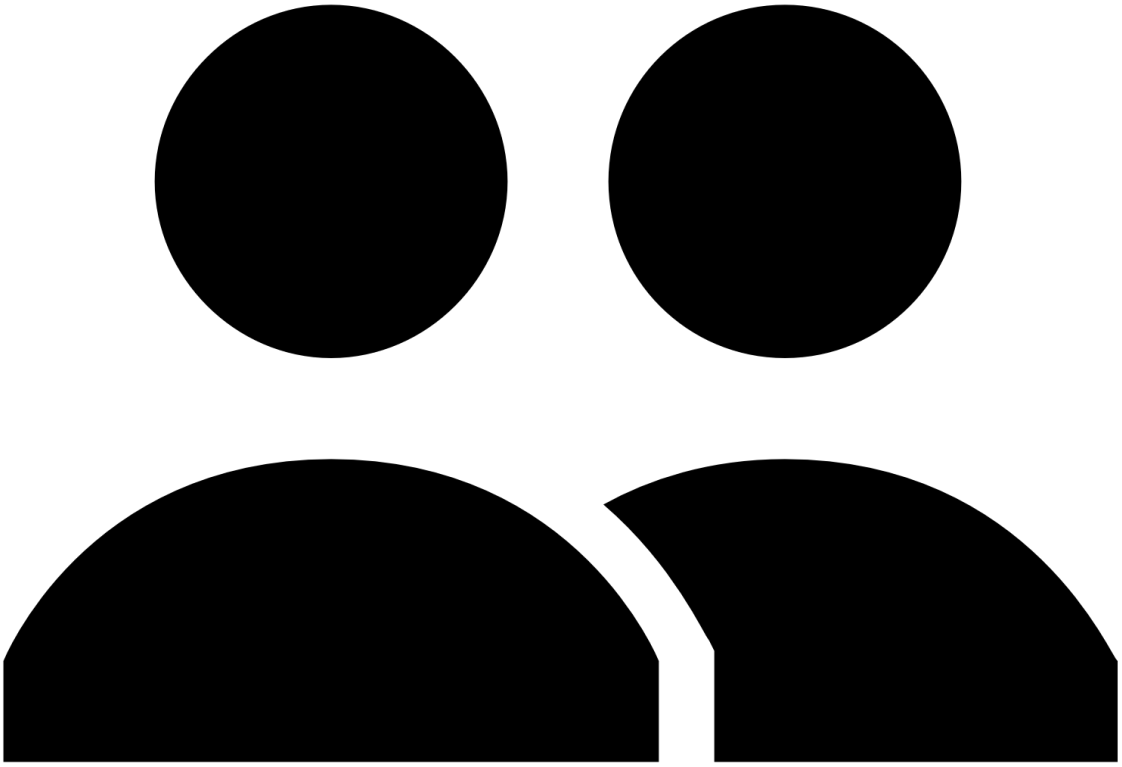
Philippe Krüttli
16.06.2017

information
user management



Project WISSLearnCards - User Management

Employees of this project:

- Frithjof Hoppe
- Philippe Krüttli
- Hugo Lucca (Learning Coach)

Table of Contents

1 starting position

In this document all the necessary information about user management with respect to privileges and opportunities are recorded and erklärt.Die permissions are always awarded to groups and a data management object (with the included tables Card table. Stack Table and Door Table can (by the owner creators) with different permissions to different groups are distributed.

The permissions are in the following sections permissions are more closely erklärt.Grundsätzlich allocated on the stack, so that all cards can be processed within a stack without problems (if the rights are available).

Furthermore, it would be unnecessary to assign the rights to Doors, since not all stacks of a door must have the same privileges and downloading all the Doors would take too much time and data volume.

1.1 expandability

The extensibility of the authorization concept is certainly somewhat restricted since permission is controlled with certain attributes in the roaming database.

Nevertheless successor or optional continued development at any time to add new authorization attributes, or edit the function or the rights of already set permissions.

Thus Weiterführbarkeit of the program is assured and even in complete change of the concept (the part of the WISS) the re-programming would take comparatively relatively little time and effort to complete.

2 authorization concept

In this section, all permissions are explained in detail. The concept was developed by the project team, together with the representatives of the client.

2.1 default permissions

The default permissions, which maintains each user automatically to each data management object, include the following:

- Read access to all data management objects
- Download permission (to only read permission is required) Thus, a clone on the local database user is created.
- However, local clones can then be uploaded * only on its own data management object and not on the original "repository.

2.2 Derive Authority

The so-called Derive authorization is an extension of the standard authorization and thus not only contains the standard rights, but also the following permissions:

- Upload the local clone to the original data management object is allowed.
- The uploaded clones are shown in the original data management object as a new stack.
- Integrating or merging of local with roaming tables or stacks is not allowed.

2.3 Teamwork Authority

The teamwork authorization is intended to enable groups to work together on a stack. in turn, it is an extension of the Derive permission and thus contains all the standard permissions, all Derive rights and the rights listed below:

- Upload the local clone to the original data management object and integration into the existing stack is allowed.
- Mergen is always allowed for an internet connection, only limitation is that below Lock system.
- If, already another merge operation is performed (by another user) on the same stack at the time at which a merge process is started, the newly launched merge operation is canceled and the user will be asked this later (when the first merge process is complete) to try again.

3 Database Concept

All important information is recorded with respect to database concept and erklärt.Die basic structure of the already created file " Dokumentation_User-DB.pdf"In github repository is maintained, but slightly modified.

This has led us to explain here once again and to supplement innovations, the database concept.

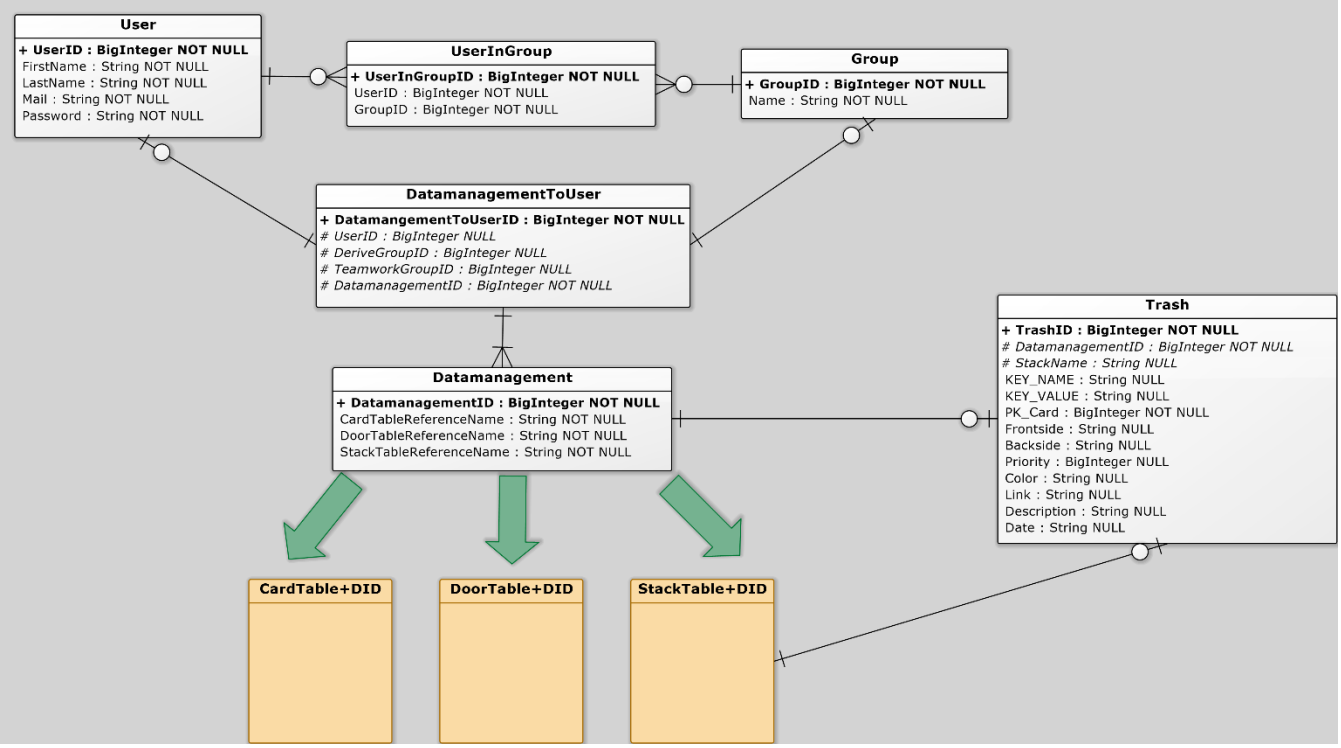
3.1 Database structure

We had to take into account various factors in our database structure. Some questions that we had had to ask ourselves:

- How should the access be regulated?
- we want the functionalities?
- Which tables are suitable?

We have revised the solution variant of the above-mentioned document again, but this time with a proper planning in advance.

Also, we have discussed the solution this time before the reaction with the representative of the client, so this solution variant is now true for all.
Below the resulting database structure is shown below:



3.1.1 Notes to the DB structure

The basic idea of our database structure is as follows:

- However, access to Cards / Stacks / Doors is stored in a central manager, the program itself manages (by authorization queries).
- Each user gets their own separate Cards-, Stacks- and Doors table
- Each has Cards-, Stacks- and Doors table a unique name (table name + DatamanagementID)
- Company / classes can control a shared access by groups

In our solution variant, these functionalities are controlled so that the tables Data Management and Data Management TOUSER access beinhalten.Die to the above levels Table Data Management always refers respectively to a Cards-, Stacks- and Doors table on which the Table Data Management TOUSER access can be set.

The access concept is in the upper portion (2 authorization concept) noted and explained.

As a user name in the user table the e-mail is used address of the user, as this is clearly in each case and thus a check is (uniqueness) pretty easy to implement.

In the table UserInGroup the group membership of all users is regulated. Thus, a user can be a member in both multiple groups and multiple users in a Gruppe.Des Furthermore, the user is not forced to be a member of a group, and there are no unnecessary groups erstellt.Auch group names must be unique in every case and be unique.

3.2 Important

There are some points that are important to consider both of us, and by our successors müssen.Diese are explained in detail below.

3.2.1 blank entries

Since one user and up to two groups may have access to a data management record that we have as RIB (Referential integrity condition) specifies that when deleting a user or a group of the entry in the Data Management table to the value "NULL" to be set.

Thus, the other groups or users retain access, certainly what the only correct solution be kann.Dadurch it but it can also happen that there are Data Management entries in which all three attributes (user ID, and ReadGroupID WriteGroupID) is "NULL "possess, this entry has no more sense.

For this reason, should be at an appropriate time (after one week, one month or one year) are reviewed if there are entries where the problem zutrifft.Die tuple described above, in which this is true, should be cleared out from the table Data Management ,

3.2.2 explicitness

In both the user name and the group name and the table name (of Tables Table Card + DID, Stack Table + DID and Door Table + DID) must be adhered to the uniqueness in each case and checked.

Otherwise, errors can occur which would make the whole user functionality unusable and therefore cost a lot of time and work.

3.2.3 waste paper bin

We have created a recycling bin, where the deleted cards is cached so that they will be restored in case of accidental deletion können.Damit this trash but is not too busy, but it should be emptied regularly or contain a maximum of entries.

A manual deletion would be, especially with maximum entries possible.Of However, the deletion process can also be done, for example, when closing the project or always on the same day of the month.

Even an individual deletion were possible, the entry is in the checked how long now trashed exists.

3.3 additional features

There are still some additional functionality that we first really seen as additional functionality, but now almost obligatory, since downloading and uploading for the basic functionality of the user management are needed.

The above-mentioned points are listed below and explained.

3.3.1 download

If the user has (at least) read access to a Door-, stack or card element, he shall also download this and save a local können.Die new data will be integrated into the existing local database.

This functionality is made easier for the user to its own elements (which he as a user with the user ID has access) can download a lot easier by the existing division into three tables and thus integrate.

The locally stored data should (as well as downloaded from Quizlet data) can be processed, but the change to the server-stored data will not be accepted.

3.3.2 Upload

Just as with the download function, a particular stack by the user in its Data Management entry would now werden.Auch integrated (in the appropriate table) this facilitates our chosen database structure work significantly here.

Also, a merge function is planned, where you can upload a stack in the original data management object.

The upload feature, combined with the download function, it would then be possible, for example, download a Quizlet stack edit it locally or to alter and then upload them to your own (or group internal) structure on the WISS-house server.