**SOFTWARE DEVELOPMENT**

**FUNDAMENTALS-II**

**QUIZ MASTER**

**GROUP MEMBERS – BATCH B1**

* 18103304 RISHIK SOOD
* 18103024 SHRUTI GOYAL
* 18103008 ANUSHKA AGARWAL
* 18103020 RITWIK TANEJA

**PROBLEM STATEMENT AND IT’s RELEVANCE**

**Objective -** The objective of the program is to give the user some random questions which would help him to prepare for tests. The program keeps track of all the questions stored in it.

**Scope-** The program can be used by students to help test their knowledge on a particular topic by getting random questions from the system. The program can also be used by teachers to add more questions.

**Data Structures**

* **Array and Structure**

Various structs and arrays have been used to store the names, passwords, manipulate questions, etc during the execution time of the program.

**2. Circular Queue**

We have used a queue due to its favourable FIFO principle. The queue contains ‘x’ number of questions. If the students chooses to attempt the question right now, or not at all, it will be taken out of the queue, otherwise, the question is put at the end of the queue, so that it can be attempted at the end. We have used a ‘circular queue’ due to its property to save space, thus it made a very practical choice.

**3. Binary Search Tree**

The binary search tree has been used to store the questions during the execution time of the program. It made a practical choice due to its property to search for elements in O(log(n)) time. We use it to generate unique IDs for each question. Each node in the binary tree consists of the question, its options and its IDs. We also used the property that it provides an ordered output of integers after inorder traversal.

**Note:** We have avoided the use of linked list and used the Binary Tree instead as, when we generate a unique ID, we need to continuously search if a node exists with the same id. If we used a Linked list, it would’ve consumed O(n) time which might be of significance during larger input.

**Project description**

There are two users teacher (admin) and student (user). First procedure is to add questions. Teacher can log in to the application using an id and a password. Once the teacher logs in to the application he/she can add and delete the questions available to/from the system. Once there are enough questions available the student can ask for a questionnaire (in which the system itself will select random questions from questions available) and can set the number questions to be asked, which he/she will have to answer in a limited time as each question will get (set the time yourself).Each question will have four options (MCQs) the student will be rewarded +4 marks for every correct answer and -1 for every wrong answer. Student will also have an option to leave the question unaswered. At the end of the questionnaire all the attempted and unattempted questions will be displayed and if the time would allow the user will also get to attempt all the unanswered questions and modify answers of already answered questions.

After submitting the questionnaire the system will show the student his/her score based on the scheme mentioned above.

**Working Algorithm**

When the program starts, it provides the user with the option to login or quit. If the user chooses to login, then he must enter a Unique ID and password. Based on the ID and Password, we encounter 3 cases:

**1. Correct ID and Correct Password.**

When both the ID and password are correct, the user is logged in as a student or teacher, based on the database stored before in “StudentsIDPW.txt” and “TeachersIDPW.txt”.

**2. Correct ID and Wrong Password.**

When the ID exists in the database and the corresponding password is incorrect, then the user is prompted to re-enter the ID and password.

**3. New User.**

If the ID doesn’t exist in the database, then the user is

asked to enter his neccasary details and then a request is put into “Requsts.txt” and

when the user is approved as a student or teacher only when the Admin approves.

Based on the correctness of the ID and Password, we arrive at 3 cases:

**1.**

**Admin**

If the user is admin then he is shown a bunch of requests that exist in “Requests.txt” and then the he can accordingly approve or disapprove of the requests. If the admin approves the request then the user is added to “TeacherIDPW.txt” or “StudentIDPW.txt” accoridngly.

**2. Teacher**

If the user is admin, then the user can Add a question or Delete a question.

If the user chooses to add a question, then he can provide the question and 4 options. He also has to provide us with which answers in the given options are correct. Then a random ID is generated for the question and the question is put into the Binary Tree.

If the user chooses to delete a question, then he is provided with the existing questions in the database, their corresponding correct answers and their IDs that have been previously generated. Now, the user can choose an ID, if a question exists in the database with that ID, then it is deleted from the database and also from the Binary Tree.

**3. Student**

If the user is a student, then he can choose to attempt a quiz or not. If he chooses to attempt the quiz then ‘x’ (can be defined in the code easily) questions are added into a queue. For each question, the student can choose to attempt it now, later or not attempt it at all. If the student attempts the question, then he is provided with +4 or -1 marks

based on the correctness, if the question is passed on for later, then it is put back into the queue and in the case of not attempting a question, the question is removed from the queue. This process continues till the queue is not empty.

**GROUP MEMBER WISE TASK DISTRIBUTION**

* 18103304 Rishik Sood

The functions like implementing the quiz by using binary search tree ,circular queue , shuffling of questions and also calculating the time taken by the user,compiling the programme and also rectifying the error in the programme was done by Rishik.

* 18103024 Shruti Goyal

Admin ,teacher and student menu were created by Shruti also the addition and deletion of question and the code of starting and making quiz was done by her.

* 18103008 Anushka Agarwal

Sign/log in requests of student,teacher or a new user and also the approval of request by the admin and checking of existing user was done by Anushka.

* 18103020 Ritwik Taneja

Synopsis was made by him while also providing vital ideas and features which were added in the project.