6.170 Project 3: Crowd-approved Comments

PLAZA, Eddie Francis (efcplaza)

nQoot: http://nqoot.francis.ph

Problem Analysis

Purpose and Goals

The goal of the system is to provide a crowd-sourced knowledge base platform (Quora-like) for internet users to post questions ("question") and answer existing posts and/or questions ("answer").

The basic implementation of this system will allow users to create and have several questions and be able to answer questions. Users will also be able to upvote answers. A specially privileged user called a 'staff' will be able to endorse answers to questions. All endorsed answers will appear highlighted in the website.

Unusual Requirements

One of the most essential requirements of this system is auto-updating of the page as answers (or updates to answers i.e. upvotes and endorsements) come in. In this case, users do not need to refresh the page every time in order to see new answers. It is essential for the page to get or receive new information from the server. That is, the implementation approach can have a page that constantly queries the server for new information (i.e. answers, upvotes, endorsements) or the server has an open channel between all its clients and send new information as they come in. The latter approach might be a little harder to implement.

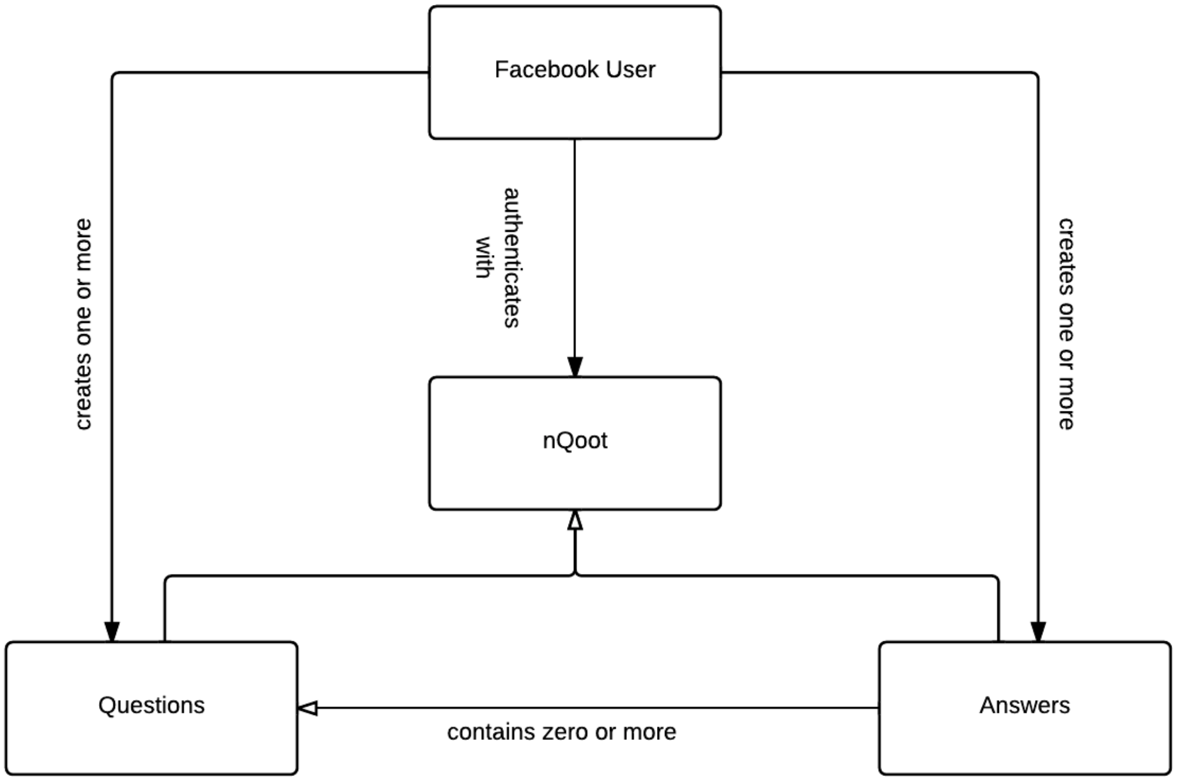
Another important aspect of this crowd-sourced knowledge base platform is user identity. How do we ensure the integrity of the answers from random users? In order to at least increase our level of confidence that the users are who they say they are, users need Facebook accounts in order to login to this system.

Key Features

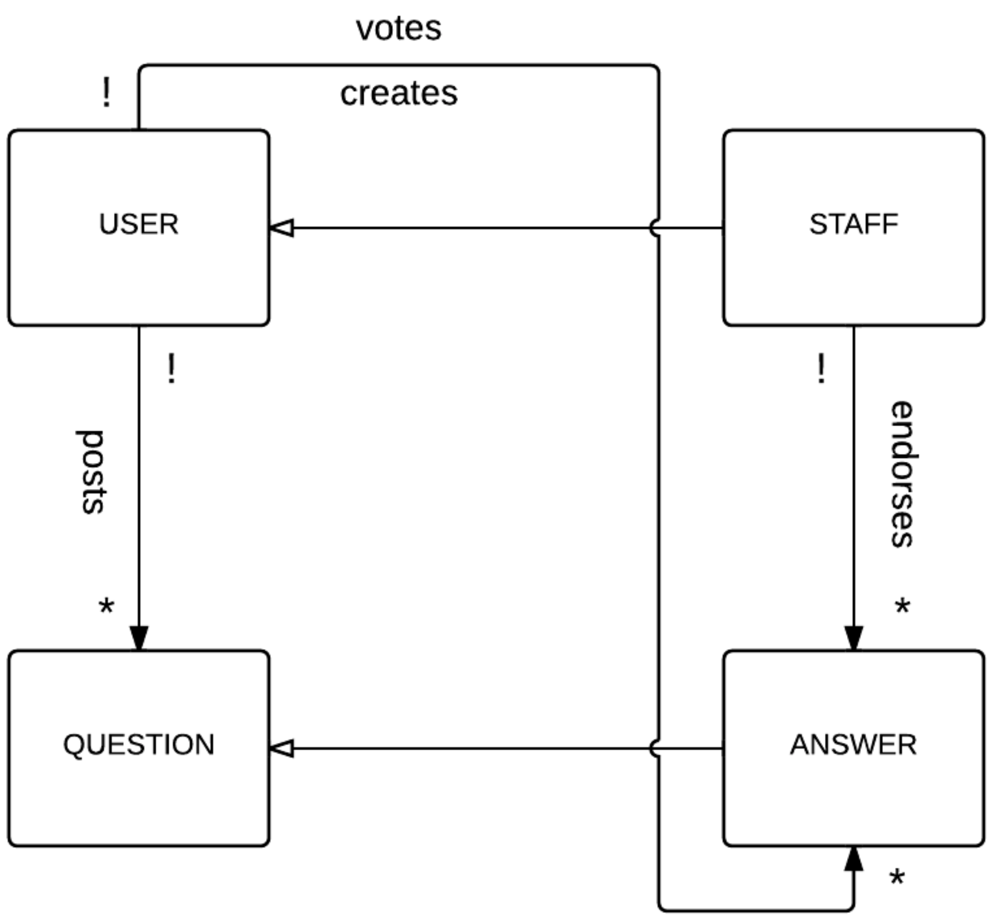
The crowd-approved comment system has the following features:

- Allow users to create questions and add answers to posts as well as existing comments
- Upvote answers to existing questions.
- For the advanced version, allow staff to endorse answers to questions.

Context Diagram



Object Model



Event Model

User events

- **add_question:** user creates new question
- **edit_question:** user edits a question that belongs to it
- **delete_question:** user removes question that belongs to it
- **up_vote:** user upvotes an existing answer to a question
- **add_answer:** user adds answer to an existing question

For the advanced version, a staff user exists. The staff inherits all user events.

Staff events

- **endorse_answer:** staff endorses an existing answer
- **delete_question:** staff deletes any question
- **delete_answer:** staff deletes any answer

Standard scenario

- user_login
- add_question
- edit_question
- add_answer
- up_vote
- user_logout

Orderings

- User ::= add_question edit_question* delete_question* add_answet* up_vote*
- Staff ::= endorse_answer* delete_question* delete_answer*

Feature descriptions

- Allow users to add, edit, and delete questions as well as answer on existing questions.
- Enable staff to monitor questions including endorsing and moderating answers to questions.

Security concerns

- While there isn't a huge security risk in this platform or system, it is necessary to ensure the credibility of the user. That is, users can only upvote answer that doesn't belong to it. It must not also have access to staff privileges. It follows that only staff accounts can only have access to staff privileges.
- Ensuring the credibility of the identity of the user can be challenging. Different systems implement their own checking mechanism. In this system, each user is required to have a Facebook account to login. While this is not foolproof, it increases our confidence that the identity of the user is credible.