

TIMA

The word "TIMA" is rendered in a bold, dark blue, sans-serif font. Two bright blue arrows are integrated into the design: one starts above the 'M', points down into its central negative space, and then continues as a horizontal line to the right; the other starts below the 'A', points up into its central negative space, and then continues as a horizontal line to the left, meeting the first arrow's path to form a continuous loop.

Abschlussarbeit

TIMA

Nathanael Philipp, Felix Rauchfuß, Kai Trott

11. August 2015



Inhaltsverzeichnis

1	Einleitung	1
2	Assoziationsdatenbank und Website	2
2.1	Backend und Datenbank	2
2.1.1	Datenmodell	2
2.2	Webfrontend	4
2.3	API	4
3	App	5
4	ausblick	6
5	Zusammenfassung	7

1 Einleitung

2 Assoziationsdatenbank und Website

Der Hauptteil von TIMA ist die Datenbank in denen die Assoziation gespeichert werden. Diese ist direkt verknüpft mit dem Webfrontend, das sowohl der Hauptanlaufpunkt für die Benutzer ist als auch für die Apps durch die Bereitstellung einer umfassenden API.

Im nächsten Abschnitt wird das Backend und die Datenbank genauer beschrieben. Dabei wird genauer auf den Aufbau der einzelnen Tabellen eingegangen, sowie die verschiedenen Designentscheidungen.

Anschließend wird die API und die dahinter stehenden Designentscheidung genauer erläutert.

2.1 Backend und Datenbank

Für das Backend der Website haben wir Django als grundlegende Technologie entschieden. Bei Django handelt es sich um ein in Python geschriebenes Webframework, das dem Model-View-Controller-Schema folgt. Django bietet unter anderem einen sehr komplexen objektrelationalen Mapper, der es ermöglicht auch komplexe Objektstrukturen abzubilden ohne die verwendete Datenbank explizit zu kennen.

2.1.1 Datenmodell

In Abbildung 2.1 ist das komplette Datenmodell von TIMA dargestellt. Dabei bildet das Modul `associations.models` den Kern.

Das Modell `Language` repräsentiert die verfügbaren Sprachen der Wörter, das Modell `Word` die einzelnen Wörter, wobei ein Wort, wenn es in mehreren Sprachen existiert, für jede Sprache einen eigenen Eintrag hat und das Modell `Association` die Assoziation zwischen zwei Wörtern. Zusätzlich wird bei einem Wort gespeichert, wie oft dieses gefragt wurde und bei der Assoziation wie häufig diese gegeben wurde.

Die Modelle des Moduls `app.models` behandeln grundlegende Funktionen des Benutzermanagements. So zum Beispiel das Modell `Profile`, das den kulturellen Hintergrund, die Punktzahl und die Sprachen für die ein Benutzer assoziiert hat,

2 Assoziationsdatenbank und Website

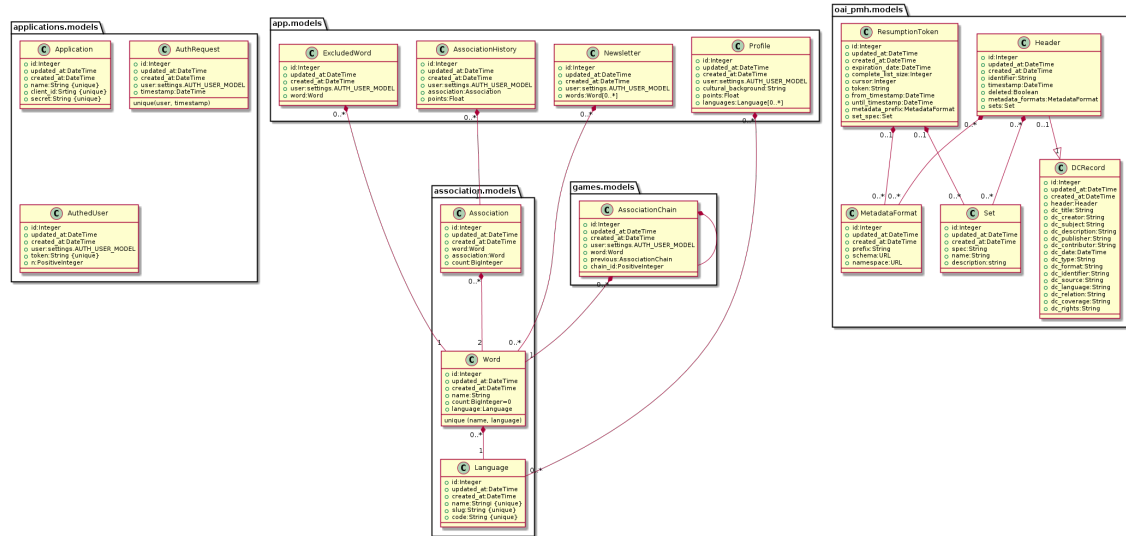


Abbildung 2.1: UML des TIMA Datenmodells

speichert. In dem Modell **AssociationHistory** wird die gesamte Assoziationsgeschichte eines Benutzers gespeichert, mit den jeweils für eine Assoziation erhaltenen Punkte. Das Modell **ExcludeWord** enthält für jeden Benutzer die Wörter, die er übersprungen hat (siehe TODO ref), diese werden automatisch nach sieben Tagen gelöscht. Das letzte Modell in diesem Modul speichert für jeden Benutzer welche Worte er in seinem Newsletter empfangen möchte.

Das Modul **games.models** enthält Modelle die für verschiedenen Spiele wichtig sind. Dies ist im Moment nur AssoziationsKette (vgl. TODO ref), hierfür werden in dem Modul **AssociationChain** die letzte/aktuelle Assoziationskette eines Benutzer gespeichert. Diese wird bei jedem Start eines Spieles gelöscht.

Für die Kommunikation zwischen App und Backend, insbesondere die schreib Zugriffe (vgl. TODO ref), zu autorisieren und dem speichert der nötigen Informationen dient das Modul **applications.models**. Das Modell **Applicaion** speichert die Apps, die Autorisiert sind, mit den nötigen Daten für die Autorisierung (vgl. TODO ref). Die beiden anderen Modell in diesem Modul **AuthRequest** und **AuthedUser** speichern die nötigen Information für einen Benutzer der sich authentifizieren möchte und hat.

Das letzte Modul und die beinhalteten Modell sind für das OAI-PMH erforderlich siehe dafür (TODO ref).

2.2 Webfrontend

Das Webfrontend ist die Hauptanlaufstelle für Benutzer. Hierüber kann sowohl anonym als auch angemeldet Assoziationen eingeben, Wörter und deren Assoziationen angesehen und weitere Funktionen (Rangliste, Statistik) aufgerufen werden.

Das Webfrontend basiert auf Django, wurde zusätzlich zu HTML mit Bootstrap und JQuery erstellt, sowie zur Visualisierung D3.

2.3 API

API.md schöner schreiben, auth uml
bisschen OAI-PMH

3 App

grundlegende APP mit Qt/Qt-quick/QML

4 ausblick

5 Zusammenfassung