C22063583 - YOUTH GROUP SCENARIO

**Database design**

I was tasked with creating a database for a scouts like youth group called SuperGeeks which should involve capabilities of handling sessions and groups with wait lists, volunteers and validating if they’ve had safety checks, handling badges including requirements, applications, validation and verification. We also have a responsibility to protect the students so we need to make sure that we have access to contact information for their parents so they can be informed if their child is ill or some other emergency.

The database will used hashed data to store the IDs of the users such as students, parents and volunteers. Hashing is when you map an input into a piece of data so that the same input should get the same output each time as described by (Aggarwal, Verma 2015) It should also be impossible to revert the hash to the inputted data. This is useful as we do not want a malicious party to access our database and seeing the passwords of all our users either in plaintext or encrypted. These hashes will be made out information such as emails, names and passwords so they can be used on login. Additionally this will be useful to prevent duplicate users as they would need to match multiple fields of data. all of these will be a varchar with a length of 40 characters as that is the result of an md5 hash which is built into MYSQL the DBMS I’m using. If a describe an ID and don’t mention its datatype it will be this.

The universe of Discourse describes the concept of a table in a database representing some aspect of the real world ( Wojnicki 2005 )The Student table represents an individual student in the universe of discourse. It contains an ID which is a result of a hash between their full name and password. It cannot be null and it must be unique. This represents the specific student so it multiple students cannot share the same ID. It also contains the studentCode which is an auto incrementing value which should be a stand in for a username. It starts at 30001 to not have easy to guess usernames such as “1”. It also has values for the first name and last name, fname and lname respectively they are varchars of length 35 as recommended by (NHS 2011). It stores the birth date as a date so that it can work out if the student can join a specific group

The parents table represents a parent in the UoD. Their ID is the result of a hash of their email and password

The prompt suggests that SuperGeeks is a popular group and has a long waiting list this means that we need a way to handle which students get added when spaces do open so my program should be able to use logic to work out which students should be added and which ones don’t and I will use a first come first served method of prioritizing students. I know institutions such as the NHS offer priority with their waitlists so I believe this db should also . The waitlist table will have the ID of the student who is on the waitlist, the Id of the group they are applying to, the day the student joined SuperGeeks (date) so in promotional emails we can say how long they’ve been a part of our organization. It also has their parents ID so we can keep data separated as they are represented by different things in the universe of discourse and we need a way to connect them. This way a parent can have multiple children and a student can only have one parent (account). It also has the groupID so we know if they are a part of the group if they try to attend an event.

A picture containing diagram, text, plan, line

Description automatically generated

References

Kirti Aggarwal., Harsh K. Verma. 2015 Hash\_RC6 — Variable length Hash algorithm using RC6

accessed 22/04/23 [https://ieeexplore.ieee.org/document/7164747](https://ieeexplore.ieee.org/document/7164747%20accessed%2022/04/23)

Igor Wojnicki 2005 – Databases accessed 22/04/23 at <https://home.agh.edu.pl/~wojnicki/phd/node5.html>

NHS. 2011. Government Data Standards Catalogue accessed on 22/4/23 at <https://www.datadictionary.nhs.uk/supporting_information/government_data_standards_catalogue.html>