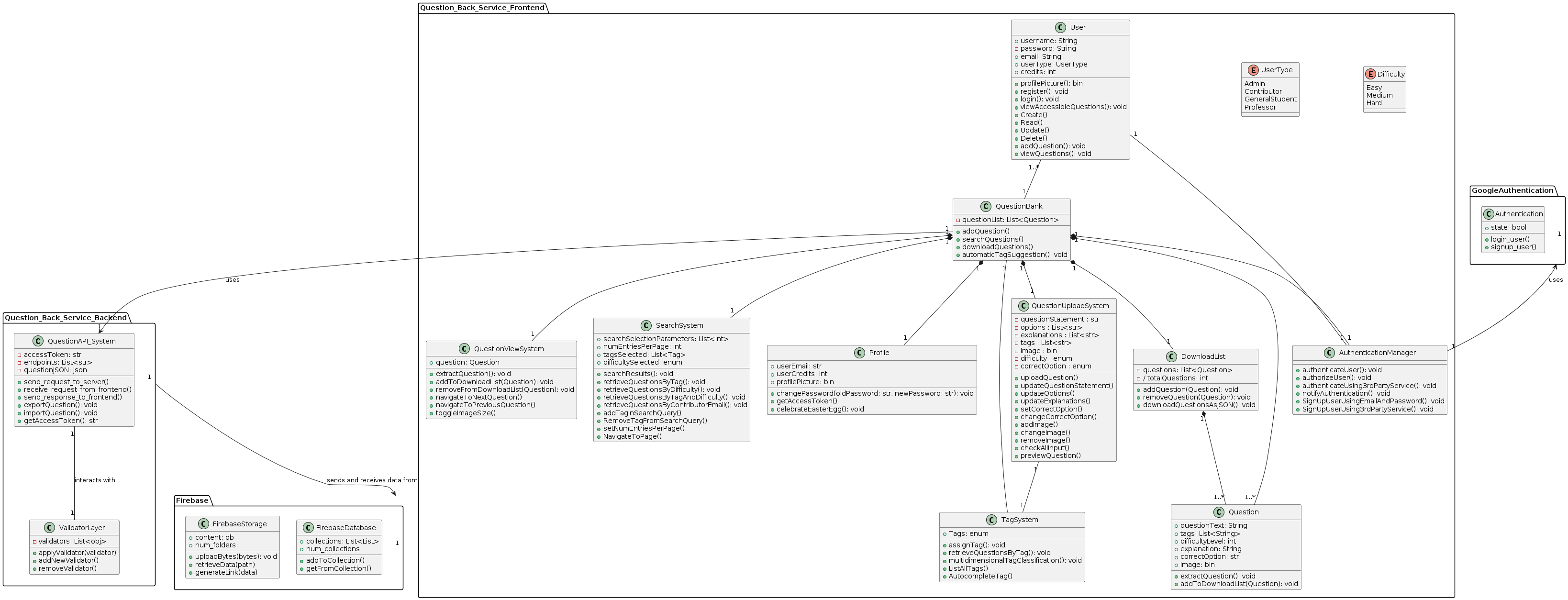
**Product Design**

|  |  |
| --- | --- |
| **Team** | **Number: 21**  **Members:**   * Ansh Chablani (2022111031) * Divyarajsinh Mahida (2022101085) * Siddharth Singh (2021114001) * Shreyas Badami (2021114003) |

# Design Model

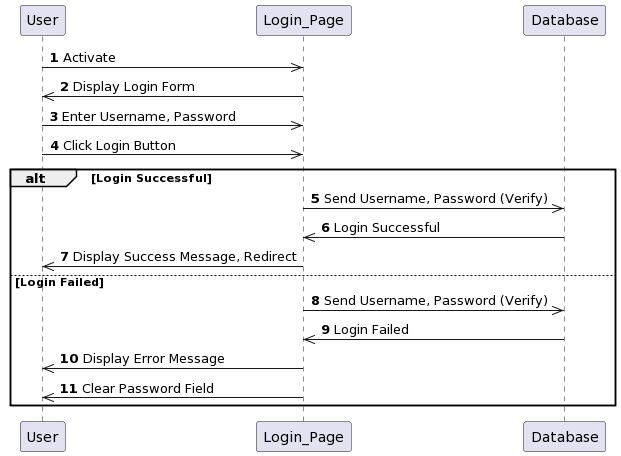
|  |  |
| --- | --- |
| Class Name | Features of Class |
| User | * **Class State:**  1. Maintains user-related information such as username, password, email, user type, and credits. 2. User Type can be three values: Admin, Contributor, Student.  * **Class Behavior:**  1. Supports user actions like registration, login, viewing accessible questions, and changing credits. 2. Credits change when a user adds/retrieves a question |
| Admin | **Class State:**   1. Inherits attributes from the User class.  * **Class Behavior:**  1. Implements CRUD operations (Create, Read, Update, Delete) for managing questions. |
| Contributor | * **Class State:**  1. Inherits attributes from the User class.  * **Class Behavior:**  1. Implements the ability to add questions to the system. |
| General Student | * **Class State:**  1. Inherits attributes from the User class.  * **Class Behavior:**  1. Implements the ability for general students to view questions. |
| Questions | * **Class State:**  1. Holds information about a question, including the question text, tags, difficulty level, and an explanation. 2. Difficulty levels can be Easy, Medium, or Hard. 3. Each option has an Explanation.  * **Class Behavior:**  1. Implements methods for extracting the question, checking for plagiarism, and adding the question to a download list. 2. Extracting Questions can be done through API or Visually. |
| Question Bank | * **Class State:**  1. Maintains a list of questions and a credit system.  * **Class Behavior:**  1. Supports adding questions, automatic tag suggestion, and plagiarism checks. 2. The Upload Question Class is a composite of this class. |
| Authentication Manager | * **Class State:**  1. Included in the User and Question Bank classes.  * **Class Behavior:**  1. Provides methods for authenticating and authorizing users. |
| Search System | * **Class State:**  1. Included in the Question Bank class.  * **Class Behavior:**  1. Implements methods for retrieving questions based on tags, difficulty, and combinations. 2. Tags must be chosen from the ones existing. 3. Multidimensional tags also applicable |
| Question API System | **Class State:**   1. Contains methods for exporting and importing questions. 2. The methodology used for exporting and importing is API calls. |
| Download List | **Class State:**   1. Manages a list of questions and keeps track of the total number of questions. 2. The total number of questions is a derived attribute.   **Class Behavior:**   1. Supports adding and removing questions from the download list and downloading questions as a JSON file. |
| Question Upload System | * **Class State:**  1. Holds information related to uploading a question, including the question statement, options, explanations, tags, difficulty, and correct option.  * **Class Behavior:**  1. Implements the ability to upload questions. |

**UML Class Diagram (too small? View at:** [**link**](https://www.plantuml.com/plantuml/svg/hLTVRnit37_EVOf3dh9kwJ3ipIY2Ed_QPKXJhtRwQj0d-g969zqadbDlo7UV99yaoxvUXs4ljlKZHP4KVwJyrX1ewXfHCZIybalxioXQg9wWnlAF3Wrn9PUN5bZEKMzuXSjtMab2oSg_YvDAW37bWq7jLYS_bftzAQ71IJadpMLjmVEojKhFIhC4MariW8k8XBsBRMltFlI_F5nfP9pCfEIIFD1gjUO2PxoYJkFP3vDonQMNQAov8TGEsoZEF2XKpMM6R3W-JwiAZU4hWI5Aa-bSQGIonltY4mBhVpwqBE7NA30kWB5WwkYqWJDUYf2-8EnJ-6U_NE1Nol93KDiit753RtReXOSPNwzvrGdQtk46HSeJVcq5I72sCZkLqXeh-jZk98QqXta3DUvosMyd3HKDncHZNQXhzIo50kOyEWjg8_5zjecORmrXSn3f9AZyFoVxGcZK1csbDjzLbR3XDH0krBqzDvrv89nft73LcGC5KdKjyDQbRix_GeiE17u9yic7V1v3TkxqTncqBewho2C6GLUF0JGzolgu3d7eI3L0l5f0FU_gUYSTAhjfHuyehQQla0yWeOuqXYJ31vC85OLAssW79MaRb_Mlcir0qxRl7PcoLCJNszoBJ67EQ_dGkaEyhHlNCAQIpVfkCgOxV6uM_jodDTPZolGdD9sWnDGacQF0ohbfRKE3X3hmaKlQNP_icXj9cgEPePvvGbbPfE_E0h9-smAEE9mqqDhgYzaPtAH6TRcr6vD_mohNqMJG735cq_hlzQ-KYt3LaTB-Jexu2VMjd8UgrTlGH3qdhT3HybXiaEuFazUBxXCBBPXpRBOR1cdyx7nARSr1L-CZPA_fLOyWQumbfaGiDs_nfrBYSuRinruZxSRAGZrXe6-50bSQ26_0aFMkheUgqQObAqMxDdq9-3KODvx7htsmCffE46UyGMci0WWhlnBEsdg8Sgugfy8fXNvY-QagrRG22ItS9xyuSQx5EUsTdhB6Z-6zyh2hzoXHWvXJnz0dtrqO6kE4By5EgZrloMPgQxy-8EEU0h-1xfELTTJJNqxBL-VdzllrwrUdQOQSEBS2wkL9w3P6XI0TReNVFEqqwnq3QhrMlDDXI__fv7oMZXiFt3YQHm-jcnoXthAvD2Um84egUmPPcMgTn9HzqxBmXSF3-s9OI50VW_w9KNgw-JLBznxxtNd-FWc27PUTuuVpiVCliW3603BfpfR9i9izzrCFkje_tCAfAGn9W34_-pF5WAI7IAw3rPFbsArik-1agt7_bTYJRAGA1YuogEVeS3s6WjX_71qKWXbudPp_N98Y404B3JL_jemyo4PK1buqK_Fx_EDzw4_ZASb2AKxsLz_i4o_V-wVZ5jaVdQ0sdTqkC_f0Qlk9DtPhgxWaazCWvTY5Doc_cF3kDNR7KgCNBqajZVK3TQoa2la6etojLUC-_7ynfvCCc5P9Wyv2BdUCLNhuNSsRN3Gorle8q_3w388p8ANlO8kwJyacW25yjVfoqG-pjXNRkEaiQaPEtUDp56T4IcXEaNGZMSNdZlbgHGqLcVAPqsEnNm7lkCOL64oN7P2vtGzrU65KFg4qATagFFYMQoKOkZZtkzJbbj2ShTpdqCY-1eAp5kXn5rxjnoJX7PTFPypA-lXoLzmcxsHqHWYizkdgFYwYOtlo8lrxkeheMJYSt9iin-tfBp4xxvMg1UR3AcNg67UrI43kPQMKA8lqHtpPcLZIxadJjH7n9nO3q_J2VG-GRrAUshtcj1YbRsOaP7BYkM9Aa2mmovGkxwNZLL4C1kmCl1aRt9FI-LCKRwtjhX7_0000)**)**

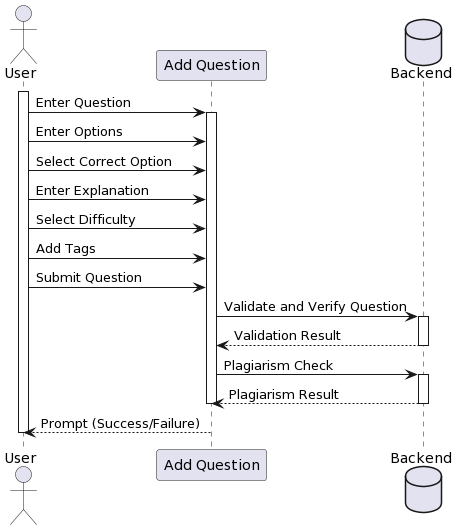


# Sequence Diagram(s)

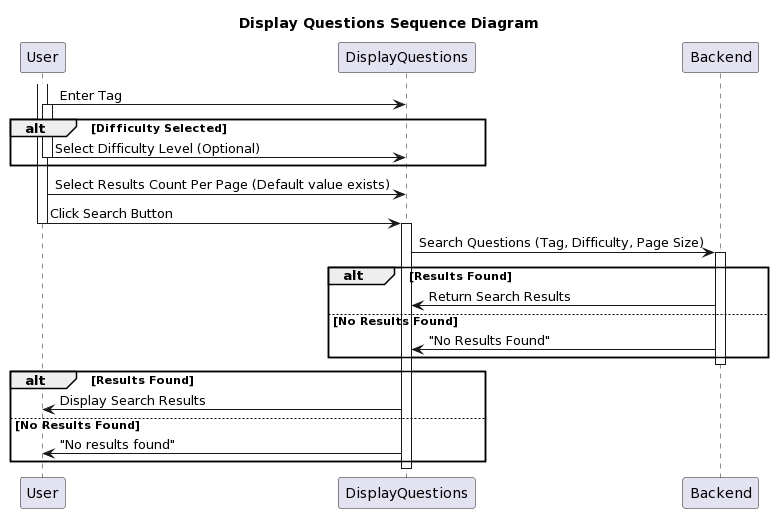
1. **Authentication**



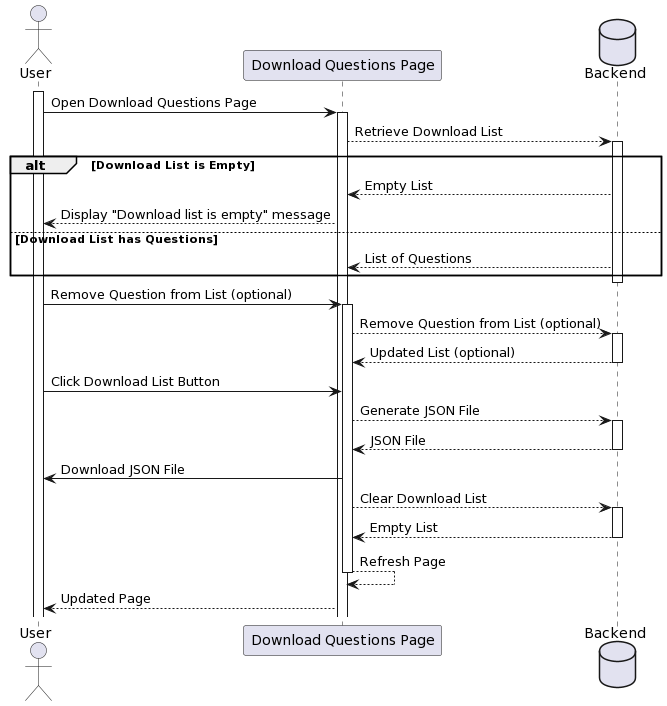
1. **Add questions.**



1. **Display Questions**



1. **Download Questions**



# Design Rationale

There were certain components we considered while designing this question bank service.

Let us see this component-wise.

1. **Login Page**

The login page had to be a standard page and a gateway for the users of the website. It had to be designed so that the first impression of the service should not be disappointing.

We wanted to make it simple and effective enough. To do this we included the following features:

1. Functionalities-wise we made it like a form which is most intuitive for people and widely used across several platforms on the web.
2. After the user hits on the login button, there is some delay before the user gets logged in. Earlier, a waiting indicator was not available. In fact, the user could also change to the sign-up option doing this time. It was later changed so that waiting times have been added and the malicious switch to the sign-up option has been disabled.
3. Retention of details on wrong credentials. If a user enters the wrong credentials, an alert is shown, however the previously entered details remain intact. This choice was made so the user can correct minor problems in the password without re-entering the email id.
4. **Display Questions Page**
5. Several iterations went into designing the display questions page.
6. Initially we designed it to take up much more real estate than what it takes as of what is planned for release – 1.
7. It is now condensed into a much-condensed form for better readability.
8. Also, viewing a particular question involves being able to only see its statement in the initial display questions screen. Only later can we see the full question parts.
9. **Add Questions Page**
10. The add-questions page is built like a form. The form’s format is like some of the popular form providers. This was done so that users could add their questions in an environment whose look and feel is like something that they have used before.
11. This is in line with the methods adopted to keep the target audience in mind.
12. **Download Questions**
13. The download questions page had to not be a hassle.
14. We tried to make it a minimalist page, taking inspiration from the other pages around, wherein we found that in a crowded downloads page, the user has a lot of inconvenience in locating the correct button for download, causing frustration at the very last step for some users.
15. We also thought of displaying all the questions that are in the download list in a top to bottom sequential order. This was done so that the user can quickly navigate the headers of the questions.
16. Note that the questions while being displayed in the download list only include their statements. The options can be viewed by clicking the header. This was in line with the minimalist approach, while adding functionality to view the details of the questions also.