**Software Requirements Specification (SRS) Document**

**Project Name:** Question Bank Service

**Team’s Name:** Team-21

**Members in the team: Divyarajsinh Mahida (202201085)**

**Ansh Chablani (2022111031)**

**Shreyas Badami (2021113003)**

**Siddharth Singh (2021113001)**

# **Brief problem statement**

Design and develop a web application that allows users to add, manage, and view questions, sorted by tags, which can also be accessed via an API, to facilitate efficient knowledge sharing and learning within a community. Additionally, implement a credit-based system to incentivize users to add new and higher quality questions.

# **System requirements**

The following are needed to run the software:

1. React
2. @mui
3. Ajv
4. @emotion
5. @testing-library
6. Web-vitals
7. React-dom.
8. React-helmet.
9. React-markdown.
10. React-markdown-editor-lite.
11. React-router-dom.
12. React-scripts.
13. Firebase
14. Google storage
15. Selenium
16. Pytest
17. Express JS
18. Node JS

# 

# 

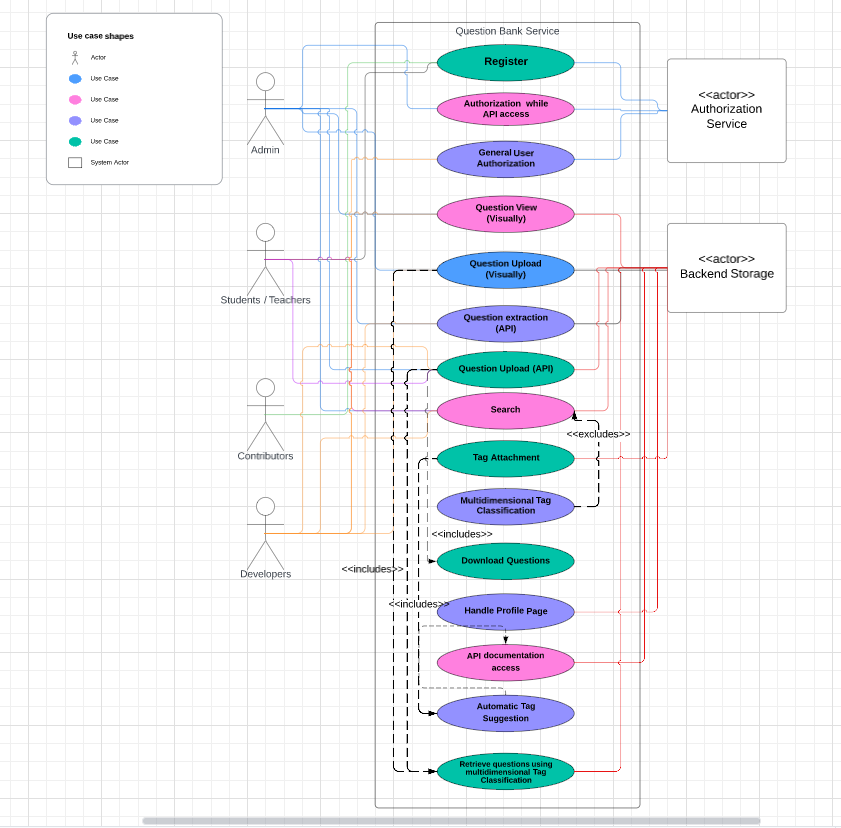
# **Users profile**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **User** | **Purpose** | **Mode** | **Profile** |
| **1.** | **Admin** | **Admin is responsible for overseeing the work of question bank. Admin has permissions to CRUD any questions in the bank.This is done only using API calls** | **Admin must be a System Administrator. Admin must be authenticated by two-factor authentication** | **Admin must be familiar with Schema of questions and must have knowledge on handling questions which are invalid/plagiarized. Admin must have complete knowledge of question validation, where and how they are stored and how are they displayed to users** |
| **2.** | **Contributors** | **Contributors are the specific/permissible users who add new questions to database** | **Contributors must operate on the Add-Question Screen of the Website. Contributors must be authenticated** | **Contributors must be familiar with the steps of adding a question. If they wish to add images/Markdown/Latex to questions, they must have knowledge about them/must read Documentation for more details. Contributors must adhere to plagiarism policy and questions added by them need to be original and not taken from other sources. Also, they must add questions of the Tags allowed by website** |
| **3.** | **General Students/Teachers** | **The Question Bank Service is accessible to all learning enthusiasts. This includes students as well as teachers** | **The Students/Teachers can view and retrieve the questions of the Service available on Retrieve-Screen after being authenticated.** | **The Teachers can retrieve the questions of service along with solutions based on Tags. This can be helpful for them in in-Class-activities and exams for their subject. The Retrievers do not have many prerequisites except that they should know how to run website (explained in documentation)** |
| **4.** | **Developers** | **The developers are primarily responsible for maintenance of the website. The website may update with time and Users. Developers must be able to extract data from the website and modify it whenever necessary.** | **Developers are people who have knowledge about the core working of the website. They must be authenticated, and their access must be approved by Admin** | **The Developers must have knowledge (of both the code and the flow) of the working of the website.** |

# **Feature requirements (described using use cases)**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Use Case Name** | **Description** | **Release** |
| **1** | **Question Upload (Visually)** | **The user(s) can upload question(s) to the service by entering the question in Add-Question Screen** | **R1** |
| **2** | **Preview Question while adding the question** | **Users can see the preview of how question will look** | **R2** |
| **3** | **Image Upload in Questions** | **Users can upload image to question** | **R2** |
| **4** | **Markdown Input of Questions** | **Users can use Markdown while adding Question Text that can faciliate mathematical equations and formulae** | **R2** |
| **5** | **Question View (Visually)** | **The user(s) can view the question visually by visiting the Display-Question Screen.** | **R1** |
| **6** | **Navigating Questions** | **The users(s) can navigate between questions just by clicking on buttons** | **R1** |
| **7** | **Question extraction (API)** | **The user(s) can extract questions individually and in bulk. API here refers to an Application Programming Interface that will be used to export questions in a prescribed format.** | **R2** |
| **8** | **Question Upload (API)** | **Questions can be uploaded individually or in bulk by the users. API will be used to import questions in a prescribed format.** | **R2** |
| **9** | **General User Authorization** | **Authorization for general users. General users would have authorization to view the questions accessible to them. Right now , Client requirement is to give access of all questions to all users, irrespective of credits** | **R1** |
| **10** | **Authorization while API access** | **1. Authorization for general users. General Users are authorized by the access tokens to be passed as HEADERS to API calls  2. Also other applications can as well make API calls to the service by using these access tokens** | **R2** |
| **11** | **Tag Attachment** | **The tag system assigned attribute(s) to the questions in the form of “tags.” These are keywords that describe properties of a question.** | **R2** |
| **12** | **Search** | **The search system allows the user to see all the relevant matching entries by typing initials/related information.** | **R1** |
| **13** | **Retrieve questions using multidimensional Tag Classification** | **This feature allows users and other applications to retrieve questions based on tags, contributor’s EMAIL , or difficulty or the combination of these via API calls** | **R2** |
| **14** | **Multidimensional Tag Classification** | **This feature allows the user to retrieve questions based on multiple levels of filtering based on tags, difficulty, and contributor’s email.** | **R2** |
| **15** | **Automatic Tag Suggestion** | **Users can type initials of tags and the matching tags are automatically displayed** | **R2** |
| **16** | **Download Questions** | **Download selected questions singly or in bulk** | **R1** |
| **17** | **API documentation access** | **Users can access API documentation of access from the Web Application** | **R2** |
| **18** | **User Profile Management** | **1. Allow users to change passwords 2. Also copy Access token used in API calls** | **R2** |
| **19** | **Register** | **New users/contributors can register to the system** | **R1** |

**Use case diagram**



**Use case description.**

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-01** |
| **Use Case Name:** | **Question Fill and Upload (Visually)** |
| **Overview:** | **The user(s) can upload question(s) to the service by entering the question in Add-Question Screen** |
| **Actors:** | **Admin, Contributors, Teachers, Students, Backend System** |
| **Pre-condition:** | **The add question screen for question upload form should be opened.** |
| **Flow:** | **Main (success) Flow:**   1. **User enters the question.** 2. **User enters the options.** 3. **User enters the explanations.** 4. **The user may / may not add tag(s) from the existing tags.** 5. **The user clicks on the submit button.** |
| **Alternate Flows** | **Alternate Flow- 1**  **5. User clicks on submit without entering the mandatory fields of question, options, and explanations** |
| **Post Condition:** | **The question is filled in and uploaded.**  **Alternate Flow - 1**  **5.The screen will display error message and will end up to the same screen with the entered details** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-02** |
| **Use Case Name:** | **Preview Question while adding the question** |
| **Overview:** | **Users can see the preview of how question will look** |
| **Actors:** | **Admin, Contributors, Teachers, Students, Backend System** |
| **Pre-condition:** | **The add question screen for question upload form should be opened.** |
| **Flow:** | **Main (success) Flow:**   1. **User enters the question.** 2. **User enters the options.** 3. **User enters the explanations.** 4. **The user may / may not add tag(s) from the existing tags.** 5. **The user clicks on the Preview Page button** |
| **Alternate Flows** | **Alternate Flow- 1**  **5. User clicks on Preview, entering the mandatory fields of question, options, and explanations** |
| **Post Condition:** | **The preview of the question is Displayed to user**  **Alternate Flow - 1**  **5.The Preview of only the fileds entered is displayed** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-03** |
| **Use Case Name:** | **Image Upload in Questions** |
| **Overview:** | **Users can use Markdown while adding Question Text that can faciliate mathematical equations and formulae** |
| **Actors:** | **Admin, Contributors, Teachers, Students, Backend System** |
| **Pre-condition:** | **The add question screen for question upload form should be opened.** |
| **Flow:** | **Main (success) Flow:**   1. **User click on Upload Image button 12. User chooses an image from his device 13. User clicks on done** |
| **Alternate Flows** | **Alternate Flow- 1**  **5. User clicks on Upload image after uploading an image and then chooses a new image** |
| **Post Condition:** | **The image is uploaded**  **Alternate Flow - 1**  **The new image overwrites the existing one** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-04** |
| **Use Case Name:** | **Markdown Input of Questions** |
| **Overview:** | **The user(s) can upload question(s) to the service by entering the question in Add-Question Screen** |
| **Actors:** | **Admin, Contributors, Teachers, Students, Backend System** |
| **Pre-condition:** | **The add question screen for question upload form should be opened.** |
| **Flow:** | **Main (success) Flow:**   1. **User enters the question the question in markdown format** 2. **User clicks on Preview** |
| **Alternate Flows** | **None** |
| **Post Condition:** | **The Markdown is rendered and displayed to the user** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-05** |
| **Use Case Name:** | **Question View (Visually)** |
| **Overview:** | **The user(s) can view the question visually by visiting the Display-Question Screen.** |
| **Actors:** | **Admin, Students, Teacher, Contributors, Backend System** |
| **Pre-condition:** | **User must be on the view question screen and must have the permission to view the question(s)** |
| **Flow:** | **Main (success) Flow:**   1. **User selects a question.** 2. **User may search for a question.** 3. **Users may also apply tags to level up search.** 4. **User clicks on the matching result** |
| **Alternate Flows** | **Alternate Flows - 1**  **4. If none of the requested questions match the user’s requirements, “No results found” is displayed.** |
| **Post Condition:** | **The question is viewed.\**  **Alternate Flows - 1**  **4. “No results found” is displayed.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-06** |
| **Use Case Name:** | **Navigating Questions** |
| **Overview:** | **The users(s) can navigate between questions just by clicking on buttons** |
| **Actors:** | **Admin, Contributors, Teachers, Students, Backend System** |
| **Pre-condition:** | **The View Question screen must be opened** |
| **Flow:** | **Main (success) Flow:**  **1. User clicks on the forward button on question** |
| **Alternate Flows** | **User clicks on backward button** |
| **Post Condition:** | **The User is shown the next question (next in terms of numbering)**  **Alternate Flows:- User is shown the previous question** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-07** |
| **Use Case Name:** | **Question Extraction (API)** |
| **Overview:** | **The user(s) can extract questions individually and in bulk. API here refers to an Application Programming Interface that will be used to export questions in a prescribed format.** |
| **Actors:** | **Admin, Teachers, Students, Developers, Backend System** |
| **Pre-condition:** | **Users should be authenticated and must have the permissions to view the question(s).** |
| **Flow:** | **Main (success) Flow:**   1. **User enters an API key.** 2. **User enters details on the question (details to be expanded later)** 3. **User submits request.** |
| **Alternate Flows** | **Alternate Flow - 1**   1. **The API key is wrong. An error message saying that the API key is wrong is displayed and the user is given more chances.**   **Alternate Flow - 2**   1. **The details of the question return no valid question(s); hence an error message is displayed, and the user is given the option to retry or exit the application.** |
| **Post Condition:** | **The user is provided by a JSON file.**  **Alternate Flow - 1**  **User exits from the service.**  **Alternate Flow - 2**  **User exits from the service.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-08** |
| **Use Case Name:** | **Question Upload (API)** |
| **Overview:** | **Questions can be uploaded individually or in bulk by the users. API will be used to import questions in a prescribed format.** |
| **Actors:** | **Admin, Contributors, Developers, Backend System** |
| **Pre-condition:** | **Users must be authenticated, must have a valid JSON file.** |
| **Flow:** | **Main (success) Flow:**   1. **User enters an API key.** 2. **User attaches a file containing the JSON data for the questions to be uploaded.** 3. **A validator checks for the validity for the JSON file and returns its state of being correct or not correct.** 4. **If the state is correct, the user submits the request.** |
| **Alternate Flows** | **Alternate Flow - 1:**   1. **The API key is wrong. An error message saying that the API key is wrong is displayed and the user is given more chances.**   **Alternate Flow - 2:**   1. **File has incorrect format for the JSON data to be uploaded. The state of correctness is returned to be incorrect, and the user is prompted with this error. At this point, the user has the option to put another file or can exit the application.** |
| **Post Condition:** | **Questions are uploaded.**  **Alternate Flow – 1**  **User is prompted again.**  **Alternate Flow – 2**  **User is prompted to add another file.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-09** |
| **Use Case Name:** | **General User Authorization** |
| **Overview:** | **Authorization for general users. General users would have authorization to view the questions accessible to them. Accessible questions are a subset of the set of uploaded questions on the website.** |
| **Actors:** | **Authorization System, Students, Teachers, Developers** |
| **Pre-condition:** | **User must not be a super user.** |
| **Flow:** | **Main (success) Flow:**   1. **The user enters their username followed by password.** 2. **The system verifies the credentials and the status of not being a super user.** 3. **The user is logged in.** |
| **Alternate Flows** | **Alternate Flow - 1**  **2. If the credentials are wrong, the same Login screen appear displaying Error message**  **Alternate Flows - 2**  **2. If the password entered is not of a minimum length of six characters, the same login screen appears displaying an error message.** |
| **Post Condition:** | **User is authenticated.**  **Alternate Flow – 1**  **Error message is displayed.**  **Alternate Flow – 2**  **Password length error message is displayed.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-10** |
| **Use Case Name:** | **Authorization while API access** |
| **Overview:** | **1. Authorization for general users. General Users are authorized by the access tokens to be passed as HEADERS to API calls  2. Also other applications can as well make API calls to the service by using these access tokens** |
| **Actors:** | **Authorization System, Admin** |
| **Pre-condition:** | **User must be on an API HTTP API calling application User must be having access token of application** |
| **Flow:** | **Main (success) Flow:**   1. **The user enters request along with body(if applicable)** 2. **The user attaches the access token in Authorixation “Bearer” of the service** 3. **The user sends the request** |
| **Alternate Flows** | **Alternate Flows - 1**  **1 User enters invalid Access token**  **Alternate Flows – 2**  **1 User enters invalid API call** |
| **Post Condition:** | **User recieves the API response with success (Code : 200)**  **Alternate Flows - 1**  **2. If the access token is invailid, the request is denied with return status 400**  **Alternate Flows - 2**  **2. Return status is not OK** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-11** |
| **Use Case Name:** | **Tag Attachment** |
| **Overview:** | **The tag system assigned attribute(s) to the questions in the form of “tags.” These are keywords that describe properties of a question.** |
| **Actors:** | **Students, Teachers, Contributors** |
| **Pre-condition:** | **The user must be on the Question Upload Page in the case of visual uploading.** |
| **Flow:** | **Main (success) Flow:**   1. **User starts typing a tag.** 2. **The system provides tag suggestions.** 3. **User completes typing the tag and submits.** 4. **The submitted tag is added to the list of the added tags.** 5. **Process is repeated until the user is satisfied with the tags or max-tag limit is reached.** |
|  | **Alternate Flows:**  **Alternate Flow 1:**   1. **User starts typing a tag.** 2. **The system provides tag suggestions.** 3. **User selects a suggested tag.** 4. **The selected tag is added to the list of tags.** 5. **Process is repeated until the user is satisfied with the tags or max-tag limit is reached.** |
| **Post Condition:** | **The tags are added and are visible on screen** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-12** |
| **Use Case Name:** | **Search** |
| **Overview:** | **The search system allows the user to see all the relevant matching entries by typing initials/related information.** |
| **Actors:** | **Students, Teachers, Admin, Contributors, Developers** |
| **Pre-condition:** | **The user must be on the search page and the search query must not be empty.** |
| **Flow:** | **Main (success) Flow:**   1. **User enters a search query.** 2. **The search query is processed, and valid outputs are generated.** 3. **The user views the search results.** |
| **Alternate Flows** | **Alternate Flow - 1:**  **3. If no search results match the query, “No match found” is displayed on screen** |
| **Post Condition:** | **The search results are returned.**  **Alternate Flow - 1**  **A no result found message is shown to the user and the user gets another try for the same.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-13** |
| **Use Case Name:** | **Retrieve questions using multidimensional Tag Classification** |
| **Overview:** | **This feature allows users and other applications to retrieve questions based on tags, contributor’s EMAIL , or difficulty or the combination of these** |
| **Actors:** | **Students, Teachers, Admin, Contributors, Developers** |
| **Pre-condition:** | **The user must be on an application calling API The user must be authorixed and thus have access token in Authorization Bearer** |
| **Flow:** | **Main (success) Flow:**  **1. User enters a API call**  **2. The call must have either of tags or difficulty or contributor’s email after ?**  **3. The user views the search results.** |
| **Alternate Flows** | **Alternate Flow - 1:**  **1 User enters invalid tags or difficulty in API call** |
| **Post Condition:** | **The search results with status 200 OK**  **Alternate Flow - 1**  **The search results return error 400** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-14** |
| **Use Case Name:** | **Multidimensional Tag Classification** |
| **Overview:** | **Allows the users to search for combinations of tags allowing a multidimensional classification over a linear one.** |
| **Actors:** | **Students, Teachers, Developers, Admin, Contributors** |
| **Pre-condition:** | **Users must be on the search page and multiple tags must be selected.** |
| **Flow:** | **Main (success) Flow:**   1. **User enters a search query with multiple tags.** 2. **The search query is processed, and valid outputs are generated.** 3. **The user views the search results.** |
|  | **Alternate Flows:**  **Alternate Flow - 1**  **2. The search query returns with no result found.** |
| **Post Condition:** | **The search results are displayed.**  **Alternate Flow - 1**  **A no result found message is shown to the user and the user gets another try for the same.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-15** |
| **Use Case Name:** | **Automatic Tag Suggestion** |
| **Overview:** | **Users can type initials of tags and the matching tags are automatically displayed** |
| **Actors:** | **Students/Teachers, Admin, Contributors** |
| **Pre-condition:** | **User has started searching for a tag and typed something** |
| **Flow:** | 1. **User signs in** 2. **User tries to search for questions by tag.** 3. **User starts typing the initials of a tag.** 4. **All similar tags are displayed to user to choose from** |
|  | **Alternate Flows:**  **4. No tag is displayed; the user continues to manually write a tag.** |
| **Post Condition:** | **User has been given tag options like the initial letters he has typed.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-16** |
| **Use Case Name:** | **Download Questions** |
| **Overview:** | **Download selected questions singly or in bulk** |
| **Actors:** | **All Users** |
| **Pre-condition:** | **User add single/multiple questions to Download Bucket User navigates to download Question screen** |
| **Flow:** | **Main (success) Flow:**   1. **The user must click on donwload button from the screen** |
|  |  |
| **Post Condition:** | **User gets downloaded questions in JSON format.** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-17** |
| **Use Case Name:** | **API documentation access** |
| **Overview:** | **Users can access API documentation of access from the Web Application** |
| **Actors:** | **Teacher/Students, Contributors** |
| **Pre-condition:** | **User has signed in and landed on any page containing NAVBAR** |
| **Flow:** | 1. **User clicks on SMILEY button in navbar** 2. **User clicks on API documentation from the DROP-DOWN menu** |
|  | **Alternate Flows:**  **None** |
| **Post Condition:** | **User lands on API documentation page .** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-18** |
| **Use Case Name:** | **User Profile Management** |
| **Overview:** | **1. Allow users to change passwords 2. Also copy Access token used in API calls** |
| **Actors:** | **Teachers/Students, Contributors** |
| **Pre-condition:** | **User islogged in . User is on profile page** |
| **Flow:** | 1. **User signs in** 2. **User tries to copy access token** |
|  |  |
| **Post Condition:** | **The access token copied message is displayed** |

|  |  |
| --- | --- |
| **Use Case Number:** | **UC-19** |
| **Use Case Name:** | **Register** |
| **Overview:** | **New Users/contributors register in system with the help of Register screen** |
| **Actors:** | **Teacher/Students, Contributors** |
| **Pre-condition:** | **User should not have an already existing account** |
| **Flow:** | **Main (success) Flow:**   1. **The user enters their username followed by password.** 2. **The system verifies the credentials and the status of not being a super user.** 3. **The system creates user of the username and password.** 4. **The user is logged in.** |
|  | **Alternate Flows:**  **None** |
| **Post Condition:** | **The user is registered and logged in.** |