THM | CAMPUS GIESSEN | MNI
Mathematik, Naturwissenschaften und Informatik
TECHNISCHE HOCHSCHULE MITTELHESSEN

CS1023 - Softwaretechnik Projekt

TALS
nice to see you.

Von
Christian Breternitz
Lars Herwegh
Johannes Meintrup
Jonas Nimmerfroh

Dozent: Dr. habil. Frank Kammer
Wintersemester 2017/2018

Inhaltsverzeichnis

Um der Geschlechtervielfalt auch in der Sprache gerecht zu werden, wird im vorliegenden Dokument der sog. Gender_Gap[1] verwendet.

1 https://de.wikipedia.org/wiki/Gendergap_(Linguistik) (Stand: 29. Dezember 2017)

1 Projektvorbereitung

Die Durchführung des Projekts fand im Wintersemester 2017/2018 im Rahmen des Bachelor-Moduls „CS1023 Softwaretechnik Projekt“ unter Anleitung von Herrn Dr. habil. Frank Kammer statt.

Das Team bestand aus den Studierenden

- Christian Breternitz, Bio-Informatik
- Lars Herwegh, Ingenieur-Informatik
- Johannes Meintrup, Informatik
- Jonas Nimmerfroh, Ingenieur-Informatik

Zur Realisierung wurde das Vorgehensmodell Scrum[2] für die agile Softwareentwicklung angewandt. Hierbei übernahm Herr Kammer die Funktion des Stakeholders. Herr Herwegh übernahm die Funktion des Scrum-Masters. Die Rolle des Product Owners wurde aufgrund der Umstände kollektiv von allen beteiligten Studierenden übernommen. Zudem waren alle beteiligten Studierenden als Entwickler tätig.

Um das Projekt gemeinsam zu realisieren wurde als Versionsverwaltung Git[3] verwendet. Dabei wurde von Herr Herwegh ein zentrales Repository[4] eingerichtet und für die Entwicklung auf eine Feature-Branch-Strategie[5] gesetzt. In dieser Strategie wird für die Entwicklung von neuen Funktionen nicht auf dem `master`-Branch gearbeitet, sondern jeweils ein neuer Branch erstellt. Dadurch bleibt der `master` immer stabil. Erst, wenn das neue Feature fertig entwickelt und stabil ist, wird es in den `master`-Branch gemerged.

Zu Beginn wurde die Software Jira[6] für das Projektmanagement eingesetzt. Mit Jira können sog. Sprints[7] nach Vorbild von Scrum erstellt und abgearbeitet werden. Dabei ist sichtbar, wer welches Arbeitspaket gerade bearbeitet und in welchem Stadium sich das Arbeitspaket befindet. Dies ist besonders hilfreich bei der Organisation von verzahnten Arbeitsprozessen die aufeinander aufbauen bzw. voneinander abhängig sind. Im Laufe des Projekts stellte sich heraus, das die Arbeitsgebiete sehr gut voneinander abgegrenzt sind und vom Arbeitsumfang für eine Person geeignet sind. Daher kristallisierte sich recht

2 Scrum ist ein Vorgehensmodell des Projekt- und Produktmanagements, insbesondere zur agilen Softwareentwicklung. Mehr Informationen unter https://de.wikipedia.org/wiki/Scrum (abgerufen am 29. Dezember 2017).
3 Git ist eine freie Software zur verteilten Versionsverwaltung von Dateien, die durch Linus Torvalds initiiert wurde. Mehr Informationen unter https://de.wikipedia.org/wiki/Git (abgerufen am 29. Dezember 2017).
4 Repository (englisch für *Lager*, *Depot* oder auch *Quelle*; deutsch Plural *Repositorys*). Mehr Informationen unter https://de.wikipedia.org/wiki/Repository (abgerufen am 29. Dezember 2017).
5 Mehr Informationen unter https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow (abgerufen am 29. Dezember 2017).
6 Jira ist eine Webanwendung zur Fehlerverwaltung, Problembehandlung und operativem Projektmanagement. Mehr Informationen unter https://de.wikipedia.org/wiki/Jira_(Software) (abgerufen am 29. Dezember 2017).
7 Siehe Fußnote zu „Scrum“

schnell eine wohldefinierte Arbeitsteilung heraus. Dabei bearbeitete Herr Breternitz die Entwicklung einer Smartphone-Applikation für iOS[8]. Herr Meintrup übernahm die Entwicklung der Smartphone-Applikation für Android[9]. Die Entwicklung des Moodle-Plugin wurde von Herr Herwegh durchgeführt. Mit der Verantwortung für das Gebiet der Kommunikation zwischen den Komponenten übernahm Herr Nimmerfroh die koordinierende Aufgabe im Entwicklungs-Prozess, indem er sich die Funktionsweise der REST[10]-API[11] als Expertenwissen aneignete und das restliche Team bei der Herstellung der Kommunikation unterstützte.

Da sich Herr Herwegh um die Entwicklung des Plugin kümmerte, wurde durch ihn auch ein Test-System installiert, welches von allen Beteiligten genutzt werden konnte. Um das Test-System dem späteren Anwendungsfall an der Technischen Hochschule Mittelhessen (THM) so nah wie möglich zu bringen, wurde die zum Zeitpunkt der Installation aktuellste Version (Moodle 3.3.1) verwendet und der Central Authentication Service (CAS)[12] der THM eingebunden.

Um innerhalb des Teams einen simplen und vielseitigen Kommunikationskanal nutzen zu können, wurde von Herr Breternitz ein Slack-Workspace[13] eingerichtet. Dadurch konnte das Team jederzeit wichtige Themen diskutieren und den Verlauf des Projektes abstimmen. Durch die vielseitigen Funktionen von Slack konnten mehrere Probleme produktiv besprochen und gelöst werden.

Für das Projekt standen ca. 12 Wochen zur Verfügung. Das erste Treffen, bei dem das Team zum ersten Mal aufeinander traf und Herr Kammer das Projekt vorstellte, fand am 12. Oktober 2017 statt. Für die Zeit ab dem 15. Januar 2018 wurde ein größerer Beta-Test in Kursen von Herr Kammer angekündigt. Dadurch leitete sich die Projektwoche (08. bis 12. Januar 2018) als Deadline für das Projekt ab.

Während der Durchführung des Projekts traten mehrere unerwartete und teils große Probleme auf. Diese werden ausführlich im jeweiligen Abschnitt der Dokumentation behandelt.

---

8 iOS ist ein von Apple entwickeltes mobiles Betriebssystem für das iPhone, das iPad und den iPod touch. Mehr Informationen unter https://de.wikipedia.org/wiki/IOS_(Betriebssystem) (abgerufen am 29. Dezember 2017).

9 Android ist sowohl ein Betriebssystem als auch eine Software-Plattform für mobile Geräte wie Smartphones, Mobiltelefone, Mediaplayer, Netbooks und Tabletcomputer, die von der von Google gegründeten Open Handset Alliance entwickelt wird. Mehr Informationen unter https://de.wikipedia.org/wiki/Android_(Betriebssystem) (abgerufen am 29. Dezember 2017).

10 Representational State Transfer (abgekürzt REST) bezeichnet ein Programmierparadigma für verteilte Systeme, insbesondere für Webservices. Mehr Informationen unter https://de.wikipedia.org/wiki/Representational_State_Transfer (abgerufen am 08. Januar 2018).

11 Eine Programmierschnittstelle, kurz API, ist ein Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird. Mehr Informationen unter https://de.wikipedia.org/wiki/Programmierschnittstelle (abgerufen am 08. Januar 2018).

12 Central Authentication Service (CAS) ist ein föderiertes Identitätsmanagement, das ursprünglich von der Yale Universität entwickelt wurde. Mehr Informationen unter https://de.wikipedia.org/wiki/Central_Authentication_Service (abgerufen am 29. Dezember 2017).

13 Slack ist ein webbasierter Instant-Messaging-Dienst des US-amerikanischen Unternehmens Slack Technologies zur Kommunikation innerhalb von Arbeitsgruppen. Mehr Informationen unter https://de.wikipedia.org/wiki/Slack_(Software) (abgerufen am 29. Dezember 2017).

2 Abstrakt

Das „THM Attendance Logging System“ (engl., kurz ‚TALS‘; deutsch „THM Anwesenheits Protokollierungs System“) ist eine als Moodle[14]-Plugin realisierte Anwendung zur digitalen Erfassung und Auswertung der Anwesenheit von Teilnehmenden bei Veranstaltungen. Das besondere Merkmal dieser Anwendung ist die Verwendung einer PIN[15], die zur Eintragung benötigt wird.

2.1 Bestandsaufnahme

An der Technischen Hochschule Mittelhessen (THM) wird fachbereichs-übergreifend die E-Learning-Plattform ‚Moodle‘ eingesetzt. Diese verfügt über ein modulares System zur (auch nachträglichen) Installation von Erweiterungen.

Die Dozierenden der THM können ihre Lehrveranstaltungen in Moodle-Kursen organisieren. Diese Kurse können genutzt werden, um Lehrmaterialien für Studierende bereit zu stellen, aber auch um die Abgabe von Hausübungen zentral zu verwalten oder Studierende individuell zu benoten.

Um die Anwesenheit in Kursen kontrollieren zu können, müssen Dozierende auf Anwesenheitslisten zurückgreifen, die in ausgedruckter Form während der Veranstaltung herum gegeben werden und z.B. unterschrieben werden müssen. Die Auswertung dieser Listen ist im Nachhinein vergleichsweise mühsam. Ebenso ist die Pflege dieser Listen, z.B. wenn ein_e Student_in ein Attest einreicht oder anderweitig als entschuldigt einzutragen ist, sowie deren Archivierung zumeist umständlich.

Die papiergestützte Anwesenheitskontrolle hat zudem auch weitere Nachteile. Zum Einen besteht ein besonderer Arbeitsaufwand für Dozierende, da entsprechende Anwesenheitslisten zu Beginn einer Veranstaltung in ausgedruckter Form vorliegen müssen. Dazu müssen diese Listen zunächst entsprechend formatiert und dann schließlich gedruckt werden. Zum Anderen ist der besondere Aspekt der ökonomischen Vertretbarkeit zu beachten. Durch das Erstellen und Drucken der Listen entstehen Kosten, die sich in der Beschaffung von Druckerpapier und der, zum Teil kostenintensiven, Wartung von Druckern widerspiegelt. Zudem hat eine moderne Hochschule, wie es die THM ist, auch eine Verantwortung gegenüber der Umwelt. Diese Verantwortung wird bereits im Bestreben zur Minimierung des Energieverbrauchs der THM umgesetzt. Daher ist es nur ein logischer Schritt, diese Verantwortung auch beim Papierverbrauch aufzugreifen. Schließlich ist auch der Aspekt des Datenschutzes zu beachten. Gedruckte Listen können abhanden kommen (gleich ob gestohlen oder verloren). Die darauf ablesbaren Daten können Studierenden zum Nachteil werden, abgesehen von der simplen Tatsache, dass häufig Matrikelnummern sowie der volle Name der Studierenden auf diesen Listen vermerkt sind.

14 Moodle ist eine freie Lernplattform. Mehr Informationen siehe https://moodle.org/
15 Personal Identification Number

2.2 Anforderungsanalyse

Eine Veränderung des aktuellen Verfahrens zur Anwesenheitskontrolle stellt besondere Anforderungen. Zum Einen muss ein neues Verfahren einfach in der Handhabung sein und zum Anderen aber komplex genug, um viele Studierenden verwalten zu können. Schließlich muss es auch sicher und robust sein, um den Datenschutz zu gewährleisten.

Die Gebrauchstauglichkeit[16], auch *Usability*, steht hierbei im Mittelpunkt. Ein neues Verfahren muss Studierende sowie Dozierende in die Lage versetzen, in möglichst wenigen Schritten alles notwendige zu erledigen um die Anwesenheit für eine Veranstaltung erfassen zu können. Auf die Dozierenden kommt hierbei üblicherweise ein höherer Aufwand zu als auf Studierende. Dennoch muss dieser Aufwand so gering wie möglich, aber dennoch so komplex wie nötig gestaltet werden. Es ergibt z.B. für Dozierende keinen Sinn, eine Veranstaltung nicht im Vorfeld komplett zu Ende planen zu können. Ebenso ergibt es für Studieren keinen Sinn, erst während der Veranstaltung über alle wichtigen Informationen zur Anwesenheitskontrolle in Kenntnis gesetzt zu werden.

Eine gewisse Komplexität ist für ein neues Verfahren nicht zu vermeiden. Diese Komplexität muss allerdings gekapselt werden, um die Gebrauchstauglichkeit nicht einzuschränken. Trotzdem muss das neue Verfahren in die bereits vorhandene Infrastruktur eingegliedert werden können. Dadurch kann eine Kapselung deutlich verbessert werden, da bereits eine gewohnte Umgebung vorhanden ist, in die ein neues Verfahren integriert wird. Das ist notwendig, um die Akzeptanz von Seiten Dozierender wie auch Studierender zu fördern.

Zuletzt muss auch der Datenschutz entsprechende Beachtung finden. Es ist wichtig, dass ein neues Verfahren die Daten von (vor allem, aber nicht nur) Studierenden schützt, um Dritten keinen Zugriff zu gestatten. Dabei ist der Umstand zu beachten, dass es sich um ein Kontrollverfahren handelt und dadurch bestimmte Zugriffsmöglichkeiten für Dozierende möglich sein müssen. Aber auch dieser Zugriff muss Grenzen finden, wenn es sich um sicherheitsrelevante Daten, wie z.B. Login-Daten, handelt. Nicht jede verfügbare Information sollte auch angezeigt werden. Es ist auch der Grundsatz der Datensparsamkeit zu berücksichtigen.

Zusammenfassend erfordert ein neues Verfahren zur Anwesenheitskontrolle die Beachtung folgender Punkte:

- Die Gebrauchstauglichkeit muss hoch sein.
- Dozierende müssen in möglichst wenigen Schritten vollständige Veranstaltungen erstellen können.

16 „Gebrauchstauglichkeit (englisch *usability*) bezeichnet nach DIN EN ISO 9241-11 das Ausmaß, in dem ein Produkt, System oder ein Dienst durch bestimmte Benutzer in einem bestimmten Anwendungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“ https://de.wikipedia.org/wiki/Gebrauchstauglichkeit_(Produkt)

- Studierende müssen in möglichst wenigen Schritten alle notwendigen Informationen für Veranstaltungen erhalten.
- Studierende müssen in möglichst wenigen Schritten ihre Anwesenheit erklären können.
- Es soll bereits vorhandene Infrastruktur genutzt werden.
- Es dürfen nur notwendige Daten erhoben werden (Grundsatz der Datensparsamkeit).
- Gespeicherte Daten müssen entsprechend vor dem Zugriff unbefugter Dritter geschützt werden.

2.3 Lösungsansatz

In Anbetracht der Anforderungen sowie der bereits vorhandenen Infrastruktur ist es naheliegend, eine digitale Lösung in Form eines Moodle-Plugin zu implementieren.

Dieses Plugin ist entsprechend an den Bedürfnissen von Dozierenden und Studierenden zu orientieren. Da Moodle ein rollen basiertes System ist, werden nachfolgend auch Dozierende und Studierende als Rollen behandelt. Moodle-intern wird dies häufig durch die Rollen ‚Manager_in' und ‚authentifizierte_r Nutzer_in' abgebildet, wobei es noch weitere Abstufungen gibt.

Dozierende müssen dabei die Möglichkeit haben, das Plugin zur Verwaltung von Veranstaltungen zu nutzen. Das bedeutet, dass es möglich sein muss, Veranstaltungen anzulegen, diese zu editieren und, wenn nötig, auch wieder zu entfernen. Zudem ist eine Authentifizierung in Form einer PIN zu realisieren, damit nur anwesende Studierende sich eintragen können. Eine solche PIN muss von dem_der Dozierenden zum Zeitpunkt der Veranstaltungs-Erstellung angefordert werden und wird dann automatisch zufällig erzeugt. Zusätzlich muss eine Gültigkeitsdauer für die PIN angegeben werden, um den Studierenden für eine vordefinierte Dauer die Eintragung zu ermöglichen. Wenn die Veranstaltung stattfindet, muss die_der Dozierende die PIN manuell freischalten können. Für jede Veranstaltung braucht die_der Dozierende eine Übersicht, wie viele Studierende in Summe sich bereits für die aktuelle Veranstaltung eingetragen haben. Zudem muss es möglich sein, eine Liste der bereits eingetragenen Studierenden zu erhalten und den Anwesenheitsstatus einzelner Studierender ändern zu können. Eine solche Übersicht muss auch für bereits vergangene Veranstaltungen möglich sein. Die Funktionen für Dozierende können in drei Bereiche gegliedert werden:

1. Veranstaltungen allgemein:

   Eine Übersicht aller Veranstaltungen eines Kurses sowie eine Übersicht der Bearbeitungsmöglichkeiten.

2. Veranstaltung anlegen:

Eine neue Veranstaltung anlegen und eine PIN sowie deren Gültigkeitsdauer bestimmen.

3. Berichte:
   Eine Übersicht aller vergangenen Veranstaltungen sowie Kurz-Informationen über die Anwesenheit. Durch Klicken auf eine Veranstaltung wird die Liste der Studierenden im Kurs angezeigt und deren Anwesenheitsstatus für die betreffende Veranstaltung.

Studierende müssen die Möglichkeit haben, das Plugin simpel zu nutzen, um ihre Anwesenheit bei einer Veranstaltung eintragen zu können. Die Eingabe einer PIN muss an eine aktuelle Veranstaltung gebunden sein und darf erst dann möglich sein, wenn für diese Veranstaltung die PIN freigegeben wurde. Wenn die PIN einer Veranstaltung freigegeben wurde, muss die_der Studierende die PIN eintragen und die bisherigen Fehltage bestätigen. Ist die PIN korrekt und wurden die bisherigen Fehltage bestätigt, kann die_der Studierende eingetragen werden. Zudem müssen Studierende die Möglichkeit haben, eine Übersicht über ihre Anwesenheit bei Veranstaltungen eines Kurses einsehen zu können. Dazu müssen folgende Informationen zusammengeführt werden:

1. Vor- und Nachname des_der Studierenden.
2. Anzahl der bisherigen Fehltage in diesem Kurs.
3. Eine Liste aller bisherigen Veranstaltungen dieses Kurses und den dazugehörigen Anwesenheitsstatus des_der Studierenden.

Studierende können keine Änderungen an ihren Daten vornehmen. Dies ist dem_der Dozierenden vorbehalten.

Um die Gebrauchstauglichkeit weiter zu erhöhen, soll es ermöglicht werden, die jeweilige Anwesenheit auch mittels einer Smartphone-Applikation eintragen zu können. Eine solche Applikation sollte auf mehreren Smartphone-Betriebssystemen nutzbar sein. Studierende müssen sich mit ihren gewohnten Anmeldedaten über die Applikation in der bereits vorhandenen Infrastruktur anmelden können. Es muss möglich sein, eine Liste aller Kurse des_der Studierenden zu erhalten, welche das Plugin verwenden. Außerdem muss eine Liste aller zukünftiger Veranstaltungen aus solchen Kursen abrufbar sein. Schließlich müssen Studierende in der Applikation für eine bestimmte Veranstaltung ihre Anwesenheit durch Eingabe der (korrekten) PIN und Bestätigung ihrer bisherigen Fehltage eintragen können.

2.4 Namensgebung

Die Anwendung trägt den Namen „THM Attendance Logging System“, kurz „TALS“.

Der Name bildet die Funktion sowie die Herkunft der Anwendung direkt ab.

3 Das Moodle-Plugin

Im Zentrum des Projektes steht das Moodle-Plugin. Es fungiert als zentraler Server für die Verwaltung der Veranstaltungen und Anwesenheits-Protokolle. Es handelt sich dabei um ein Plugin, welches im Rahmen des Projektes grundlegend neu entwickelt wurde.

3.1 Systemanforderungen

Moodle-Version	Moodle 3.0 (2015111600) oder höher
Webservices aktiviert	Ja
Protokolle	REST

Berechtigungen	Nutzer_innen des Plugins müssen eigene Tokens erstellen dürfen.
	`moodle/webservice:createtoken` für Rolle `user`
	`webservice/rest:use` für Rolle `user`

Login	CAS

3.2 Kurzbeschreibung

Das Plugin stellt vielfältige Funktionen zur Verfügung. Es bietet für Studierende eine umfassende Übersicht über die individuelle Anwesenheit bei Veranstaltungen eines Kurses und die Möglichkeit, sich mittels Smartphone-Applikation einzutragen. Für Dozierende bietet es die Möglichkeit die Veranstaltungen eines Kurses zu verwalten. Das umfasst das Erstellen, Bearbeiten und Entfernen von Veranstaltungen. Außerdem können Dozierende umfassende Informationen über bisherige Veranstaltungen und deren Teilnehmende erhalten. Der Anwesenheitsstatus von Studierenden kann im Nachhinein von Dozierenden bearbeitet sowie kommentiert werden. Dadurch ist es möglich, fehlerhafte Eintragungen zu korrigieren und ggf. weiterführende Informationen zu hinterlegen.

3.3 Entwicklungsverlauf

Das Plugin ist realisiert als sog. activity module[17]. Dies rührt daher, dass das Plugin für die Verwendung als sog. activity, bzw. *Aktivität*, angelegt ist. Dabei handelt es sich um grundlegende Bausteine von Moodle. Kurse sind definiert als eine Sammlung von Aktivitäten. Diese Aktivitäten stellen Funktionen bereit, wie z.B. ein Forum, die Abgabe einer Hausübung oder die Bereitstellung von Lehrmaterial.

Bei der Installation registriert sich das Plugin im Moodle-System als sog. Webservice[18]. Dadurch erhält es eine Schnittstelle nach außen, durch die die Kommunikation zu den Smartphone-Applikationen möglich wird. Für diesen Zweck werden verschiedene

17 https://docs.moodle.org/dev/Activity_modules

18 Webservices ermöglichen es externen Systemen, sich in Moodle anzumelden und Aktionen auszuführen. Mehr Informationen unter https://docs.moodle.org/33/de/Webservices (abgerufen am 29. Dezember 2017).

Funktionen registriert, welche von außen mittels REST-API aufgerufen werden können. Hierbei übernimmt das Moodle-System die Validierung der übergebenen Parameter gegen die vom Plugin vorausgesetzte Definition. Die Antwort wird vom Moodle-System, je nach Anfrage, als JSON[19]- oder XML[20]-Objekt zurückgegeben.

Innerhalb des Moodle-Systems verfügt das Plugin über einen eigenen Datenbankbereich, der beliebig viele Tabellen enthalten kann. Lediglich eine Tabelle ist vom System vorgegeben um das Plugin korrekt verwalten zu können.

Am Anfang der Entwicklung steht das **Datenbankdesign**.

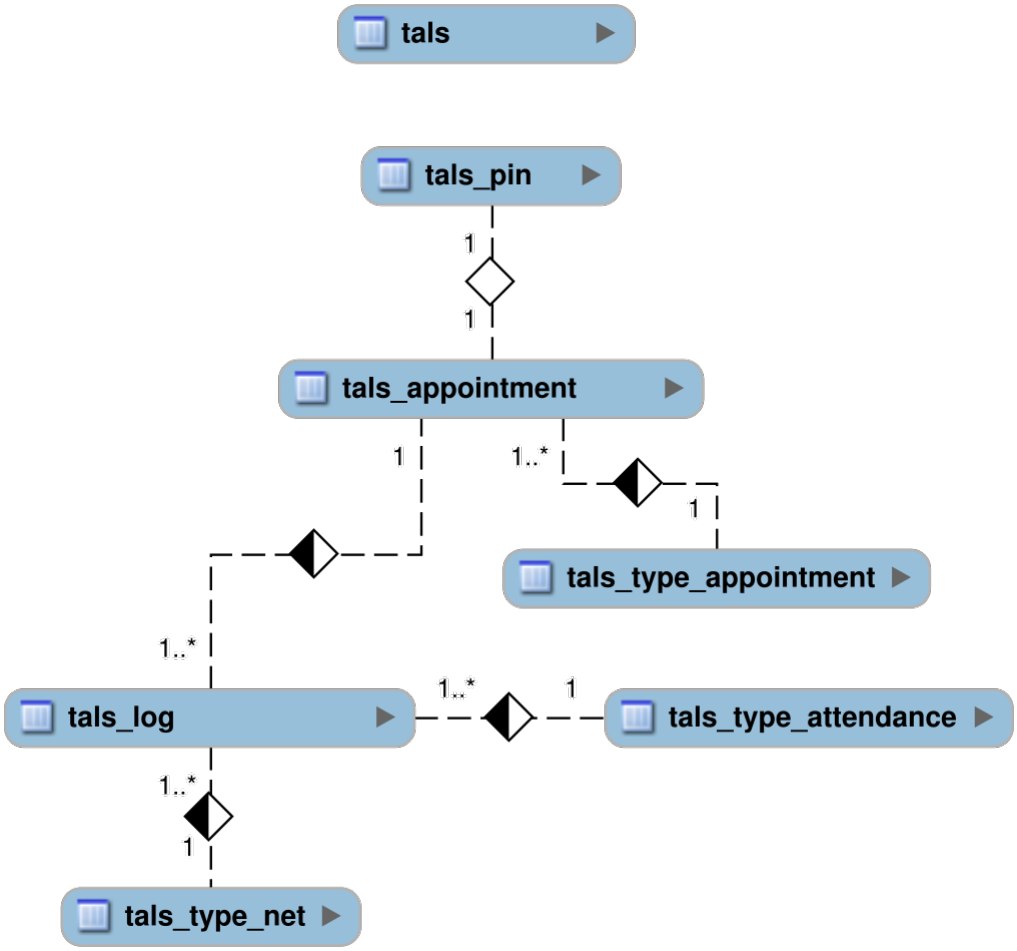

Abbildung 1: ER-Diagramm des Plugin.
Quelle: eigene Darstellung

Hierbei wurde eine recht pragmatische Struktur gewählt. Die Namensgebung erfolgte anhand der von Moodle vorgegebenen Konvention.

19 Die JavaScript Object Notation, kurz JSON, ist ein kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen. Mehr Informationen unter https://de.wikipedia.org/wiki/JavaScript_Object_Notation (abgerufen am 08. Januar 2018).

20 Die erweiterbare Auszeichnungssprache, abgekürzt XML, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten im Format einer Textdatei, die sowohl von Menschen als auch von Maschinen lesbar ist. Mehr Informationen unter https://de.wikipedia.org/wiki/Extensible_Markup_Language (abgerufen am 08. Januar 2018).

Der sog. Frankenstyle[21] wird von Moodle bei der Benennung u.a. der Tabellen eines Plugins eingesetzt. Dies ist auch zwingend notwendig, da es sehr viele Tabellen gibt. Grundsätzlich ergänzt Moodle die Namen noch zusätzlich mit einem vorangestellten ‚mdl_', was eine Abkürzung für ‚Moodle' darstellt. Somit ist ein Tabellen-Name wie folgt aufgebaut:

mdl_plugin_beschreibung

Da Moodle auf verschiedenen Datenbanksystemen verwendet werden kann und somit nicht ausschließlich auf nur MySQL oder nur PostreSQL ausgelegt ist, muss auch die Datenbankstruktur entsprechend SQL-übergreifend formuliert sein. Das bedeutet vor allem, dass z.B. keine MySQL-eigenen Datentypen verwendet werden können.

Im nächsten Schritt muss die **Installation des Plugin** behandelt werden. Während dieses Prozesses wird die benötigte Datenbankstruktur angelegt und das Plugin im System registriert. An dieser Stelle ist es wichtig, dass Moodle alle notwendigen Daten an den erwarteten Stellen vorfinden kann. Das bezieht sich in erster Linie auf die Namenskonvention von Dateien wie auch Funktionen bzw. Klassen.

Danach muss das **Hinzufügen zu einem Kurs** behandelt werden. Dazu muss ein Formular definiert werden, welches die notwendigen Daten erfasst, die für eine Instanz notwendig sind. Das sind vor allem Titel und Beschreibung der Instanz für den betreffenden Kurs (beides ist frei wählbar bei jeder einzelnen Instanziierung). Während dieses Prozesses wird eine neue Instanz in der Datenbank registriert und so einem Kurs hinzugefügt.

Erst jetzt kann die eigentliche **Funktionalität** für das Plugin, entsprechend den Anforderungen, implementiert werden. Diese Funktionalität kann frei wählbar organisiert werden. Im vorliegenden Plugin wurde eine minimalistischer Ansatz gewählt, um wenige Dateien zu erhalten.

Schließlich kann nun auch der **Webservice** für das Plugin umgesetzt werden. Für diesen müssen zunächst die Funktionen, die später von extern aufgerufen werden sollen, implementiert werden. Dabei bietet es sich an, bereits vorhandene Funktionen des Plugin zu verwenden. Als Besonderheit bei dieser Implementierung ist darauf hinzuweisen, dass die Parameter sowie die Rückgabetypen für jede Funktion mittels je einer zusätzlichen Funktion definiert werden müssen. Bei diesen Sonderfunktionen ist die von Moodle vorgegebene Namenskonvention unbedingt zu befolgen. Ebenso ist der Aufbau dieser Funktionen unbedingt an den Vorgaben von Moodle auszurichten. Andernfalls wird Moodle diese Funktionen nicht für externe Zugriffe zulassen bzw. nutzen können. Nachdem die Funktionen implementiert sind, muss das Plugin selbst noch als Webservice und die Funktionen als Webservice-Funktionen registriert werden.

21 Die ‚Frankenstyle Komponenten Namen' bezeichnen die Namenskonvention, welche genutzt wird um ein Moodle-Plugin, basierend auf dem Typ und den Namen des Plugins, eindeutig zu identifizieren. Mehr Informationen unter https://docs.moodle.org/dev/Frankenstyle (abgerufen am 30. Dezember 2017).

Außerdem muss eine **graphische Oberfläche** implementiert werden, da das Plugin sonst nicht bedient werden kann. Dieser Schritt sollte erst relativ spät begonnen werden, da hier die Funktionalität des Plugin genutzt wird.

3.4 Probleme

Während der Entwicklung traten unterschiedliche Probleme auf, welche für eine, zum Teil nicht unerhebliche, Verzögerung sorgten. Nachfolgend sind Problem-Komplexe beschrieben.

Dokumentation von Moodle

Die Dokumentation seitens Moodle ist vordergründig auf die Bedürfnisse von Anwender_innen zugeschnitten. Es ist zwar eine Dokumentation für Entwickler_innen vorhanden, nur ist diese an entscheidenden Stellen unvollständig oder missverständlich. Da ein kollaboratives System (ähnlich der Wikipedia) eingesetzt wird, ist dies aber nicht verwunderlich. Funktionen, die häufig Anwendung finden, wie z.B. der Datenbankzugriff, sind gut dokumentiert und ermöglichen eine effiziente Nutzung. Andere Funktionen, die weniger verwendet werden bzw. nur als Feature im Moodle-Kern-System vorhanden sind und von Moodle selbst gar nicht genutzt werden, sind vergleichsweise rudimentär dokumentiert. Dadurch musste viele Moodle-reverseengineering[22] zur Entwicklung eines neuen Plugin durchgeführt werden.

Erstellen der Datenbankstruktur

Da Moodle, wie bereits erwähnt, mit unterschiedlichen Datenbanksystemen verwendet werden kann, muss die Definition der Datenbankstruktur für das Plugin generalisiert werden. Dazu setzt Moodle auf ein XML-Format um die notwendige Struktur zu beschreiben. In diesem Format sind nur primitive Datentypen zugelassen und es müssen besondere Konventionen befolgt werden, die allerdings nirgends dokumentiert sind. Stattdessen stellt Moodle einen Dienst, den XMLDB-Editor[23], zur Verfügung. Mit diesem Dienst können XML-Beschreibungen von Datenbankstrukturen bearbeitet werden. Allerdings funktioniert dies nur mit einem bereits installierten Plugin. Somit musste an dieser Stelle ein Henne-Ei-Problem gelöst werden, da die Datenbankstruktur benötigt wurde um das Plugin zu installieren, aber ohne Installation keine Datenbankstruktur erstellt werden konnte. Nachdem weder in den Einstellungen und Funktionen des Editors noch in der Dokumentation eine Möglichkeit gefunden wurde, für ein Plugin ohne XML-Beschreibung eine solche zu erstellen, wurde schließlich die Beschreibung eines anderen Plugin kopiert und das neue Plugin mit dieser installiert. Nun konnte diese fremde Beschreibung geladen und bearbeitet werden. In der späteren Betrachtung erweckte die

22 Reverse Engineering bezeichnet den Vorgang, aus einem bestehenden fertigen System durch Untersuchung der Strukturen, Zustände und Verhaltensweisen die Konstruktionselemente zu extrahieren. Mehr Informationen unter https://de.wikipedia.org/wiki/Reverse_Engineering (abgerufen am 11. Februar 2018).

23 https://docs.moodle.org/dev/XMLDB_editor (abgerufen am 30. Dezember 2017)

XML-Beschreibung den Eindruck unnötig komplex gestaltet zu sein. Eine umfassende Dokumentation des Formats sollte kein Problem sein um eine manuelle Erstellung ohne einen Editor zu ermöglichen.

Grundgerüst für ein Plugin

In der Dokumentation von Moodle befindet sich ein Tutorial[24] für die Entwicklung eines Plugin. Dabei handelt es sich in erster Linie um eine Link-Sammlung und kurze Zusammenfassungen. Leider gibt dieses Tutorial nicht mehr als einen rudimentären Überblick, was für die Entwicklung eines Plugin benötigt wird. Das tatsächliche Wissen musste zum Teil mühsam zusammengesucht werden. Oft beinhaltet die Dokumentation auch nur Beispiel-Code, welcher rudimentär in Kontext gesetzt ist. Allgemein entsteht der Eindruck, diese Beispiele sind nicht auf die Bedürfnisse von Moodle-Anfänger_innen ausgelegt.

Vor allem musste die logische Verbindung zwischen den benötigten Dateien zum Teil durch Ausprobieren erst in Erfahrung gebracht werden. Beim Verständnis des Grundgerüstes eines Plugin mussten oft andere Plugins betrachtet und deren Funktionsweise nachvollzogen werden.

Bei der Instantiierung des Plugins trat zunächst auch ein allgemeiner Fehler auf, welcher keiner bestimmten Ursache zugeordnet werden konnte. Nach aufwendiger Fehlersuche gelang es zunächst nicht, die Ursache herauszufinden. Eine Internetrecherche war auch nicht zielführend, da die Fehlermeldung zu allgemein war. Erst nach Tipps aus der Moodle-Community konnte die Fehlersuche zielführend fortgesetzt und schließlich erfolgreich zum Abschluss gebracht werden.

Webservice

Wie bereits erwähnt, muss das Plugin als Webservice registriert werden. Das geschieht automatisch bei der Installation, sofern die entsprechenden Dateien mit entsprechendem Code vorliegen. Hier ist es besonders wichtig die Namenskonvention einzuhalten, da jeder Fehler zu einem Abbruch führt. Dieser Abbruch wird allerdings nicht durch eine Fehlermeldung kenntlich gemacht. Das Plugin wird im Fehlerfall einfach ohne Webservice installiert und diese Installation als erfolgreich zurückgemeldet. Da die Moodle-Dokumentation dieses Verhalten nicht enthält, musste durch umständliche Fehlersuche und Internetrecherche dieses Verhalten erst erklärt werden, um es in Zukunft vermeiden zu können. Zudem ist die Definition eines Webservice mit allen Funktionen zum Teil redundant und dadurch für ungeübte Entwickler_innen umständlich. Im Fehlerfall kann es auch passieren, dass ein Webservice registriert wurde, die zugehörigen Funktionen ebenfalls korrekt in der Datenbank eingepflegt wurden, aber der Webservice dennoch nicht funktioniert. Der Grund dafür ist dann häufig ein nicht-offensichtlicher Tippfehler in der Definition der Funktionen.

24 https://docs.moodle.org/dev/Tutorial

Des Weiteren ist die Implementierung von Funktionen umständlich, da die benötigten Definitionen für Parameter und Rückgabetypen nicht selbsterklärend sind. An dieser Stelle ist ebenfalls die Dokumentation verbesserungswürdig, da das Grundprinzip zwar durchaus gut erläutert wird, die konkrete Umsetzung allerdings vernachlässigt wurde. Auch hier wird wieder sichtbar, dass die Dokumentation nicht an den Bedürfnissen von Moodle-Anfänger_innen orientiert ist.

Um einen Webservice verwenden zu können, wird zudem ein sog. Token[25] benötigt. Dieses wird benötigt, damit Nutzer_innen auf die Funktionen eines Webservice zugreifen dürfen. Es muss für jede_n Nutzer_in erzeugt werden und ist nur für diese_n Nutzer_in und diesen Webservice gültig. In der Moodle-Dokumentation ist lediglich die manuelle Erzeugung dieses Tokens durch eine_n Administrator_in beschrieben[26]. Erst durch eine gründliche zeitintensive Recherche im Moodle-Quelltext konnte eine Funktion ausfindig gemacht werden, welche einen Token generiert. Diese Funktion kann auch mit nicht-administrativen Nutzungsrechten aufgerufen werden. Dennoch müssen besondere Rechte für die Nutzungsrolle gesetzt sein.

Daten erhalten

Grundlegend ist es simpel, Daten aus der Datenbank zu erhalten. Ebenso können unkompliziert Daten über den_die aktuelle Nutzer_in oder den aktuellen Kurs erhalten werden. Komplex wird es allerdings, wenn über diese Informationen hinaus gegangen werden soll. So gibt es z.B. keine dokumentierte Vorgehensweise, um an eine Liste aller Nutzer_innen zu gelangen, die aktuell in einem bestimmten Kurs angemeldet sind. Eine Internetrecherche verwies auf eine komplizierte SQL-Abfrage, welche manuell formuliert und ausgeführt werden müsste. Da es von der Moodle-Datenbankstruktur keine Beschreibung gibt, konnten die Tabellen nur anhand ihrer Namen einem (mutmaßlichen) Nutzen zugeordnet werden. Die gefundene SQL-Abfrage war allerdings zu komplex, um sie für das, doch simple, Vorhaben zu nutzen. Zunächst wurde versucht, das Verhalten von bereits vorhandenen Nutzer_innen-Listen für Kurse nachzuvollziehen. Dies schlug jedoch fehl, weswegen an dieser Stelle der Moodle-Quelltext nach einer geeigneten Funktion durchsucht werden musste. Hierbei wurden einige wenige Funktionen gefunden, die eine Nutzer_innen-Liste von Kursen zurückgaben. Eine dieser Funktionen stellte sich schließlich als sinnvoll heraus.

3.5 Problembewältigung

Um die auftretenden Probleme bewältigen zu können, wurden unterschiedliche Strategien eingesetzt.

Die einfachste Strategie war es, simple Textausgaben in den Code einzubauen. Dadurch konnte der Programmablauf nachvollzogen und so die Fehlerquelle eingegrenzt werden.

25 Hier: Eine Public-Private-Key Architektur zur Identifizierung von Nutzer_innen.
26 https://docs.moodle.org/34/de/Webservices_nutzen#Ein_Token_erzeugen

Später wurde zudem auf das Moodle-eigene Debug-System zurückgegriffen. Hierbei lieferte das System häufig brauchbare Anhaltspunkte, an welcher Stelle ein Problem auftrat.

Zeitweise wurde die Datenbank genutzt, um ein Protokoll der SQL-Anweisungen anzufertigen. Dabei wurde zunächst das Protokoll aktiviert, dann die fragliche Funktion ausgeführt und anschließend das Protokoll ausgewertet. In keinem der Anwendungsfälle wurde durch dieses Vorgehen ein Problem gelöst. Es führte lediglich zu einer gewissen Eingrenzung der Fehlerquelle. Zudem führte ein simpler Funktionsaufruf zu einer großen Anzahl von SQL-Anweisungen (~500 Zeilen Protokoll).

3.6 Screenshots

Nachfolgend wird das fertige Plugin mit einigen Screenshots gezeigt.

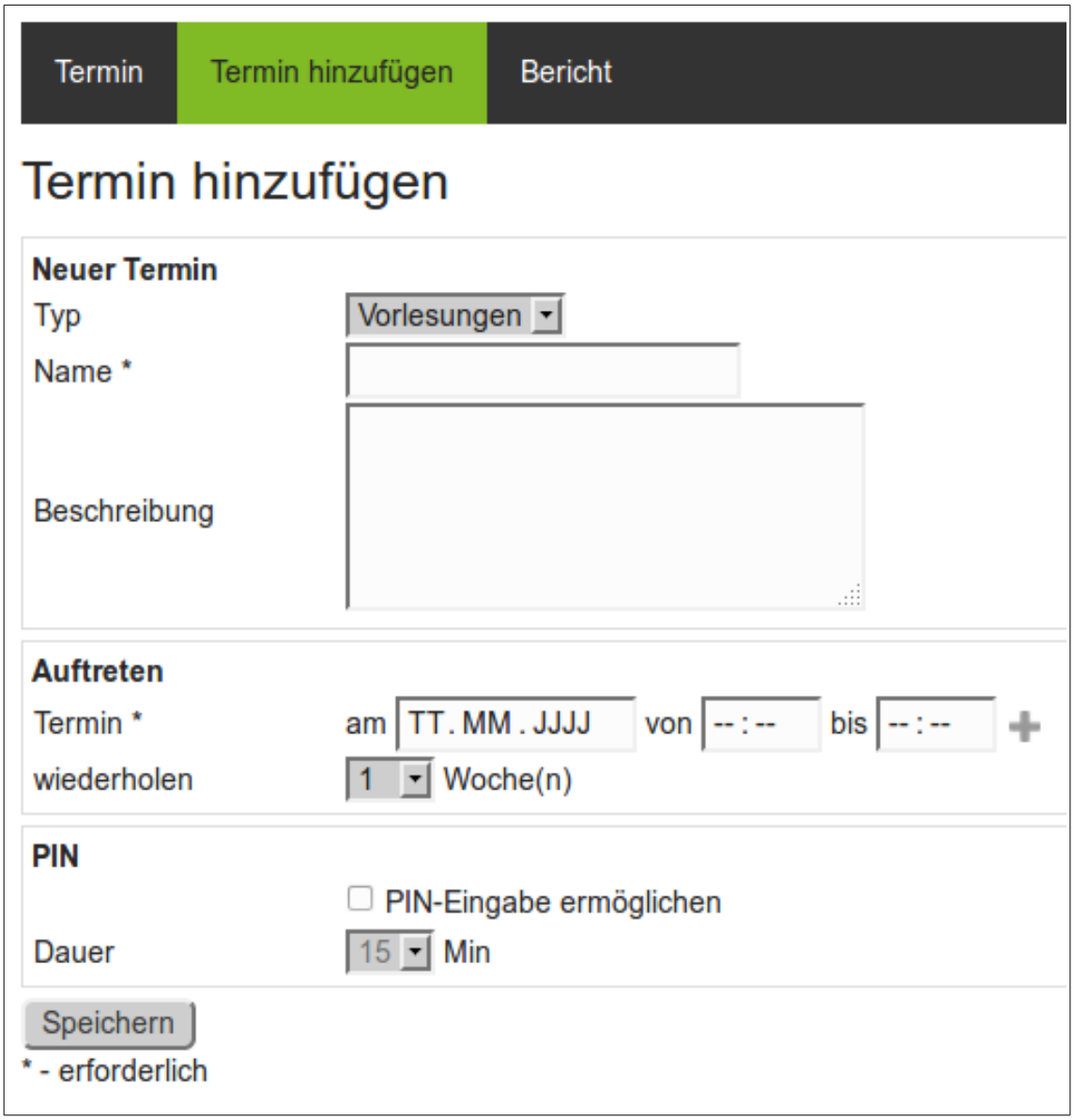

Abbildung 2: Dozierenden-Ansicht zum hinzufügen eines Termines.

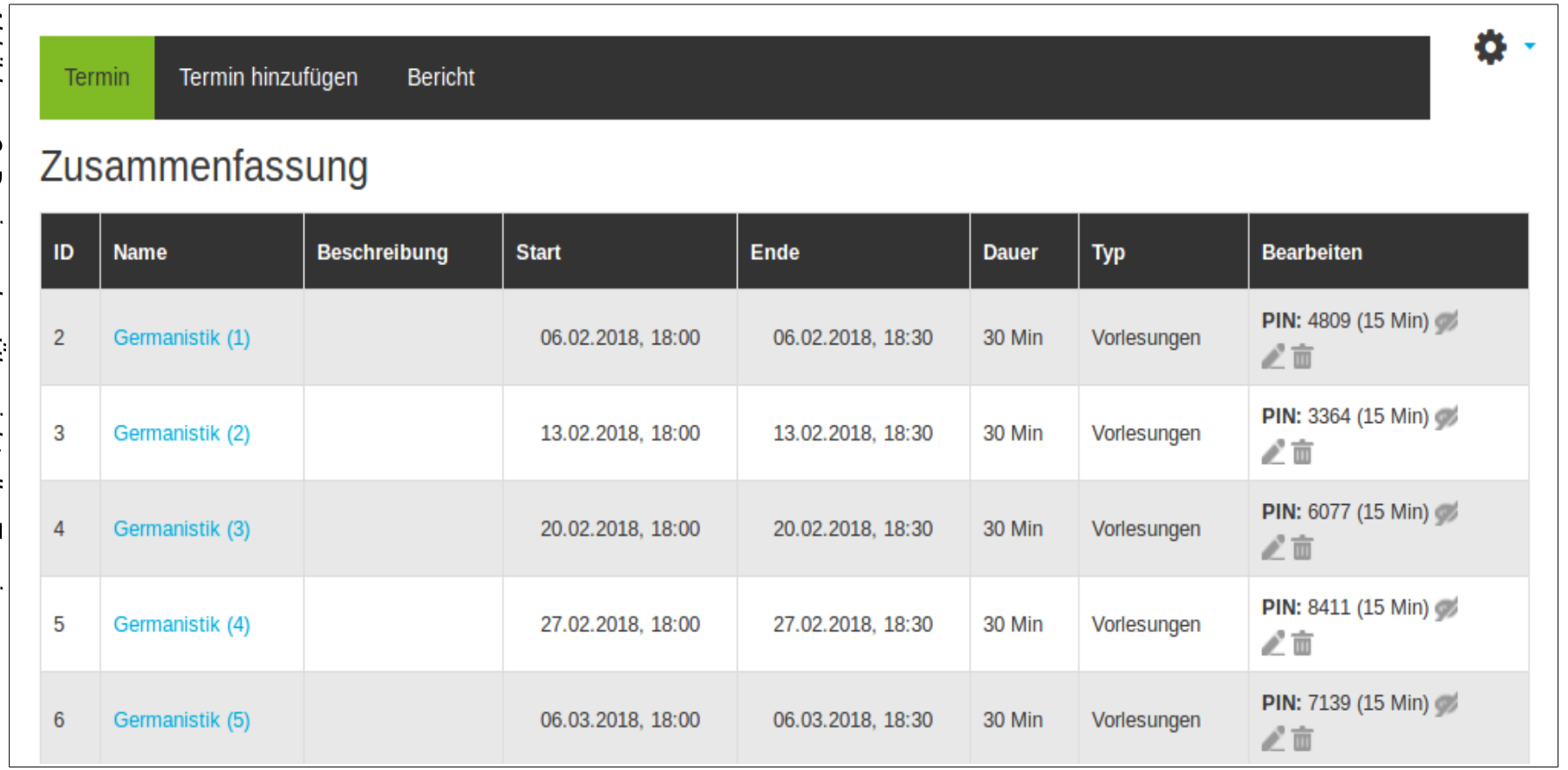

Abbildung 3: Dozierenden-Übersicht aller Termine.

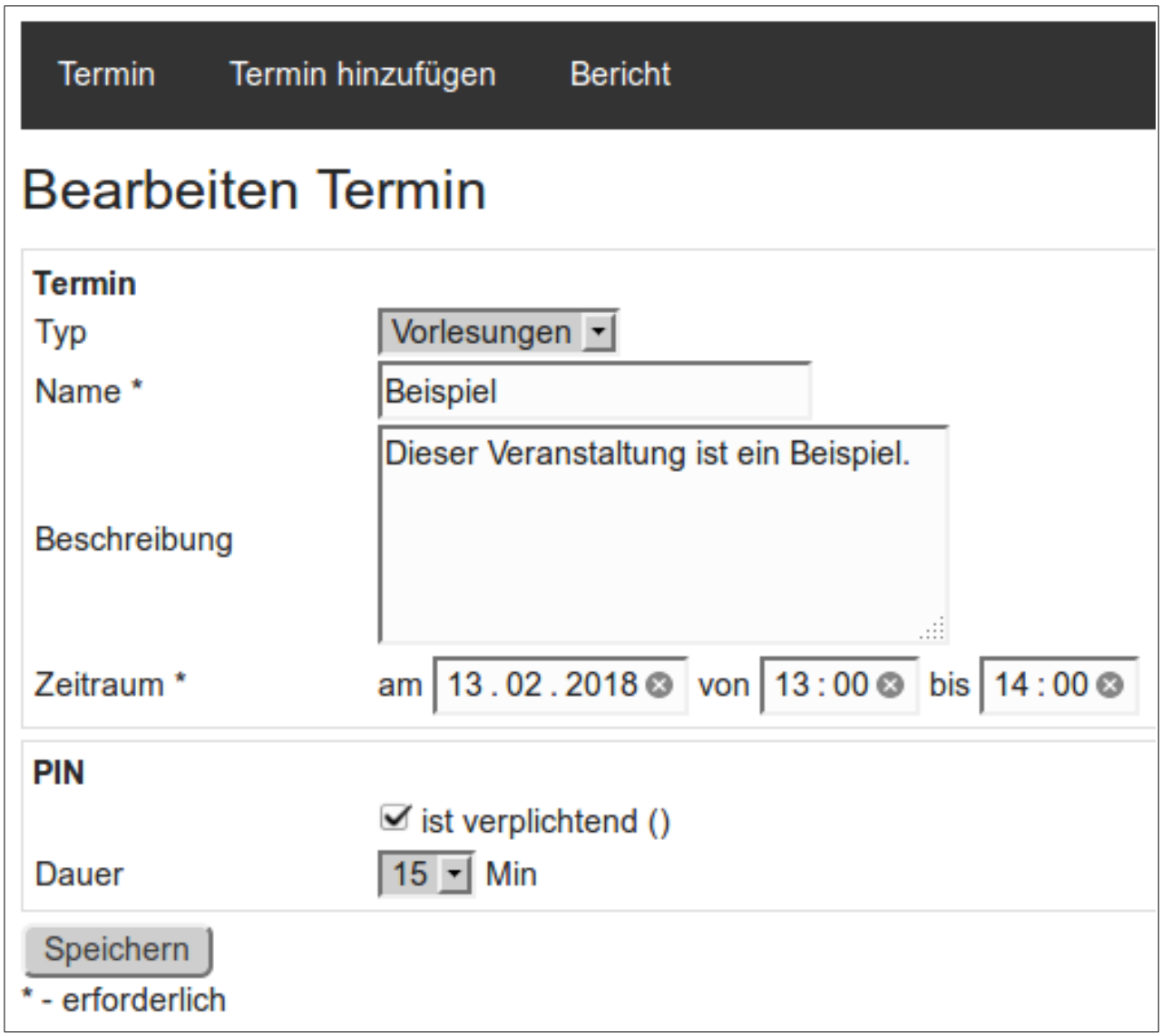

Abbildung 4: Dozierenden-Ansicht zum Bearbeiten eines Termines.

Abbildung 5: Dozierenden-Sicht in der Kurs-Übersicht bei nicht-aktiver PIN.

Abbildung 6: Dozierenden-Sicht in der Kurs-Übersicht bei aktiver PIN.

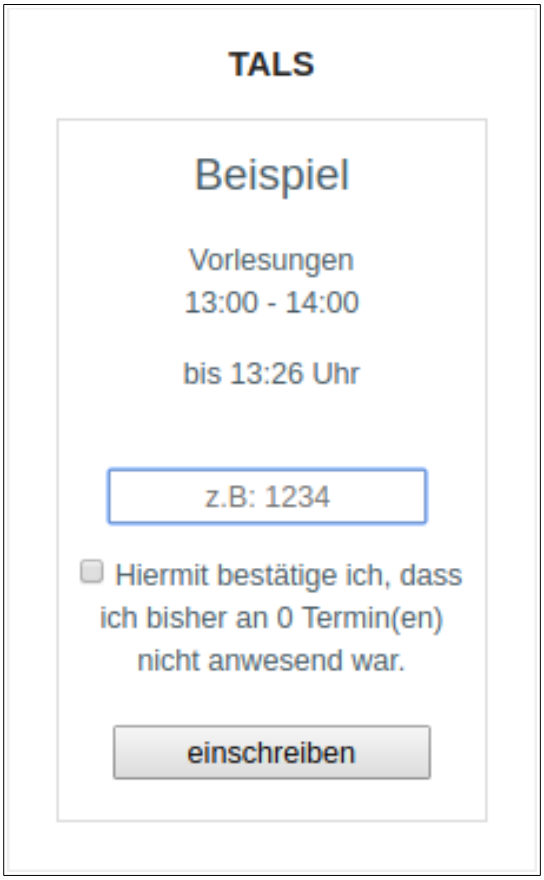

Abbildung 7: Studierenden-Ansicht in der Kurs-Übersicht, wenn die PIN aktiv ist.

Abbildung 8: Studierenden-Ansicht in der Kurs-Übersicht, wenn die Anwesenheit eingetragen wurde.

Emmanuel Goldstein

emmanuel.goldstein@mni.thm.de

Anzahl der Termine: 15 (5 sind verpflichtend)

Anwesenheit: 0

Verpasst: 1

Entschuldigt: 0

Meine Anwesenheit | Kursübersicht

Name	Beschreibung	Start	Ende	Dauer	Typ	Verplichtend
Germanistik (1)		06.02.2018, 18:00	06.02.2018, 18:30	30 Min	Vorlesungen	
Germanistik (2)		13.02.2018, 18:00	13.02.2018, 18:30	30 Min	Vorlesungen	
Germanistik (3)		20.02.2018, 18:00	20.02.2018, 18:30	30 Min	Vorlesungen	
Germanistik (4)		27.02.2018, 18:00	27.02.2018, 18:30	30 Min	Vorlesungen	
Germanistik (5)		06.03.2018, 18:00	06.03.2018, 18:30	30 Min	Vorlesungen	
Übung (1)		06.02.2018, 18:30	06.02.2018, 18:45	15 Min	Übung	
Übung (1)		06.02.2018, 18:45	06.02.2018, 19:00	15 Min	Übung	
Übung (2)		13.02.2018, 18:30	13.02.2018, 18:45	15 Min	Übung	
Übung (2)		13.02.2018, 18:45	13.02.2018, 19:00	15 Min	Übung	
Übung (3)		20.02.2018, 18:30	20.02.2018, 18:45	15 Min	Übung	
Übung (3)		20.02.2018, 18:45	20.02.2018, 19:00	15 Min	Übung	
Übung (4)		27.02.2018, 18:30	27.02.2018, 18:45	15 Min	Übung	
Übung (4)		27.02.2018, 18:45	27.02.2018, 19:00	15 Min	Übung	
Übung (5)		06.03.2018, 18:30	06.03.2018, 18:45	15 Min	Übung	
Übung (5)		06.03.2018, 18:45	06.03.2018, 19:00	15 Min	Übung	

ist verplichtend

4 Die Smartphone-Applikationen

Das neue Verfahren zur Erfassung der Anwesenheit soll nicht nur direkt in Moodle nutzbar sein, sondern auch auf Smartphones verwendet werden können. Um dabei die Gebrauchstauglichkeit deutlich zu erhöhen, wurden Smartphone-Applikationen für die verbreiteten Betriebssysteme iOS und Android konzipiert.

Die Auswahl der Ziel-Plattformen geschah auf Grundlage folgender Punkte:

- Aufwand: Kann der Aufwand vom Team in der gegebenen Zeit bewältigt werden?
- Nutzen: Wird die Ziel-Plattform von Studierenden ausreichend genutzt?

4.1 Die Android-Applikation

Im ersten Schritt musste die Android-Version festgelegt werden, für die eine Applikation entwickelt werden soll. Bei der Entscheidung half die Betrachtung der aktuellen Statistik zur Verbreitung der verschiedenen Android-Versionen.

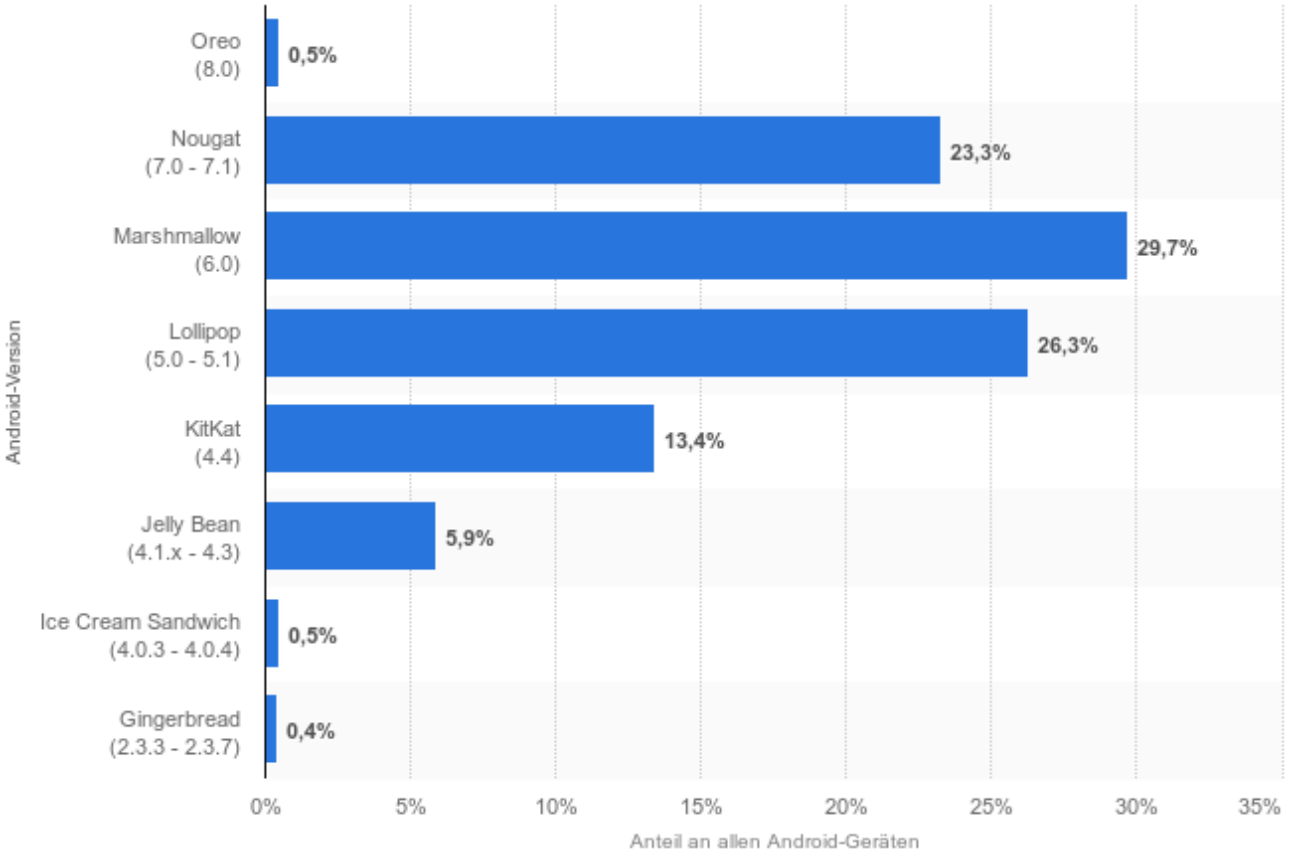

Tabelle 1: Verbreitung der Android-Versionen im Zeitraum 05. bis 11. Dezember 2017.
Quelle: https://de.statista.com/statistik/daten/studie/180113/umfrage/anteil-der-verschiedenen-android-versionen-auf-geraeten-mit-android-os/ (abgerufen am 31. Dezember 2017)

Auf Grundlage dessen wurde entschieden, die Applikation für die Android-Version 4.4 (KitKat) und höher zu entwickeln. Dadurch wird eine möglichst große Abdeckung gewährleistet.

4.1.1 Systemanforderungen

Android-Version	4.4 (KitKat) oder höher
Berechtigungen	Voller Internetzugriff

4.1.2 Kurzbeschreibung

Diese Applikation ermöglicht es Studierenden der THM mit Hilfe ihres Android-Smartphones ihre Anwesenheit während einer Veranstaltung einzutragen.

Studierende können sich mit der Applikation im THM-eigenen Moodle anmelden und dann eine Liste aller Kurse abrufen, in denen sie eingeschrieben sind und die das Plugin verwenden. Außerdem kann eine Liste aller Veranstaltungen, die am aktuellen Tag stattfinden, abgerufen werden.

Findet eine Veranstaltung statt und ist die PIN freigegeben worden, können Studierende die PIN in der Applikation eingeben, ihre bisherigen Fehltage bestätigen und sich dann als ‚Anwesend' eintragen.

Andere Funktionen, wie z.B. sich als ‚Krank' o.ä. einzutragen oder einem Kurs beitreten, sind bewusst nicht implementiert worden. Die Applikation ist schlank gehalten um die Gebrauchstauglichkeit zu erhöhen.

4.1.3 Entwicklungsverlauf

Die Applikation ist als native Android-Applikation implementiert. Eine solche native Applikation ist effizienter und arbeitet schneller. Zudem kann sie die Hardware des Smartphone einfacher nutzen.

Am Anfang der Entwicklung steht das **Datenbankdesign**.

Abbildung 9: ER-Diagramm der Applikation für Android.

Die notwendige Datenbankstruktur für die Applikation ist vergleichsweise simpel. Da kaum Daten gespeichert werden müssen, wird auch nur eine simple Datenbank benötigt. Bei Android kommt native SQLite[27] zum Einsatz. Die benötigten Daten (Token des_der Nutzer_in) werden durch Schlüssel-Wert-Paare repräsentiert.

27 SQLite ist eine gemeinfreie Programmbibliothek, die ein relationales Datenbanksystem enthält. Mehr Informationen unter https://de.wikipedia.org/wiki/SQLite (abgerufen am 06. Januar 2018).

Danach folgt das **Anmelden im THM-eigenen Moodle**. Da die THM für die Anmeldung bei ihren Diensten CAS verwendet, muss ein CAS-Client implementiert werden. Dieser muss es möglich machen, die Nutzer_innenkennnung und das Passwort einzugeben und sicher zu übertragen. Dies benötigt eine durch Transport-Layer-Security (TSL)[28] gesicherte Verbindung.

Um nun an **die benötigten Daten** zu kommen, muss die Applikation das Nutzer_innen-Token abfragen. Daraufhin kann mithilfe dieses Tokens die Funktionalität des Moodle-Plugin voll genutzt werden. Zu Beginn muss die Kurs-Liste des_der Nutzer_in abgefragt werden. Direkt im Anschluss ist es möglich, alle am aktuellen Tag stattfindende Veranstaltungen in Erfahrung zu bringen. Für einzelne Veranstaltungen muss abgefragt werden können, ob die PIN bereits verfügbar ist. Es können folgende Daten abgerufen werden:

- Kurs-Liste
- Veranstaltungen des aktuellen Tages
- Bisherige Fehltage
- Ob eine PIN bereits/noch verfügbar ist

Damit der_die Nutzer_in **die Anwesenheit eintragen** kann, muss es eine Oberfläche geben, auf der eine PIN für eine bestimmte Veranstaltung eingegeben und abgeschickt werden kann.

Das **Design** der Applikation soll nach dem THM Corporate Design[29] gestaltet werden. Zudem soll es begleitend mit der Implementierung erstellt werden, da bei Android-Applikation der Code und das Design enger verzahnt sind als z.B. beim Moodle-Plugin.

Letztlich muss die Applikation im **Google Play Store veröffentlicht** werden. Dadurch können die Studierenden die Applikation wie üblich installieren. Für die Veröffentlichung ist ein Developer-Konto notwendig.

4.1.4 Probleme

Nachfolgend sind Problem-Komplexe beschrieben, die während der Entwicklung auftraten.

Android Studio

Für die Entwicklung von Android-Applikationen wird die integrierte Entwicklungsumgebung (IDE)[30] „Android Studio“ bereit gestellt. Mit dieser können alle notwendigen Arbeitsschritte

28 Transport Layer Security (TLS), weitläufiger bekannt unter der Vorgängerbezeichnung Secure Sockets Layer (SSL), ist ein hybrides Verschlüsselungsprotokoll zur sicheren Datenübertragung im Internet.

29 https://www.thm.de/site/hochschule/service/ag-kommunikation-und-marketing/corporate-design.html (abgerufen am 06. Januar 2018)

30 Eine integrierte Entwicklungsumgebung ist eine Sammlung von Computerprogrammen, mit denen die Aufgaben der Softwareentwicklung (SWE) möglichst ohne Medienbrüche bearbeitet werden können. Mehr Informationen unter https://de.wikipedia.org/wiki/Integrierte_Entwicklungsumgebung (abgerufen am 08. Januar 2018).

erledigt werden. Das macht diese IDE jedoch sehr mächtig, was in Komplexität resultiert. Die Einarbeitung in die Umgebung fiel vergleichsweise schwer, da es nur für einfache Aufgaben ein intuitives Verhalten gibt. Um diese Einarbeitung in größerer Tiefe zu ermöglichen, stehen im Internet viele Anleitungen zur Verfügung. Bei der Bearbeitung dieser Anleitungen und dem daraus resultierenden Einarbeiten in die IDE verging jedoch einige Zeit.

Design

Das Erstellen des Designs für die Applikation hielt ebenfalls mehrere Stolpersteine bereit. Zunächst ist es für Entwickler_innen, welche zuvor keinen Kontakt mit Design-Erstellung hatten, schwer sich in die Konzepte einzuarbeiten. Die Schnittmenge zwischen Informatik-Studium und Design ist relativ gering. Hier werden Aspekte wichtig, die zuvor bei der Entwicklung von Software keine Rolle spielten. Das sind z.B. die Lesbarkeit von Schrift, die Farbkonzeption und Anpassung an unterschiedliche Bildschirmgrößen. Zudem bereitete die sog. „Action Bar“ einige Probleme. Diese Leiste befindet sich am oberen Rand von Android-Applikationen und enthält Menü-Knöpfe. Änderungen daran können nicht ohne weiteres durchgeführt werden. Stattdessen muss eine neue Leiste definiert und die alte mit dieser ersetzt werden. Letztlich ist auch ausreichendes Wissen über die XML-Syntax von Nöten, da die Design-Definition im XML-Format beschrieben wird.

Im Hinblick auf die Verwendung des THM Corporate Design ergaben sich ebenfalls einige Probleme, da die Richtlinien zum Teil unklar oder schlicht unpraktikabel waren. So musste an mehreren Stellen von der Richtlinie abgewichen werden, um z.B. die Lesbarkeit von Text deutlich zu erhöhen.

Debugging

Es stellte sich heraus, dass der Android-eigene Debugger nicht auf Smartphones funktionierte. Dadurch konnte dieser nicht genutzt werden und es musste auf andere Techniken zurückgegriffen werden. Dies kostete viel Zeit.

4.1.5 Problembewältigung

Die vorher genannten Probleme konnten in erster Linie durch Einarbeitung in Beispiel-Code und Ausprobieren verschiedener Möglichkeiten gelöst werden. An einigen Stellen musste durch Ausprobieren und dem Einfügen von Test-Ausgaben auf mögliche Fehlerquellen geschlossen werden, um diese dann durch ausgiebige Internetrecherche zu beheben.

4.1.6 Features

Zwar bietet der Google Play Store bereits vielfältige Möglichkeiten an Nutzungs-Statistiken und Absturzberichte der eigenen Applikation zu gelangen, dennoch sind diese rudimentär, verglichen mit einem vollständigen Bericht, welcher von dem_der Nutzer_in gesendet wurde.

Zu diesem Zweck wurde ein Berichts-System eingebaut, welches ermöglicht, vollständige Berichte zu senden. Dazu wurde das bereits vorhandene Open-Source-Plugin ‚Shaky-Android'[31] genutzt. Wird während der laufenden Applikation das Smartphone geschüttelt, öffnet sich ein Popup-Dialog und es kann ein Bericht gesendet werden.

4.1.7 Screenshots

Nachfolgend wird die Android-Applikation mit einigen Screenshots gezeigt.

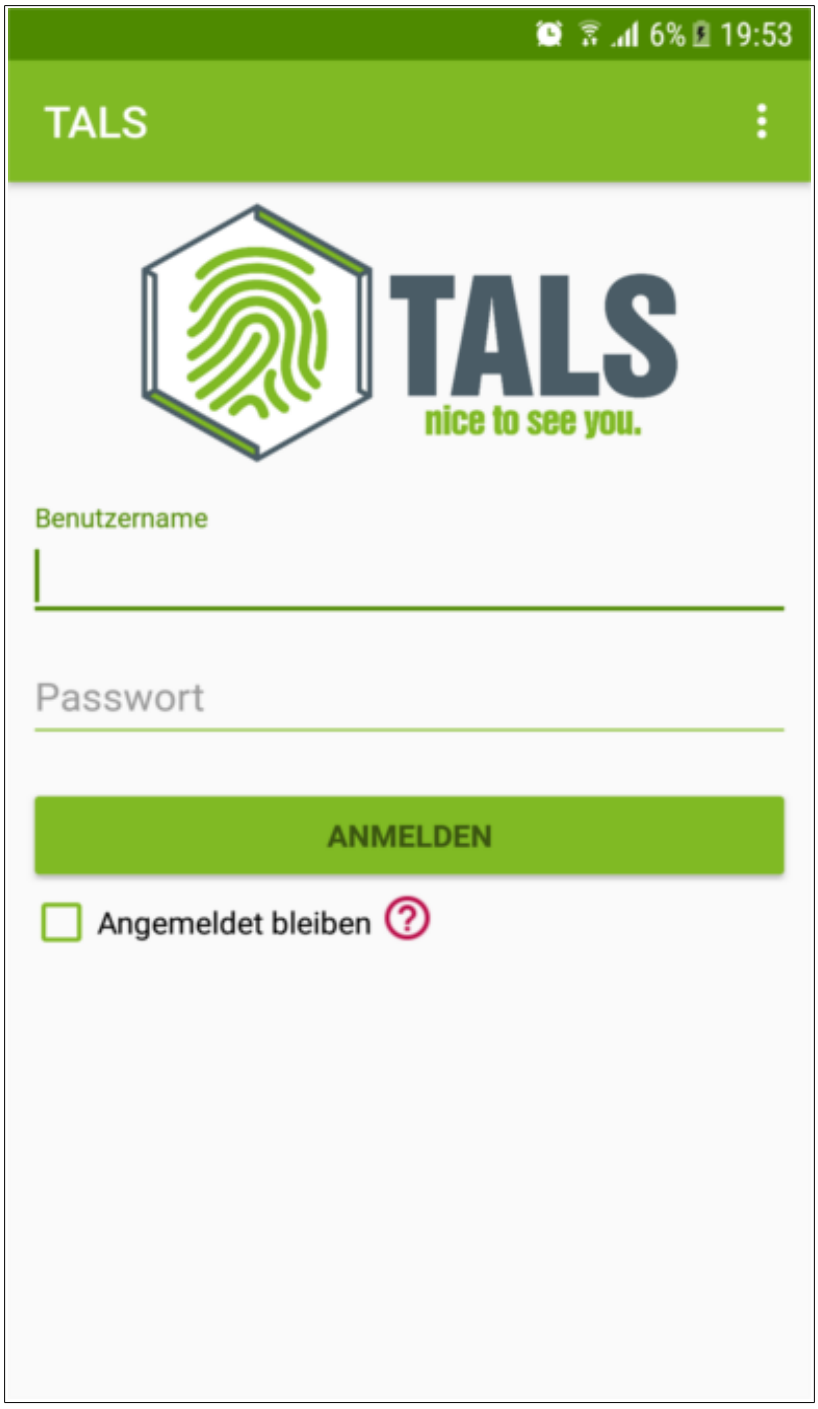

Abbildung 10: Anmelde-Bildschirm der Android-Applikation.

31 https://github.com/linkedin/shaky-android (Abgerufen am 05. Februar 2018).

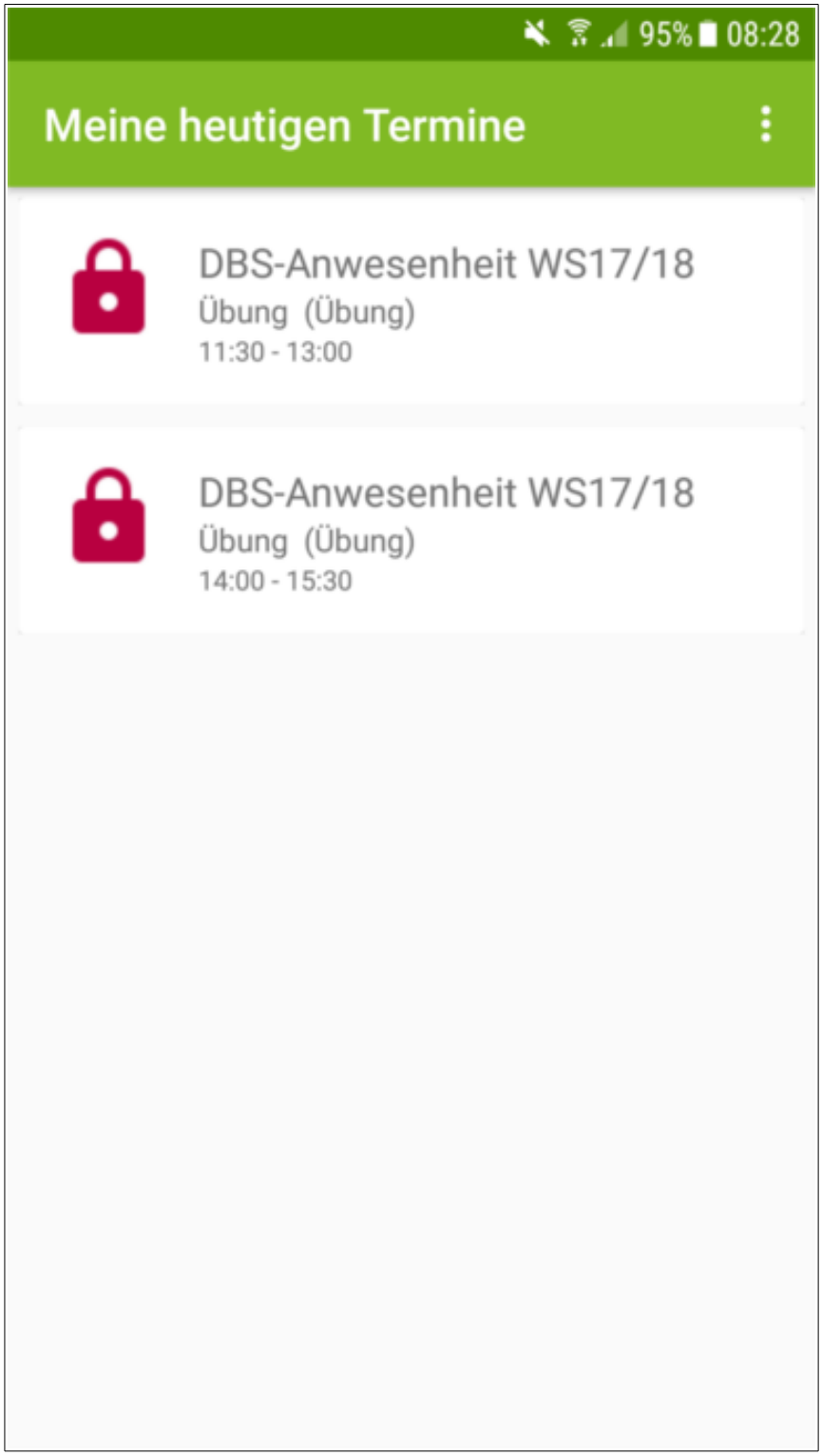

Abbildung 11: Terminliste der Android-Applikation.

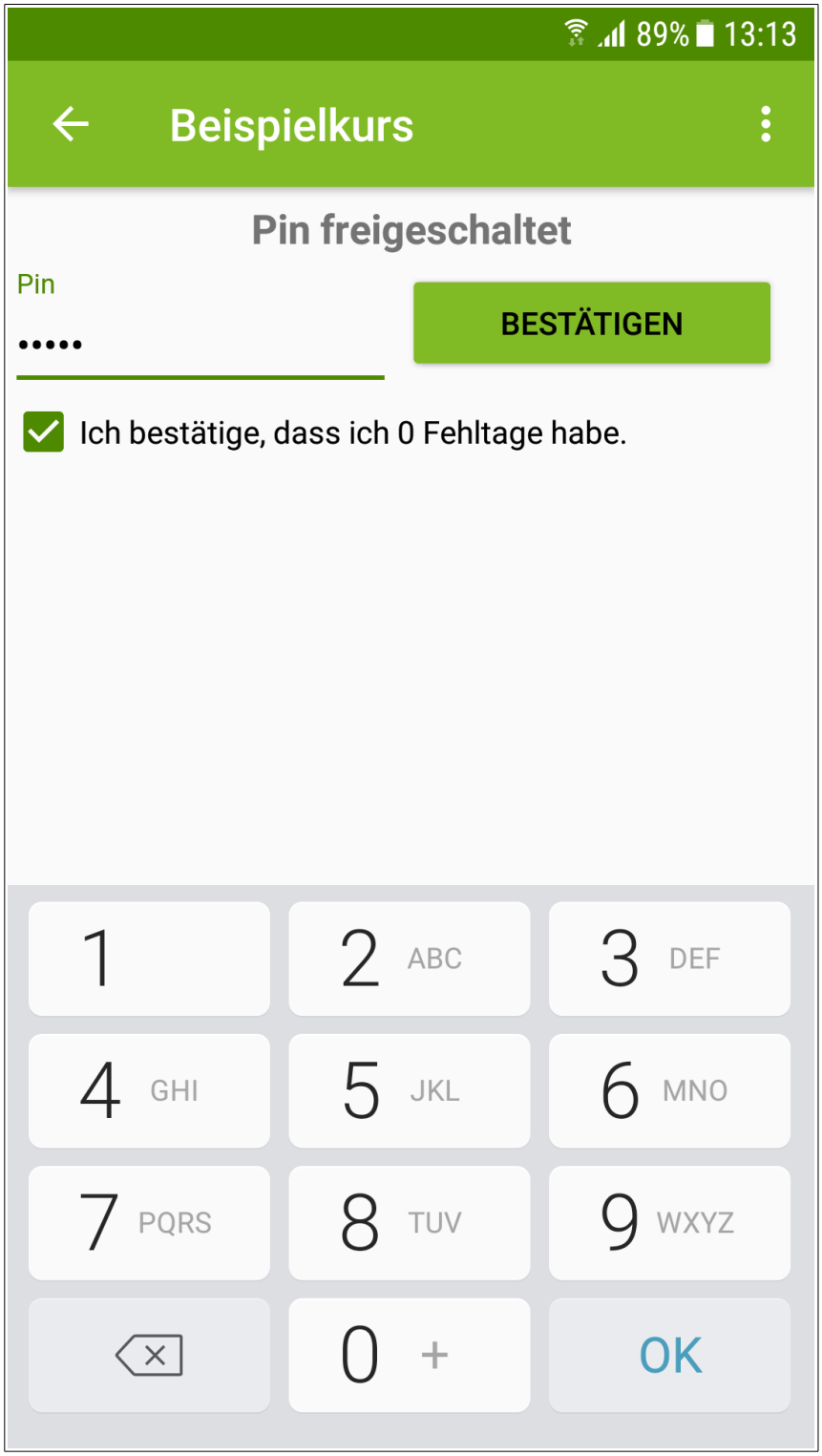

Abbildung 12: PIN-Abfrage in der Android-Applikation.

Abbildung 13: Wenn ein_e Student_in sich bereits eingetragen hat, kann dies nicht wiederholt werden.

4.2 Die iOS-Applikation

Im ersten Schritt musste die iOS-Version festgelegt werden, für die eine Applikation entwickelt werden soll. Bei der Entscheidung half die Betrachtung der aktuellen Statistik zur Verbreitung der verschiedenen iOS-Versionen.

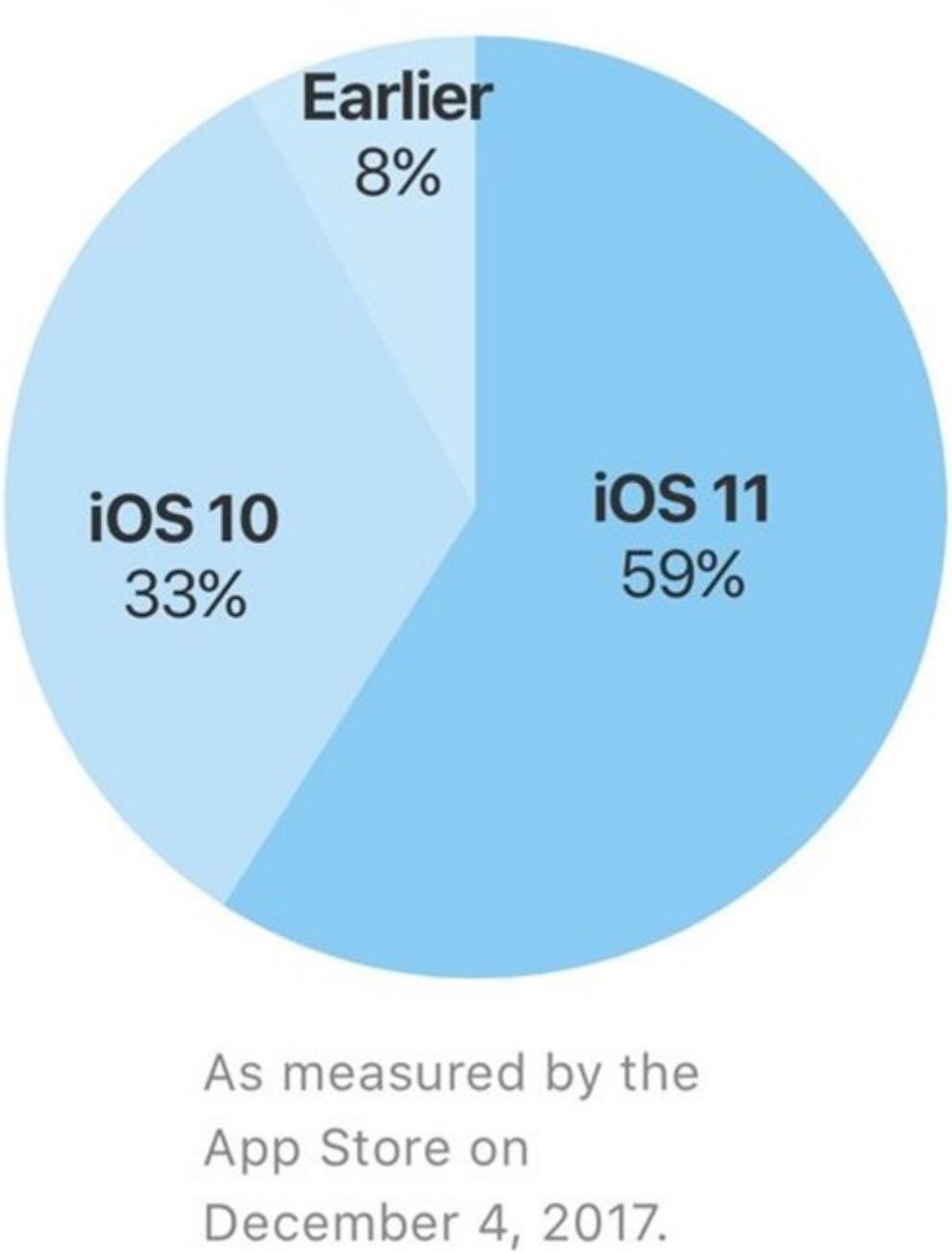

Tabelle 2: Verbreitung der iOS-Versionen vom 04. Dezember 2017.
Quelle: https://www.mactechnews.de/news/article/iOS-11-hinkt-bei-Verbreitung-hinterher-168488.html (abgerufen am 05.02.2018)

Auf Grundlage dessen wurde entschieden, die Applikation für die iOS-Version 10 und höher zu entwickeln. Dadurch wird eine möglichst große Abdeckung gewährleistet.

4.2.1 Systemanforderungen

iOS-Version	10 oder höher

4.2.2 Kurzbeschreibung

Diese Applikation ermöglicht es Studierenden der THM mit Hilfe ihres iPhones® ihre Anwesenheit während einer Veranstaltung einzutragen.

Die iOS-Applikation soll die selbe Funktionalität umfassen wie bereits die Android-Applikation.

4.2.3 Entwicklungsverlauf

Die Applikation ist als native iOS-Applikation implementiert. Dadurch ist der volle Funktionsumfang der Plattform nutzbar. Aus diesem Grund stand noch vor Entwicklungsbeginn die Einarbeitung in Objective-C[32].

Am Anfang steht das **Design einer Speicherstruktur**. Anders als bei z.B. Android bietet iOS, bzw. das Cocoa Framework[33], sog. „NSUserDefaults“ an. Dabei handelt es sich um vom Betriebssystem zur Verfügung gestellte Speicherstrukturen, die von Applikationen genutzt werden können. Sie sind Schlüssel-Wert-Paare und ermöglichen das persistente Abspeichern von Daten in verschiedenen Formaten.

Danach folgt das **Anmelden im THM-eigenen Moodle**. Da die THM für die Anmeldung bei ihren Diensten CAS verwendet, muss ein CAS-Client implementiert werden. Dieser muss es möglich machen, die Nutzer_innenkennnung und das Passwort einzugeben und sicher zu übertragen. Dies benötigt eine durch Transport-Layer-Security (TSL) gesicherte Verbindung.

Um nun an **die benötigten Daten** zu kommen, muss die Applikation das Nutzer_innen-Token abfragen. Daraufhin kann mithilfe dieses Tokens die Funktionalität des Moodle-Plugin voll genutzt werden. Zu Beginn muss die Kurs-Liste des_der Nutzer_in abgefragt werden. Direkt im Anschluss ist es möglich, alle am aktuellen Tag stattfindende Veranstaltungen in Erfahrung zu bringen. Für einzelne Veranstaltungen muss abgefragt werden können, ob die PIN bereits verfügbar ist. Es können folgende Daten abgerufen werden:

- Kurs-Liste
- Veranstaltungen des aktuellen Tages
- Bisherige Fehltage

32 Objective-C, auch kurz ObjC genannt, erweitert die Programmiersprache C um Sprachmittel zur objektorientierten Programmierung. Mehr Informationen unter https://de.wikipedia.org/wiki/Objective-C (abgerufen am 05. Feburar 2018).

33 Cocoa ist eine objektorientierte Programmierschnittstelle zur Programmierung unter dem Betriebssystem macOS von Apple. Mehr Informationen unter https://de.wikipedia.org/wiki/Cocoa_(API) (abgerufen am 05. Februar 2018).

- Ob eine PIN bereits/noch verfügbar ist

Damit der_die Nutzer_in **die Anwesenheit eintragen** kann, muss es eine Oberfläche geben, auf der eine PIN für eine bestimmte Veranstaltung eingegeben und abgeschickt werden kann.

Das **Design** der Applikation soll nach dem THM Corporate Design gestaltet werden.

Letztlich muss die Applikation im **Apple App Store veröffentlicht** werden. Dadurch können die Studierenden die Applikation wie üblich installieren.

4.2.4 Probleme

Objective-C

Die Einarbeitung in die Programmiersprache Objective-C nahm mehrere Wochen in Anspruch. Erst danach konnte mit der Entwicklung der iOS-Applikation begonnen werden. Die fehlende Erfahrung mit dieser Sprache führte im Entwicklungsverlauf immer wieder zu Verzögerungen, da sich erst in neue Paradigmen zur Problemlösung eingearbeitet werden musste.

Xcode

Um eine iOS-Applikation entwickeln zu können, wird von Apple die IDE Xcode[34] zur Verfügung gestellt. Allerdings ist diese nur für MacOS verfügbar. Dies stellte zunächst ein Problem dar, da kein Mac-Gerät zur Verfügung stand. Im Verlauf der Entwicklung machte Xcode immer wieder Probleme, da sich erst in diese IDE eingearbeitet werden musste.

Design

Das Oberflächendesign der iOS-Applikation bereitete immer wieder Probleme. So passierte es, dass die Oberfläche auf unterschiedlichen Bildschirmgrößen anders bzw. fehlerhaft dargestellt wurde. Da dies die Gebrauchstauglichkeit mindert, war die Oberflächengestaltung eine besondere Herausforderung.

Es ist für Entwickler_innen, welche zuvor keinen Kontakt mit Design-Erstellung hatten, schwer sich in die Konzepte einzuarbeiten. Die Schnittmenge zwischen Informatik-Studium und Design ist relativ gering. Hier werden Aspekte wichtig, die zuvor bei der Entwicklung von Software keine Rolle spielten.

4.2.5 Problembewältigung

In der Entwicklung wurde stark auf den Ansatz der testgetriebenen Entwicklung[35] zurück gegriffen. Durch die Anbindung an eine verfügbare Test-Installation der Moodle-Plattform

34 Xcode ist eine integrierte Entwicklungsumgebung von Apple, mit der man Programme für macOS, iOS, tvOS und watchOS schreiben kann. Mehr Informationen unter https://de.wikipedia.org/wiki/Xcode (abgerufen am 05. Februar 2018).

35 Testgetriebene Entwicklung ist eine Methode, die häufig bei der agilen Entwicklung von Computerprogrammen eingesetzt wird. Mehr Informationen unter https://de.wikipedia.org/wiki/Testgetriebene_Entwicklung (abgerufen am 05. Februar 2018).

konnten die neu entwickelten Funktionalitäten durchgehend getestet und mit dem zu erwartenden Ergebnis verglichen werden.

Ebenso konnte die Entwicklung anhand von Beispiel-Code zielgerichtet geführt werden.

4.2.6 Screenshots

Nachfolgend wird die iOS-Applikation mit einigen Screenshots gezeigt.

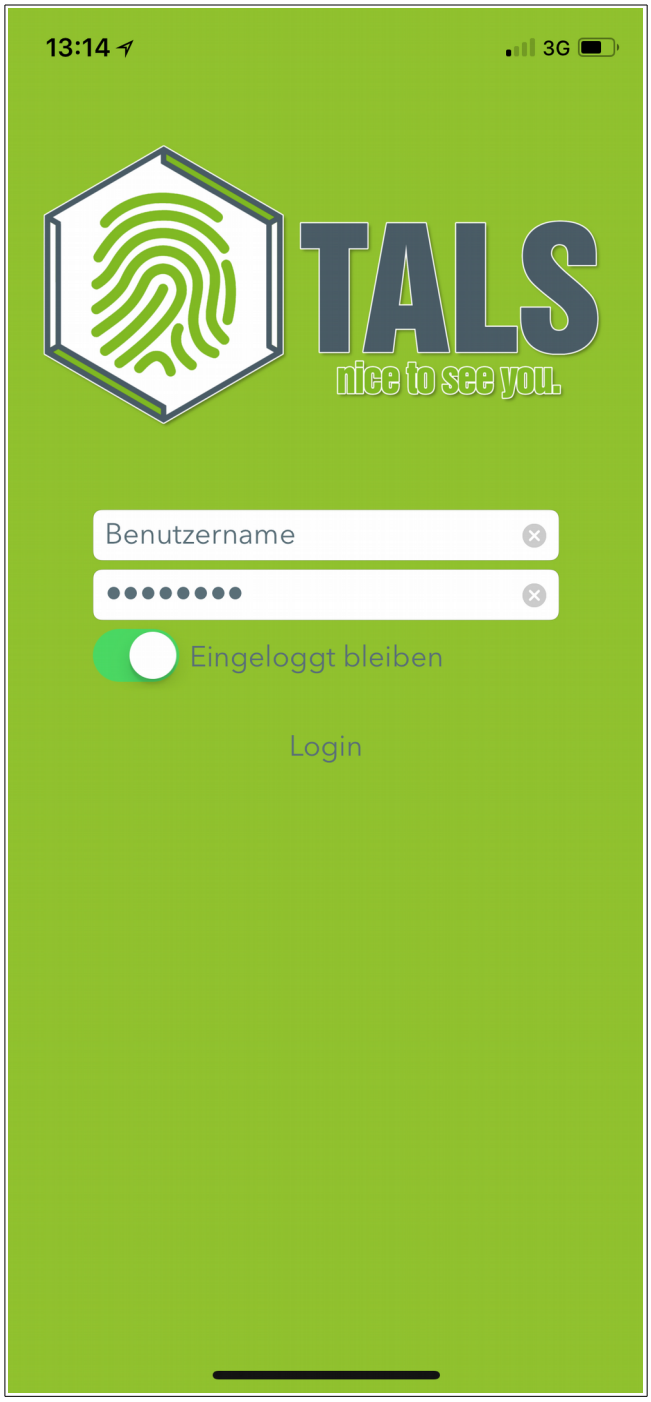

Abbildung 14: Anmelde-Bildschirm der iOS-Applikation.

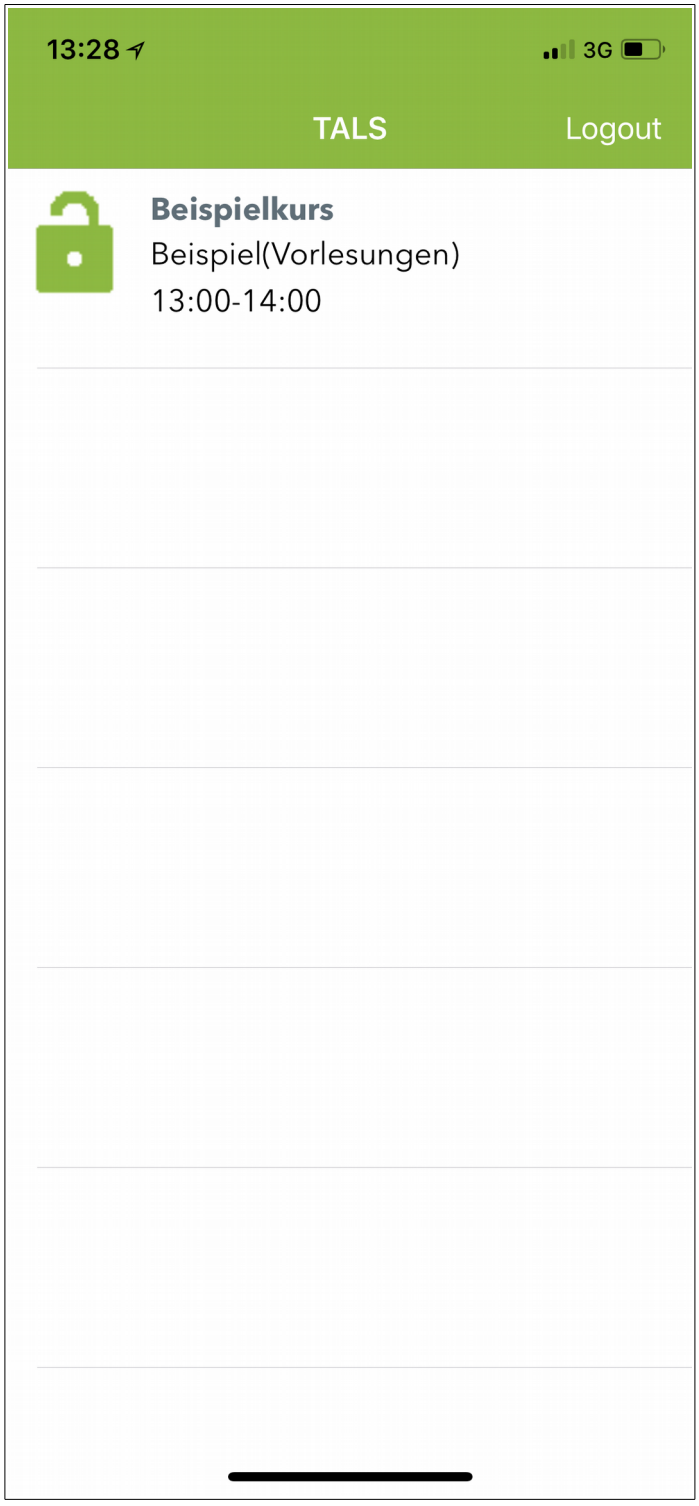

Abbildung 15: Terminliste in der iOS-Applikation.

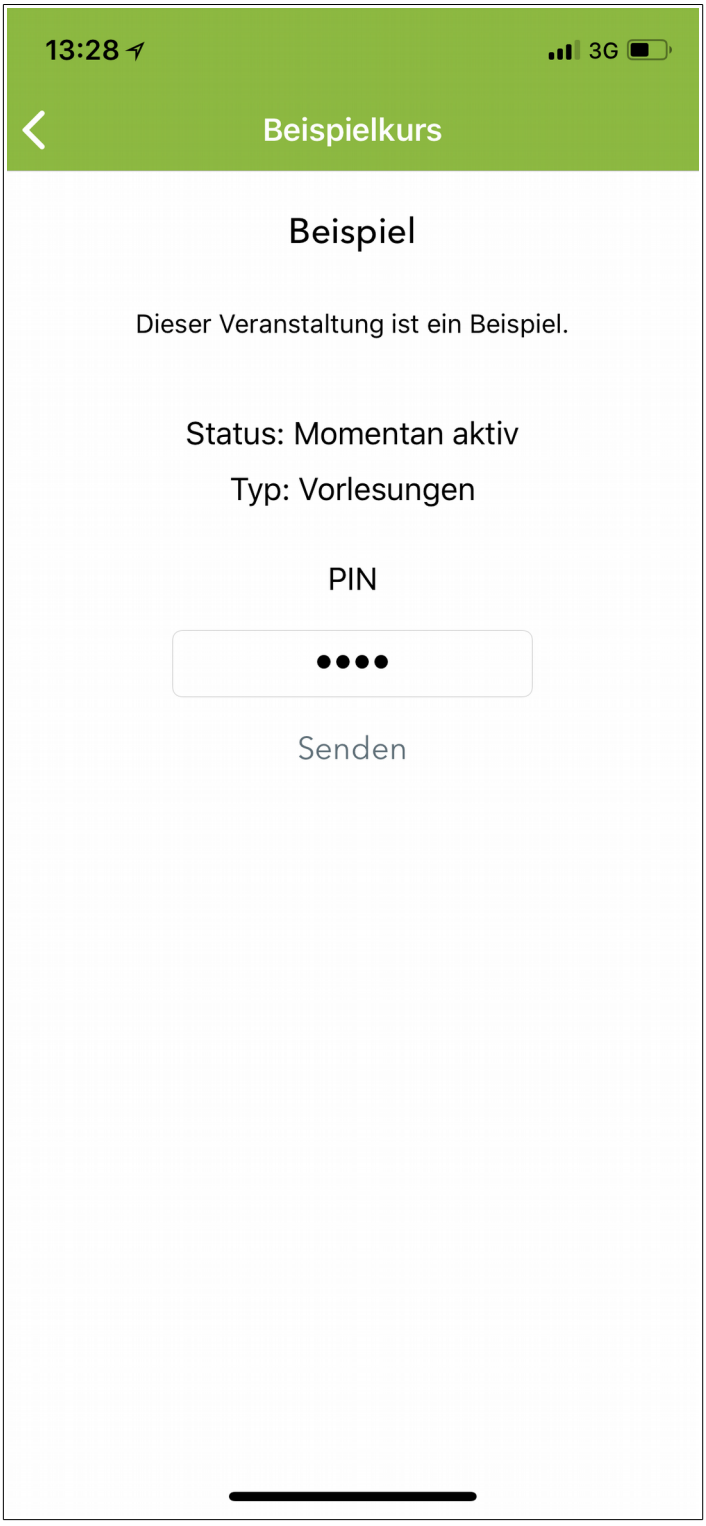

Abbildung 16: PIN-Eingabe in der iOS-Applikation.

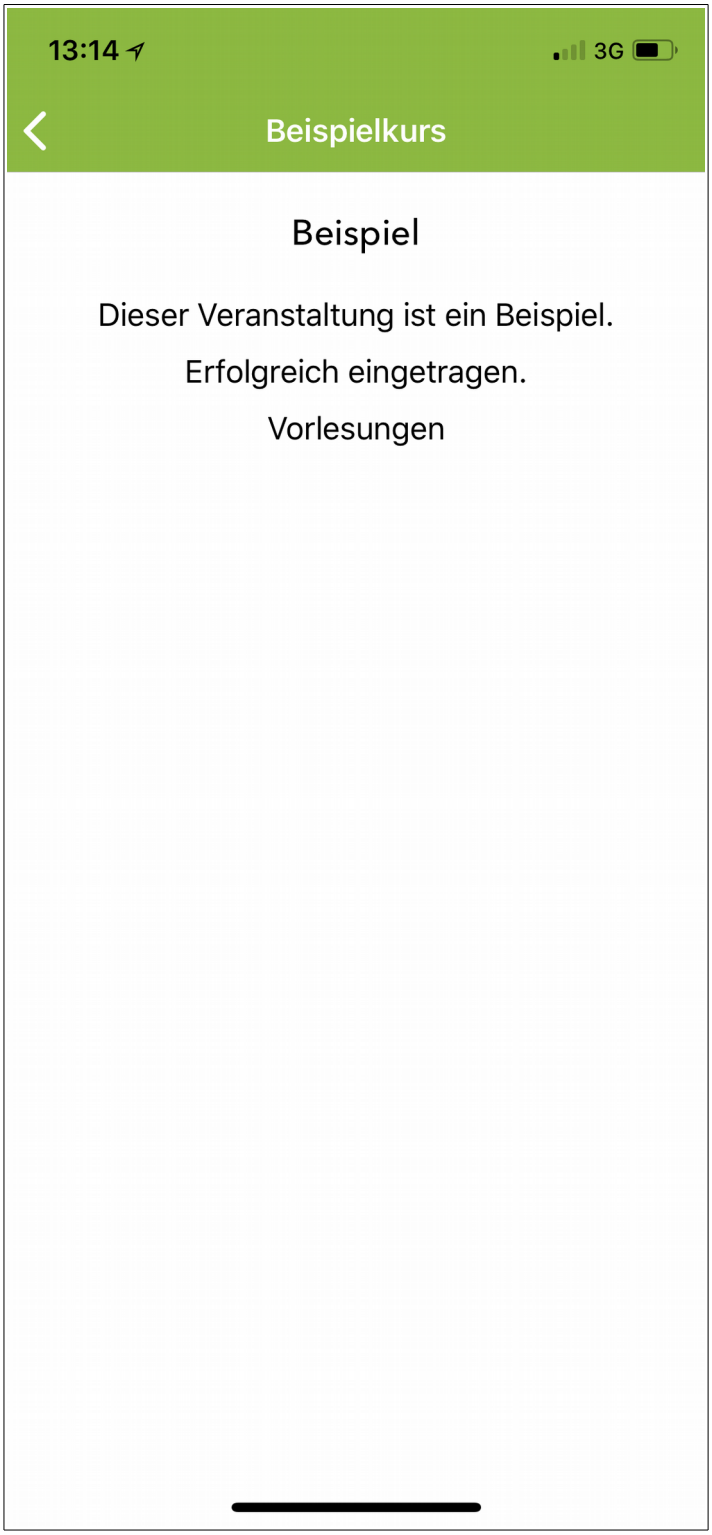

Abbildung 17: Wenn ein_e Student_in sich schon eingetragen hat, kann dies nicht wiederholt werden.

5 Kommunikation

Die Kommunikation zwischen den verschiedenen Teilen findet mit einer HTTPS-Verbindung und der RESTful-API statt. Moodle bietet dafür bereits eine passende Schnittstelle, welche lediglich aktiviert werden musste. Für Antworten hält Moodle nativ die Formate XML und JSON bereit, wovon wir letzteres nutzen.

5.1 Authentifizierung

Um eine sichere Übertragung der notwendigen Daten zu gewährleisten und um die Gebrauchstauglichkeit möglichst hoch zu halten, wird zur Kommunikation auf einen CAS sowie auf Token zurückgegriffen.

5.1.1 Central Authentication Service – CAS

Die THM verwendet für ihre Dienste einen CAS. Dadurch können sich Nutzer_innen an einer zentralen Stelle im System anmelden und dann auf alle Dienste zugreifen. Es lag nahe, dieses Verhalten auch für die Smartphone-Applikationen zu übernehmen.

CAS war ursprünglich eine Entwicklung der Yale University und bildet ein föderiertes Identitätsmanagement für beliebig viele Dienste. Mittlerweile wird das System von dem Unternehmenszusammenschluss JA-SIG/Apereo weitergeführt. Es soll vor allem Bildungseinrichtungen zu gute kommen.

JA-SIG/Apereo stellt eine umfassende Dokumentation des CAS sowie Client-Implementierungen für verschiedene Programmiersprachen kostenlos bereit. Allerdings ist diese Dokumentation unvollständig und die zur Verfügung stehenden Client-Implementierungen sind fehlerhaft bzw. nicht ohne weiteres verwendbar. Es stellte sich schließlich heraus, dass es keine geeignete Implementierung für unseren Anwendungsfall gab. Während der Entwicklung der iOS-Applikation traten hierbei allerdings keine größeren Probleme auf. Deswegen wird nachfolgend der Entwicklungsprozess für Android geschildert.

Zunächst musste das Verhalten des CAS nachvollzogen werden. Dazu wurde die Dokumentation studiert wie auch durch das Beobachten des an der THM verwendeten Web-Clients wichtige Rückschlüsse auf die Cookie[36]-Verwaltung gezogen.

Es stellte sich folgendes Verhalten heraus:

- Anmelden:

    1. Bei Aufruf eines Dienstes leitet dieser auf den CAS weiter.

36 Ein Cookie ist eine Textinformation, die die besuchte Website über den Webbrowser auf dem Computer des_der Betrachter_in platziert. Mehr Informationen unter https://de.wikipedia.org/wiki/HTTP-Cookie (abgerufen am 06. Februar 2018).

2. Bei dieser Weiterleitung wird die URL[37] des Dienstes als Parameter übergeben. An diese URL leitet CAS nach (erfolgreicher) Anmeldung zurück.
3. CAS fragt nun Benutzer_innenname sowie Passwort (oder ggf. eine andere Authentifizierung) ab. Diese wird dann geprüft.
4. Ist die Anmeldung erfolgreich, leitet CAS an die angegebene URL aus 2. zurück und liefert ein sog. Ticket-Granting-Ticket (TGT) mit. Dieses stellt den Nachweis dar, dass die Anmeldung erfolgreich war und ist jeweils einzigartig.

- Validieren:
  1. Bei Aufruf eines Dienstes leitet dieser auf den CAS weiter.
  2. Bei dieser Weiterleitung wird die URL des Dienstes sowie das TGT als Parameter übergeben.
  3. CAS prüft das TGT auf Gültigkeit.
  4. Bei einem gültigen TGT leitet CAS auf den Dienst zurück und übergibt eine Antwort im CAS-XML oder SAML-XML[38] Format. Darin ist mindestens der Nutzer_innenname enthalten.
- Abmelden:
  1. Bei Aufruf des Abmelde-URL eines Dienstes leitet dieser auf den CAS weiter.
  2. Bei dieser Weiterleitung wird die URL des Dienstes sowie das TGT als Parameter übergeben.
  3. CAS prüft das TGT auf Gültigkeit und markiert es als ungültig.

Alle Schritte lassen sich auch ohne einen Dienst direkt mit dem CAS Web-Client durchführen. In diesem Fall entfällt die Weiterleitung an einen bestimmten Dienst.

Obwohl nun das Verhalten bekannt war, gab es sehr viele Schwierigkeiten einen funktionierenden Client für Android zu entwickeln. Dies lag an verschiedenen Eigenheiten des CAS, die von Außen nicht sofort erkennbar waren.

Die Theorie hinter der Authentifizierung mittels CAS ist in der offiziellen Dokumentation zwar gut beschrieben, dennoch fehlen wichtige Hinweise zur programmiertechnischen Umsetzung. Erst durch Reverseengineering stellte sich z.B. heraus, dass es zwischen dem CAS und dem Dienst sehr viele Weiterleitungen gab. Diese wurden von den zunächst verwendeten Android-Funktionen nicht unterstützt. Hier war es eine Herausforderung, sich das Wissen trotz der fehlenden Dokumentation zu erarbeiten. Zudem wird nicht nur das (zentrale) TGT verwendet, sondern noch weitere Cookies.

37 Uniform Resource Locator. Mehr Informationen unter https://de.wikipedia.org/wiki/Uniform_Resource_Locator (abgerufen am 06. Februar 2018).

38 Die Security Assertion Markup Language (SAML) ist ein XML-Framework zum Austausch von Authentifizierungs- und Autorisierungsinformationen. Mehr Informationen unter https://de.wikipedia.org/wiki/Security_Assertion_Markup_Language (abgerufen am 06. Februar 2018).

Die Cookies werden allerdings nicht bei jedem Aufruf einzeln übergeben. Es wird stattdessen durch einen sog. ‚CookieStore‘[39] ein ständiger Speicher verwendet, aus dem bei Bedarf Cookies abgerufen werden können.

Da es immer wieder zu Fehlern kommen kann, z.B. wenn die Anmelde-Daten nicht stimmen, und es seitens CAS keine besondere Fehlerbehandlung gibt, werden diese Fehler von dem CAS-Client für Android abgefangen und behandelt. Die Fehlerbehandlung wird dann so geführt, wie es für die Gebrauchstauglichkeit am passendsten ist.

Um den eigens entwickelten CAS-Client für Android möglichst modular zu halten, sind die zu verwendenden URLs zentral abgelegt und so leicht anpassbar. Dadurch lässt sich der CAS-Client auch für andere CAS‘ als das der THM verwenden.

Für die Entwicklung des CAS-Clients für Android waren mehrere Wochen intensiver Auseinandersetzung mit CAS, Java, Android, HTTP, REST und Web-Technologien nötig.

5.1.2 Token

Wie bereits erwähnt werden von Moodle selbst Tokens verwendet. Diese werden genutzt um eine_n Nutzer_in zu identifizieren. Allerdings sind diese Tokens kein Ersatz für eine korrekte Anmeldung im System. Der Gebrauchsumfang dieser Tokens ist, absichtlich, sehr beschränkt. So braucht ein_e Nutzer_in für jeden zu nutzenden Webservice ein eigenes Token. Diese Tokens sind zwangsweise maximal drei Monate gültig und erlauben lediglich einen Zugriff auf den zuvor angegebenen Webservice.

Greif die Smartphone-Applikation das erste Mal auf Moodle zu, wird zunächst ein Token abgefragt. Das Moodle-Plugin stellt eine Schnittstelle bereit, um ein solches Token abzufragen bzw. automatisiert zu erzeugen. Dabei wird lediglich eine interne Moodle-Funktion genutzt.

39 Ein CookieStore ist ein Speicher für Cookies. Es können Cookies gespeichert und abgerufen werden. Mehr Informationen unter https://docs.oracle.com/javase/9/docs/api/java/net/CookieStore.html (abgerufen am 06. Februar 2018).

6 Beta-Test

Am 24. und 25. Januar 2018 wurde in vier Übungs-Blöcken mit insgesamt 70 Studierenden ein Beta-Test des Systems durchgeführt.

6.1 Aufbau

Im Rahmen des Kurses „CS1020 – Datenbanksysteme“ im Wintersemester 2017/2018 bei Herr Dr. habil. Frank Kammer fanden am 24. und 25. Januar 2018 insgesamt vier Übungen à 90 Minuten als Pflichtveranstaltungen statt.

Dabei wurde das System zunächst den Studierenden kurz präsentiert. Da zu dieser Zeit keine Installation im THM-eigenen Moodle bestand, wurde auf eine Test-Installation zurückgegriffen. Dazu hatte Herr Kammer ein Moodle auf einem Server installiert und Herr Herwegh hatte dann das TALS-Plugin installiert und gemeinsam mit Herr Kammer die Veranstaltung erstellt.

Die Studierenden sollten sich dann zunächst in diesem Test-Moodle anmelden, was durch die Integration von CAS einfach funktionierte. Zudem waren die Studierenden angehalten, sich die Android-Applikation zu installieren.

6.2 Durchführung

Zu einem gegebenen Zeitpunkt während der jeweiligen Veranstaltung wurde den Studierenden die PIN bekannt gemacht und wie lange diese aktiv sein würde.

Die Studierenden kamen gut mit dem System zurecht. Einige Studierende verwendeten die Anmeldung via Browser auch auf ihrem Smartphone. Auch dieser Weg war gut in der Handhabung.

Insgesamt nahmen 70 Studierende an diesem Test teil.

6.3 Entdeckte Fehler

Dank diesem Test wurden einige kleinere Fehler bekannt und behoben.

Dabei handelte es sich nur um geringe Fehler. Ein Fehler war ein falsch beschrifteter Button. Ein anderer Fehler betraf die Gebrauchstauglichkeit insofern, dass in der Android-Applikation beim Anmelden nach dem Klick auf den Button „Anmelden“ die Textfelder geleert wurden. Zudem fehlte bei bestimmten Bildschirmauflösungen das Logo.

Der schwerwiegendste gefundene Fehler, der gefunden wurde, war, dass die Android-Applikation beim Starten auf einigen wenigen Geräten direkt abstürzte. Dies lag an einer falsch skalierten Bild-Ressource für das Logo, welches beim Anmelde-Bildschirm angezeigt wird.

7 Ausblick

Das System ist bewusst offen gestaltet, damit Erweiterungen in Zukunft möglich sind.

Als Erweiterung für das Moodle-Plugin wäre z.B. denkbar:

- Integration des vom Fachbereich MNI verwendeten Stundenplan-Tools.
- Auswertung der Veranstaltungen durch bessere, statistische Methoden.
- Ein Feedback-System, damit die Dozierenden den Studierenden eine Rückmeldung geben können, ob die bisherige Anwesenheitsquote erfüllt ist oder ob Handlungsbedarf besteht.
- Handhabung von Testaten.

Für die Smartphone-Applikation wäre als Erweiterung denkbar:

- Eine Dozierenden-Ansicht, damit ein PIN auch über das Smartphone aktiviert werden kann.
- Push-Nachrichten für kommende/laufende Termine.
- Popup für den Sperrbildschirm, um die PIN einfacher und schneller eingeben zu können.