Students Message Board
Software Requirement Specification Document
Team 38

ANIRUDH SRINIVASAN
CS20BTECH11059

MADDILETIGARI MADHU DHEERAJ
ES19BTECH11002

LELLA MANOJ KUMAR REDDY
CS19BTECH11041

GODA SUBHADEEP
CS19BTECH11018

22nd February 2023

Contents

1 Introduction

1.1 Purpose

1. Student Message Board (SMB) is intended to help the users post questions and get it clarified within the community
2. This document is meant to delineate the features of SMB, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

1.2 Scope

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

1.2.1 In Scope

1. User Authentication
2. Interface for the user to post the question and answer the questions posted
3. Tag system for the questions to categorize questions into different sections
4. Search Option to look out for similar questions posted in the past
5. Points system for the active people in the community and special privilege to the faculty users
6. Upvotes and Downvotes for both the questions and answers
7. Comment option for both the questions and answers
8. Accepting an answer (by user) and giving bounties

1.2.2 Out of Scope

1. Chat option between different users
2. Auto-answering the posted questions using an NLP Model that can generate the answer from previous accepted answers and based on information retrieval on the documentation provided to it
3. Automated tagging based on a classification model to classify questions into different categories based on the previous set of tags
4. Forming teams inside the SMB community and asking relevant questions in the sub-community created
5. Integrating multiple languages in the Q/A and make it more broader in terms of the use cases

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Acronyms, and Abbreviations

1. SMB: Student Message Board
2. SRS: Software Requirements Specification
3. UI: User Interface
4. GUI: Graphical User Interface
5. IITH: Indian Institute of Technology, Hyderabad

1.3.2 Definitions

1. Thread: The collection of messages grouped together in a meaningful conversation, with a title, listed in a category, beginning with an Original Post, and including all replies in chronological order.
2. Post: Each element of a thread is referred to as a post. Each post has its own author and can be moved to a different thread if necessary, or even become the start of a new thread.
3. Original Post (or OP): The first post in any thread. This is key since it determines the focus of the thread and is what the title and tags will link to.
4. Category: The primary means of organizing threads. Each thread is placed in exactly one category. Categories have permissions which can restrict which users can create, reply, and see its threads.
5. Tag: A marker placed on a topic to describe it. While a topic can only have one category, it can have multiple tags. New tags can be created by some users.
6. User: Only IITH students, faculty and staff will be allowed to access the sites. You will always be required to register in order to create posts and record any actions such as bookmarking, liking, tagging, and flagging. All the site activity, therefore, is based around registered users.
7. Bounties: The special rewards given by people with high reputation - moderators or faculties for very well-framed answers.
8. Reputation: The amount of points of an user which is obtained through good answers, good questions and good comments based on the number of upvotes and bounties.

1.4 References

1. Stack Overflow: Stack Overflow is a question and answer website for programmers. It is the flagship site of the Stack Exchange Network. It was created in 2008 by Jeff Atwood and Joel Spolsky. It features questions and answers on certain computer programming topics.

1.5 Overview

The rest of this SRS is organized as follows:

1. Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, somegeneral constraints while making the software and some assumptions and dependencies that are assumed.
2. Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.
3. Section 4 gives some possible future extensions of the system.
4. Finally the appendices in Section 5 resolves few doubts and take that up for clarification.

2 Overall Description

2.1 Product Perspective

1. SMB is aimed towards the entire community to post and resolve others' doubts regarding various topics like Academics, Hostels, Campus, Mess, Canteen, Food Court, Clubs, Gymkhana, Sports, and many more.
2. SMB should be user-friendly, 'quick to learn' and reliable software for the above purpose.
3. SMB is intended to be a stand-alone product and should not depend on the availability of other software. It will be a web application and hence is portable across multiple platforms.

2.2 Product Functions

SMB should support the following use cases:

1. Use cases related to System Authorization
   - (a) Login using Google Account
   - (b) Signup and Create an Account
   - (c) Login using existing Account

   (d) Change Password for the Account

2. Use cases related to asking a question
   (a) Post a question
   (b) Edit a question
   (c) Accepting an answer

3. Use cases related to answering a question
   (a) Answer the question
   (b) Edit the answer

4. Use cases related to commenting
   (a) Comment on the question
   (b) Edit a Comment on the question
   (c) Comment on the answer
   (d) Edit a Comment on the answer

5. Use cases related to upvotes, downvotes and bounties
   (a) Upvote or Downvote the question
   (b) Upvote or Downvote the answer
   (c) Upvote or Downvote the comment
   (d) Bounty a question
   (e) Bounty an answer

6. Use cases related to searching
   (a) Searching the question based on title
   (b) Searching the question based on tags
   (c) Searching a profile

7. Use cases related to flagging
   (a) Flagging a question
   (b) Flagging an answer
   (c) Flagging a comment

2.3 User Characteristics

1. The user should be either student/faculty/staff of the IITH Community

2. The UI will be self-explanatory, but it would be preferred if the user reads the manual and help pages.

2.4 Principal Actors

The principal actors of SMB are 'user' and 'system'. The types of users can be categorized as follows:

2.4.1 Normal Users

1. Student
2. Faculty
3. Staff

2.4.2 Privileged Users

1. Moderator
2. Admin

2.5 General Constraints

1. SMB requires Internet connection for its complete working
2. SMB is multi-user software
3. The environment should be such that we can run JavaScript files

3 Specific Requirements

3.1 Functional Requirements

We describe the functional requirements by giving various use cases

3.1.1 Use cases related to System Authorization

1. **Use Case 1: Login using Google Account**
   - Primary Actor: User
   - Pre Condition: Nil
   - Main Scenario:
     (a) Start the application. User prompted for login using Google account
     (b) User selects the account and gives password.
     (c) System does authentication.
     (d) Main screen is displayed.
   - Remarks: Signing in using Google Account, we can make sure only IITH accounts are able to access our SMB.

2. **Use Case 2: Signup and Create an Account**
   - Primary Actor: User
   - Pre Condition: Nil
   - Main Scenario:
     (a) Start the application.
     (b) User clicks sign-up
     (c) User enters information like name, role in the organisation, institute ID number, proof of belongingness to IITH community as a student/faculty/staff, contact details, emaid ID, password, etc.
     (d) System creates an account and authorizes the new account
     (e) Main screen is displayed.
   - Alternate Scenario : Mandatory Fields not filled
     (a) Prompt the user that all the mandatory fields haven't been filled
     (b) Allow the user to re-enter them

3. **Use Case 3: Login using existing Account**
   - Primary Actor: User
   - Pre Condition: Nil
   - Main Scenario:
     (a) Start the application. User prompted for login credentials
     (b) User enters username and gives password.
     (c) System does authentication.
     (d) Main screen is displayed.
   - Alternate Scenario : Authorization fails
     (a) Prompt the user that he typed the wrong password
     (b) Allow the user to re-enter the password. Give the user 3 chances

4. **Use Case 4: Change Password**
   - Primary Actor: User
   - Pre Condition: User logged in
   - Main Scenario:
     (a) User initiates the password change command.
     (b) User is prompted for old password, new password and confirm new password.
     (c) User gives the old password, new password and confirm new password.
     (d) System does authentication.

    (e) New password is registered with the system.

- Alternate Scenario :
    (a) Authorization fails
        i. Prompt the user that a wrong password was typed
        ii. Allow the user to re-enter the password. Give 3 chances.
    (b) New password and confirm new password do not match.
        i. Allow the user to re-enter the attributes. Give 3 chances.

3.1.2 Use cases related to asking a question

1. **Use Case 5: Post a question**

    - Primary Actor: User
    - Pre Condition: User logged in
    - Main Scenario:
        (a) User clicks ask a question button
        (b) User is prompted to enter the specifics of the question including title, description, (an optional field will be present for file upload for images or code snippets)
        (c) The description should ideally include the attempts in solving the problem (if applicable)
        (d) User has to select the mandatory tag (called category in the SMB terminologies), and then he/she can choose infinite number of optional tags.
        (e) While the user can select from the existing tags, he/she can choose to create new tags (if neccessary). The new tags can be created while posting the question itself and it will be automatically added to the list of tags by the system
        (f) The question is posted with the date and time of the asking
    - Alternate Scenario : Title or Description is empty or Category isn't selected
        (a) Question won't be posted
            i. Prompt the user regarding the empty fields
            ii. Allow the user to re-enter the above details
            iii. If the fields are filled, then the system posts the question publicly

2. **Use Case 6: Edit a question**

    - Primary Actor: User who asked the question / Priveleged users
    - Pre Condition: User logged in and has asked a question
    - Main Scenario:

    (a) User clicks asked questions tab in his/her profile
    (b) User can see the option to edit the question
    (c) On clicking the edit question button, each of the individual fields become editable
    (d) The user can save changes once done. The moderator reviews the edit and approves it.
    (e) The system updates the question posted with the timing of the edit

- Alternate Scenario : Title or Description is empty or Category isn't selected
    (a) Question won't be posted
        i. Prompt the user regarding the empty fields
        ii. Allow the user to re-enter the above details
        iii. If the fields are filled, then the system posts the question publicly

- Remarks: The people who have answered the question before the edit will be notified about the change in the question so that they can verify whether the answer still holds true based on the contextual change.

3. **Use Case 7: Accepting an answer**

    - Primary Actor: User who asked the question
    - Pre Condition: User logged in and has asked a question
    - Main Scenario:
        (a) User clicks asked questions tab in his/her profile
        (b) User can see all the answers in the same thread
        (c) The user can accept the appropriate answer that best answers his/her question
        (d) The system will mark that answer with a tick mark and that will be the first answer displayed when the question is searched publicly
        (e) The user can choose to accept further answers or not even after accepting an answer
    - Remarks: There is no need for the user to accept an answer unless he is satisfied. But the moderator has special privilege to close the question from further answering if he/she feels the need to.

3.1.3 Use cases related to answering a question

1. **Use Case 8: Answer the question**

    - Primary Actor: User

- Pre Condition: User logged in and clicks to answer a question that has been already posted
- Main Scenario:
  (a) User clicks answer the question button
  (b) User can add a detailed description of answer. There will be provision to add links and files (again an optional field, the one answering can choose to use it at his/her's own convenience)
  (c) Once he saves the changes, the answer will be posted with date and time of answering
- Alternate Scenario : Answer is empty
  (a) Answer won't be posted
    i. Prompt the user regarding the empty fields
    ii. Allow the user to re-enter the above details
    iii. If the fields are filled, then the system posts the answer publicly
- Remarks: An user can answer his/her's own question. This will be marked as the accepted answer by default, but the user gets no reputation points for this. This can be used as a medium to report his findings of his/her research regarding the question.

2. **Use Case 9: Edit the answer**
   - Primary Actor: User who has already posted the answer / Moderator
   - Pre Condition: User logged in and has answered a question
   - Main Scenario:
     (a) User can navigate to the answer he wants to edit from the answered questions in his profile
     (b) User clicks edit the answer button
     (c) User can edit the previously posted answer. He can upload new files as well (optional)
     (d) Once he saves the changes, the answer will be posted. The thread will show the answer was edited with date and time of answering.
   - Alternate Scenario : Answer is empty
     (a) Answer won't be posted
       i. Prompt the user regarding the empty fields
       ii. Allow the user to re-enter the above details
       iii. If the fields are filled, then the system posts the answer publicly

3.1.4 Use cases related to commenting

1. **Use Case 10: Comment on the question**

    - Primary Actor: User
    - Pre Condition: User logged in and previously posted question
    - Main Scenario:
        (a) User can see comment option below the question
        (b) User can enter his comment related to the question
        (c) Once he saves the changes, the comment will be posted with date and time
    - Alternate Scenario : Comment is empty
        (a) Comment won't be posted
            i. Prompt the user regarding the empty fields
            ii. Allow the user to re-enter the above details
            iii. If the fields are filled, then the system posts the comment publicly
    - Remarks: The user who posted the question can answer to queries related to the question in the comments section. Comments should ideally be only for clarification purposes, long comments should be voiced as answers unless it is still clarifications related to the question

2. **Use Case 11: Edit a Comment on the question**

    - Primary Actor: User
    - Pre Condition: User logged in and commented on a previously posted question
    - Main Scenario:
        (a) User can see edit comment option in the comment he had previously written
        (b) User can edit the comment
        (c) Once he saves the changes, the comment will be posted with date and time
    - Alternate Scenario : Comment is empty
        (a) Comment won't be posted
            i. Prompt the user regarding the empty fields
            ii. Allow the user to re-enter the above details
            iii. If the fields are filled, then the system posts the comment publicly

3. **Use Case 12: Comment on the Answer**

    - Primary Actor: User

- Pre Condition: User logged in and a previously posted answer
- Main Scenario:
  - (a) User can see comment option below the answer
  - (b) User can enter his comment related to the answer
  - (c) Once he saves the changes, the comment will be posted with date and time
- Alternate Scenario : Comment is empty
  - (a) Comment won't be posted
    - i. Prompt the user regarding the empty fields
    - ii. Allow the user to re-enter the above details
    - iii. If the fields are filled, then the system posts the comment publicly

4. **Use Case 13: Edit a Comment on the Answer**
   - Primary Actor: User
   - Pre Condition: User logged in and commented on a previously posted answer
   - Main Scenario:
     - (a) User can see edit comment option in the comment he had previously written
     - (b) User can edit the comment
     - (c) Once he saves the changes, the comment will be posted with date and time
   - Alternate Scenario : Comment is empty
     - (a) Comment won't be posted
       - i. Prompt the user regarding the empty fields
       - ii. Allow the user to re-enter the above details
       - iii. If the fields are filled, then the system posts the comment publicly

3.1.5 Use cases related to upvotes, downvotes and bounties

1. **Use Case 14: Upvote or Downvote the question**
   - Primary Actor: User
   - Pre Condition: User logged in and a previously posted question
   - Main Scenario:
     - (a) User can see upvote and downvote buttons next to the question
     - (b) He/She can upvote the question if the question is noteworthy while they can downvote it if it is ambiguous or lame.

(c) Once the the button is clicked, the effective number of upvotes are shown (The number can be negative if the number of downvotes are more than the number of upvotes)

2. **Use Case 15: Upvote or Downvote the answer**
   - Primary Actor: User
   - Pre Condition: User logged in and a previously posted answer
   - Main Scenario:
     (a) User can see upvote and downvote buttons next to the answer
     (b) He/She can upvote the answer if the answer is noteworthy while they can downvote it if it is ambiguous or irrelevant or wrong.
     (c) Once the the button is clicked, the effective number of upvotes are shown (The number can be negative if the number of downvotes are more than the number of upvotes)

3. **Use Case 16: Upvote or Downvote the comment**
   - Primary Actor: User
   - Pre Condition: User logged in and a previously posted comment
   - Main Scenario:
     (a) User can see upvote and downvote buttons next to the comment
     (b) He/She can upvote the comment if the comment is noteworthy while they can downvote it if it is ambiguous or irrelevant or wrong.
     (c) Once the the button is clicked, the effective number of upvotes are shown (The number can be negative if the number of downvotes are more than the number of upvotes)

4. **Use Case 17: Bounty a question**
   - Primary Actor: Moderator or Faculty
   - Pre Condition: User logged in and a previously posted question
   - Main Scenario:
     (a) The user has the discretionary power to bounty a question
     (b) It will be a special privelege visible to only these users. To the side of upvote, they can see bounty the question option
   - This encourages good questions to be posted in the SMB and promotes a healthy competition between those who are posting the questions

5. **Use Case 18: Bounty an answer**
   - Primary Actor: Moderator or Faculty

- Pre Condition: User logged in and a previously posted answer
- Main Scenario:
  (a) The user has the discretionary power to bounty an answer
  (b) It will be a special privelege visible to only these users. To the side of upvote, they can see bounty the answer option
- This encourages good answers to be posted in the SMB and promotes a healthy competition between those who are posting the answers

3.1.6 Use cases related to searching

1. **Use Case 19: Searching the question based on title**
   - Primary Actor: User
   - Pre Condition: User logged in and a previously posted question
   - Main Scenario:
     (a) User can search the question posted in the community using the search button
     (b) Relevant search entries will appear while searching, and the user can navigate to the question of his/her interest
2. **Use Case 20: Searching the questions based on tags**
   - Primary Actor: User
   - Pre Condition: User logged in and pre-existing tag
   - Main Scenario:
     (a) User can choose the tag of interest
     (b) All the questions related to the tag will be displayed
     (c) The user can then navigate to the question of interest from the entire list
   - Alternate Scenario: Tag doesn't exist. An error message will be displayed for the same
   - Remarks: This can be used by the user to check if a similar question has been posted and addressed previously. Will be really helpful to prevent duplicate questions. The moderators however periodically look into this, and redirect duplicate questions to the OP.
3. **Use Case 21: Searching a profile**
   - Primary Actor: User
   - Pre Condition: User logged in and a pre-existing user
   - Main Scenario:
     (a) User can type the username of the user to be searched

   (b) All vital information including when the user joined the community will be present
   (c) The reputation points, all the questions asked and all the answers posted by the user will be visible publicly
   (d) The user can then navigate to the question or answer of interest from the entire list

3.1.7 Use cases related to flagging

1. **Use Case 22: Flagging a question**

   - Primary Actor: User
   - Pre Condition: User logged in and a previously posted question
   - Main Scenario:

     (a) A user can flag a question inappropriate if they don't follow the community restrictions
     (b) The moderator can assess the flag and can delete the question

2. **Use Case 23: Flagging an answer**

   - Primary Actor: User
   - Pre Condition: User logged in and a previously posted answer
   - Main Scenario:

     (a) A user can flag an answer inappropriate if they don't follow the community restrictions
     (b) The moderator can assess the flag and can delete the answer

3. **Use Case 24: Flagging a comment**

   - Primary Actor: User
   - Pre Condition: User logged in and a previously posted comment
   - Main Scenario:

     (a) A user can flag a comment inappropriate if they don't follow the community restrictions
     (b) The moderator can assess the flag and can delete the comment

   Remarks: Any flagging that's approved, leads to a black mark in the profile and can be seen publicly

3.2 Performance Requirements

1. Should run on 500 MHz, 64 MB machine.

2. 90% of the responses should be within 2 sec, except for corner cases like deleting an account and then seeing the questions posted by them. More time is acceptable in such cases.

3.3 Design Constraints

1. Security: The files in which the information regarding securities and portfolios should be secured against malicious deformations.

2. Fault Tolerance: Data should not become corrupted in case of system crash or power failure.

4 Future Extensions

1. Chat option between different users

2. Auto-answering the posted questions using an NLP Model that can generate the answer from previous accepted answers and based on information retrieval on the documentation provided to it

3. Automated tagging based on a classification model to classify questions into different categories based on the previous set of tags

4. Forming teams inside the SMB community and asking relevant questions in the sub-community created

5. Integrating multiple languages in the Q/A and make it more broader in terms of the use cases

6. Tag Request specific to an user to be handled by the moderators and the request to be sent in the form of notifications

7. Anonymous questions: The identity of the user who asked the question won't be revealed, but the user will be able to see the questions he/she had asked in his questions asked section

8. Duplicate check for the new questions to be automatically done by a similarity search: Machine Learning Model.

5 Appendix

5.1 Clarifications

5.1.1 Categories vs Tags

1. Category is the mandatory tag that has trivial classes that the questions go into. It can be ranging from Academics, Sports, Culturals, Clubs, etc.,

2. Tag can be system defined or user-defined. User defined tags can be more specific and can give more intuitive understanding over what the entire thread is about

3. All categories are tags but all tags need not be categories

5.1.2 What special priveleges do faculty get?

1. The faculties start with a higher reputation in the community. Their base number of points is 10000 points

2. Their answer will most probably be in the top as the answers will be ordered in terms of upvotes and reputation of the answerer

3. There will be a special marking on top of the answer to indicate the answer was written by an expert faculty user

5.2 Point System for the Users

1. The reputation field in the user's profile gives an indication of how trustworthy the person is, with respect to the community.

2. A user's reputation can increase if he asks a good question, posts a good answer or comment, based on the number of upvotes, downvotes and bounties

3. Upvotes or downvotes can be done by any logged-in user for any question, answer and comment.

4. Upvotes:
   (a) Each upvote on a question gives +10 points
   (b) Each upvote on an answer gives +20 points
   (c) Each upvote on a comment gives +1 point

5. Downvotes:
   (a) Each downvote on a question gives -2 points
   (b) Each downvote on an answer gives -4 points
   (c) Each downvote on a comment gives -0.2 point

6. Bounties:
   (a) Each bounty on a question gives +100 points
   (b) Each bounty on an answer gives +200 points

   Remarks: Bounties can be given to only questions or answers and not for comments.

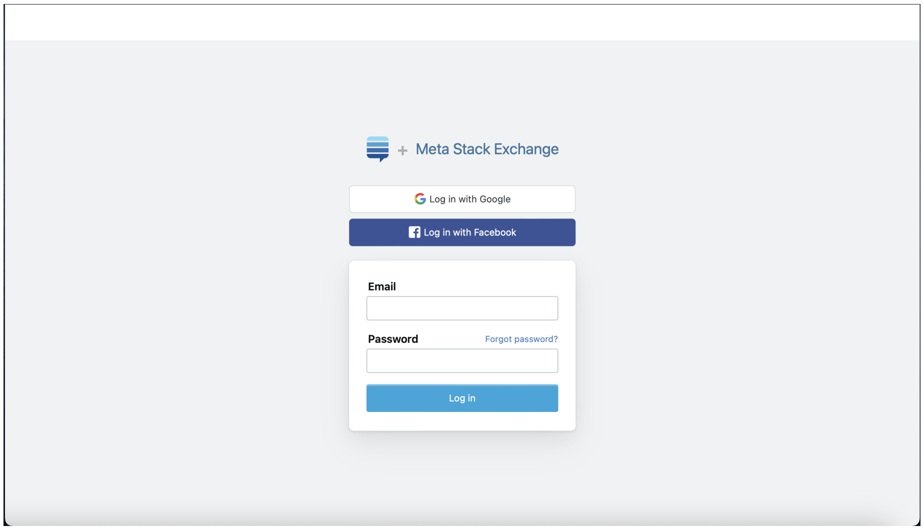
5.3 User Interface

5.3.1 Log in page

1. Reference of Stack Exchange website has been taken in order for us to proceed.

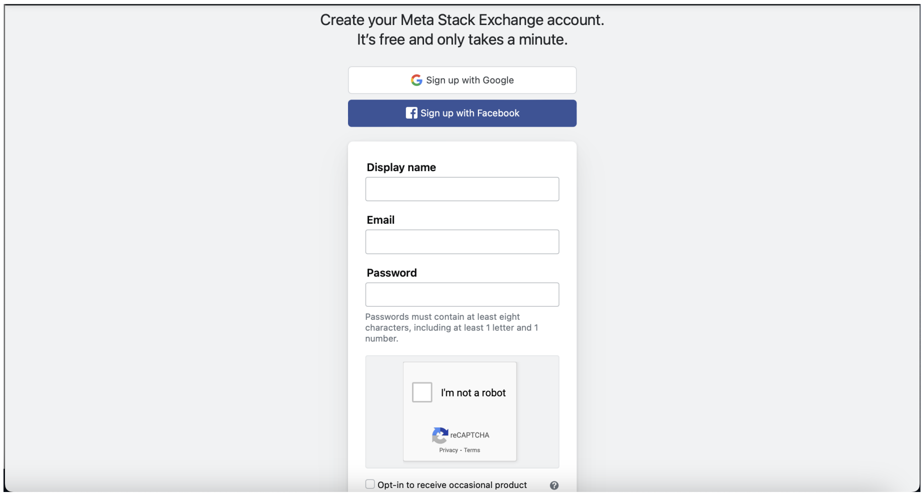
2. The user sees a log in page where he can log in with the google account or via other email options.

3. If the user forgets the password then he/she can reset the password using the registered mail.



5.3.2 Sign up page

1. If the user is trying to log in for the first time then the user have sign up by entering his/her details as shown in the example figure [There are some other categories that can be added in the future extensions].


Create your Meta Stack Exchange account.
It's free and only takes a minute.

Display name

Email

Password

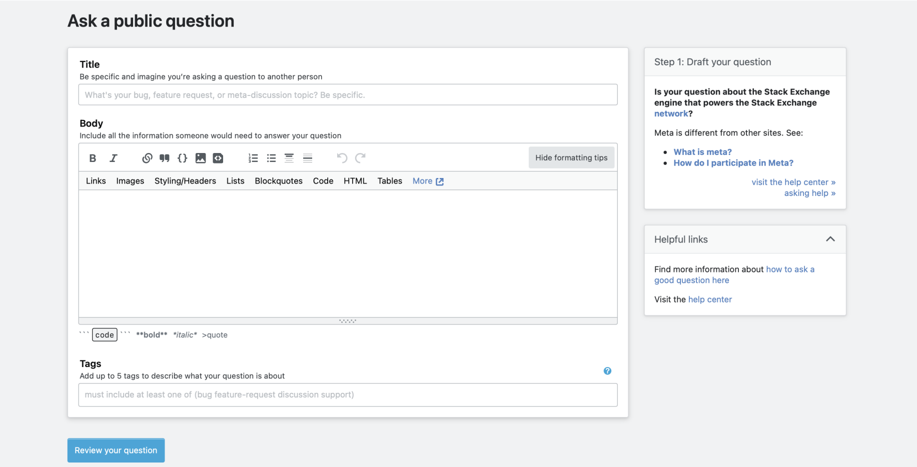
Passwords must contain at least eight characters, including at least 1 letter and 1 number.

□ I'm not a robot

□ Opt-in to receive occasional product

5.3.3 Posting a Question

1. After the user logs in, if the user has to ask a question, then he/she can describe about the question if needed with an attached file[image or document]/can insert a link as shown in the referencial image.


Ask a public question

Title

Be specific and imagine you're asking a question to another person

What's your bug, feature request, or meta-discussion topic? Be specific.

Body

Include all the information someone would need to answer your question

Tags

Add up to 5 tags to describe what your question is about

must include at least one of (bug feature-request discussion support)

Step 1: Draft your question

Is your question about the Stack Exchange engine that powers the Stack Exchange network?

Meta is different from other sites. See:

- **What is meta?**
- **How do I participate in Meta?**

visit the help center »
asking help »

Helpful links

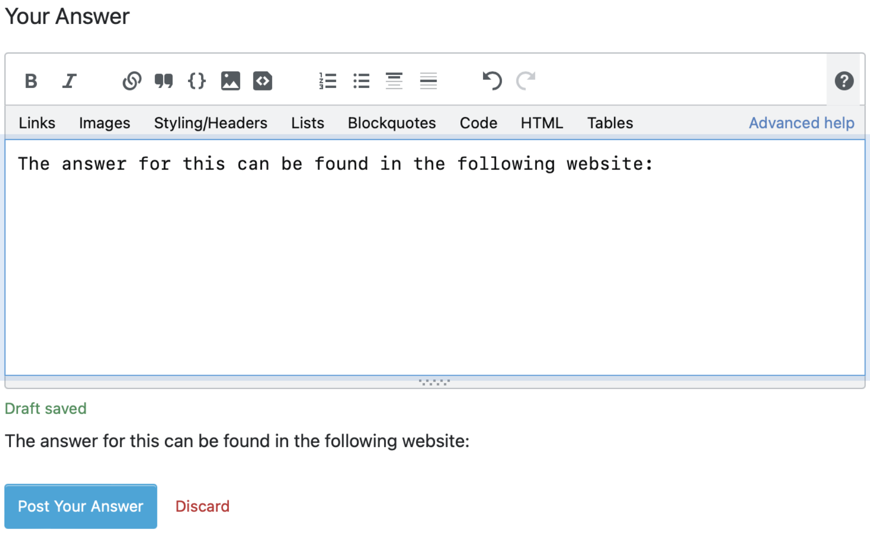
Find more information about how to ask a good question here

Visit the help center

5.3.4 Posting an Answer

1. A user can answer to a posted question as shown in the referential image using an image or a document etc.

2. If the question is modified by the questioner then the user is notified.



```
The answer for this can be found in the following website:
```

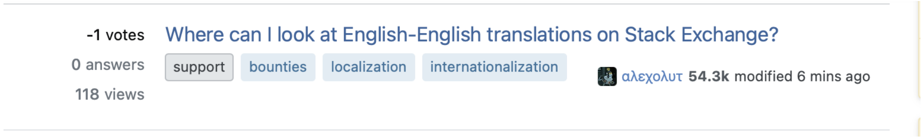
Draft saved

The answer for this can be found in the following website:

5.3.5 View a Question

1. Any posted question has two features namely upvote and downvote as shown in the referential image.

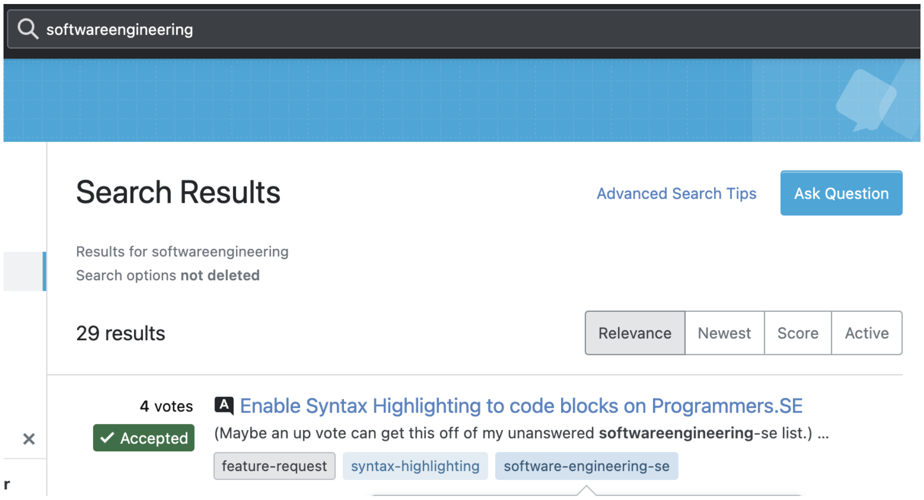
2. We are not going to add the total views feature in this final project but it can be added as a future extension.


-1 votes
0 answers
118 views

Where can I look at English-English translations on Stack Exchange?

5.3.6 Search by title

1. Any user can search for any question using the title of that question as shown in the image.



softwareengineering

Search Results

Advanced Search Tips | Ask Question

Results for softwareengineering
Search options **not deleted**

29 results

Relevance | Newest | Score | Active

4 votes ✓ Accepted

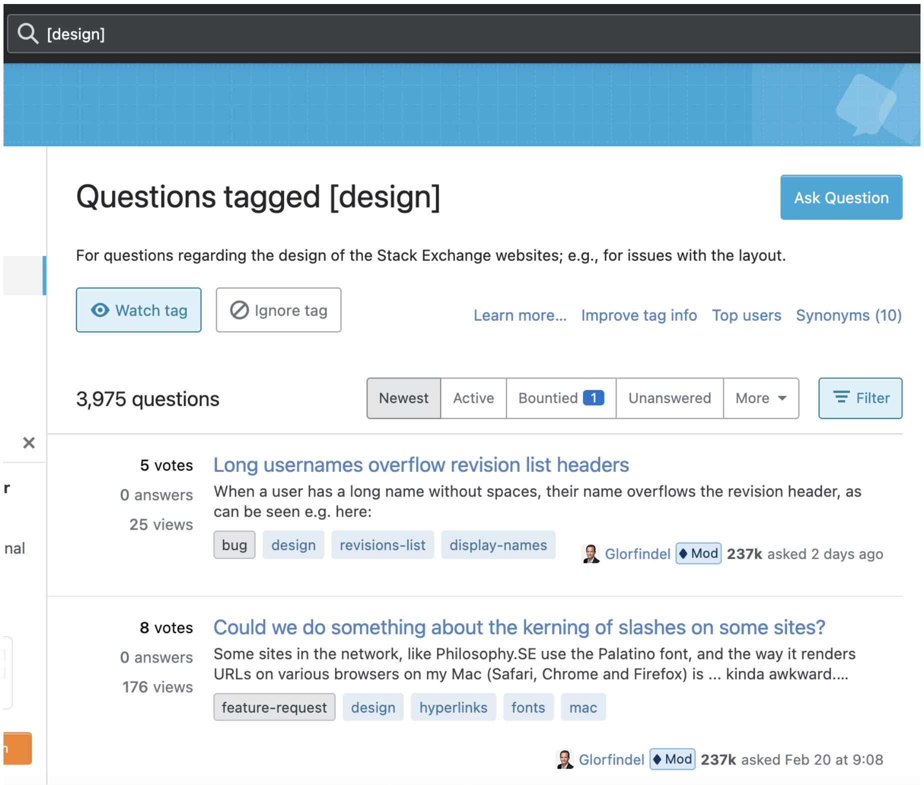
Enable Syntax Highlighting to code blocks on Programmers.SE

(Maybe an up vote can get this off of my unanswered **softwareengineering**-se list.) ...

feature-request | syntax-highlighting | software-engineering-se

5.3.7 Search by tag

1. Any user can search for any question using one of the tags of that question as shown in the image.



[design]

Questions tagged [design]

Ask Question

For questions regarding the design of the Stack Exchange websites; e.g., for issues with the layout.

Watch tag | Ignore tag

Learn more... | Improve tag info | Top users | Synonyms (10)

3,975 questions

Newest | Active | Bountied 1 | Unanswered | More | Filter

5 votes, 0 answers, 25 views

Long usernames overflow revision list headers

When a user has a long name without spaces, their name overflows the revision header, as can be seen e.g. here:

bug | design | revisions-list | display-names

Glorfindel ♦ Mod **237k** asked 2 days ago

8 votes, 0 answers, 176 views

Could we do something about the kerning of slashes on some sites?

Some sites in the network, like Philosophy.SE use the Palatino font, and the way it renders URLs on various browsers on my Mac (Safari, Chrome and Firefox) is ... kinda awkward....

feature-request | design | hyperlinks | fonts | mac

Glorfindel ♦ Mod **237k** asked Feb 20 at 9:08