Dokumentation zur User-Datenbank

Projekt WISSLearnCards – Benutzerverwaltung

Mitarbeiter dieses Projekts:

- Frithjof Hoppe
- Philippe Krüttli
- Hugo Lucca (Lerncoach)

Inhaltsverzeichnis

1 Ausgangslage

Grundsätzlich war das Programm WISSLearnCards bereits von unserern Vorgängern erstellt worden.
Zuerst haben wir also sogenannte Issues bearbeitet, damit wir besser ins Projekt hineinkommen und damit wir die ganzen Abläufe sowie die Struktur besser verstehen können.

Mit der Zeit war unser Ziel jedoch, auch noch etwas Eigenleistung in das Projekt einzubringen.
Aus diesem Grund haben wir uns für einige weiterführende User-Stories entschieden, welche unter anderem das hier beschriebene Projekt beinhalten.

1.1 Projektbeschrieb

Wir haben uns für eine Implementierung der User-Funktionalität entschieden, welche es dem Benutzer ermöglichen sollte, sich einzuloggen und somit Zugriff auf alle seine Karten, Stapel und Fächer zu haben.

Damit dies auch umgesetzt werden konnte, mussten wir natürlich eine Datenbank planen und erstellen.
Die Planung hat uns (aufgrund einiger Komplikationen) knapp einen Tag gekostet. Die Umsetzung der Datenbankstruktur war in phpmyadmin auf unserem Server (Testumgebung) in weniger als einer halben Stunde implementiert.

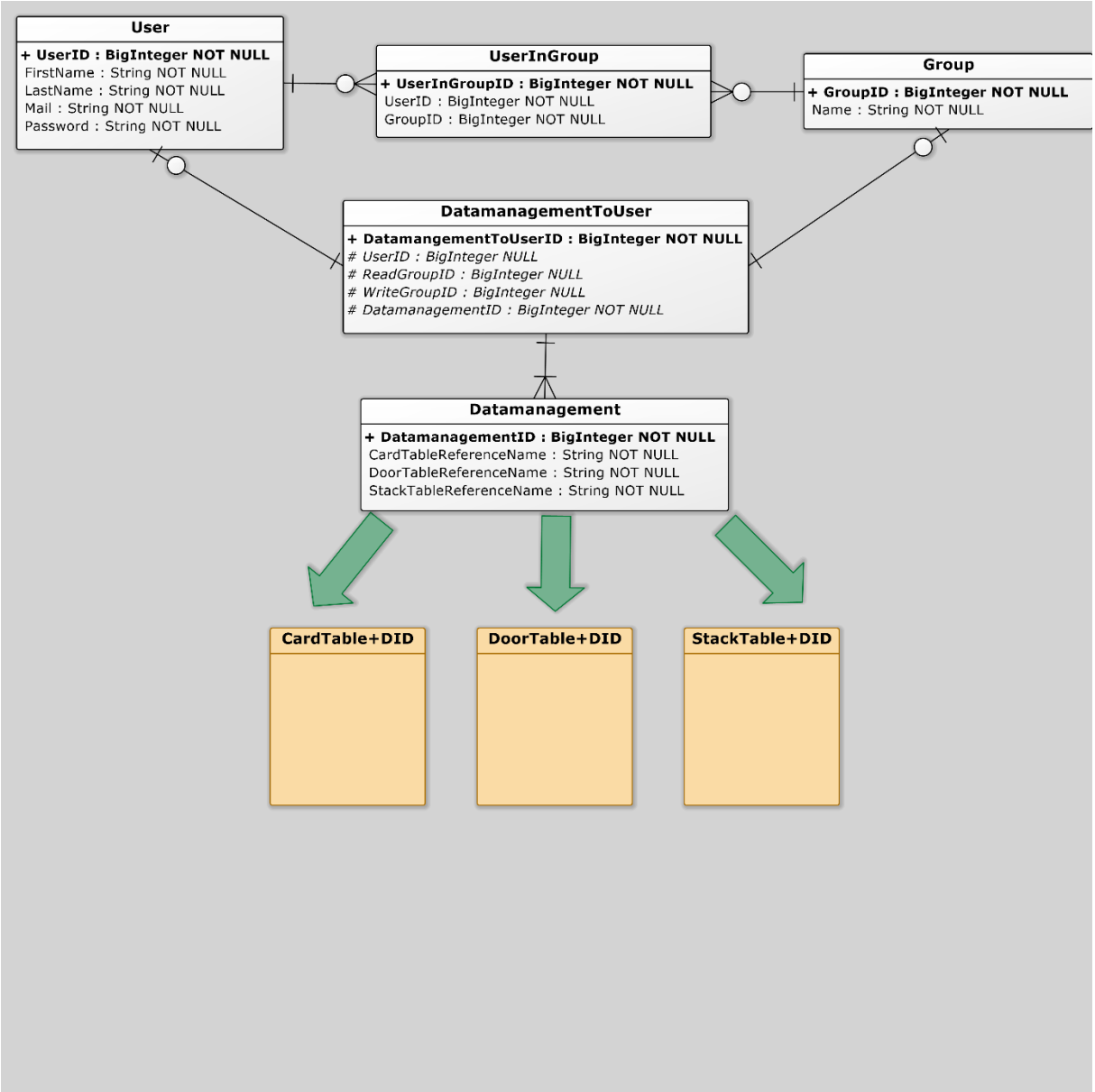
2 Datenbankstruktur

Wir hatten in unserer Datenbankstruktur verschiedenste Faktoren zu berücksichtigen. Einige Fragen, die wir uns stellen mussten waren:

- Wie soll der Zugriff geregelt sein?
- Welche Funktionalitäten wollen wir?
- Welche Tabellen sind dazu geeignet?

Anschliessend mussten wir die Vor- und Nachteile von verschiedenen Lösungen abwägen. Dieser Punkt hat uns am meisten Zeit gekostet, da es hierbei einige Unstimmigkeiten gab.
Schlussendlich haben wir uns jedoch für eine verbesserte Version unseres ursprünglichen Lösungsvorschlags entschieden.

Untenstehend ist die Datenbankstruktur abgebildet:



2.1 Erläuterungen zur DB-Struktur

Die Grundidee unserer Datenbank-Stuktur ist die folgende:

- Der Zugriff auf Cards / Stacks / Doors wird von einem zentralen Manager verwaltet
- Jeder Benutzer erhält jeweils eine eigene Cards-, Stacks- und Doors-Tabelle
- Jede Cards-, Stacks- und Doors-Tabelle hat einen eindeutigen Namen (Tabellenname + DatamanagementID)
- Firmen / Klassen können einen gemeinsamen Zugriff mittels Gruppen regeln

In unserer Lösungsvariante sind diese Funktionalitäten so geregelt, dass die Tabellen *Datamanagement* und *DatamanagementToUser* den Zugriff auf die oben beschriebenen Ebenen regelt.
Die Tabelle *Datamanagement* beinhaltet immer jeweils eine Cards-, Stacks- und Doors-Tabelle, auf welche mit der Tabelle *DatamanagementToUser* der Zugriff geregelt werden kann.

Das Zugriffskonzept ist wie folgt:

- Der Benutzer mit der UserID, welche in der Tabelle *DatamanagementToUser* mit der DatamanagementID verknüpft ist, hat alle Rechte auf die enthaltenen Tabellen. Somit kann er die Datensätze darin sowohl einsehen, erstellen, bearbeiten wie auch löschen.
- Die Gruppe WriteGroupID hat auf die enthaltenen Tabellen sowohl Lese-, als auch Schreibrechte. Somit kann man die Datensätze einsehen, erstellen und bearbeiten. Der Löschvorgang ist jedoch nicht ausführbar.
- Die Gruppe ReadGroupID hat auf die Tabellen CardTable, StackTable und DoorTable nur Leserechte und kann somit die Daten zwar einsehen, jedoch weder erstellen, noch bearbeiten, noch löschen.

In der Tabelle *Datamanagement* wird des Weiteren auf die eindeutigen Tabellennamen verwiesen, auf die die Zugriffe geregelt werden sollen.

Als Benutzername in der Tabelle *User* wird die E-Mail Adresse des Benutzers verwendet, da diese in jedem Falle eindeutig ist und somit eine Überprüfung (auf Eindeutigkeit) recht einfach zu implementieren ist.

In der Tabelle *UserInGroup* wird die Gruppenzugehörigkeit aller User geregelt. Somit kann ein Benutzer sowohl in meheren Gruppen Mitglied sein, als auch mehrere Benutzer in einer Gruppe.
Des Weiteren wird so der Benutzer nicht dazu gezwungen in einer Gruppe Mitglied zu sein, und es werden keine unnötigen Gruppen erstellt.

3 Wichtiges

Es gibt einige Punkte, die sowohl von uns, als auch von unseren Nachfolgern unbedingt berücksichtigt werden müssen.
Diese werden im Folgenden genauer erläutert.

3.1 Leere Einträge

Da auf einen Datamanagement-Eintrag jeweils ein Benutzer sowie bis zu zwei Gruppen Zugriff haben können, haben wir als RIB (Referentielle Integritätsbedingung) festgelegt, dass beim Löschen eines Benutzers bzw. einer Gruppe der Eintrag in der *Datamangement*-Tabelle auf den Wert „NULL" gesetzt werden soll.

Somit behalten die anderen Gruppen bzw. Benutzer den Zugriff, was sicherlich die einizig richtige Lösung sein kann.
Dadurch kann es jedoch auch vorkommen, dass es Datamanagement-Einträge gibt, bei denen alle drei Attribute (UserID, ReadGroupID sowie WriteGroupID) den Wert „NULL" besitzen, hat dieser Eintrag keinen Sinn mehr.

Aus diesem Grund sollte zu einem geeigneten Zeitpunkt (nach einer Woche, einem Monat oder einem Jahr) jeweils überprüft werden, ob es Einträge gibt bei denen das oben beschriebene Problem zutrifft.
Die Tupel, bei denen dies zutrifft, sollen aus der Tabelle *Datamanagement* herausgelöscht werden.

3.2 Eindeutigkeit

Sowohl beim Benutzernamen, als auch beim Gruppennamen und bei den Tabellennamen (der Tabellen CardTable+DID, StackTable+DID und DoorTable+DID) muss die Eindeutigkeit in jedem Falle eingehalten und überprüft werden.

Ansonsten können Fehler auftreten, welche die ganze Benutzer-Funktionalität unbrauchbar machen und somit viel Zeit und Arbeit kosten würden.

4 Zusatzfunktionalitäten

Es gibt noch einige Zusatzfunktionalitäten, welche wir bisher nicht umsetzen konnten, welche aber die Arbeit als Benutzer deutlich erleichtern bzw. verbessern würde.
Dazu gehören die nachstehenden Punkte.

4.1 Download

Hat der Benutzer (zumindest) Lese-Rechte auf ein Door-, Stack- oder Card-Element, so soll er dieses auch herunterladen und lokal abspeichern können.
Die neuen Daten werden in die bestehende lokale Datenbank integriert.

Diese Funktionalität wird dadurch erleichtert, dass der Benutzer seine eigenen Elemente (bei denen er als User mit der UserID Zugriff hat), durch die bereits bestehende Gliederung in drei Tabellen viel einfacher herunterladen und somit integrieren kann.

Die lokal abgespeicherten Daten sollten (wie auch die von Quizlet heruntergeladenen Daten) bearbeitet werden können, jedoch soll die Änderung auf den servergespeicherten Daten nicht übernommen werden.

4.2 Upload

Genau wie bei der Download-Funktion, würde nun hier ein bestimmter Stapel, ein bestimmtes Fach oder auch nur eine bestimmte Karte, vom Benutzer in seiner Datamanagement-Eintrag (in die entsprechende Tabelle) integriert werden.
Auch hierbei erleichtert die von uns gewählte Datenbank-Struktur die Arbeit deutlich.

Durch die Upload-Funktion, kombiniert mit der Download-Funktion wäre es dann möglich z.B. einen Quizlet-Stapel herunterzuladen, diesen lokal zu bearbeiten oder zu verändern und anschliessend in die eigene Struktur auf den WISS-internen Server hochzuladen.