



# Inception Phase

*Team Super Ghost*

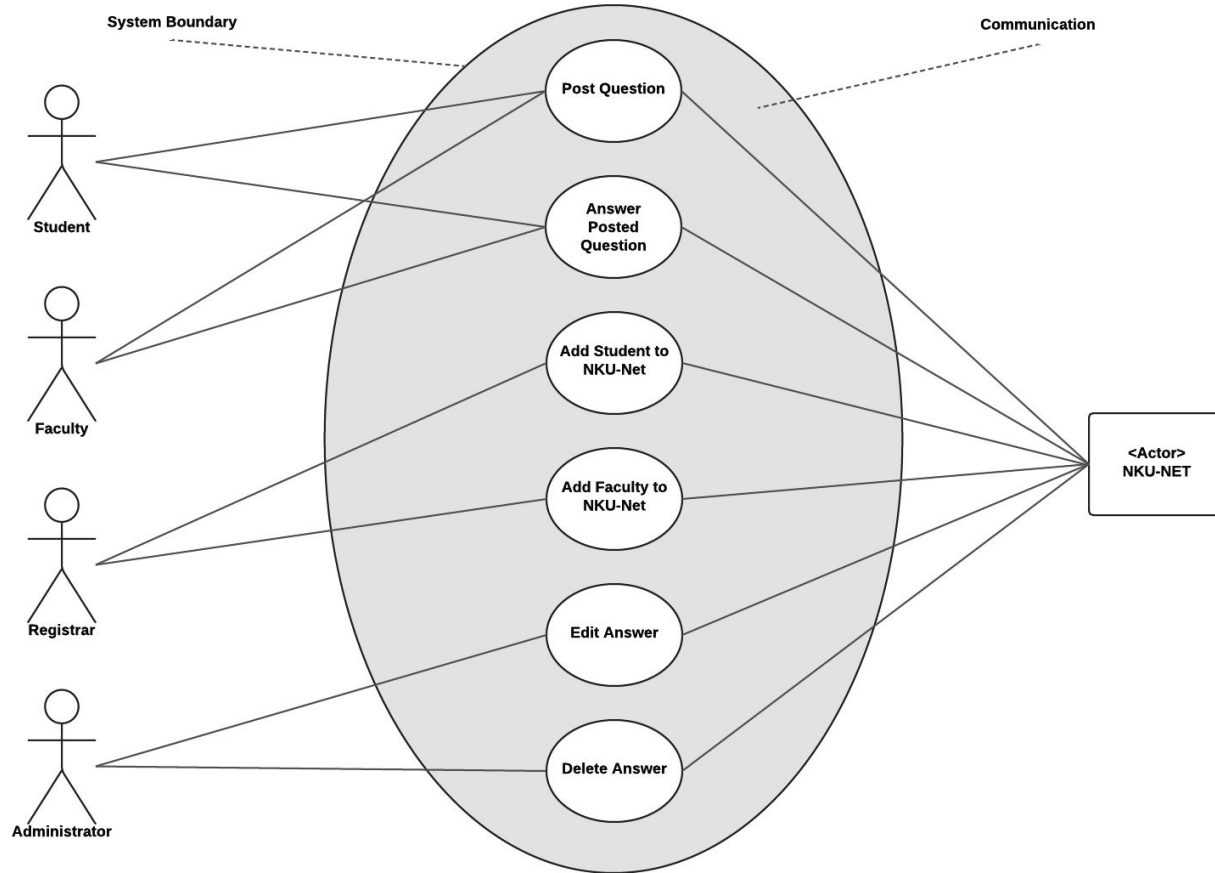
Hudson, Nathaniel • Jansen, Shane • Owens, Kirby • Zumwalde, Zach



# Use Case Diagram

# USE CASE DIAGRAM

Team Super Ghost



# Overall Use Case Design Process

# Overall Use Case Design Process

- We discussed and chose the system boundary (NKU-Net software).
- Identified the primary actors: Student, Faculty, Registrar, System Administrator
- Identified goals for each actor:
  - **Student:** post questions, answer questions, upvote and downvote answers, modify individual profile page, upload documents.
  - **Faculty:** post questions, answer questions, upvote and downvote answers, modify individual profile page, upload documents.
  - **Registrar:** add students to NKU-Net, add faculty members to NKU-Net, add courses to NKU-Net, add student schedules each semester.
  - **Administrator:** Edit/delete questions and answers, add/edit/delete forms to add special users to system, add students to NKU-Net, add faculty members to NKU-Net, add courses to NKU-Net, add student schedules each semester, post questions, answer questions, upvote and downvote answers, modify individual profile page.

# Use Cases

# Add Student

## Design:

- Identified Registrar as the primary actor.
- Problem: How to handle multiple input methods?
- Modeled manual input as the “Basic Flow” and the .CSV input as an alternate flow.

## 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 faculty members.*

# Add Faculty

## Design:

- Identified Registrar as the primary actor.
- Same problem existed in add student to NKU-Net use case.
- Modeled manual input as the “Basic Flow” and the .CSV input as an alternate flow.

## 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.*



# Post Question

## Design:

- Identified Student and Faculty Member as the primary actors.
- Problem: How to handle determining which answers are inappropriate.

## 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.*

# Answers Question

## Design:

- Identified Student and Faculty Member as the primary actors.
- Problem: How to handle determining which answers are inappropriate.

## 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.*

# Edit answer

## Design:

- Identified Administrator as the primary actor
- Problem: How to handle determining which answers are inappropriate.

## 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.
6. The system uploads the edited answer to NKU-Net.

# Delete answer

## Design:

- Identified Administrator as the primary actor.
- Problem: How to determine answers that need to be deleted.

## 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.
5. The system removes the answer from NKU-Net.

# Vision Statement

# Positioning

Product	Strengths	Weaknesses
Blackboard	<ul style="list-style-type: none"><li>• Intuitive Interface</li><li>• Well known by students/faculty</li><li>• Not buggy</li></ul>	<ul style="list-style-type: none"><li>• Design not responsive. Does not work with mobile.</li><li>• Limited discussion post features</li><li>• Lack of personal pages</li><li>• Too many features</li></ul>
Canvas	<ul style="list-style-type: none"><li>• Plenty of discussion board features (especially multimedia options)</li><li>• Good, responsive mobile site coverage</li><li>• Basic profile functionality (can upload files to profile, create “eportfolio” to show work.</li><li>• Has a “commons” area where courses from other universities can share lessons/files.</li></ul>	<ul style="list-style-type: none"><li>• Cluttered/disorganized user interface (too many features all at once).</li><li>• Like system on the discussion boards but not an “upvote/downvote” system.</li><li>• Disorganized discussions. No tags/groups to organize discussions.</li></ul>

# Stakeholder Description

## ***Stakeholder (Non-User) Summary***

- Non-users of this system are members of administration at Northern Kentucky University (NKU).

## ***User Summary***

- Users of this system, NKU-Net, will largely be students at NKU, in addition to faculty members at NKU. Other users will include registrars—who will be tasked with adding courses and students to NKU-Net—and administrators—who will be tasked to monitoring and editing the system.

## ***Stakeholder Summary***

- Student/Faculty Member: post/answer questions, upvote/downvote answers, have individual profile page, upload documents.
- Registrar: add data, students, and courses to NKU-Net.
- Administrator: Edit/delete questions & answers, add/edit/delete forms to add special users to system, in addition to every other action other users can perform.

# Other Requirements and Constraints

- All user passwords should be encrypted.
- The application should function on all major web browsers (Chrome, Firefox, Safari, Internet Explorer).
- The application should function under a heavy load.
- Backups should be taken on a fixed schedule to prevent data loss.
- The application should have high availability as students and faculty could depend on the application during any time of the day.



# Product Overview

Supporting Feature	Stakeholder Benefit
Functionality, the site will have familiar components that will improve the ease of use.	Ease of use.
No sign-up, the registrar will be held responsible for adding new faculty and students. The main user base will not have to sign up for anything.	Users will be more open to using the product.

# Summary of System Features

- An upvote and downvote system of determining best answers to questions
- The ability to look at questions other students have posted for a class and the responses.
- The ability for faculty to answer questions that other students have posted.
- The ability for there to be special users with permissions to add users and classes.
- The ability for there to be special users with permissions to remove unwanted questions or answers.