The Borne Collective: Use Case Model

NKUNet

Use Case UC1: Post Thread
Scope: NKUnet System

Level: user goal

Primary Actor: User
Stakeholders and Interest:
- User: Wants the System to be responsive, receiving the input and displaying the posted question or answer accurately.
- NKU: Wants students and faculty to be able to post questions and answers to share knowledge which will be mutually beneficial to the NKU community.
- System Administrator: Wants NKUNet to behave as designed allowing the user to successfully submit questions and answers to questions.

Preconditions: User has an account and is logged into NKUNet
Postconditions: Question or answer to question is entered. The input is received by NKUNet and saved into the database.
Basic Flow:
1. User arrives in a forum with intent to either ask or answer a question.
2. User enters input into textbox.
3. User submits question/answer.
4. System saves the input into the NKUNet database.

Alternate Flow:
2a. The NKUNet crashes.
    1. User contacts a System administrator to report the issue.
        a. User waits for the for the system to be back up before continuing.
    1a. User decides to use StackOverflow instead.
3a. The page times out while attempting to submit question/answer.
    1. User checks to see if the question/answer was submitted successfully by refreshing the page.
        a. If the question/answer did not submit then the User starts over.
    1a. User contacts a System Administrator to report the issue.
4a. There is a backend database error and the entry is not persisted into the database.
    1. User realizes the issue when they find that their question/answer did not submit.
    1a. User contacts a System Administrator to report the issue.

Use Case UC2: Manage Thread

Scope: NKUnet System

Level: user goal

Primary Actor: System Administrator

Stakeholders and Interest:

- System Administrator: Wants the ability to edit or delete any of the Users posts.
- Users: Wants the ability to manage forums so that NKUNet is free of inappropriate content.
- NKU: Wants NKUNet to be free of inappropriate content.

Preconditions: System Administrator is logged into NKUNet

Postconditions: System Administrator is able to select any question or answer and edit or delete it.

Basic Flow:

1. System Administrator monitors forums for inappropriate content.
2. System Administrator finds and selects question/answer with inappropriate content.
3. System Administrator edits or deletes inappropriate content.

*Repeats step 1.

Alternate Flow:

1a. If no forums exits, then the System Administrator has nothing to monitor.
1b. System Administrator is warned by user of inappropriate content.
   1. Goes to step 3.
2a. System Administrator doesn't find inappropriate content.
   1. Skips step 3.
2b. Website times out when attempting to edit/ delete content.
   1. System Administrator troubleshoots issue.
3a. System Administrator's submission of edit/delete fails.
   1. System Administrator tries again.
         a. System Administrator troubleshoots issue.
   1a. System Administrator fixes issue

Use Case UC3: 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 posts 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 (UC5)
3. User clicks on a thead
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'

Use Case UC4: Manage Users

Scope: NKUnet System

Level: user goal

Primary Actor: Admin

Stakeholders and Interest:

    - Admin: Wants to view and manage user's details and accounts

Preconditions: Admin Authenticated

Postconditions: System displays a list of users where the admin can take further action to view or modify a user's account.

Summary:

Basic Flow:

1. Admin logs into the system
2. From an admin menu, they can select manage users
3. From there a user can be searched for and selected
4. Admin can take various actions to manage user accounts (add, delete, modify, etc.)

Alternate Flow:
> 1a. Admin tries to modify another admin account
>> 1. System rejects the request without proper admin rights

Use Case UC5: Search

Scope: NKUnet System

Level: user goal

Primary Actor: User (Student or Faculty)

Stakeholders and Interest:

- **User:** Wants to find questions previously asked based on subject, user, post title, and/or keyword
- **Admin:** Wants to log user and invalid search strings in case of malicious users

Preconditions: User Authenticated

Postconditions: Search results are generated and all 'Question' threads meeting the search query are listed with a summary and direct link.

Basic Flow:

1. User enters text into the search interface that they want to find relevant question threads on
2. System generates results based on search query
3. User can further sort search results or start another search

Alternate Flow:

1a. User selects 'advance search' extending the search interface to include optional 'subject' , 'user' , 'post title' , and 'keyword' search parameters.

> 1. User selects the optional search parameters they want to narrow the search results by
>> 1a. Invalid user ID
>>> 1. System signals error and rejects entry.
>>> 2. User responds to error:
>>>> 2. User checks that the user ID is correct and tries again.

> 1b. Invalid search string:
>> 1. System logs user and search string, signals error of invalid characters, and rejects entry.

2. 2. User responds to error:

   2a. User reformats search string and ensures the invalid characters are removed before trying again

1c. Blank input error:

   1. System signals error and rejects entry.

Alternate Flow:

Use Case UC6: Manage Schedule
Scope: NKUnet System
Level: user goal
Primary Actor: Registrar
Stakeholders and Interest:
- Registrar: Wants to quickly alter a student's schedule in a transaction-safe manner.
- User: Wants schedule accurately updated to reflect the courses that they are currently enrolled in.
- Admin: Wants to log all changes of users (faculty and student) and their respective schedules

Preconditions: Registrar is authenticated
Postconditions: System updated to reflect submitted user schedule, modifications to user schedule are logged

Basic Flow:
1. Registrar enters user control panel and begins schedule management process
2. Registrar completes and submits user (student or faculty) schedule update web form
3. System altars database to reflect newly updated student schedule and alerts user of success

Alternate Flow:

2a. 'New User' indicated:
   1. System signals confirmation to registrar
   2. Registrar responds to error:
      2a. Registrar responds to confirmation request and accepts
         1. System updates to reflect addition of new user
      2b. Registrar cancels confirmation, enters student's ID and tries again

2b. Invalid university ID:
1. System signals error and rejects entry.
   1a. Registrar responds to error:
      1. Registrar ensures user's university ID is correct and tries again.

2c. Invalid Course ID:
1. System signals error and rejects entry.
2. Registrar responds to error:
   2a. Registrar ensures Course ID is that of an active course and tries again.
   2b. Registrar determines Course ID to be inaccurate, and contacts student to resolve issue

2d. Database error:
1. Registrar submits ticket to system administrator regarding failure and tries again later.