

HAUSARBEIT WAHLPFLICHTMODUL MOBILE ANWENDUNGEN Q4 2016

Stundenplan App an der Nordakademie

Daniel Bormann, Daniel.Bormann@nordakademie.de, 14150
Leif Johannsen, Leif.Johannsen@nordakademie.de, 13939
Arvid Ottenberg, Arvid.Ottenberg@nordakademie.de, 14374
Timo Daniel Schlatter, Timo.Schlatter@nordakademie.de, 14349

Table of Contents

1	Beschreibung des Projekts	3
1.1	SQLite-Datenbank	3
1.1.1	TimetableContract	3
1.1.2	TimetableDbHelper	3
1.2	ContentProvider	3
1.2.1	TimetableProvider	3
1.3	Tests.....	3
1.4	Preference Screen	4
1.5	Sync Adapter	4
1.6	Das Layout	4
1.6.1	Activities	4
1.6.2	Fragments.....	4
1.6.3	Adapter.....	4
2	Bedienungsanleitung „NAK Timetable“	5
2.1	Der Startbildschirm	5
2.1.1	Einstellungen	5
2.1.2	Neu-Laden	6
2.2	Tagesansicht.....	6
2.3	Veranstaltungsansicht	7
2.4	Dozentenansicht	8
3	Matrix nicht funktionaler Anforderungen	9
4	Selbstreflexionen.....	10
4.1	Daniel Bormann (14150).....	10
4.2	Leif Johannsen (13939)	10
4.3	Arvid Ottenberg (14374).....	10
4.4	Timo Schlatter (14349)	10
5	Quellen& Hilfsmittel.....	10
6	Eigenständigkeitserklärung.....	11

1 Beschreibung des Projekts

Das Repository ist unter folgender Adresse zu erreichen:

https://gitlab.nordakademie.de/14349/stundenplan_app/

Hier liegt auch die APK-Datei in dem Verzeichnis stundenplan_app/APK-FileDirectory zu finden. Sie heißt „**NakTimetable-release.apk**“.

1.1 SQLite-Datenbank

1.1.1 TimetableContract

In der TimetableContract-Klasse werden Konstanten für die frei verschiedenen Datenbanktabellen definiert. Es gibt eine Tabelle für Module, eine für Dozenten und eine für die Veranstaltungen. Dazu zählen z.B. die Tabellen- und Spaltennamen, sowie (Bestandteile von) Uris. Des weiteren werden Methoden für die Erzeugung von Uris und zum Extrahieren von Informationen aus Uris zur Verfügung gestellt.

Die TimetableContract-Klasse ist eine angepasste Version der WeatherContract-Klasse aus dem Sunshine Source Code des MOOC.

1.1.2 TimetableDbHelper

Die TimetableDbHelper-Klasse dient der Erzeugung der Datenbanktabellen. Die Benennung der Tabellen und Spalten erfolgt nach den im TimetableContract definierten Werten.

Die TimetableContract-Klasse ist eine angepasste Version der WeatherContract-Klasse aus dem Sunshine Source Code des MOOC.

Die TimetableDbHelper-Klasse ist eine angepasste Version der WeatherDbHelper-Klasse aus dem Sunshine Source Code des MOOC.

1.2 ContentProvider

1.2.1 TimetableProvider

Die TimetableProvider-Klasse stellt Methoden für CRUD-Zugriffe auf die Datenbank zur Verfügung. Die Methoden werden immer mit einer Uri als Parameter aufgerufen, welche die Information enthält, auf welcher Tabelle die Operation ausgeführt werden soll. Diese Zuordnung erfolgt durch den UriMatcher. Des weiteren kann die Uri z.B. Parameter für eine WHERE-Bedingung enthalten. Diese Bedingungen werden ebenfalls in dieser Klasse definiert.

Die TimetableProvider-Klasse ist eine angepasste Version der WeatherProvider-Klasse aus dem Sunshine Source Code des MOOC.

1.3 Tests

Die korrekte Funktionalität der Klassen des data-Packages sowie der DateFormatConverter-Klasse wird mit Tests sichergestellt.

Die Test-Klassen sind angepasste Versionen der Test-Klassen aus dem Sunshine Source Code des MOOC.

1.4 Preference Screen

Der Preference-Screen umfasst neben der darstellenden XML, auch die Settingsactivity. Diese wird bei Auswählen des „Settings“-Button wird diese ausgeführt.

Während über die `pref_general.xml` nur der Rahmen gesetzt wird (welche Preferences es gibt und welchen Datentypen diese implementieren), verwaltet die Settings-Activity die Logik bei der Eingabe. Hier wird bspw. die Eingabe der Kalenderwoche überprüft. Dem User wird bei Eingabe einer ungültigen KW (bspw. 345 oder 0) eine Ausgabe gegeben. Hierbei werden nach erfolgreicher Validierung die Daten auf den übergebenen Wert geändert.

Ebenso ist hier die Logik für ein Checkbox-Feld („Sync Calendar Week“) hinterlegt, welche die manuelle Eingabe der Kalenderwoche unmöglich macht. Hierbei wird bei Aktivierung der Checkbox „Sync Calendar Week“ das EditText-Feld „Calendar Week“ gesperrt.

In diversen anderen Methoden wird auf diese Preferences zugegriffen um weitere (kontextbezogene) Logik auszuführen.

1.5 Sync Adapter

Der SyncAdapter ist für das Synchronisieren von Daten zwischen der App und dem Web-Server zuständig. Das Framework unterstützt die Verwaltung und Automatisierung von Datenübertragungen und koordiniert Synchronisationsoperationen über verschiedene Apps hinweg. Durch die Verwendung des SyncAdapters, können mehrere Funktionen genutzt werden. Er ermöglicht die Automatisierung der Datenübertragung basierend auf verstrichener Zeit. Das System führt Ihre Datenübertragung nur aus, wenn das Gerät über eine Netzwerkverbindung verfügt. Es werden noch eine Reihe an weiteren Funktionalitäten geboten, die in dieser App allerdings nicht implementiert wurden.

1.6 Das Layout

1.6.1 Activities

Die App besteht aus zwei Activities. Dabei wurde das grundlegende Design analog zur Sunshine-App gehalten. Die MainActivity zeigt beinhaltet auf Smartphone im Hochkant-Modus nur die Master-Ansichten (siehe Fragments). Die DetailActivity kommt nur in diesem Fall zum Einsatz um die Detail-Ansicht in einer gesonderten Activity anzuzeigen. Im Landscape-Modus, sowie auf Geräten mit einer DPI größer 600, beinhaltet die MainActivity auch die Detail-Fragmente.

1.6.2 Fragments

Das MainWeekOverviewListFragment stellt die Hauptansicht dar und ist mit den Wochentagen der ausgewählten Woche befüllt.

Das MainWeekOverviewSwipeFragment stellt einzelnen Wochentage in einem ViewPager dar. Dieser ist pro Tag mit einem ListLessonFragment befüllt, der die einzelnen Unterrichtseinheiten am jeweiligen Tag darstellt.

Das LessonDetailFragment und das LecturerDetailFragment stellen die Details zu den Vorlesungen und Dozenten dar. Es werden neben den generellen Informationen auch Änderungen angezeigt. Außerdem wird durch Intents die Möglichkeit gegeben den Dozenten anzurufen oder ihm eine Mail zuschreiben.

1.6.3 Adapter

Zum Befüllen des MainWeekOverviewListFragment wurde nicht bewusst kein Cursoradapter gewählt, sondern ein ArrayAdapter, da es mit der geforderten API nicht möglich war einen Cursor mit dem Flag „distinct“ zu erstellen. Dies wurde durch ein Set kompensiert.

Das MainWeekOverviewSwipeFragment wird mit einer Implementation des PagerAdapters befüllt. Dieser generiert für jede Seite ein eigenes Fragment (ListLessonFragment). Dies war nicht Teil des Unterrichtsinhalts.

Analog zur Sunshine-App wird das ListLessonFragment mit einem CursorAdapter befüllt.

2 Bedienungsanleitung „NAK Timetable“

2.1 Der Startbildschirm

Vom Startbildschirm lässt sich mit einfachem Tippen auf den gewünschten Tag eine detailliertere Ansicht des Tages aufrufen. Weitere Informationen hierzu finden Sie unter 1.2 Tagesansicht.

Durch das Tippen auf den „...“-Button (neuere Android Versionen) oder den „Optionen“-Button des jeweiligen mobilen Endgerätes (ältere Android Versionen) lassen sich weitere Funktionen anzeigen (siehe 1.1.1 und 1.1.2).

Durch das Drücken des Zurück-Buttons wird immer zur vorherigen Ansicht gesprungen. Vom Startbildschirm wird zum Home-Bildschirm gesprungen.

2.1.1 Einstellungen

In den Einstellungen kann der User sich die App nach seinen eigenen Vorstellungen einstellen! Hier sind die Funktionen „Auswahl der Zenturie“, „Automatische Synchronisation der Kalenderwoche“, „Manuelle Konfiguration der Kalenderwoche“ und „Push-Benachrichtigungen zulassen“.

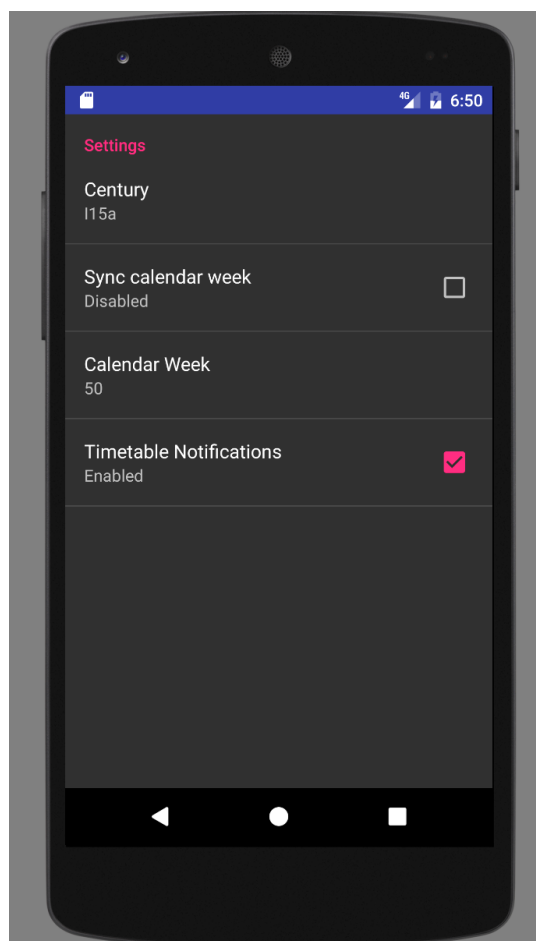


Abbildung 1: Settings

2.1.2 Neu-Laden

Sollte man ein manuelles Update vornehmen wollen, so kann man über den „Optionen“- Button des mobilen Endgeräts oder den „...“-Button in der App das Dropdown-Menü öffnen. Hier kann über den Button „Refresh“ eine Aktualisierung der Daten vorgenommen werden.

2.2 Tagesansicht

In dieser Sicht sind alle Vorlesungen des betreffenden Tages zu finden. Hierbei wird oben-links der jeweilige Tag angezeigt.



Abbildung 2: Tagesansicht

Auf jedem Bildschirm werden nun die Vorlesungstermine und Namen angezeigt. Durch horizontales Swipen, kann in dieser Ansicht durch die jeweiligen Tagesansichten navigiert werden. Das Symbol auf der rechten Seite jedes Moduls zeigt an, ob es hier kürzlich Änderungen gab. Es gibt die verschiedenen Symbole:



- Zeitänderung



- Raumänderung



- Raum und Zeitänderung

-  Ausfall


Durch Auswählen einer Veranstaltung wird die Veranstaltungsansicht geöffnet (siehe 1.3 Veranstaltungsansicht).

2.3 Veranstaltungsansicht

In dieser Sicht werden alle relevanten Informationen für die ausgewählte Vorlesung angezeigt. Hierbei wird zuerst der Änderungscode aufgelöst und rot angezeigt, sollte eine Änderung stattgefunden haben. Wenn keine Änderung der Veranstaltung stattfand, wird dieses Feld nicht angezeigt.

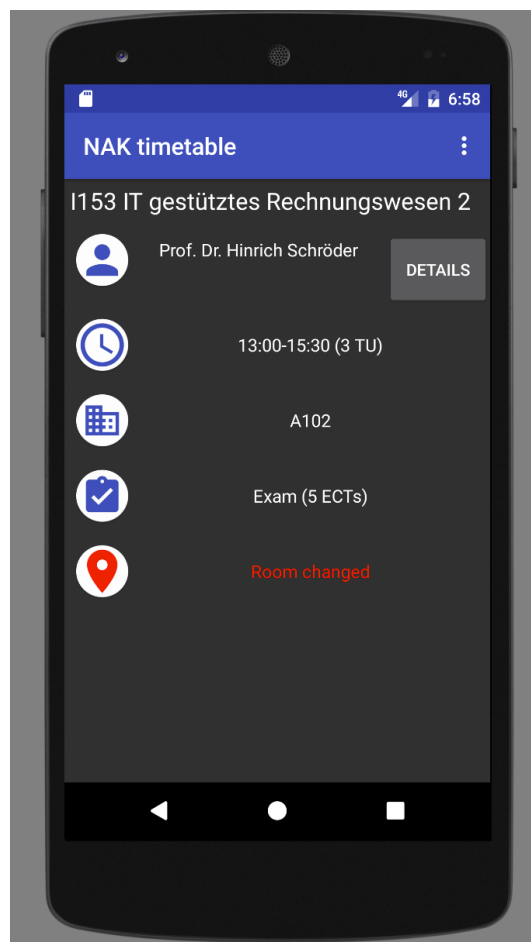


Abbildung 3: Vorlesungsansicht

Als letztes ist hier der Dozent zu sehen. Neben dessen Namen ist der „Details“ – Button zu finden, der den User zur Dozentenansicht führt (siehe 1.4 Dozentenansicht).

2.4 Dozentenansicht

Sollte es Informationen zu dem jeweiligen Dozenten geben, werden sie in folgender Ansicht angezeigt. Andernfalls ist die Ansicht leer. Die folgenden Felder werden jedoch nur angezeigt, wenn die Informationen zum Dozenten vorliegen.

Hier ist die Telefonnummer zu sehen. Durch Tippen auf diese wird man an die Telefon-App (mit bereits gewählter Nummer) weitergeleitet.

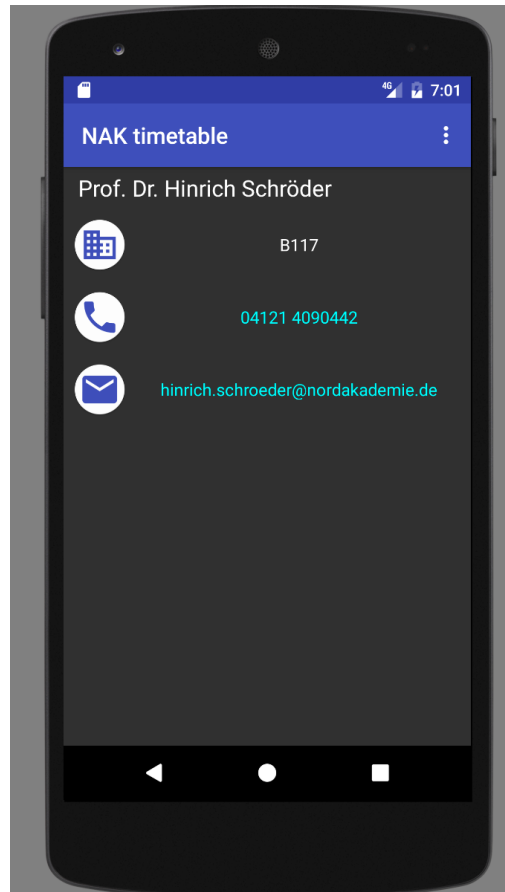


Abbildung 4: Dozentenansicht

Als letztes ist das Feld „E-Mail“ zu sehen. Hierbei kann durch Tippen auf diese, der bevorzugte E-Mail Client ausgewählt werden um dem Dozenten direkt eine Mail schreiben zu können.

Viel Spaß bei der Nutzung!

3 Matrix nicht funktionaler Anforderungen

Anforderung	Maßnahme
Datenqualität	Validierung der eingegebenen Daten durch Abfrage des eingegebenen Wertes. Der eingegebene Wert muss zwischen 1 und 53 liegen. Erweiterung auf die Erkennung von Schaltjahren ist möglich.
Usability	Bei Änderung des Wertes von „Sync calendar week“ wird ein Aufruf der EditTextPreference „Calendar Week“ ausgeführt. Wenn der Wert dieser ungleich der aktuellen Kalenderwoche ist, wird dieser Wert automatisch immer verändert. Man sieht somit die Kalenderwoche, in dem deaktivierten Feld.
Performance	Erst nach Abschluss eines Syncs (bzw. der letzten Einfügeoperation in die Datenbank) wird die Oberfläche aktualisiert. Dazu wird das Interface Updatable von den betroffenen Fragmenten implementiert. Die update()-Methode ist vergleichbar mit der onChange()-Methode, jedoch würde diese nach jedem Einfügen einer Zeile aufgerufen.
Usability	Das Design wurde für eine Tablet- und Landscape-Ansicht optimiert.
Usability	Anstatt von Texten werden in der Detailansicht Icons angezeigt.
Usability	Die Funktionalität des Zurück-Buttons wurde so angepasst, dass anstatt einer Activity zunächst zu dem vorigen Fragment gewechselt wird.
Usability	Die Texte in der App sind in Deutsch und in Englisch verfügbar. Weitere Sprachen können einfach hinzugefügt werden.

4 Selbstreflexionen

4.1 Daniel Bormann (14150)

Die Zusammenarbeit in meinem Team für die Hausarbeit hat meiner Meinung nach sehr gut geklappt. Durch einen ständigen Austausch und die hohe Verfügbarkeit aller Teammitglieder konnte eine stetige und effiziente Weiterentwicklung gewährleistet werden. Die klare Abgrenzung der Verantwortlichkeiten hat den individuellen Fortschritt gefördert. Wenn es Herausforderungen in den jeweiligen Bereichen gab, waren diese schnell behoben.

Im Vergleich zu PdSe war die Arbeit deutlich umfangreicher, jedoch führte der sehr ausführliche MOOC und die Android-Dokumentation zu einer guten Weiterentwicklung. Letztendlich denke ich, dass Team Hane eine gute Zusammenarbeit geleistet und das erwartete Ergebnis abgeliefert hat.

4.2 Leif Johannsen (13939)

Die Arbeit in Team Hane hat mir sehr gut gefallen! Auch wenn ich anfangs stark durch noch ausstehende Klausuren und den AKdWi-Vortrag eingeschränkt war, hat es meine Gruppe geschafft mich stetig auf dem aktuellen Stand zu halten. Besonders die aktive Nutzung von Trello als Kanban-Board hat die Arbeit erleichtert, da der Aufgabenpool stets noch offene Punkte bot.

Im Vergleich zu der PdSE-Hausarbeit lässt sich nur sagen, dass die Koordination dieses Projekts deutlich einfacher durchzuführen war. Dies zeigte sich besonders in der Terminabsprache. Hier musste nämlich auf Rahmenbedingungen, wie die jeweilige Ortszeit, keine Rücksicht genommen werden.

4.3 Arvid Ottenberg (14374)

Die Hausarbeit im Fach „Mobile Anwendungen“ fand ich sehr gut was Aufwand und Komplexität anging. Man konnte viele Dinge aus der „Sunshine App“ aus Udacity übernehmen, wobei bei jeder Komponente zwei bis drei kleinere bis größere Änderungen nötig waren. Somit konnte ich mein auf den Udacity Kurs beschränktes Wissen über Android deutlich ausbauen.

Mit Blick auf die „Praxis der Softwareentwicklung“ Hausarbeit hatte diese Arbeit zwar deutlich weniger Umfang, jedoch waren die einzelnen Aufgaben nach meinem Gefühl teils komplexer. Ich empfand es zudem sehr angenehm, dass alle Team Mitglieder hier an einem Standort waren, was die Koordination deutlich erleichtert hat. Ich fand die Stimmung im Team sehr positiv und denke, dass wir zusammen eine coole App, die sich sehr gut benutzen lässt entwickelt haben.

4.4 Timo Schlatter (14349)

Meiner Meinung nach war die Hausarbeit ein voller Erfolg, nicht zuletzt weil ich auf die Erfahrungen aus der Hausarbeit für Praxis der Softwareentwicklung zurückgreifen konnte. Dies betrifft hauptsächlich die Team-übergreifende Organisation, wie z.B. die Aufgabenverteilung, regelmäßige Rücksprachen, Nutzung von Trello für Fortschrittsübersicht und Ideensammelstelle. Die Tatsache, dass Android-Entwicklung komplexer ist, als bisherige von mir durchgeführte Projekte, hat zwar durchaus für Frust gesorgt, jedoch habe ich mich umso mehr gefreut, wenn meine Mühen mit Erfolg gekrönt wurden.

5 Quellen& Hilfsmittel

- Stackoverflow.com
- Udacity „Developing Android Apps“ inkl. zugehörigem Code
- Android Developer Guide
- Trello
- Skype

6 Eigenständigkeitserklärung

Hiermit bestätigen wir, dass wir die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt haben. Die Stellen der Arbeit die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quellen kenntlich gemacht.

Daniel Bormann, Elmshorn der 23.12.2016

Leif Johannsen, Elmshorn der 23.12.2016

Arvid Ottenberg, Elmshorn der 23.12.2016

Timo Schlatter, Elmshorn der 23.12.2016
