**Dokumentation DBS2**

**Teamvorstellung**

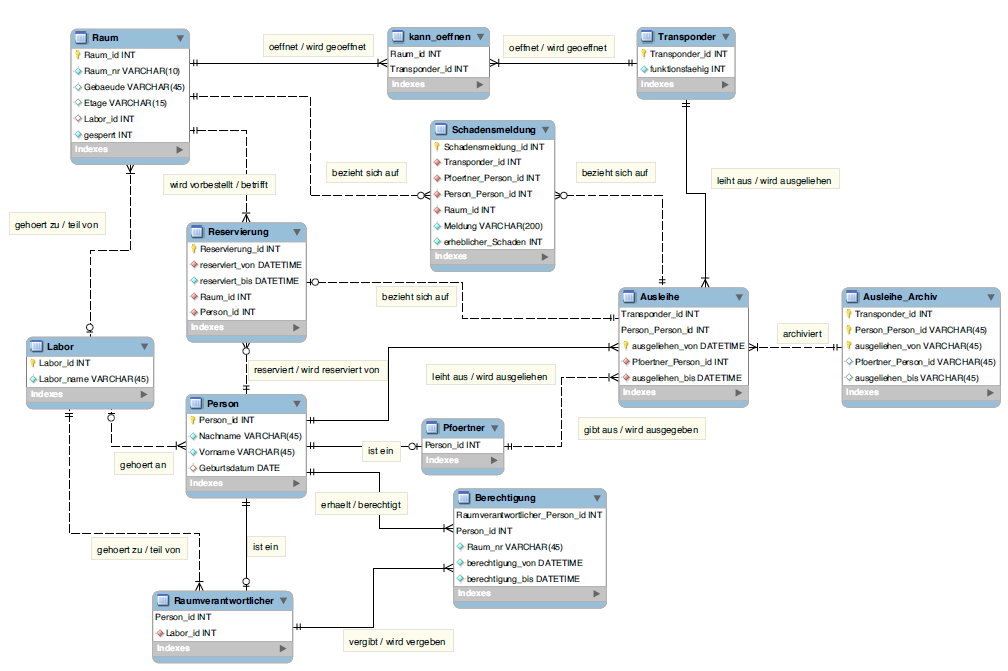
Unser Team besteht aus den Mitgliedern Tobias Adler(AI), Marcel Bröcher(AI), Muhammed Kozan(WI), Daniel Oberle(AI) sowie Lukas Radik(AI). Als Teamleiter haben wir Marcel Bröcher ausgewählt.

**Projektidee**

Unser Projekt befasst sich damit, die Datenbank bezüglich des Transponderverleihs der TH zu optimieren/erstellen. Dementsprechend haben wir uns dazu entschieden es "Transponderverleih" zu nennen. Zurzeit werden jegliche Daten auf Papier gespeichert. Es werden per Hand Listen geführt, was dementsprechend ein hohes Maß an Arbeitsaufwand mit sich bringt, als auch einfach einer "Technischen Hochschule" im Jahr 2020 nicht gerecht wird. Hauptziel ist es somit, die Verwaltung zu digitalisieren und infolgedessen zu vereinfachen.

An erster Stelle stehen aus datenbanktechnischer Sicht also die Transponderaus und Transponderrückgabe, sowie die Verwaltung der des Personals und deren Berechtigungen. Unter dem Punkt Funktionen/Prozeduren fassen wir die benötigten Funktionalitäten zusammen, die dementsprechend benötigten Entitäten und Eigenschaften/Daten können aus dem nachfolgenden ERM entnommen werden.

**ERM**



**Benutzergruppen/Use Cases**

**Ein Bild, das Text, Karte enthält.

Automatisch generierte Beschreibung**

1. Pförtner

Bei der Ausgabe und Reservierung der Transponder, prüft der Pförtner die Berechtigung. Der Pförtner ist für die komplette Verwaltung der Transponder zuständig.

2. Mitarbeiter/Student

Mitarbeiter und Studenten arbeiten entweder an der Hochschule oder sind dort immatrikuliert, sie können Transponder ausleihen, reservieren und zurückgeben.

3. Raumverantwortlicher

Raumverantwortliche kümmern sich um die Räume, sie gehören beispielsweise einem bestimmten Labor an. Sie vergeben die Berechtigung einen Transponder für die dementsprechenden Räume auszuleihen.

**Funktionen/Prozeduren**

1. Transponder ausleihen.

Bei der Ausleihe eines Transponders wird geprüft ob dieser aktuell Funktionsfähig und nicht ausgeliehen ist. Für den entsprechenden Mitarbeiter/Studenten wird geprüft ob dieser berechtigt ist den entsprechenden Transponder auszuleihen. Liegt aktuell eine Schadensmeldung für den Raum vor wird eine Info über eben jene ausgegeben. Liegt ein erheblicher Schaden vor kann der Transponder für diesen Raum nicht ausgeliehen werden. Zusätzlich wird ein Record Ausleihe\_Archiv erstellt, welches identisch mit der Ausleihe ist, jedoch permanent bestehen bleibt.

2. Transponder zurückgeben.

Bei der Rückgabe wird der Status des Transponders auf nicht ausgeliehen zurückgesetzt. Zusätzlich wird für den entsprechend Raum geprüft ob eine Schadensmeldung vorliegt die chronologisch nach dem Datum der Ausleihe des Transponders liegt. Optional: Falls eine Schadensmeldung vorliegt und es sich nicht um Personal, sondern um einen Studenten handelt, wird dieser für die weitere Ausleihe eines Transponders für einen der Räume des betreffenden Labors gesperrt. Liegt ein erheblicher Schaden vor wird zusätzlich eine Warnung ausgegeben sodass der Raumverantwortliche kontaktiert werden kann.

3. Berechtigungen vergeben.

Mitarbeiter vergeben Berechtigungen an das Personal der Hochschule, diese gelten dann als Raumverantwortliche. Da die meisten Räume zu einem Labor gehören, wird geprüft ob das entsprechende Personal zum entsprechenden Labor gehört.

4. Transponder reservieren

Bei der Reservierung eines Raumes wird geprüft, ob die entsprechende Berechtigung vorliegt und der Raum nicht schon reserviert ist. Dabei muss geprüft werden, ob das "von" Datum vor dem "bis" Datum liegt. Ebenfalls muss geprüft werden ob der Raum für den Zeitraum noch gesperrt ist.

**Trigger**

1. Bei der Ausleihe oder Reservierung eines Raumes, wird geprüft, ob die entsprechende Berechtigung noch gültig ist. Ist dies nicht der Fall, wird eine Exception geworfen.
2. Ebenfalls bei der Ausleihe sowie Reservierung wird geprüft, ob für den entsprechenden Zeitraum nicht bereits eine Reservierung/Ausleihe vorliegt. Ist dies der Fall, wird dem Benutzer ein alternativer Termin vorgeschlagen. Ist er mit diesem einverstanden, wird Transponder ausleihen ausgeführt.
3. Wird einer Person seine Berechtigung entzogen, werden alle betreffenden, noch offenen Reservierungen gelöscht.
4. Wird ein neuer Raum eingefügt, bekommen die entsprechenden Raumverantwortlichen eine Nachricht.
5. Werden Personen exmatrikuliert oder arbeiten nicht mehr für die Hochschule, werden sie nach einem Jahr aus der Datenbank gelöscht. Dementsprechend werden auch alle Berechtigungen gelöscht.
6. Wird eine Schadensmeldung erstellt, wird geprüft ob der Schaden erheblich ist. Ist dies der Fall wird der Raum als gesperrt markiert. Alle Reservierung, die im Zeitraum der Sperrung liegen, werden gelöscht und die Benutzer erhalten dementsprechend eine Nachricht.

**View mit Instead-of-Trigger**

Unsere View zeigt uns für einen bestimmten Raumverantwortlichen, alle Personen, für die er eine Berechtigung erteilt hat, sowie deren Ausleihhistorie. Personen können aus der View entfernt werden, um die Entfernung von Berechtigungen umzusetzen. Dies darf jedoch nur passieren, wenn die Person aktuell keinen Transponder ausgeliehen hat. Unser Instead-of-Trigger ersetzt die Delete Funktion, und benachrichtigt stattdessen den Hausmeister. Dieser kann sich nun darum kümmern, die Person aufzuspüren und den Transponder einzufordern.

**Aufgabenplanung**

|  |  |  |
| --- | --- | --- |
| **#** | **Bezeichnung** | **Verantwortlich** |
| **Function/Procedure** |  |  |
| Proc 1 | Transponder ausleihen | Daniel |
| Proc 2 | Transponder zurückgeben | Tobias |
| Proc 3 | Berechtigung vergeben | Muhammed |
| Proc 4 | Reservierung erstellen | Daniel |
| **Trigger** |  |  |
| Instead-of-Trigger/View | Berechtigung entfernen | Lukas |
| Trig 1 | Prüfung Berechtigung | Daniel |
| Trig 2 | Prüfung Ausleihe | Tobias |
| Trig 3 | Reservierung löschen | Muhammed |
| Trig 4 | Nachricht Raumverantwortliche | Marcel |
| Trig 5 | Records löschen | Lukas |
| Trig 6 | Schadensmeldung prüfen | Marcel |
| **Implementierung** |  |  |
| ERM MYSQL | ERM in MYSQL implementieren | Lukas |
| EERM Oracle | EERM in Oracle implementieren | Tobias |
| Trigger MYSQL | Trigger & Testfälle in MYSQL implementieren | Lukas, Marcel |
| Trigger Oracle | Trigger & Testfälle in Oracle implementieren | Tobias, Daniel |
| **Doku/Formalien** |  |  |
| Dokumentation | Dokumentation aktuell halten | Marcel |
| Upload | Fristgerechter Upload aller Artefakte | Marcel |