

## TABLE OF CONTENTS

I.	Discussion for Iteration 2.....	2
II.	Discussion for Iteration 3.....	3
III.	Use Case Model.....	4
IV.	Fully Dressed Use Cases.....	5
V.	Domain Model.....	10
VI.	System Sequence Diagrams.....	11

VII.

### **Discussion for Iteration 2**

For iteration 2, we added four new use cases to meet requirements of this iteration. The use cases we added were “Add Student to Course”, “Manage Course”, “View Forum”, and “Answer Voting”. “Add Student to Course” means that the faculty member approves the student for the student to be added in the course. The student must be currently enrolled in the course to be added. “Manage Course” is for the admin to add, delete, and edit courses based on the need. “View Forum” is simply viewing a forum after clicking on it. “Answer Voting” is for the student and faculty to say that they agree, disagree, or think the answer is inappropriate. This iteration we focused on getting the courses set up and finishing the question and answer functionality. Sequence Diagrams were added for all of these use cases to reflect their functionality. Nothing that was in iteration 1 was modified or deleted because we implemented the basic use cases behind NKUNet.

### **Discussion for Iteration 3**

For iteration 3, we further implemented the use cases for iteration 2 and finalized anything that was previously added.

## Use Case Model



## Fully Dressed Use Cases

### Use Case 1: Post Questions

**Primary Actor:** Faculty and Student

**Stakeholders and Interests:**

- Faculty and Student: Post questions in forum to be answered

**Preconditions:** Finding the corresponding forum related to the question and posting the question

**Postconditions:** Question has been posted to the correct forum for other users to participate in the forum

**Summary:** A user wants to post a question to a forum related to their question. The question is posted and is a part of the forum. Other users can now answer and or ask follow up questions.

**Basic Flow:**

1. User has a question that they want to discuss
2. User locates the forum related to their question
3. User posts question

**Alternate Flow:**

2. User has a question that doesn't relate with any created forum
  - User can post it in the incorrect forum
  - Or, User contacts Administrator with request to create a forum

### Use Case 2: Answer Questions

**Primary Actor:** Faculty and Student

**Stakeholders and Interests:**

- Faculty and Student: Post answers and vote on answers that were posted by other users

**Preconditions:** Observing different questions that are posted in a given forum and coming across a question that the user has an answer to or an opinion on an answer already posted

**Postconditions:** An answer to a question that was posted is posted

**Summary:** A user comes across a question that they have an answer to and post it to the forum where other users can observe the forum after it has been updated. A user sees an answer they have an opinion about and vote on the answer.

**Basic Flow:**

1. User visits different forums.
2. User reads posted questions.
3. User discovers a question that they have an answer to.
4. User posts the answer to the question on the forum.

**Alternate Flow:**

3. User never comes across a question that they have an answer to
  - User doesn't post anything
4. User sees similar answer already
  - User votes yes to agree with answer
4. User sees an answer that is incorrect
  - User votes no to disagree with answer

### Use Case 3: Manage Users

**Primary Actor:** System Admin

**Stakeholders and Interests:**

- Students: want up to date answers from other users that, for example, a 20 year alum won't be able to provide.
- Faculty and Board of Regents: want a safe platform for students to converse, so multiple time offenders should lose privileges to post.

**Preconditions:** User is authenticated and identified as admin.

**Postconditions:** Users are added if they are a freshman and deleted from the system if they are a graduating senior

**Basic Flow:**

1. Admin checks status of student
2. Admin determines what to do based on student status

**Alternate Flow:**

2. Student is a freshman
  - Student is added to the system
3. Student is a graduating senior
  - Student is deleted from system

### Use Case 4: Manage Courses

**Primary Actor:** Registrar/Admin

**Stakeholders and Interests:**

- Students: want to know who is in their class

**Preconditions:** User is identified and authenticated as a registrar.

**Postconditions:** Class page is generated for classes which don't already exist in system.

**Summary:** Registrar creates classes for the current semester, so that students can connect their current courses to their profile page.

**Basic Flow:**

1. Registrar inputs class identifier.
2. Registrar inputs faculty identifier for instructor.
3. Registrar inputs class information.
4. Registrar saves class.

**Alternate Flow:**

1. Class already exists.
  - Registrar edits class information.
  - Registrar saves class.
2. System can't find faculty member.
  - Registrar adds faculty user.
  - Proceeds to step 3.
4. Registrar wants to delete course
  - Registrar deletes course

### Use Case 5: Manage Posts

**Primary Actor:** Administrator

**Stakeholders and Interests:**

- Students: wants appropriate answers and questions to get the most help in a course

**Preconditions:** A question or answer has been submitted as inappropriate

**Postconditions:** Question/answer is deleted if determined inappropriate

**Summary:** Administrator reviews question or answer that has been marked inappropriate, admin determines if inappropriate.

**Basic Flow:**

1. Question/answer is reviewed
2. Question/answer is inappropriate
3. Question/answer is deleted
4. Marker is removed

**Alternating Flow:**

2. Question/answer is appropriate
- Question/answer is not deleted

**Use Case 6: Add Student to Course**

**Primary Actor:** Faculty

**Stakeholders and Interests:**

- Students: want to be added to course to get access to questions and answers pertaining to course
- Faculty: want students to excel in course

**Preconditions:** Student has requested to be added to the course

**Postconditions:** Faculty adds student to course

**Summary:** Faculty member who is teaching course reviews student requests to be added to course. Student is given access if they are currently enrolled in course.

**Basic Flow:**

1. Faculty reviews student request
2. Faculty checks if student is currently enrolled in course
3. Faculty gives student access to course

**Alternating Flow:**

2. Student isn't enrolled in course
- Student is denied access to course

**Use Case 7: View Forums**

**Primary Actor:** Student, Faculty, Admin

**Stakeholders and Interests:**

- Students: want to be able to view topics

**Preconditions:** User is identified and authenticated

**Postconditions:** Forum is displayed

**Summary:** Click on forum, a list of questions and summary information (who asked question, date and time question was asked, number of answers, date and time of last answer)

**Basic Flow:**

1. User clicks on a forum.
2. System presents list of questions.
3. System presents summary information, which includes who asked question, date and time question was asked, number of answers, date and time of last answer.

**Alternating Flow:**

1. User doesn't click on a forum
  - Forum isn't presented to user

**Use Case 8: Answer Voting**

**Primary Actor:** Student and Faculty Member

**Stakeholders and Interests:**

- Student: wants reliable answers to questions
- Faculty Member: wants students to have access to correct answers

**Preconditions:** Student and Faculty are identified and authenticated

**Postconditions:** Question is given an up or down vote by user, user's reputation is calculated

**Summary:** Student and/or Faculty member reads question, decides if they agree with the answer given, then clicks the up or down arrow to vote appropriately. User's reputation is calculated

**Basic Flow:**

1. Student/Faculty reads answer.
2. User decides if they agree with answer.
3. User clicks up or down arrow to show their decision.
4. User's reputation is calculated.

**Alternating Flow:**

2. User doesn't think answer makes sense
  - User doesn't vote up or down
  - Marks as inappropriate

**Use Case 9: Authenticate User**

**Primary Actor:** User

**Stakeholders and Interests:**

- Student, Faculty, Admin: wants access to features only those specific users can have

**Preconditions:** N/A

**Postconditions:** User is logged in

**Summary:** User logs into site and is given access depending on what kind of user they are

**Basic Flow:**

1. User enters login credentials

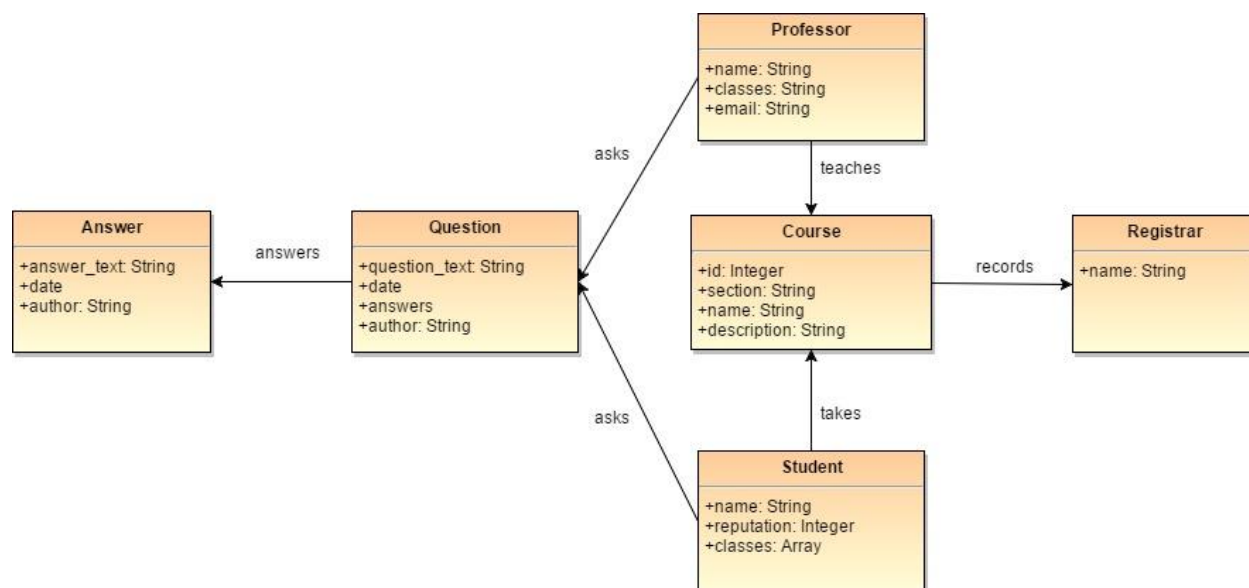


2. System verifies with database that they are a member
3. User given appropriate access

**Alternating Flow:**

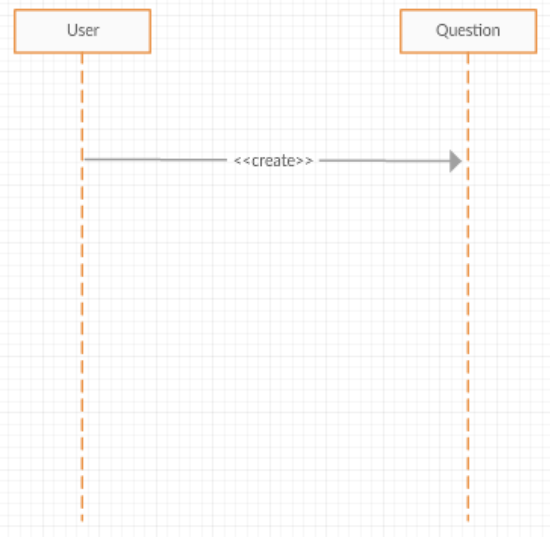
2. User is not in the system
  - Error message is presented
- 3a. User is a student
  - Student is given access to posting and answering questions and answers
- 3b. User is a faculty member
  - Faculty member can do what student does with addition of adding students to courses
- 3c. User is an admin
  - Admin gets all access the a faculty member has with the addition of being able to delete and edit posts and questions

## **Domain Model**

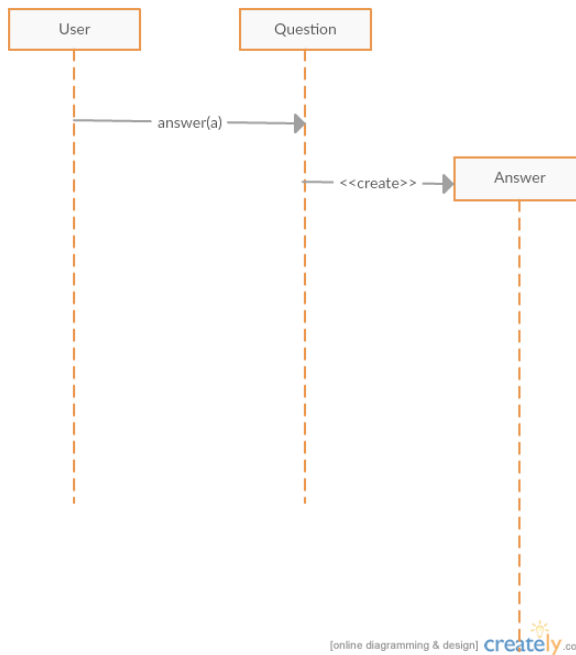


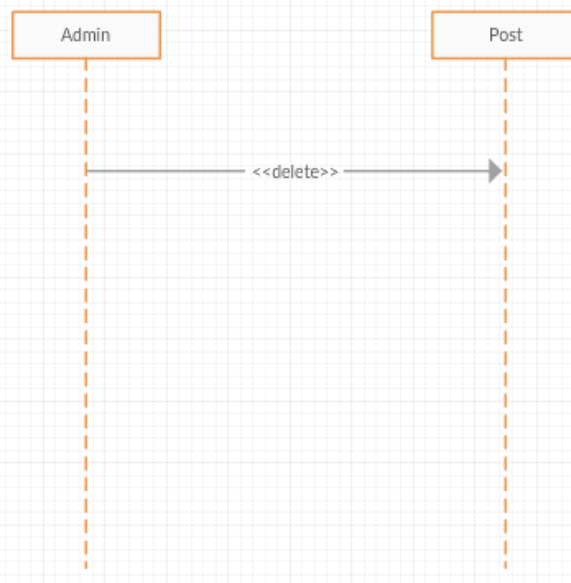
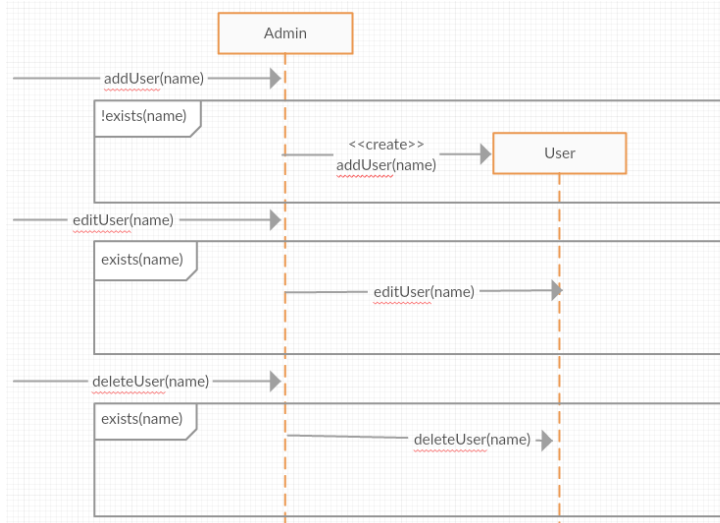
## System Sequence Diagrams

### UC Post Question

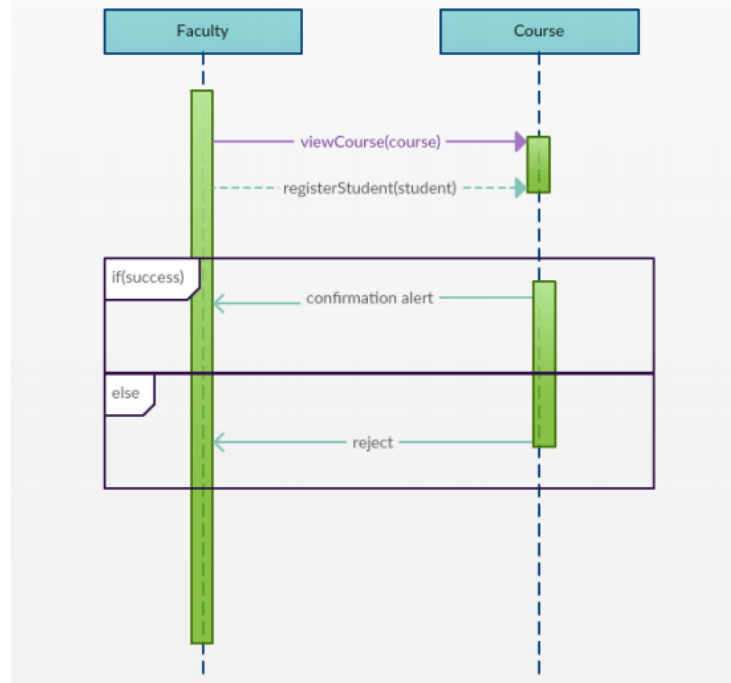


### UC Answer Question

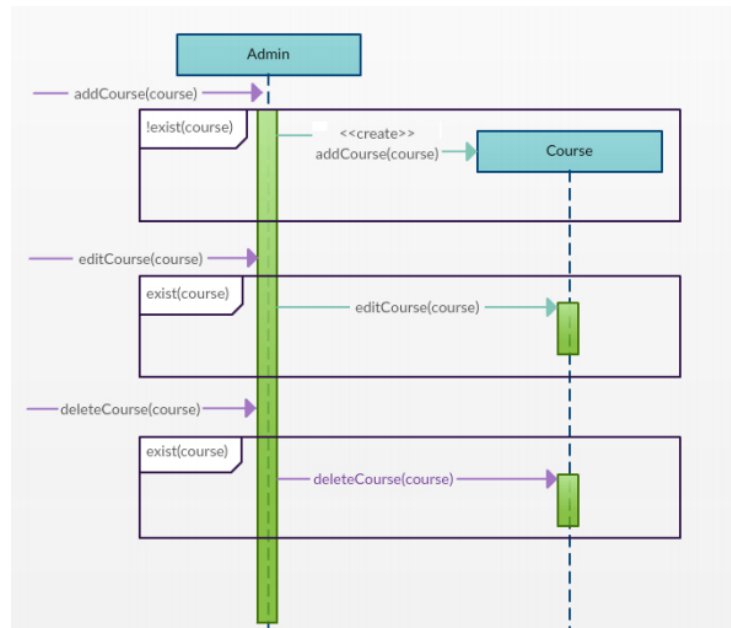


**UC Manage Post****UC Manage Users**

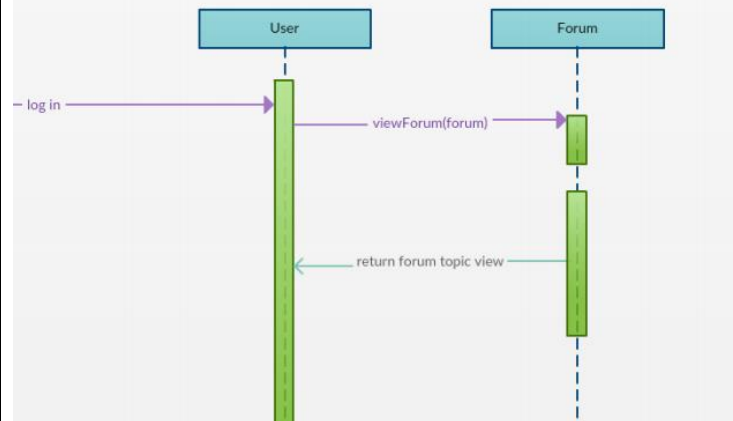
## UC Add Student to Course



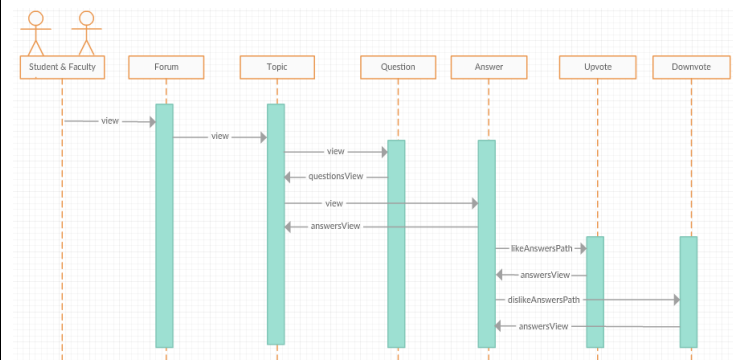
## UC Manage Courses



## UC View Forums



## UC Answer Voting



## UC Authenticate User

