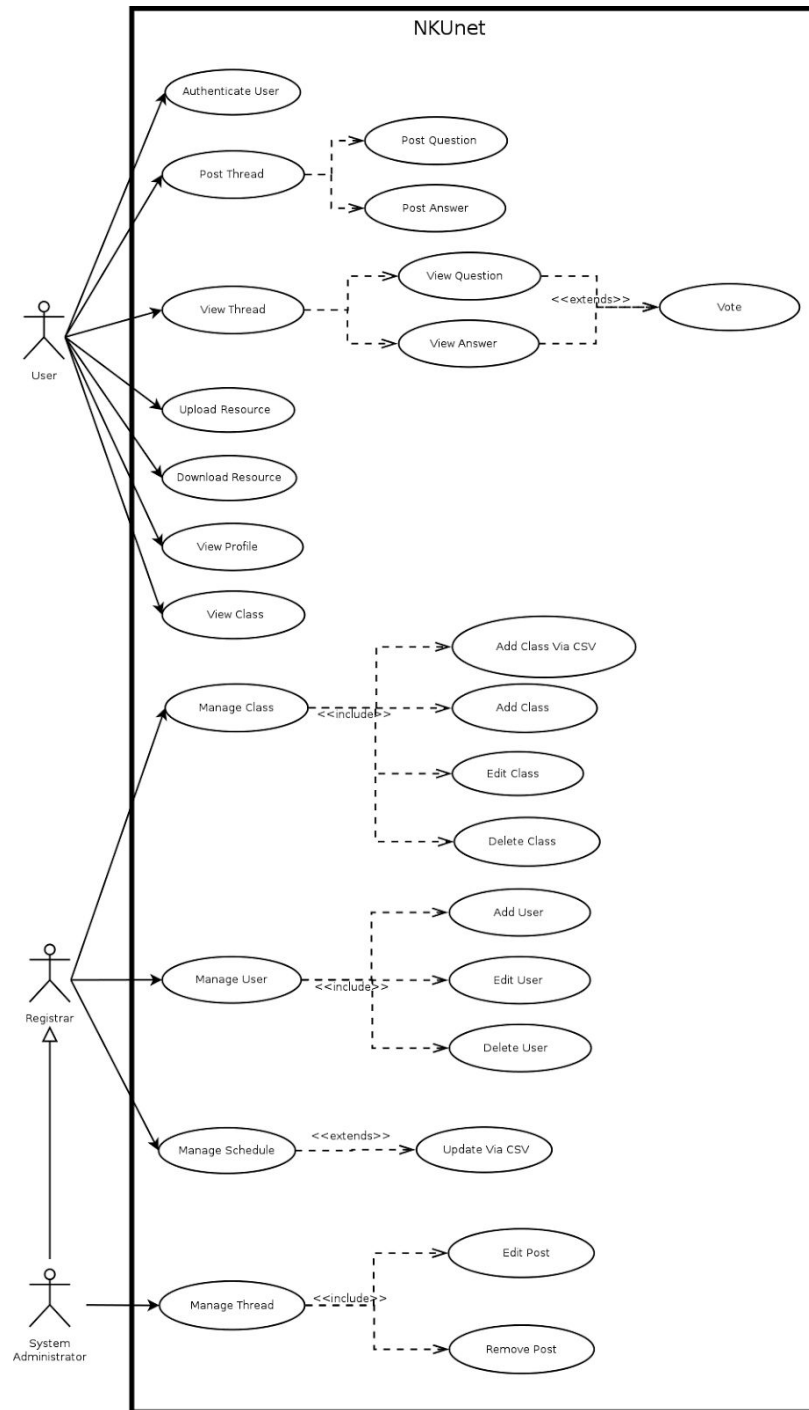
Change Discussion

Iteration 2

For this iteration, we implemented the following use cases: Manage Thread, Manage Users, Search, Manage Schedule, Authenticate User, Vote, Manage Class and View User. In addition to these we also updated our View Forum use case to View Class in order to align more closely to the project requirements. Additions to this document includes: an updated use case context diagram, fully dressed use cases for UC5-11 and their associated sequence diagrams.

Use Case Model

Context Diagram



Use Cases

UC1: View Class

Scope: NKUNet System

Level: user goal

Primary Actor: User

Stakeholders and Interest:

- User: Wants to view a forum of questions related to a specific topic
- NKU: Wants to group questions by forums to organize topics.
- System Administrator: Wants questions grouped by forums for site organization.

Preconditions: User has an account and is logged into NKUNet

Postconditions: View of specific forum is open, showing list of questions.

Basic Flow:

1. User picks a forum to view and selects it.
2. A new page is displayed with a list of questions in that forum.

Alternate Flow:

2a. No questions have been asked in that forum yet.

1. User can ask a question or return to home page.

UC2: Ask Question

Scope: NKUNet System

Level: user goal

Primary Actor: User

Stakeholders and Interest:

- User: Wants to ask a question in a forum about a topic.
- NKU: Wants students to be able to ask questions to have answered.
- System Administrator: Wants question to be placed in forum with the ability to be answered and managed.

Preconditions: User has an account and is logged into NKUNet

Postconditions: Question is posted and others can respond.

Basic Flow:

1. User enters forum from home page.
2. User initiates process to ask question.
3. User enters question title and body.
4. User submits question.

Alternate Flow:

4a. User forgets to enter title

1. System throws error and prompts user to correct mistake.
- 4b. User forgets to enter body.
 1. System throws error and prompts user to correct mistake.

UC3: Answer Question

Scope: NKUnet System

Level: user goal

Primary Actor: User

Stakeholders and Interest:

- User: Wants to share an answer to a question from another user.
- NKU: Wants users to be able to easily answer questions to share knowledge.
- System Administrator: Wants answer process to be quick and responsive.

Preconditions: User has an account and is logged into NKUNet

Postconditions: Response is posted and others can reply or vote.

Basic Flow:

1. User selects forum to view questions.
2. User selects question to view its body.
3. User initiates process to answer question.
4. User enters body of response.
5. User submits response.

Alternate Flow:

5a. User enters nothing in response body.

1. System throws error and prompts user to correct mistake

4a. User cancels answering question.

1. System exits answer process and returns to initial question view.

UC4: View Thread

Scope: NKUnet System

Level: user goal

Primary Actor: User (Student or Faculty)

Stakeholders and Interest:

- User: Wants to view the contents of a thread to read or answer other user questions
- Admin: Wants to log threads visited by users

Preconditions: User Authenticated

Postconditions: Thread post and replies are all displayed, separated by pages if necessary. A user can reply to comments or the main post via an appropriate input field along with buttons to vote up or down a particular replies or posts.

Basic Flow:

1. User searches for a thread
2. System displays search results

3. User clicks on a thread
4. System displays all post and replies for the selected thread
5. User can choose to reply to the thread if they desire

Alternate Flow:

- 1a. User wants to delete a post created by them.
 1. User selects the delete option on desired post.
 2. System asks user to confirm action
 3. System sets a delete flag in database for the post if the user accepts
- 1b. User wants to edit a post created by them.
 1. User selects the edit option on desired post
 2. System moves to an edit form with previous post filled in
 3. User edits their post then confirms edits
 4. System updates post and labels post as 'edited'

UC5: Manage Thread

Scope: NKUnet System

Level: user goal

Primary Actor: Administrator

Stakeholders and Interest:

- User: Wants inappropriate content to be removed.
- Administrator: Wants an interface to edit and delete threads.

Preconditions: Administrator Authenticated

Postconditions: Administrator edits or deletes a thread

Basic Flow:

1. Administrator searches for a thread
2. System displays search results
3. Administrator clicks on a thread
4. System displays all post and replies for the selected thread
5. Administrator selects a post or reply to edit or delete
6. System asks Administrator to confirm action.

Alternate Flow:

- 1a. Administrator is given a direct link to a post/reply from the User
 1. Administrator skips to step 5
- 1b. Administrator searches for an invalid (nonexistent) thread.
 1. Administrator finds correct search string
 2. Administrator searches again.
- 6a. Administrator selects cancel from confirmation window
 1. Administrator backtracks and fixes mistake
- 6b. Administrator selects okay from confirmation window
 1. System sends update or delete flag to database to modify selected post/reply

UC6: Manage Users

Scope: NKUnet System

Level: user goal

Primary Actor: Administrator

Stakeholders and Interest:

- Administrator: Wants the ability to add, update and delete users.
- Registrar: Wants the ability to add new users via bulk upload (CSV) or a web form
- User: Wants their their account to be added into the system.

Preconditions: Administrator/Registrar Authenticated

Postconditions: User account has been added, updated or deleted

Basic Flow:

1. Administrator opens Admin page
2. Administrator selects Manage Users
3. System displays list of current users
4. Administrator chooses to add a new user
5. Administrator enters user information into the form
6. System creates new user in the database

Alternate Flow:

- 1b. Registrar opens Admin page
 1. Registrar Manage Users
 2. Registrar selects to add new users via a CSV file
- 2b. Registrar selects to add new user via the web form
 1. Registrar enters user information.
 2. System creates new user in the database
3. Registrar uploads CSV files
4. System adds new users into the database
- 4a. Administrator chooses to modify an existing user
 1. Administrator selects a user
 2. Administrator edits user account information
 3. System updates user in the database
- 4b. Administrator chooses to delete an existing user
 1. Administrator selects a user
 2. Administrator selects to delete the user
 3. System deletes the selected user in the database

UC7: Manage Schedule

Scope: NKUnet System

Level: user goal

Primary Actor: Registrar

Stakeholders and Interest:

- Registrar: Wants the ability to manage the class schedules of student users each semester.
- User: Wants to see their schedule on their profile page

Preconditions: Registrar Authenticated

Postconditions: Registrar has entered class schedule(s) into the system.

Basic Flow:

1. Registrar opens Admin page
2. Registrar selects Manage Schedule
3. System displays the option to manage schedule(s) via web form or CSV
4. Registrar selects to add a new schedule via the web form
5. Registrar enters schedule information into the form
6. System adds the schedule to the users profile

Alternate Flow:

4a. Registrar selects to add a new schedule via CSV

1. Registrar uploads CSV file
2. System adds multiple schedules to multiples users' profiles

4b. Registrar chooses to edit a user's existing schedule

1. Registrar uses the form to edit the user's schedule
2. System updates the schedule on the user's profile

UC8: Authenticate User

Scope: NKU Net System

Level: user goal

Primary Actor: Users (Faculty, Student, Administrator and Registrar)

Stakeholders and Interest:

- NKU: wants user accounts to be secure
- User: wants authentication so that their account is secure

Preconditions: User is on the NKU Net website

Postconditions: User has been successfully logged into the system

Basic Flow:

1. User selects login from the NKU Net home page
2. System displays login form
3. User enters username and password
4. System validates the username and password
5. System directs the user back to the home page

Alternate Flow:

3a. User does not have an account in the system

1. User contacts an NKU Net administrator or a registrar to get an account created

3b. User forgot their username/password

1. User clicks 'Forgot Username/Password'

2. System prompts the user with options to reset their username or password
- 4a. User entered an incorrect username/password
 1. User fixes their mistakes and resubmits login form

UC9: Vote

Scope: NKUnet System

Level: user goal

Primary Actor: User

Stakeholders and Interest:

- User: Wants the ability to rank the quality of a thread answer

Preconditions: User Authenticated

Postconditions: Thread answer displays the user's vote

Basic Flow:

1. User searches for a thread (UC7)
2. System displays thread
3. User upvotes or down an answer
4. Systems saves vote in the database
5. System displays number of votes next to the answer

Alternate Flow:

- 4a. System returns an error while attempting to the vote into the database
 1. User contacts an Administrator
 - 1a. User tries again to enter vote

UC10: Manage Class

Scope: NKUnet System

Level: user goal

Primary Actor: Registrar

Stakeholders and Interest:

- User: Wants to have access to all the classes (forums) they are enrolled in.
- Registrar: Wants users to be added to the classes (forums) they are enrolled in.

Preconditions: Registrar Authenticated

Postconditions: Registrar has created/modified/deleted a class (forum)

Basic Flow:

1. Registrar opens Admin page
2. Registrar selects Manage Class
3. System displays a list of all classes
4. Registrar selects to edit a class
5. Registrar makes changes to the class and submits
6. System updates the class in the database

Alternate Flow:

- 4a. Registrar selects to create a new class via a form
 1. Registrar enters class information into form

2. System create new class in the database
- 4b. Registrar selects to create a new class(es) via CSV upload
 1. Registrar uploads a CSV file.
 2. System parses the file and adds the class(es) into the database
- 4c. Registrar selects to delete a class
 1. Registrar confirms that they want to delete this class
 2. System deletes this entry from the database

UC11: View User

Scope: NKUnet System

Level: user goal

Primary Actor: User

Stakeholders and Interest:

- User: Wants their profile to display information about their activity in NKUNet

Preconditions: User is on NKUNet

Postconditions: User profile displays: user schedule, recent posts, user reputation and

Basic Flow:

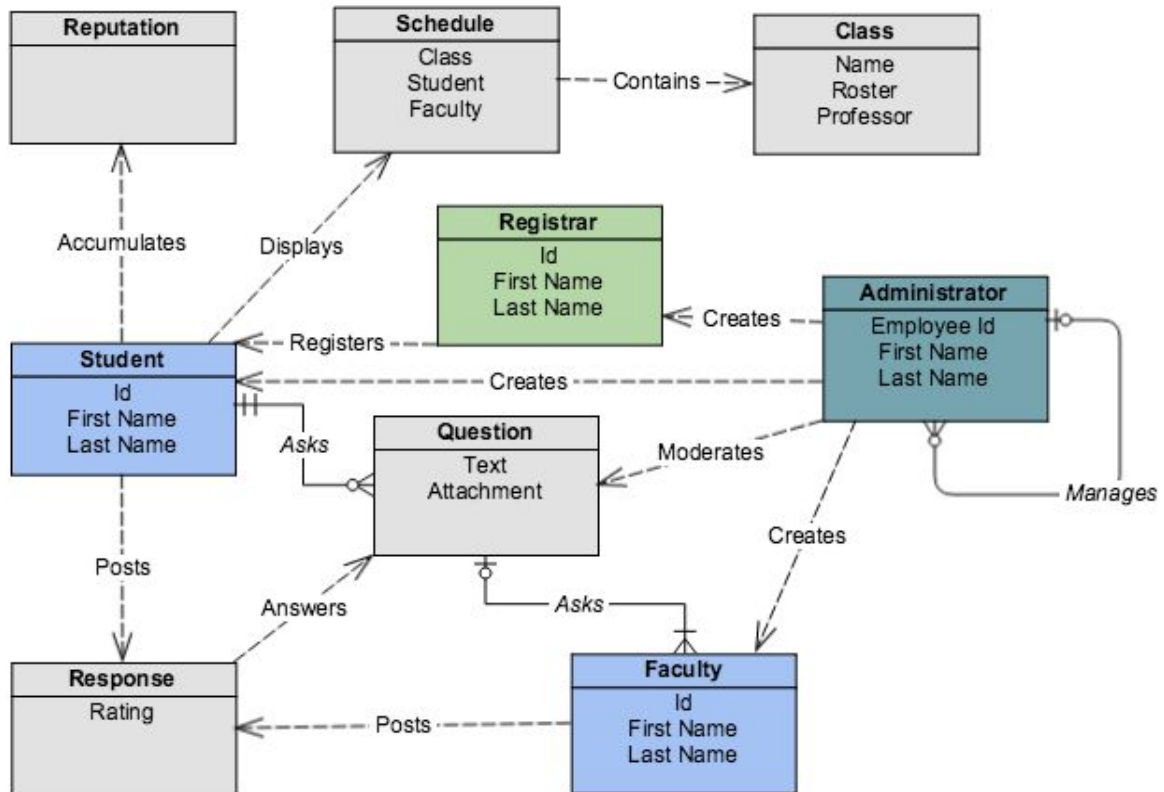
1. User clicks on a user's name
2. System opens that user's profile
3. System displays the user's reputation, schedule and recent posts

Alternate Flow:

3a. System has database error trying to populate the user's profile.

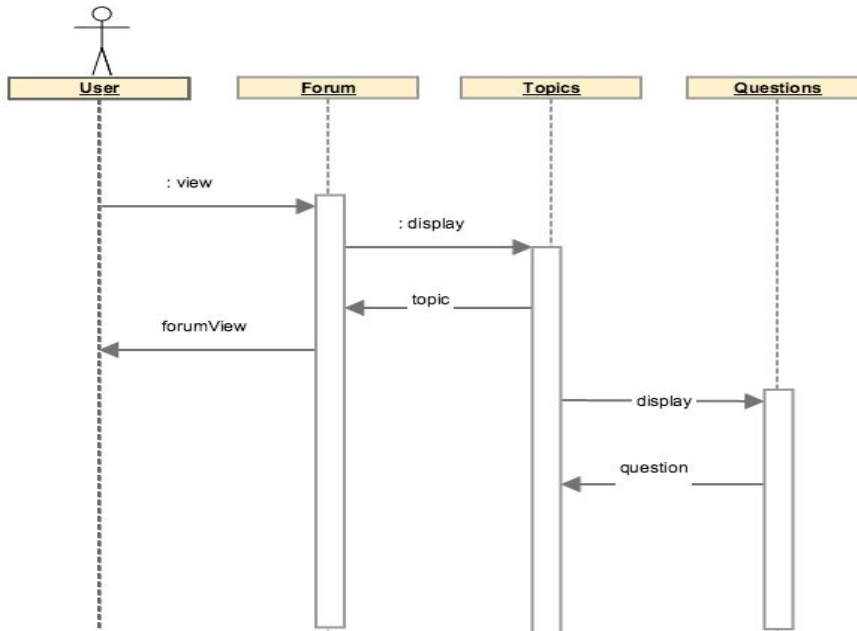
1. User contacts an NKUNet Admin
 - 1a. User tries to access the user profile again

Domain Model

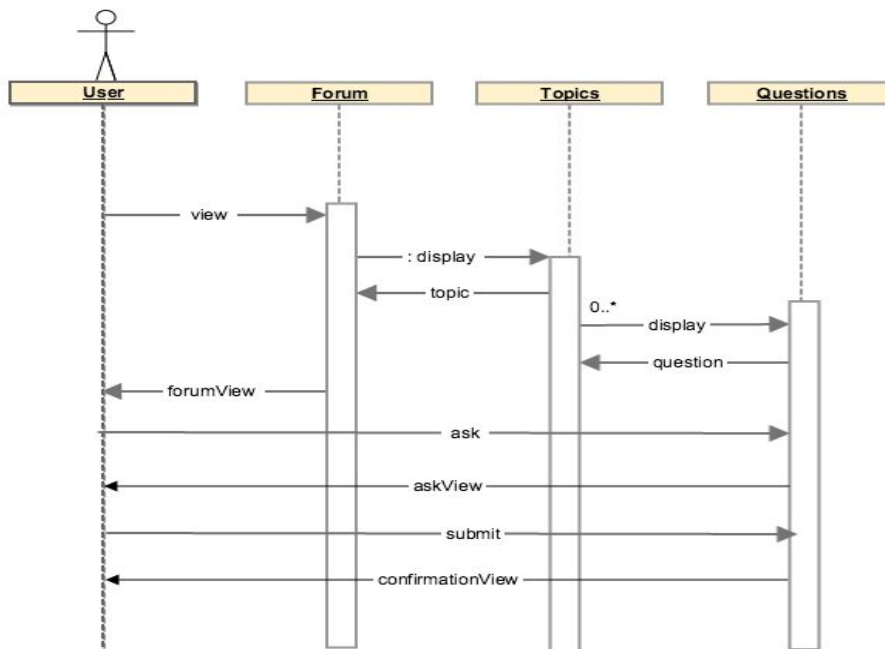


System Sequence Diagrams

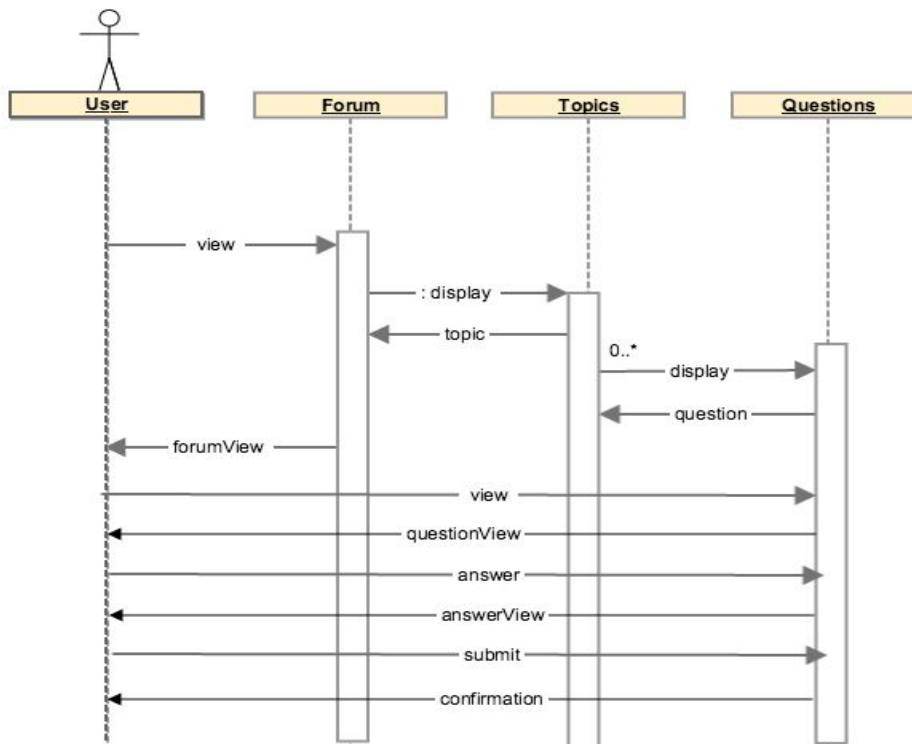
1. View Class



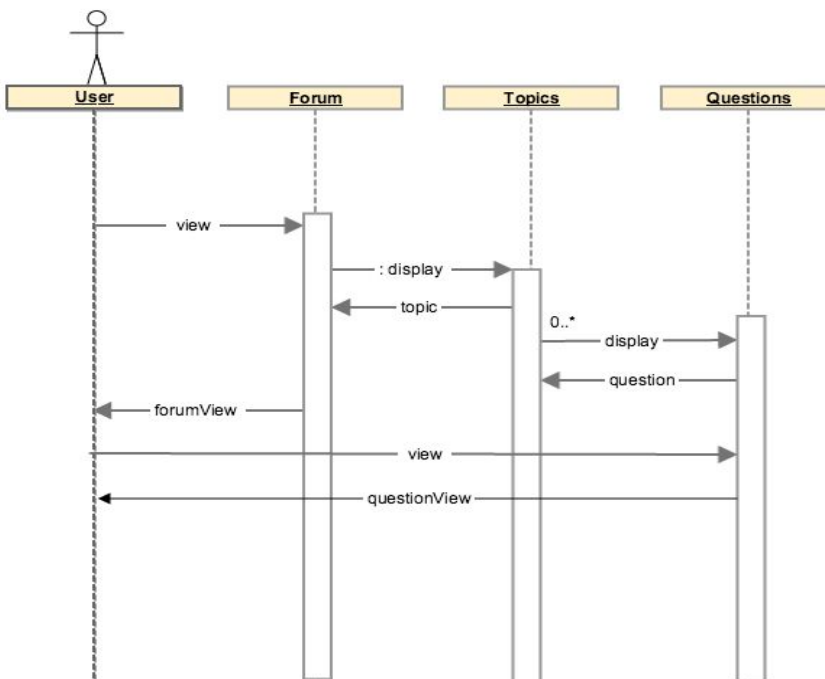
2. Ask Question



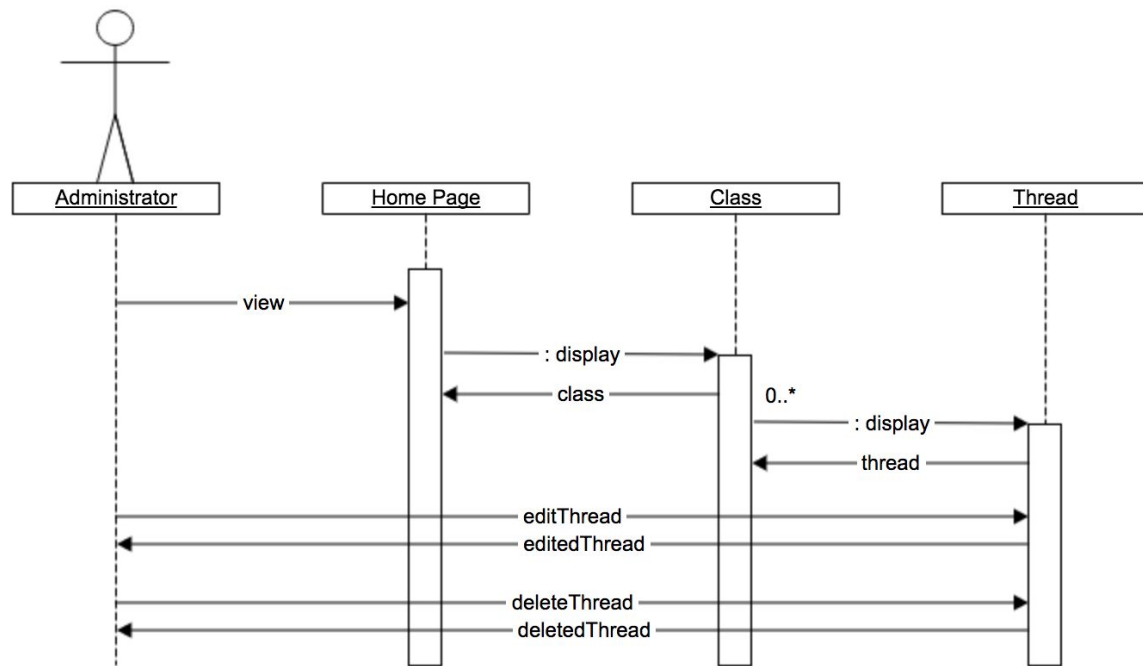
3. Answer Question



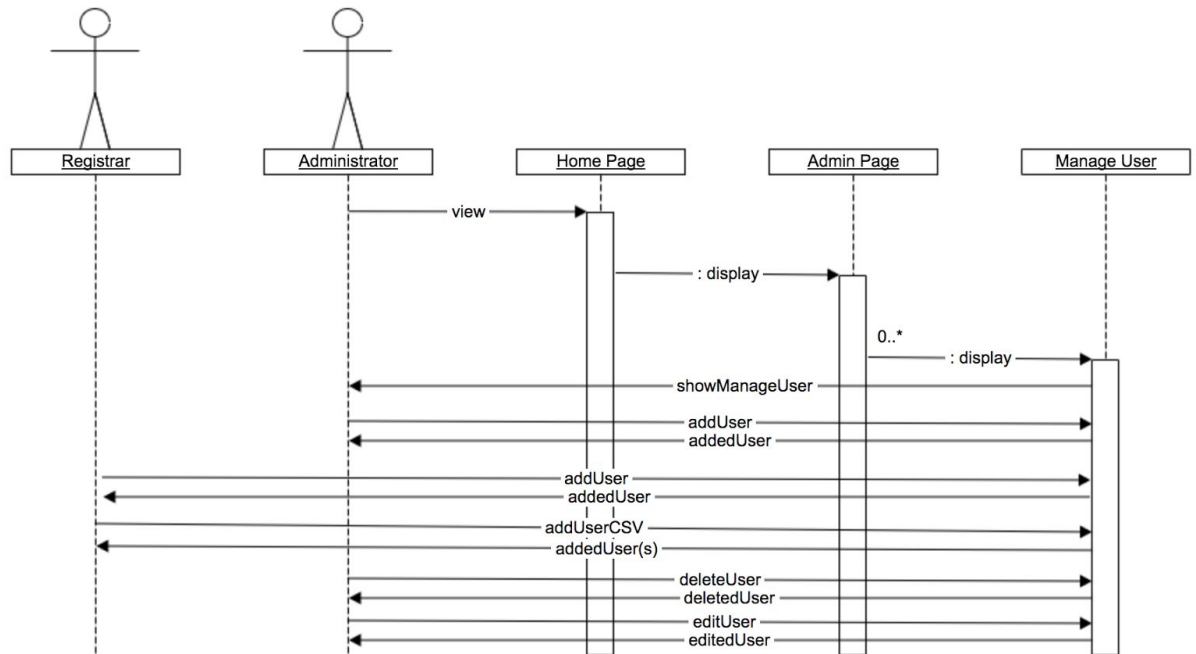
4. View Thread



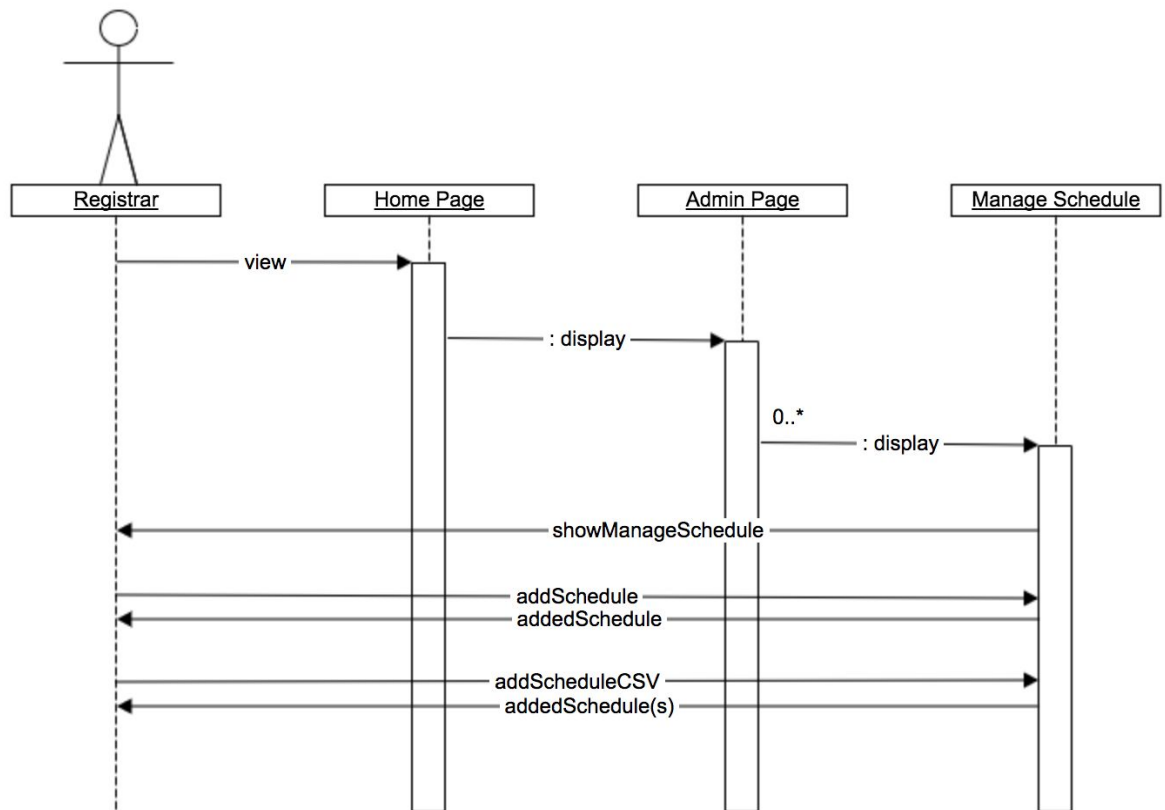
5. Manage Thread



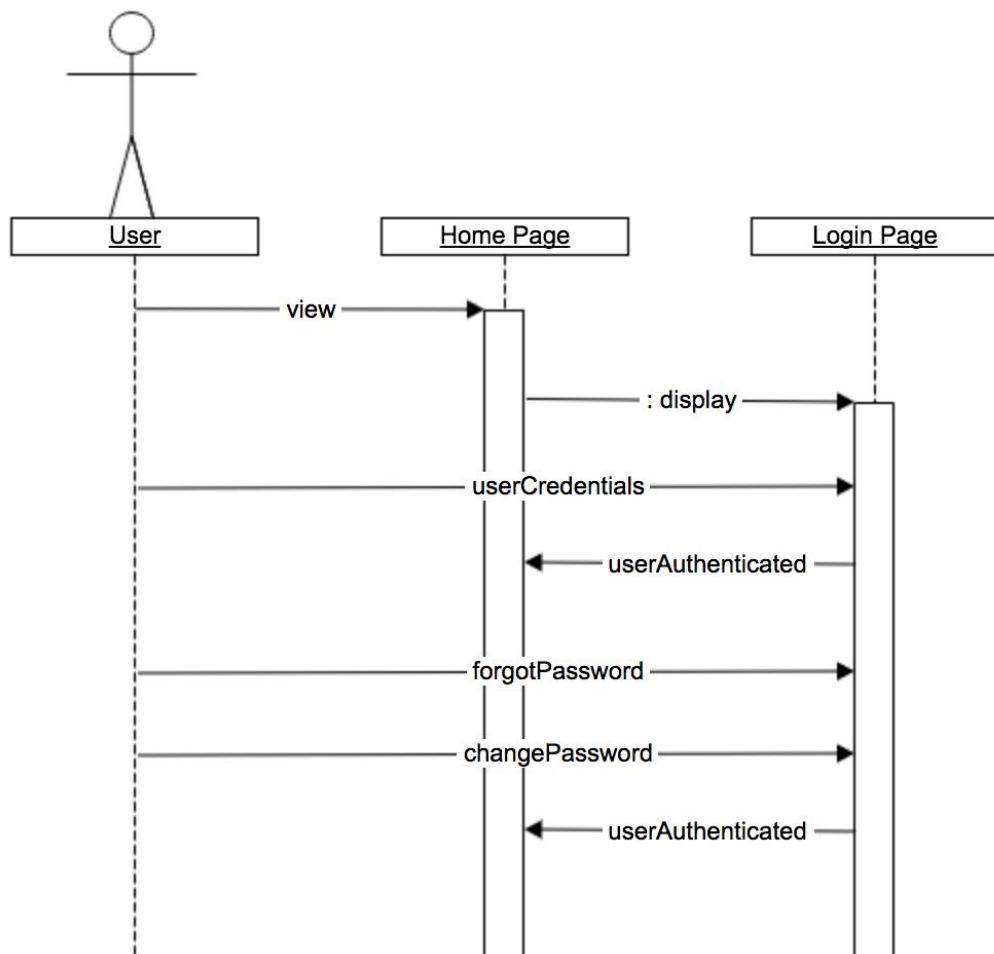
6. Manage Users



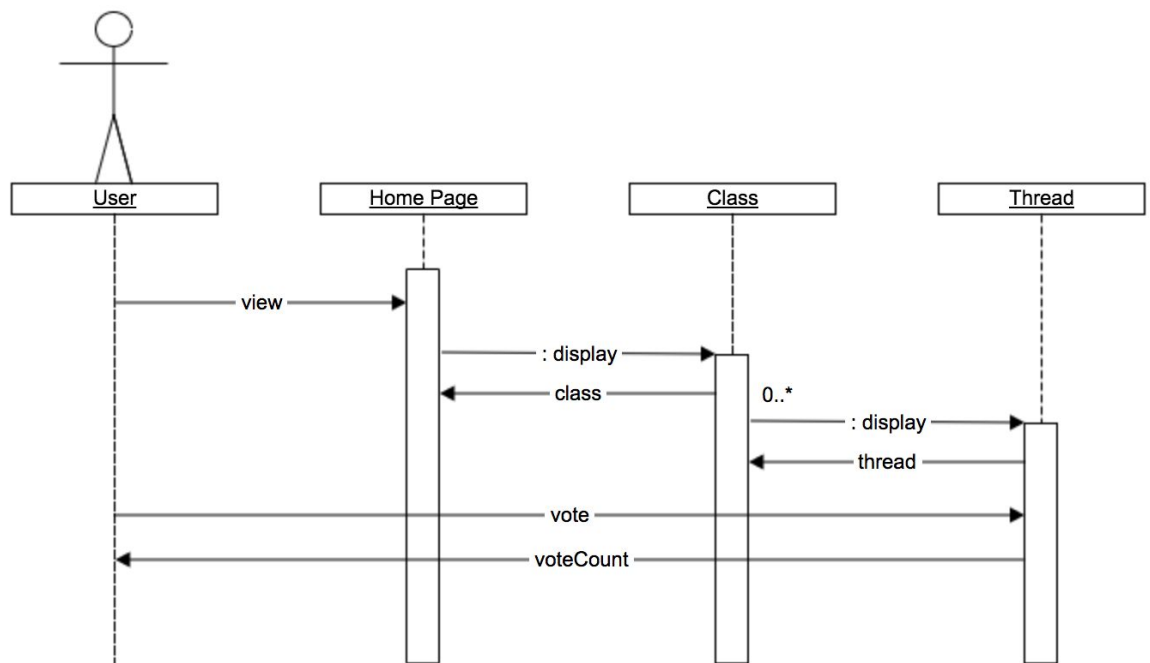
7. Manage Schedule



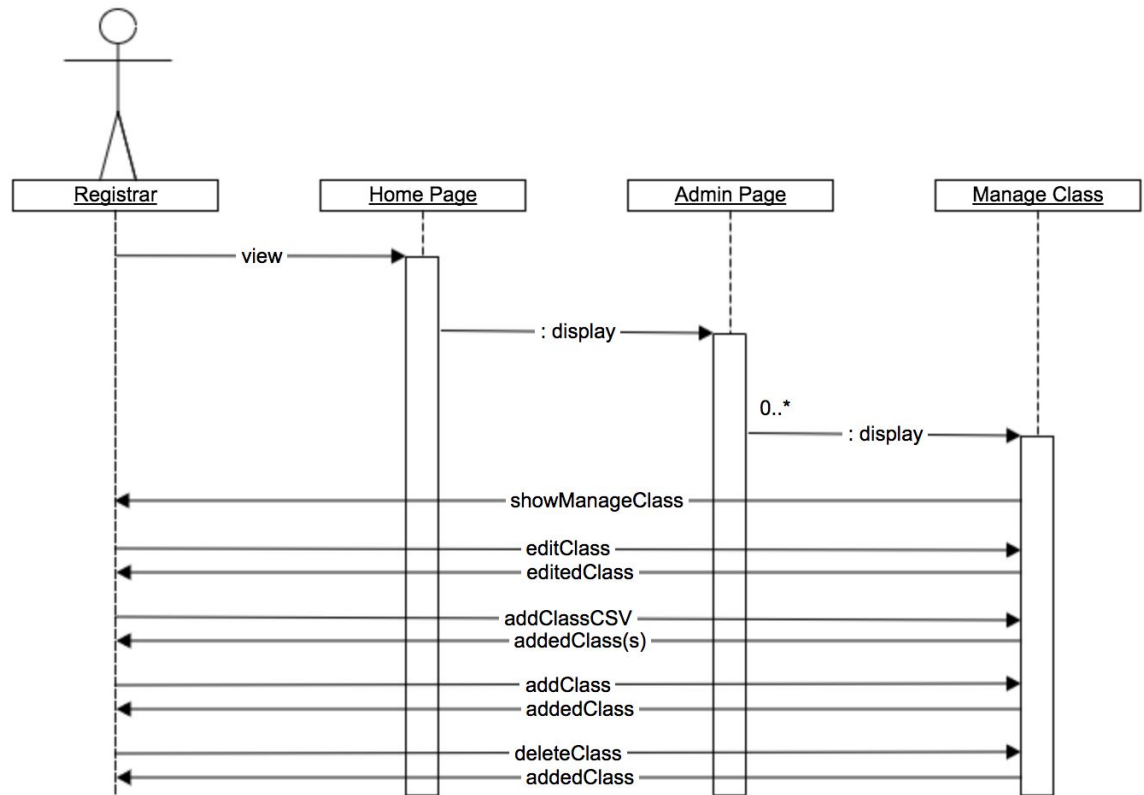
8. Authenticate User



9. Vote



10. Manage Class



11. View User

