Projektdokumentation

TeamII1

ai-Card

Verantwortliche für das Dokument: Antonio Blechschmidt

Erstellt am: 22.12.2020

Zuletzt geändert am: 23.12.2020

**Inhalt**

1. **Beschreibung des Projekts SEITENZAHLEN FOLGT**
2. **Pakete**
   1. **Account**
      1. **Account**
      2. **Student**
      3. **Professor**
      4. **LearnSetAbo**
   2. **Card**
      1. **Card**
      2. **CardContent**
      3. **Mediaklassen**
      4. **CardStatus**
   3. **LearnSet**
      1. **LearnSet**
      2. **LearningSession**
      3. **CardList**
   4. **Chat**
      1. **Chat**
      2. **Groups**
      3. **MessageList**
      4. **Message**
      5. **AccountList**
   5. **Enums**
3. **Ausgangssituation**

Für die Projektarbeit in Java 1 im dritten Fachsemester soll die Verarbeitungslogik für ein Java Programm entwickelt werden. Das macht man in Gruppenarbeit mit fünf Personen alles Ingenieurinformatiker mit entsprechender Kreativität. Eine der fünf Personen hatte die Idee ein Kartenlernsystem zu programmieren, um das Lernen im Hochschulstudium zu digitalisieren. Das Kartenlernsystem hat viele Vorteile wie, wie z.B. dass Studiumsunterlagen einfach mit Studierenden geteilt werden können. Zudem bietet die Platform eine Möglichkeit zur Kommunikation zwischen den Lehrenden und Studierenden.

Also entschieden wir uns ein Multiuser Kartenlernsystem zu entwickeln.

1. **Beschreibung**

Es soll ein Multiuser Kartenlernsystem für Studierende und Lehrende (SuL) als Pilotprojekt an der FH Erfurt entwickelt werden. Um das System nutzen zu können, müssen die Studierenden und Lehrenden ein Benutzerkonto erstellen. Hierzu sind die Angaben Hochschul-E-mail und Passwort zwingend erforderlich, da diese zum Login benötigt werden. Studierende können sich ihrem Studienfach zuordnen. Alle können Lernsets, bestehend aus maximal 200 Karten anlegen. Ein Lernset muss beim Erstellen benannt und beschrieben werden. Lernsets können privat, publik (jedes Mitglied der Hochschule) oder geschützt (auswählbar, wer es sehen darf) sein. Eine Karte besteht aus zwei Seiten (Vorder- und Rückseite).

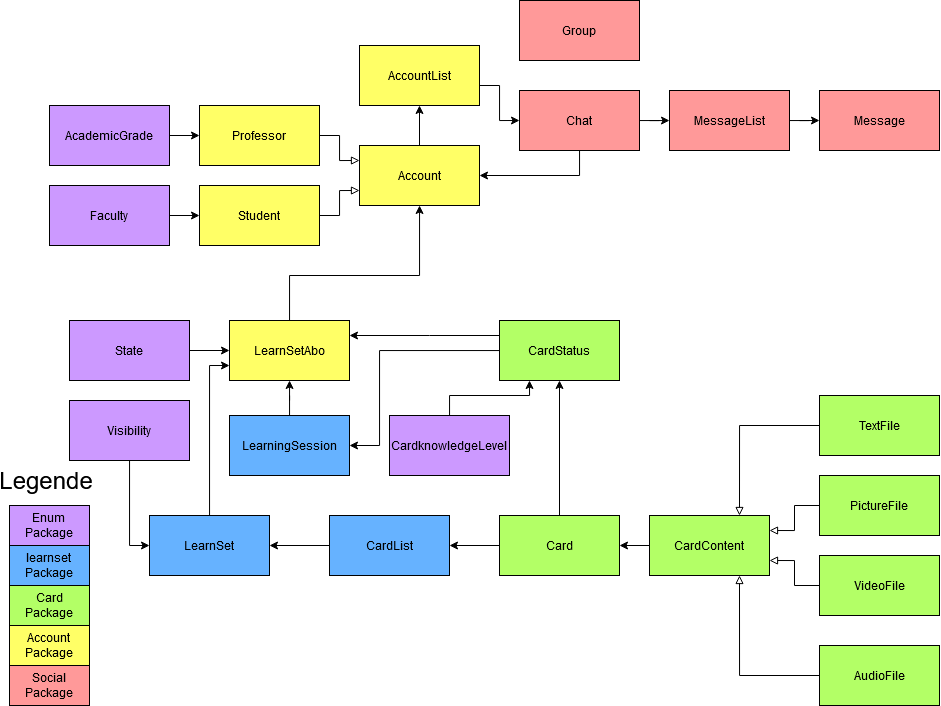
User können über eine Suchfunktion Lernsets finden, die für sie freigegeben sind, und sie zu ihrem Benutzerkonto hinzufügen (abonnieren). Hinzugefügte Lernsets können als Favorit markiert werden oder archiviert werden. Innerhalb eines Lernsets können einzelne Karten ausgeblendet werden. Mit den einem Benutzerkonto hinzugefügten Lernsets können Abfragen gestartet werden. Hierbei kann man auswählen, wie viele Karten in einer Abfrage abgefragt werden sollen: 10, 20, 30, 40 oder 50. Während der Abfrage ist von dem Benutzer selbst anzugeben, ob er die Karte gewusst hat oder nicht. Anhand dieser Angaben über alle bisherigen Abfragen wird der Karte ein Status der Kategorien 1 bis 5 im Benutzerkonto hinzugefügt. Hierbei ist 1 die Kategorie für die am wenigsten gekonnten Karten, 5 die für die fertig gelernten Karten.

Benutzerkonten können sich befreunden, sowie Gruppen bilden. Desweiteren können gemeinsame Gesprächsräume nutzen. Gruppen können gemeinsam Lernsets anlegen und bearbeiten.

1. **Pakete**

ai-Card ist in logische Komponenten aufgeteilt. Es gibt folgende Pakete mit ihrer Funktionalität:

* Account Paket: verwaltet nutzerspezifische Daten
* Card Paket: bietet Grundstruktur für Lernkarten
* LearnSet Paket: ermöglicht strukturierten Zugriff auf Lernkarten
* Social Paket: Kommunikation zwischen Nutzern
* Enums Paket: Enummeration



In den folgenden Abschnitten werden diese näher erläutert.

* 1. **Account**

Das Account Paket soll nutzerspezifische Daten verwalten. Sie ist in folgende Klassen unterteilt:

* Account: Grundlage für Nutzerdaten
* Student: accountspezifische Daten für Studenten
* Professor: accountspezifische Daten für Professoren
* LearnSetAbo verwalten der accountspezifischen Lernkartendecks
* AccountList Gruppierung von Benutzerkonten
  + 1. **Account**

Grundlage für das festhalten von Nutzerdaten

|  |
| --- |
| Account : abstrakt |
| - email : String  - password : String  - name : String  - description : String  -ownedLearningSets : List<LearnSetAbo>  -favoriteSets : List<LearnSetAbo>  -friends : AccountList  -chats: List<Chat>  -groups : List<Groups> |
| +setter/getter  +getOwnedLearnSetAboByPosition(int \_position) : LearnSetAbo  +createNewOwnedLearnSet (String \_title, String \_description, Faculty \_faculty)  +deleteFromOwnedLearningSetsByIndex(int \_index)  +deleteFromOwnedLearningSetsLastElement  +deleteAllFromOwnedLearningSets  +getFavoriteSetByPosition(int \_position) : LearnSetAbo  +addNewFavoriteSets(LearnSetAbo \_favoriteSet)  +deleteFromFavoriteSetsByIndex(int \_index)  +deleteFromFavoriteSetsLastElement(Account \_friend)  +deleteAllFromFavoriteSets(Account \_friend)  +addFriend(Account \_friend)  +removeFriend(Account \_friend)  +removeFriend(int \_friend)  +createGroup(String \_name , Visibility \_publicity)  +joinGroup(Group \_group)  +leaveGroup(Group \_group)  +likeMessage (Message \_message) : boolean  +dislikeMessage(Message \_message) : boolean  +editMessage(Message \_message, String message)  +deleteMessage(Message \_message, MessageList \_messagelist)  +deleteChat(Chat \_chat) |

* + 1. **Student**

Spezialisierung von Account

|  |
| --- |
| Student |
| -semester : int  -faculty : enum |
| +setter/getter |

* + 1. **Professor**

Spezialisierung von Account

|  |
| --- |
| Professor |
| -AcademicGrade : enum |
| +setter/getter |

* + 1. **LearnSetAbo**

verwalten der accountspezifischen Lernkartendecks

|  |
| --- |
| LearnSetAbo |
| -m\_cardStatus : ArrayList<CardStatus>  -m\_learnSetStatus : State  -m\_learnSet : LearnSet  -m\_evaluation : int |
| +getter/setter  +delete\_evaluation  -getCardStatusOfKnowledgeLevel(CardKnowledgeLevel level) : ArrayList<CardStatus>  -createCardStatusListForSession(int \_numOfCards) : ArrayList<CardStatus>  +createLearningSession(int \_numOfCards) : LearningSession |

**2.1.5. AccountList**

Sammlung an Accounts für Gruppen und Gespräche mittels einer Arraylist. Wir haben uns gemeinsam entschieden(gemeinschaftlich) dass die Benutzerkontoliste in das Account set gehört, obwohl es auch in das Sozial Paket passen könnte.

|  |
| --- |
| AccountList |
| -Accounts : ArrayList<Account> |
| + setter/getter  + addPerson(Account \_acc)  + removePerson(Account \_acc)  + removePerson(int \_acc)  + contain(Account \_acc) : boolean |

* 1. **Card**

Das Card Packet ist die Grundstruktur für alle Lernkarten. Sie ist in folgende Klassen unterteilt:

* Card – Grundlage für eine Lernkarte
* CardContent – Superklasse der Medienklassen
* Medienklassen – ermöglicht Zugriff auf die unterschiedlichen Medien
  + TextFile
  + PictureFile
  + AudioFile
  + VideoFile
* CardStatus - hält den Lernfortschritt einer Karte fest für einen Account
  + 1. **Card**

Grundlage für eine Lernkarte

|  |
| --- |
| Card |
| -cardFront : CardContent  -cardBack : CardContent |
| +setter/getter |

* + 1. **CardContent**

Interface aller Medienklassen

|  |
| --- |
| CardContent : Abstract |
|  |
|  |

* + 1. **Mediaklassen**

Ermöglicht den Zugriff auf die unterschiedlichen Medien. So können TextFile, PictureFile, AudioFile, VideoFile, CardStatus in die Karten eingebunden werden

|  |
| --- |
| AudioFile |
| - m\_AudioData : String - m\_isPlaying : boolean - m\_Title : String |
| +setter/getter  + pauseAudioData : String  + playAudioData : String |

Rückgabe String ist ein Platzhalter

|  |
| --- |
| PictureFile |
| - m\_PictureData : String  - m\_Title : String |
| +setter/getter |

|  |
| --- |
| TextFile |
| - m\_TextData : String |
| +setter/getter |

|  |
| --- |
| VideoFile |
| - m\_VideoData : String - m\_isPlaying : boolean - m\_Title : String |
| +setter/getter  + pauseVideoData : String  + playVideoData : String |

Rückgabe String ist ein Platzhalter

**2.2.4 Cardstatus**

Speichert den Lernfortschritt einer Karte fest für einen Account innerhalb eines LernSetAbos.

|  |
| --- |
| CardStatus |
| - m\_CardKnowledgeLevel : CardKnowledgeLevel - m\_Card : Card |
| +setter/getter  +increaseCardKnowledgeLevel  +decreaseCardKnowledgeLevel  +resetCardKnowledgeLevel |

* 1. **LearnSet**

Das Learnset Paket ist zuständig für den Zugriff auf die Karten und bildet so die Brücke zwischen Account und Karten. Folgende Klassen sind im LearnSet Packet.

* LearnSet – ist eine Liste an Karten, zu einer bestimmten Thematik
* LearningSession – zuständig für das Lernen der Karten
* CardList – Speichert Karten
  + 1. **LearnSet**

Ist eine Liste an Karten, zu einer bestimmten Thematik,welche von Benutzern angelegt werden können.

|  |
| --- |
| LearnSet |
| - m\_Title : String - m\_Description : String  - m\_Faculty : Faculty - m\_CardList : CardList - m\_CommentList : MessageList  - m\_Owner : Account - m\_Visibility : Visibility - m\_Admins : ArrayList<Account> - m\_Evaluation : double - m\_NumberOfEvaluations : int |
| +setter/getter  +createCardList()  +addEvaluation(int \_newEvaluation)  +deleteEvaluation(int \_EvaluationToDelete)  +increaseNumberOfEvaluations  +decreaseNumberOfEvaluations  +addAdmin(Account \_newAdmin)  +removeAdminByIndex(int \_IndexToRemove)  +removeAdminByAccount(Account \_AccountToRemove)  +addMessage(Message \_newMessage)  +removeMessageByMessage(Message \_MessageToRemove) |

* + 1. **LearningSession**

In einer LearningSession werden, als noch nicht gelernte, markierte Karten, in einer temporären Sitzung einsortiert um von einem Benutzer gelernt zu werden.

|  |
| --- |
| + LearningSession |
| - currentCard : int  - cardStatusList : ArrayList<CardStatus> - isActive : boolean |
| + setter/getter  +cardKnown(): void  +cardUnKnown(): void |

* + 1. **CardList**

Speichert Karten in einer Arraylist ab und ist die Grundstruktur für LearningSeassion und LearnSet**.** Benutzer können Karten hinzufügen,bearbeiten und löschen.

|  |
| --- |
| CardList |
| - m\_cardList : ArrayList<Card>  - m\_ListIndex : int |
| +setter/getter  +addToList(Card)  +removeFromList(Card)  +removeFromList(int)  +getCurrentCard():Card  +getListLength():int  +getCardByIndex(int) : Card  +next()  +previous() |

* 1. **Social**

Das Chat Packet soll die Möglichkeit bieten, das sich Nutzer untereinander austauschen. Im Packet sind folgende Klassen:

* Chat – Konversation zwischen mehreren Benutzerkonten
* Group – Eine Gemeischaft an Benutzerkonten – kann später Implementiert werden aber das Grundgerüst ist bereits vorhanden
* MessageList – speichert Nachrichten
* Message – eine Nachricht in einen Chat
* AccountList – Sammlung an Benutzerkonten
  + 1. **Chat**

Ist eine Konversation zwischen mehreren Benutzerkonten. Enthält einen vollständigen Chatverlauf. Das erste Benutzerkonto in der Liste participants ist der Eigentümer der Gruppe.

|  |
| --- |
| Chat |
| -chathistory : MessageList  -participants : AccountList |
| +setter/getter  +void addParticipant(Account \_participant)  + Account getChatCreator()  + clearHistory()  + sendmessage(String \_message, Account \_sender)  + removemessage (Message \_message) |

* + 1. **Group**

Enthält eine Liste der Mitglieder. Diese können gemeinschaftlich Lernsets erstellen und bearbeiten. Zudem haben sie einen gemeinsamen Gesprächsraum. Das erste Benutzerkonto in der Liste members ist der Eigentümer der Gruppe.

|  |
| --- |
| Group |
| - name : String - members : AccountList - publicity : Visibility - groupLearnSets : ArrayList<LearnSet> |
| +setter/getter  +getGroupCreator() : Account  +addToGroup(Account \_acc)  +removefromGroup(Account \_acc) |

* + 1. **MessageList**

Grundstruktur für Chat um Nachrichten zu speichern. Enthält eine Liste von Message.

|  |
| --- |
| MessageList |
| - messagelist : ArrayList<Message> |
| +setter/getter  + addMessage(Message \_message)  + removeMessage(Message \_message)  + getNumberMessages() |

* + 1. **Message**

Eine Message kann von einem Account geschrieben und geliked werden.

|  |
| --- |
| Message |
| - message : String - sender : Account - time : LocalDateTime - likes : int - likedby : AccountList |
| +setter/getter  +raise\_likes()  +newliker (Account \_account)  +remove\_Like()  +lostliker (Account \_account) |

* 1. **Enums**
* Faculty das sind die Fakultäten der Hochschule
  + Wird für die Sichtbarkeit der Lernsets benötigt. Z.B. damit Lernsets nur Mitgliedern einer bestümmten Fakultät angezeigt werden
* CardKnowledgeLevel enthält das aktuelle Wissenslevel der Karte
  + Wird für die Auswahl der Karte in eine LearningSession benötigt
* Visibility die Anzeigeeinstellungen für Lernsets und Gruppen
* State zeigt ob ein Lernset neu, in Bearbeitung oder fertig gelernt ist.
* AcademicGrade Akademische Grade für die Professor Klasse.

Sind Eine Sammlung an Klassen die den Code lesbarer machen

|  |  |
| --- | --- |
| Faculty | APPLIEDCOMPUTERSCIENCE,  ARCHITECTURE,  CIVILENGINEERING, MANAGEMENTANDLEADERSHIPINEARLYCHILDHOODEDUCATION,  BUSINESSADMINISTRATION, FORESTRYANDECOSYSTEMMANAGEMENT,  HORTICULTURE,  BUILDINGSERVICESENGINEERING,  LANDSCAPEARCHITECTURE,  EARLYCHILDHOODEDUCATION,  SOCIALWORK,  URBANANDSPATIALPLANNINGFOUNDATIONS,  RAILWAYENGINEERING, ENERGYENGINEERING, TRAFFICTRANSPORTATIONANDLOGISTICSENGINEERING |
| CardKnowledgeLevel | NOINFORMATION,  SOMEINFORMATION,  KNOW,  KNOWWELL,  KNOWVERYWELL |
| Visibility | PRIVATE, PUBLIC, PROTECTED |
| State | NEW, PROCESSING, LEARNED |
| AcademicGrade | UNIVERSITY\_PROFESSOR, UNIVERSITY\_LECTURER, JUNIOR\_PROFESSOR, JUNIOR\_LECTURER, ACADEMIC\_COUNCIL, SCIENTIFIC\_ASSISTANT, RESEARCH\_ASSISTANT, STUDENT\_ASSISTANT |