Projektdokumentation

TeamII1

ai-Card

Verantwortliche für das Dokument: Antonio Blechschmidt

Erstellt am: 22.12.2020

Zuletzt geändert am: 23.12.2020

Inhalt

[1. Ausgangssituation 3](#_Toc62816618)

[2. Beschreibung 3](#_Toc62816619)

[3. Pakete 4](#_Toc62816620)

[3.1 Account 5](#_Toc62816621)

[3.1.1 Account 5](#_Toc62816622)

[3.1.2 Student 5](#_Toc62816623)

[3.1.3 Professor 6](#_Toc62816624)

[3.1.4 LearnSetAbo 6](#_Toc62816625)

[3.1.5 AccountList 6](#_Toc62816626)

[3.2 Card 6](#_Toc62816627)

[3.2.1 Card 7](#_Toc62816628)

[3.2.2 CardContent 7](#_Toc62816629)

[3.2.3 Mediaklassen 7](#_Toc62816630)

[3.2.4 Cardstatus 8](#_Toc62816631)

[3.3 LearnSet 8](#_Toc62816632)

[3.3.1 LearnSet 8](#_Toc62816633)

[3.3.2 LearningSession 9](#_Toc62816634)

[3.3.4 CardList 9](#_Toc62816635)

[3.4 Social 9](#_Toc62816636)

[3.4.1 Chat 10](#_Toc62816637)

[3.4.2 Group 10](#_Toc62816638)

[3.4.3 MessageList 10](#_Toc62816639)

[3.4.4 Message 10](#_Toc62816640)

[3.5 Enums 11](#_Toc62816641)

# Ausgangssituation

Für die Projektarbeit in Java 1 im dritten Fachsemester soll die Verarbeitungslogik für ein Java Programm entwickelt werden. Hierzu haben sich in der Gruppe *TeamII1* fünf Personen zusammengefunden, die nicht nur die Vertiefungsrichtung Ingenieursinformatik gemeinsam haben, sondern auch die Kreativität eines Ingenieursinformatikers. Eine der fünf Personen hatte die Idee ein Kartenlernsystem zu programmieren, um das Lernen im Hochschulstudium zu digitalisieren. Dieses soll es ermöglichen, Lernunterlagen einfach mit anderen Studierenden zu teilen und gleichzeitig auch über die Plattform mit anderen Studierenden und Lehrenden zu kommunizieren.

Also entschieden wir uns ein Multiuser-Kartenlernsystem zu entwickeln.

# Beschreibung

Ziel des Projektes ist es, ein Multiuser-Kartenlernsystem für Studierende und Lehrende als Pilotprojekt and er FH Erfurt zu entwickeln. Um das System nutzen zu können, müssen die Studierenden und Lehrenden einen Account erstellen. Hierzu sind die Angaben Hochschul-Email 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, sogenannte Learning-Sessions. Hierbei kann man auswählen, wie viele Karten in einer Abfrage abgefragt werden sollen: 10, 20, 30, 40 oder 50. Es werden automatisiert in einer LearningSession genau die Karten abgefragt, die über den niedrigsten Status verfügen. 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 Status der Karte aktualisiert.

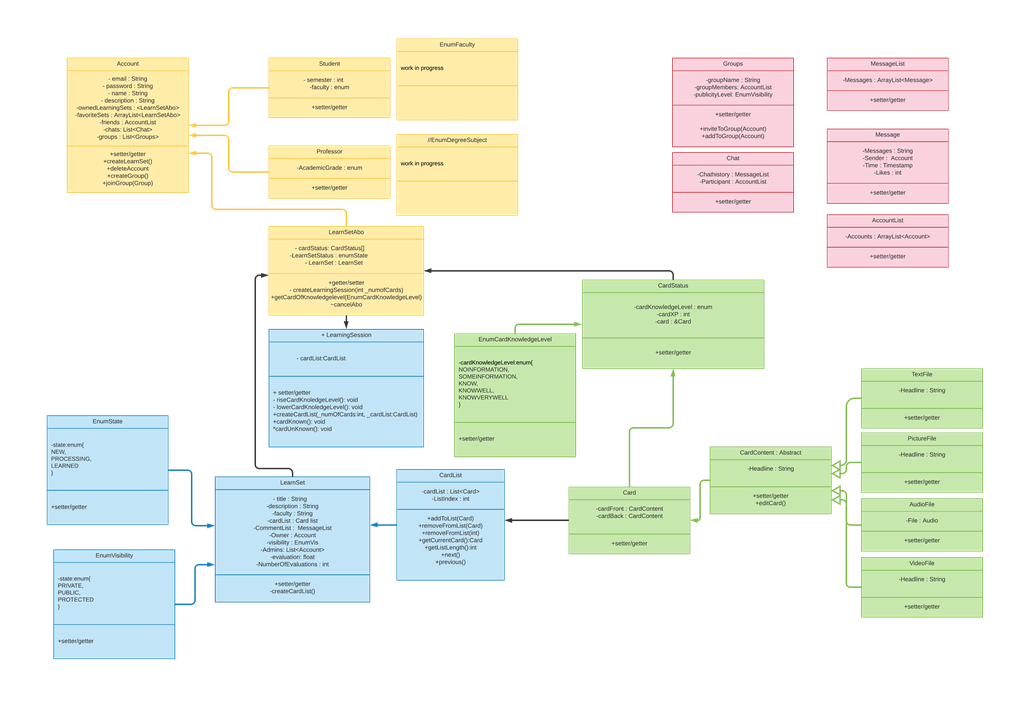
Accounts können Chats mit anderen Accounts erstellen und sich hierüber austauschen. Dem Austausch unter den Accounts dient des Weiteren die Möglichkeit, LearnSets zu kommentieren und sie zu bewerten.

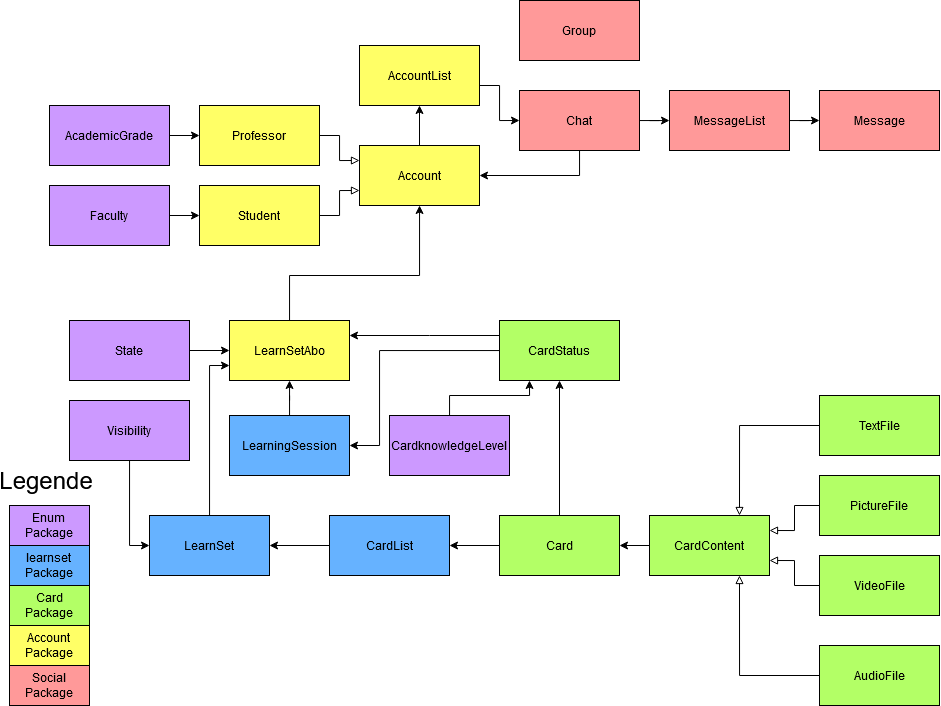
Ursprünglich wurde darüber hinaus überlegt, dem System eine Funktion hinzuzufügen, die es den Accounts ermöglicht, Gruppen mit anderen Accounts zu bilden und hierüber gemeinsam LearnSets erstellen zu können. Diese Funktion wurde jedoch nicht umgesetzt, da die Gruppe sich dazu entschied, lieber ein Augenmerk auf die Qualität der Kernfunktionen des Kartenlernsystems zu legen. Gänzlich verabschiedet haben wir uns von dieser Funktion jedoch noch nicht – eventuell erfolgt eine Implementierung in Java 2. Das bisher entstandene Grundgerüst der Gruppen-Klasse wurde im Projekt auskommentiert und alle vorhandenen Klassen wurden so gestaltet, dass eine Erweiterung des Gesamtsystems um eine Gruppen-Funktion ohne weiteres möglich ist.

# Pakete und Klassen

Ai-Card ist in funktionale Klassen aufgeteilt, die wiederum in Pakete zugeordnet sind, die einer Teilaufgabe des Gesamtsystems dienen:

* 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: enthält alle Enumeration des Projekts





## Account

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

* Account: Grundlage für Nutzerdaten
* Student: Account spezifische Daten für Studierende
* Professor: Account spezifische Daten für Lehrende

### Account

Grundlage für das Festhalten von Nutzerdaten. Neben Daten, die bereits bei Accounterstellung angegeben werden, wie Email, Passwort, Name und eine Beschreibung des Accounts, werden hier auch die Informationen gespeichert, die dazu notwendig sind, dass der Account die Kernfunktionen des Programms – das heißt die Karten-Lernfunktion sowie die Chatfunktion – nutzen kann. Dies sind zum einen zwei Listen, in denen die eigenen LearnSets sowie die abonnierten LearnSets gespeichert werden, zum anderen eine Liste der Accounts, mit denen der User befreundet ist, sowie eine Liste der Chats, an denen der User teilnimmt.

Tabelle : Klasse Account

|  |
| --- |
| Account : abstrakt |
| - email : String  - password : String  - name : String  - description : String  -ownedLearningSets : List<LearnSetAbo>  -favoriteSets : List<LearnSetAbo>  -friends : AccountList  -chats: List<Chat> |
| +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) |

### 3.1.2 Student

Spezialisierung von Account

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

### 3.1.3 Professor

Spezialisierung von Account

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

### 3.1.4 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 |

### 3.1.5 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 |

## 3.2 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

### 3.2.1 Card

Grundlage für eine Lernkarte

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

### 3.2.2 CardContent

Interface aller Medienklassen

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

### 3.2.3 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

### 3.2.4 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 |

## 3.3 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

### 3.3.1 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) |

### 3.3.2 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 |

### 3.3.4 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() |

## 3.4 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

### 3.4.1 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) |

### 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) |

### 3.4.3 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() |

### 3.4.4 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) |

## 3.5 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 |