

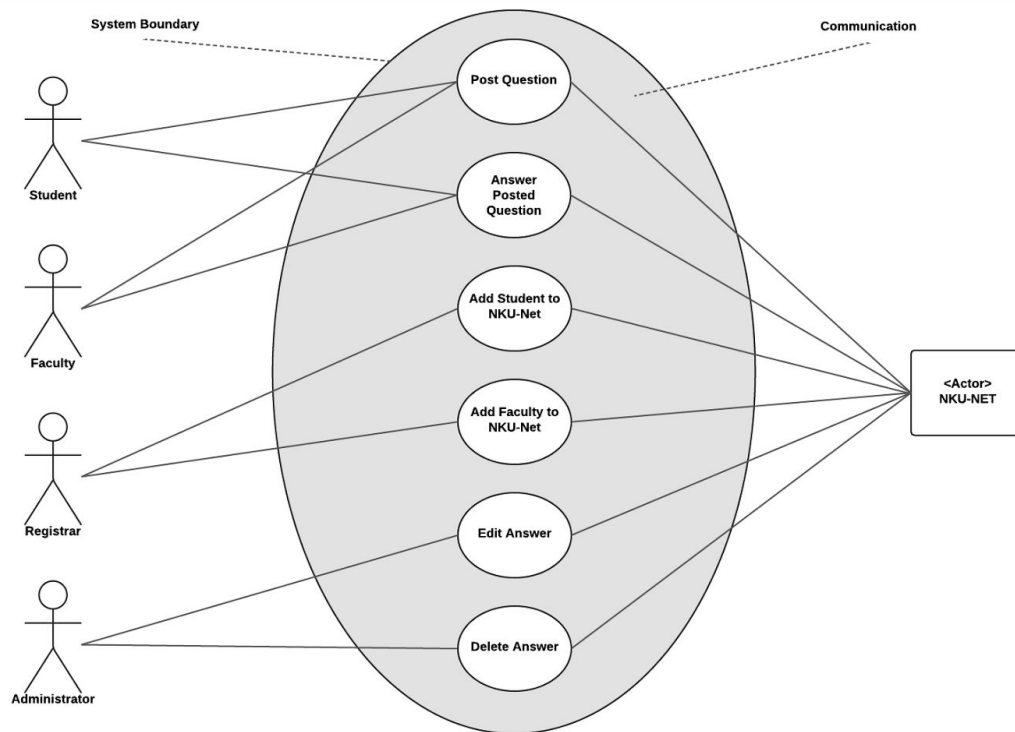
Use Cases

1. Post question.
2. Answer question.
3. Upvote an answer.
4. Downvote an answer.
5. Update profile page.
6. Upload document.
7. Add student to NKU-Net.
8. Add faculty member to NKU-Net.
9. Add class to NKU-Net.
10. Edit question.
11. Delete question.
12. Edit answer.
13. Delete answer.

Use Case Diagram

USE CASE DIAGRAM

Team Super Ghost



Use Case (UC1): Post Question.

Scope: Post Question System

Level: user-goal

Primary Actor: Student, Faculty Member

Stakeholders and Interests:

- Student: Wants to be able to post question on NKU-Net.
- Faculty Member: Wants to be able to post question on NKU-Net.
- Administrator: Wants to be able to post question on NKU-Net.
- School: Wants students to be added as immediately as possible.

Preconditions: Student or faculty member is logged into system upon being identified and authenticated.

Success guarantee (or Postconditions): Student or faculty member posts question onto NKU-Net.

Main Success Scenario (or Basic Flow):

1. Student/faculty member clicks 'post question' button.
2. Student/faculty member clicks textbox in newly displayed window.
3. Student/faculty member types out question of their choice.
4. Student/faculty member clicks 'submit' button to post question on NKU-Net.

Student/faculty member repeats steps 1-4 until done posting questions.

Extensions (or Alternative Flows):

*a. At any time, System fails:

1. To ensure security of system, student/faculty member is logged off and connection is shut off.

Special Requirements:

- System posts question within 30 seconds, 90% of the time.
- Compatible with all major Internet browsers.

Technology and Data Variations List:

*a. Student/faculty member can use any major web browser (Firefox, Google Chrome, Safari, Microsoft Edge, etc.).

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- Should there be a character or word limit on questions?
- Should there be a method in place to censor question content (such as profanity)?

~

Use Case (UC2): Answer Question.

Scope: Submit Answer System

Level: user-goal

Primary Actor: Student, Faculty Member

Stakeholders and Interests:

- Student: Wants to be able to post answer to question on NKU-Net.
- Faculty Member: Wants to be able to post answer to question on NKU-Net.
- Administrator: Wants to be able to post answer to question on NKU-Net.
- School: Wants students to be added as immediately as possible.

Preconditions: Student or faculty member is logged into system upon being identified and authenticated.

Success guarantee (or Postconditions): Student or faculty member posts answer to question on NKU-Net.

Main Success Scenario (or Basic Flow):

1. Student/faculty member clicks 'answer question' button to post answer.
2. Student/faculty member clicks textbox in newly displayed window.
3. Student/faculty member types out their answer to question.
4. Student/faculty member clicks 'submit' button to post answer on NKU-Net.

Student/faculty member repeats steps 1-4 until done posting questions.

Extensions (or Alternative Flows):

*a. At any time, System fails:

1. To ensure security of system, student/faculty member is logged off and connection is shut off.

Special Requirements:

- System posts answer to question within 30 seconds, 90% of the time.
- Compatible with all major Internet browsers.

Technology and Data Variations List:

*a. Student/faculty member can use any major web browser (Firefox, Google Chrome, Safari, Microsoft Edge, etc.).

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- Should there be a character or word limit on answers to questions?
- Should there be a method in place to censor answers to questions content (such as profanity)?

Use Case (UC3): Edit answer.

Scope: Edit Answer System

Level: user-goal

Primary Actor: Administrator

Stakeholders and Interests:

- Administrator: Wants to be able to edit answer on NKU-Net.
- School: Wants students to be added as immediately as possible.

Preconditions: Administrator is logged into system upon being identified and authenticated with admin permissions.

Success guarantee (or Postconditions): Administrator's updated edit of the answer is uploaded to NKU-Net.

Main Success Scenario (or Basic Flow):

1. Administrator selects a class forum.
2. Administrator selects a question forum.
3. Administrator selects an answer that he/she wants to edit.
4. Administrator edits the answer to his/her liking.
5. Administrator saves the edit.

Extensions (or Alternative Flows):

*a. At any time, System fails:

1. To ensure security of system, Administrator is logged off and connection is shut off.

Special Requirements:

- System posts edited answer within 30 seconds, 90% of the time.
- Compatible with all major Internet browsers.

Technology and Data Variations List:

*a. Administrator can use any major web browser (Firefox, Google Chrome, Safari, Microsoft Edge, etc.).

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- Should there be a limit on how much can be edited?

~

Use Case (UC4): Delete answer.

Scope: Delete Answer System

Level: user-goal

Primary Actor: Administrator

Stakeholders and Interests:

- Administrator: Wants to be able to delete answer on NKU-Net.
- School: Wants students to be added as immediately as possible.

Preconditions: Administrator is logged into system upon being identified and authenticated with admin permissions.

Success guarantee (or Postconditions): Administrator's chosen answer is deleted from NKU-Net.

Main Success Scenario (or Basic Flow):

1. Administrator selects a class forum.
2. Administrator selects a question forum.
3. Administrator selects an answer that he/she wants to edit
4. Administrator chooses to delete the answer.

Extensions (or Alternative Flows):

*a. At any time, System fails:

1. To ensure security of system, Administrator is logged off and connection is shut off.

Special Requirements:

- System deletes answer within 30 seconds, 90% of the time.
- Compatible with all major Internet browsers.

Technology and Data Variations List:

*a. Administrator can use any major web browser (Firefox, Google Chrome, Safari, Microsoft Edge, etc.).

Frequency of Occurrence: Could be nearly continuous.

Open Issues:

- Should there be a limit on how much can be deleted?
- Should we have a secondary check to ensure administrator is not abusing the delete function?

~

Use Case (UC5): Add Student.

Scope: NKU-Net Registration System

Level: user-goal

Primary Actor: Registrar

Stakeholders and Interests:

- Student: Wants immediate access to their course on NKU-Net.
- Faculty: Wants students to be in course before first day of classes.
- School: Wants students to be added as immediately as possible.
- Registrar: Wants to add large number of students quickly and easily.
- System Administrator: Wants registrar to add students easily so they don't have to.

Preconditions: Registrar worker is identified and authenticated.

Success guarantee (or Postconditions): Student is registered and saved under student identifier. Registrar worker is notified of registration success or failure.

Main Success Scenario (or Basic Flow):

1. Registrar worker manually enters new student information and submits to system.
2. System checks to see if student is already registered.
3. System logs student registration and notifies registrar worker of successful registration.
4. Registrar worker voluntarily disconnects from system.

Registrar worker repeats steps 1-3 until finished registering students.

Extensions (or Alternative Flows):

*a. At any time, System fails:

1. To ensure security of system, registrar worker is logged off and connection is shut off.
2. Error message is displayed.

1a. Registrar worker manually adds a student that already exists

1. System informs registrar worker that student already exists.
2. System prompts registrar worker if they want to update student information.
3. If Registrar worker says yes then student information is updated and system notifies worker.

1b. Registrar worker leaves required fields blank

1. System throws error.
2. System informs registrar worker of fields left blank.

1c. Registrar worker batch uploads student information through .CSV file

1. System checks to see if file is valid .CSV file.
2. System checks each student to see if student is already registered.

1d. Registrar worker uploads invalid file

1. System throws error.
2. System prompts worker to choose another file.

3a. If already registered system automatically updates student and notifies registrar

worker of successful update.

3b. If new system logs student registration and notifies registrar worker of successful registration.

4a. Registrar worker leaves system on unattended.

1. System disconnects registrar worker after 10 minutes of inactivity.

Special Requirements:

- Useable with various browser and computer types.
- Batch student Registration through .CSV format is available.
- Language internalization on the text displayed.
- Same entry method is used to also update student information.

Technology and Data Variations List:

*a. Registrar worker could use any web browser to connect to system (IE, Firefox, Chrome, Safari, etc).

*b. Registrar worker could upload data in .CSV format or do manual entry.

Frequency of Occurrence: Could be nearly continuous—peak up to three weeks before first day of classes through second week of classes.

Open Issues:

- What are laws on accessibility for the system?
- Should batch uploads prompt for student update or should it update automatically?

~

Use Case (UC6): Add Faculty Member.

Scope: NKU-Net Registration System

Level: user-goal

Primary Actor: Registrar

Stakeholders and Interests:

- Student: Wants access to faculty member during duration of course.
- Faculty: Wants to be able to access course/students before first day of class.
- School: Wants faculty and students to be able to communicate with each other.
- Registrar: Wants to add large number of faculty quickly and easily.
- System Administrator: Wants registrar to add faculty easily so they don't have to.

Preconditions: Registrar worker is identified and authenticated.

Success guarantee (or Postconditions): Faculty member is registered and saved under faculty identifier. Registrar worker is notified of registration success or failure.

Main Success Scenario (or Basic Flow):

1. Registrar worker manually enters new faculty information and submits to system.
2. System checks to see if faculty is already registered.
3. System logs faculty registration and notifies registrar worker of successful registration.
4. Registrar worker voluntarily disconnects from system.

Registrar worker repeats steps 1-3 until finished registering faculty members.

Extensions (or Alternative Flows):

*a. At any time, System fails:

1. To ensure security of system, registrar worker is logged off and connection is shut off.
2. Error message is displayed.

1a. Registrar worker manually adds a faculty member that already exists

1. System informs registrar worker that faculty member already exists.
2. System prompts registrar worker if they want to update faculty information.
3. If Registrar worker says yes then faculty information is updated and system notifies worker.

1b. Registrar worker leaves required fields blank

1. System throws error.
2. System informs registrar worker of fields left blank.

1c. Registrar worker batch uploads faculty information through .CSV file

1. System checks to see if file is valid .CSV file.
2. System checks each faculty member to see if member is already registered.

1d. Registrar worker uploads invalid file

1. System throws error.
2. System prompts worker to choose another file.

3a. If already registred system automatically updates faculty member and notifies registrar worker of successful update.

3b. If new, system logs faculty registration and notifies registrar worker of successful registration.

4a. Registrar worker leaves system on unattended.

1. System disconnects registrar worker after 10 minutes of inactivity.

Special Requirements:

- Useable with various browser and computer types.
- Batch faculty Registration through .CSV format is available.
- Language internalization on the text displayed.

- Same entry method is used to also update faculty information.

Technology and Data Variations List:

- *a. Registrar worker could use any web browser to connect to system (IE, Firefox, Chrome, Safari, etc).
- *b. Registrar worker could upload data in .CSV format or do manual entry.

Frequency of Occurrence: Could be nearly continuous—peak up to three weeks before first day of classes through second week of classes.

Open Issues:

- What are laws on accessibility for the system?
- Should batch uploads prompt for faculty update or should it update automatically?