



Hochschule für Telekommunikation Leipzig
University of Applied Sciences

ENTWICKLUNG EINER HFTL-APP DOKUMENTATION

Studienmodul *Software-Engineering*
der Hochschule für Telekommunikation
Leipzig

Projektarbeit - Softwareentwicklung

vorgelegt von

BKMI Matrikel 13

23.04.2015

Dozent: Profn. Dr.-Ing. Sabine Wieland

INHALTSVERZEICHNIS

1	Einführung	4
1.1	Zweck	4
1.2	Hintergründe und Ziele des Projekts	4
1.3	Produktumfang	4
1.4	Musskriterien	4
1.5	Abgrenzungskriterien	5
1.5.1	Kostenrahmen	5
1.6	Definitionen, Akronyme, Abkürzungen	5
1.7	Referenzen	5
2	Allgemeine Übersicht	6
2.1	Beschreibung der Ausgangssituation (Ist-Zustand)	6
2.2	Produkteinsatz	6
2.2.1	Anwendungsbereiche	6
2.2.2	Zielgruppen, Qualifikationsniveau	6
2.3	Produktfunktionalität	6
2.4	Randbedingungen	6
2.5	Annahmen und Abhängigkeiten	7
2.6	Verzögerungen	7
3	Anwendungsszenarien	7
3.1	Beschreibung aus der Nutzersicht	7
4	Anforderungen	7
4.1	Fachkonzept	7
4.1.1	Überblick über das Gesamtsystem	7
4.1.2	Verwendete Bibliotheken von Drittanbietern	8
4.2	Anforderungen an die Datenhaltung	9
4.2.1	allgemeine Beschreibung der Daten	9
4.3	Anforderungen an die Benutzeroberfläche	9
4.3.1	allgemeine Anforderungen an die Oberfläche	9
4.3.2	Berechtigungen	9
4.3.3	Bildschirmlayout	9
4.4	Leistungsanforderungen	10
4.5	Anforderungen für Inbetriebnahme und Einsatz	10
4.5.1	Sicherheitsziele	10
4.5.2	Installationsprozedur	10
4.5.3	Pilot- bzw. Probetrieb	10
4.5.4	Fehlerreaktion, Garantie, Service, »Wiederanlauf«	10
4.5.5	Schulungen	10
4.6	Qualitätsanforderungen	10
4.6.1	Qualitätsmerkmale	10

4.6.2	Qualitätssicherung	11
4.6.3	Qualitätsnachweise	11
4.6.4	Offenlegung der Qualitätskontrollpläne	12
4.6.5	Berichte, Protokolle zum Nachweis des Vorgehens gemäß der Qualitätskontrollpläne	12
4.7	Anforderung an die Entwicklung	12
4.7.1	Entwicklungs-Umgebung	12
4.7.2	Projekt-Organisation	12
4.7.3	Projekt-Planung	14
4.7.4	Änderungsmanagement	14
4.7.5	Testanforderungen	14
5	Anhang	14
5.1	Benutzerhandbuch	16
5.1.1	Funktionsumfang	16
5.1.2	Startbildschirm	17
5.1.3	Newsansicht	18
5.1.4	Noten	19
5.1.5	Stundenplan	21
5.2	Entwicklerhandbuch	22
5.2.1	Allgemeines	24
5.2.2	Verwendete Software	25
5.2.3	Aufbau des Projekts	26
5.2.3.1	Manifest.XML	26
5.2.3.2	Ordnerstruktur	27
5.2.4	Activities	31
5.2.4.1	NewsActivity.java	31
5.2.4.2	EinstellungsActivity.java	32
5.2.5	NewsClickedActivity.java	33
5.2.5.1	Allgemein	33
5.2.5.2	Klasse DetailHelper	33
5.2.6	Fragmente	35
5.2.6.1	NewsFragment	35
5.2.6.2	Klasse NewsHelper	37
5.2.6.3	NavigationDrawerFragment	38
5.2.6.4	Notenfragment	40
5.2.6.5	StundenplanFragment	44
5.2.7	CustomAdapter	46
5.2.7.1	Allgemein	46
5.2.7.2	CustomAdapterNews.java	48
5.2.7.3	CustomAdapterNoten.java	48
5.2.7.4	CustomAdapterStundenplan	49
5.2.8	XML-Dateien	50
5.2.8.1	fragment.xml	50

5.2.8.2	_list.xml	50
5.2.8.3	activity_news_clicked.xml	50
5.2.8.4	colors.xml	50
5.2.8.5	settings_toolbar	51
5.2.8.6	impressum.xml & activity_impressum	51
5.2.8.7	array.xml	51
5.2.8.8	einstellung.xml	51
5.2.8.9	strings.xml	51
5.2.9	Layout	53
5.2.9.1	Allgemeines	53
5.2.9.2	Schriftarten	53
5.2.9.3	Buttons	55
5.2.10	Erweiterungen und Verbesserungen für kommende Versionen	57
5.2.10.1	Pflege der array.xml	57
5.2.10.2	Raumplan	57
5.2.10.3	Portierung auf andere Systeme	57
5.3	Gantt	58
5.4	App-Layout	63
5.5	Release-Historie	70
5.6	Testprotokollentwurf	78
5.7	Testprotokolle	81

1 EINFÜHRUNG

1.1 ZWECK

Dieses Dokument dient als Grundlage zur Beauftragung des berufsbegleitenden Studienganges, Kommunikations- und Medieninformatik des Matrikel 13, mit der Programmierung einer [HFTL-APP](#). Es setzt dabei die Rahmenbedingungen fest.

1.2 HINTERGRÜNDE UND ZIELE DES PROJEKTS

Die Hochschule für Kommunikation Leipzig ([HfTL](#)) ist eine private, staatlich anerkannte Fachhochschule. Träger der [HfTL](#) ist die [HfTL-Trägersgesellschaft mbH](#), eine Beteiligungsgesellschaft der Deutschen Telekom AG. Die Schule befindet sich im Leipziger Stadtteil Connewitz. Es werden sowohl Direkt- als auch duale Studiengänge und berufsbegleitende Studiengänge angeboten. Aufgrund des umständlichen Beschaffens der Noten und Stundenpläne sowie der Termine für Teletutorings, wird eine Smartphone-Applikation benötigt. Diese stellt ein Benutzerinterface für Studenten der [HfTL](#) dar, mit dem der Zugriff auf die im [QIS](#) hinterlegten Daten vereinfacht.

1.3 PRODUKTUMFANG

Der Betrieb der [HFTL-APP](#) muss auf allen gängigen Android-Smartphones ab Version 4.0 möglich sein.

Durch die APP wird den Studenten der HFTL ermöglicht:

- die aktuellsten Nachrichten der [HfTL](#)-Webseite lesen
- auf [QIS](#) zugreifen
- Studenten sollen ihre(-n) Noten(-spiegel) aufrufen können
- Vorlesungspläne zu lesen
- Raumbellegungspläne abzurufen

Zusätzlich werden folgende Anforderungen gestellt:

- Erweiterbarkeit für weitere Funktionen und Anwender
- spätere IOS-Version

1.4 MUSSKRITERIEN

Zunächst müssen zwingend folgende Punkte des Umfangs erfüllt werden:

- NEWS
- NOTEN
- STUNDENPLAN

1.5 ABGRENZUNGSKRITERIEN

Die APP soll später auch für zusätzliche Informationen, wie ein Raumplanung erweiterbar sein. Eine spätere Version für IOS-Geräte ist ebenfalls geplant, ist hier aber nicht Bestandteil.

1.5.1 KOSTENRAHMEN

Für die Entwicklung der APP soll auf kostenfreie Opensource-Programme oder auf vordefinierte Klassen der Programmierung zurückgegriffen werden. Außer den personellen Aufwand dürfen keine Zusätzlichen Kosten entstehen.

1.6 DEFINITIONEN, AKRONYME, ABKÜRZUNGEN

HfTL	Hochschule für Kommunikation Leipzig
APP	Kurzform für Applikation
mbH	mit beschränkter Haftung
QIS	Qualitätssteigerung der Hochschulverwaltung im Internet durch Selbstbedienung
iCal	Datenformat zum Austausch von Kalenderinhalten
SoSe15	Sommersemester 2015
XML	Extensible Markup Language
HTTPS	HyperText Transfer Protocol Secure
AES	Advanced Encryption Standard
SQL	Structured Query Language
.apk	Android application package
MTBF	mean time between failure
CI/CD	Corporate Identity/Corporate Design

1.7 REFERENZEN

- QIS-System: <https://qisweb.hispro.de/tel/rds?state=user&type=0>
- News der HfTL <https://www.hft-leipzig.de/de/studierende/service/news.html>

2 ALLGEMEINE ÜBERSICHT

2.1 BESCHREIBUNG DER AUSGANGSSITUATION (IST-ZUSTAND)

Damit Studenten auf [QIS](#) zugreifen kann, müssen diese sich über einen Browser auf der [QIS](#)-Seite einloggen und über das unübersichtliche Menü ihre Daten suchen. Studenten haben die Möglichkeit ihre Noten oder einen Notenspiegel einzusehen. Für iPhone-Nutzer gibt es bereits eine kostenpflichtige [APP](#), namens Grades, die Noten und Informationen über die angemeldete Prüfungen auslesen kann. Um die auf [QIS](#) hinterlegten Vorlesungspläne abzurufen ist kein Login erforderlich, jedoch ist es notwendig sich umständlich zu dem entsprechenden Studiengang zu navigieren. Einzelne Vorlesungen oder Stundenpläne können als [iCal](#) heruntergeladen werden. Ebenso sind die Teletutorien hier zu finden. Die aktuellen Nachrichten der Hochschul-Homepage sind auf einer anderen Seite zu finden, welche öffentlich zugänglich ist.

2.2 PRODUKTEINSATZ

2.2.1 ANWENDUNGSBEREICHE

Aktuell soll die APP nur für die Studenten der [HfTL](#) zugänglich sein, welche ein Android-Smartphone besitzen.

2.2.2 ZIELGRUPPEN, QUALIFIKATIONSNIVEAU

Da bei der Nutzergruppe von Studenten mit Erfahrung im Umgang mit solchen APP's ausgegangen werden kann, wird auch die Oberfläche dementsprechend gestaltet.

2.3 PRODUKTFUNKTIONALITÄT

Die APP soll sich mittels regelmäßiger Abfragen der [HfTL](#)-Homepage, sowie von [QIS](#), die News, Noten und Stunden- und ggf. Raumbelegungspläne ziehen und die Noten für den Nutzer lokal auf dem Smartphone speichern und entsprechend darstellen. Es soll sichergestellt werden das auf sensible Daten wie z.B. Noten auch nur autorisierte Nutzer Zugang bekommen. Die APP soll in deutscher Sprache dargestellt werden. Evtl. wird sie im Nachgang in andere Sprachen übersetzt.

2.4 RANDBEDINGUNGEN

Der zeitliche Rahmen für die Entwicklung und Programmierung dieser APP begrenzt sich auf das Sommersemester 2015 ([SoSe15](#)) und dem Studienmodul Software-Engineering. Für die Entwicklung dieser APP, sowie Vertrieb, Programmierung usw. dürfen keine Kosten entstehen. Supportleistungen werden in der Projektphase über das Projektteam geleistet. Die Wartung wird, bis mit der Veröffentlichung die APP der Hochschule kostenfrei zur Verfügung gestellt wird, ebenfalls vom Projektteam übernommen. Die Dokumentation wird ebenfalls vollständig an die Hochschule übergeben. Durch das Projektteam wird es keine

weitere Softwarebetreuung, Wartung oder der gleichen geben. Es finden ebenfalls keine Schulungen oder Einweisungen statt.

2.5 ANNAHMEN UND ABHÄNGIGKEITEN

Die APP wird für Android-Geräte ab Version 4.0.3 zur Verfügung stehen. Entsprechend der Vorgaben der Deutschen Telekom AG muss bei der Programmierung der APP, explizit beim Design, auf die Konzernrichtlinien geachtet werden. Es soll zusätzlich auf die Designempfehlungen für Androidgeräte geachtet werden.

2.6 VERZÖGERUNGEN

Durch die strikte Abtrennung des zeitlichen Rahmens auf das [SoSe15](#) darf es über diesen Zeitraum hinaus nicht zu Verzögerungen kommen

3 ANWENDUNGSSZENARIEN

3.1 BESCHREIBUNG AUS DER NUTZERSICHT

Die Benutzeroberfläche muss intuitiv bedienbar sein. Der strukturierte Aufbau durch Kategorien (News, Noten, Stundenplan) soll die Übersichtlichkeit erhöhen. Die Logindaten werden verschlüsselt auf dem Smartphone gespeichert und auch verschlüsselt übertragen. Durch eine durchgehende und vollständige Dokumentation soll eine Wartung auch durch spätere Matrikel oder Administratoren der Hochschule möglich sein. Eine Implementierung weiterer Funktionen soll auch im Nachhinein möglich sein.

4 ANFORDERUNGEN

4.1 FACHKONZEPT

Die [HFTL](#)-APP wird in Java programmiert um durch Verwendung bestehender Klassen die Erweiterbarkeit und Realisierbarkeit zu vereinfachen. Für das Design werden [XML](#)-Stylesheets verwendet. Pull-und Push-Service werden zur Benachrichtigung und Abfrage verwendet.

4.1.1 ÜBERBLICK ÜBER DAS GESAMTSYSTEM

Noten

- die APP soll die Noten lokal auf dem Smartphone nach Semester aufgeschlüsselt anzeigen

- Login über die APP
- Anmeldung über gesicherte, verschlüsselte Übertragung ([HTTPS](#))
- verschlüsselte Speicherung der Daten auf dem Smartphone ([AES](#))
- verschlüsselte Speicherung der Daten auf dem Smartphone ([AES](#))
- Pull-Nachrichten (Einstellbares Intervall und/oder manuell)
- Push-Benachrichtigung
- Nutzer wird mit Hinweismeldung informiert, wenn Noten aktualisiert wurden

Optionale Angaben im Menüpunkt Noten:

- Klassenspiegel
- Notenverteilung
- Anzahl der Teilnehmer
- Notenschnitt
- Anzeige der Creditpoints
- zu erreichende Creditpoints
- erreichte Creditpoints

Stundenpläne

- Stundenplan nur für zum Nutzer passenden Studiengang
- Pull-Nachrichten (Einstellbares Intervall und/oder manuelle Abfrage)
- Synchronisierung mit dem Kalender auf dem Smartphone

News

- Pull-Nachrichten (manuelle Aktualisierung)
- News von: <https://www.hft-leipzig.de/de/studierende/service/news.html>

4.1.2 VERWENDETE BIBLIOTHEKEN VON DRITTANBIETEREN

- jsoup 1.8.1 – MIT-Lizenz
- iCal4j 1.0.6 – BSD-Lizenz

4.2 ANFORDERUNGEN AN DIE DATENHALTUNG

Alle gespeicherten Daten müssen vor unbefugtem Zugriff geschützt werden, dafür werden Benutzerdaten in einer [SQL](#)-Datenbank gespeichert und durch das Passwort und den Benutzernamen freigegeben. Passwort und Benutzername wurden mit [AES](#) verschlüsselt.

4.2.1 ALLGEMEINE BESCHREIBUNG DER DATEN

Sicherheitsrelevante Daten sind in dem Notenteil der App zu finden. Diese sind der Benutzername, das Passwort, die Prüfungs- und Prüfungsvorleistungsergebnisse, der Klassenspiegel und die Creditpoints. Die öffentlichen Daten sind in den anderen beiden Teilen der App enthalten. Dazu gehören die News mit ihren Terminen und Inhalt und der Stundenplan mit den Veranstaltungen und den dazugehörenden Informationen wie Dozent und Veranstaltungsort.

4.3 ANFORDERUNGEN AN DIE BENUTZEROBERFLÄCHE

4.3.1 ALLGEMEINE ANFORDERUNGEN AN DIE OBERFLÄCHE

Das Layout richtet sich nach dem Corporate Identity/Corporate Design ([CI/CD](#)) der Deutschen Telekom AG, speziell dem der [HFTL](#) Trägersgesellschaft, stand 14.12.12. Entsprechend sind primär die Farben, sowie Schriftarten vorgegeben. Das Layout zieht sich einheitlich (mit funktionsbedingten Abweichungen) durch alle APP-Teile und soll eine leichte Bedienung begünstigen. Die Größe der einzelnen Elemente (Buttons, Zeilen, Überschriften etc.) ist an den Designempfehlungen seitens Android angelehnt. Auch die Abstände zwischen den einzelnen Elementen wurden – sofern sie mit dem [CI/CD](#) in Einklang stehen – entsprechend der Designempfehlung festgelegt. Die Reaktionszeit der Benutzereingaben soll mit möglich wenig Verzögerung verarbeitet und dargestellt werden.

4.3.2 BERECHTIGUNGEN

Die APP wird als Open Source bereitgestellt.

Nicht angemeldete Benutzer können sich die News und den Stundenplan anzeigen lassen.

Angemeldete Benutzer können zusätzlich ihre Noten abrufen.

Die App braucht folgende Berechtigung auf dem Smartphone:

- Netzwerkstatus abrufen: zur Kontrolle ob das Handy online ist
- Internet: APP greift auf das Internet zu
- Telefonstatus: wird zur Verschlüsselung benötigt

4.3.3 BILDSCHIRMLAYOUT

siehe Anhang [5.4](#)

4.4 LEISTUNGSANFORDERUNGEN

Es wird von einer maximalen Nutzerzahl von 1000 Studenten ausgegangen. Die Reaktionszeit/Programmstart der APP soll möglichst gering gehalten werden. Das Datenaufkommen soll möglichst gering ausfallen.

4.5 ANFORDEUNGEN FÜR INBETRIEBNAHME UND EINSATZ

4.5.1 SICHERHEITSZIELE

Der Benutzername und das Passwort werden in einen gesicherten Speicherbereich (nur root), mit [AES](#) verschlüsselt, gespeichert. Die Datenbank in der die Noten gespeichert werden, liegt ebenfalls in diesem Bereich und darf nur von der [HfTL](#)-App geöffnet werden. Die Kommunikation mit der [HfTL](#)-Website und [QIS](#) erfolgt gesichert per [HTTPS](#).

4.5.2 INSTALLATIONSPROZEDUR

Der Wirkbetrieb soll nach Möglichkeit über den Android-Playstore realisiert werden. Falls es dabei zu Lizenz- und Kostenproblemen kommt, wird die APP als Android application package ([.apk](#)) über die HfTL-Homepage verteilt.

4.5.3 PILOT- BZW. PROBE BETRIEB

Hier müssen die Versionsmanagementprotokolle rein!

4.5.4 FEHLERREAKTION, GARANTIE, SERVICE, »WIEDERANLAUF«

Die Zuverlässigkeit sollte sich durch eine große mean time between failure ([MTBF](#)) darstellen.

4.5.5 SCHULUNGEN

Nach Beendigung des Projektes werden keine Schulungen usw. durchgeführt.

4.6 QUALITÄTSANFORDERUNGEN

4.6.1 QUALITÄTSMERKMALE

Folgende Qualitätsansprüche werden gestellt:

- Hohe Zuverlässigkeit der Software
- schnelle und zuverlässige Verarbeitung der gewünschten Daten
- sichere Datenspeicherung u. -übertragung mittels entsprechender Verschlüsselung
- Fehler werden mit einer entsprechenden Fehlermeldung beantwortet

- Intuitiv benutzbar
- Leicht zu warten und zu erweitern
- Vollständige Dokumentation des Projektes

4.6.2 QUALITÄTSSICHERUNG

Zur Qualitätssicherung werden einheitliche Testprotokolle und die entsprechenden Testkriterien erstellt. Es finden regelmäßige Kontrollen durch die Qualitätssicherung statt, um die Einhaltung der gegebenen Standards zu überprüfen. Es findet ebenfalls eine regelmäßige Kontrolle der Dokumentation statt. Die jeweiligen Software-Prototypen werden nach der Prototypisierung zur Fehlererkennung und zum Funktionstest an das Projektteam geschickt. Zur Verbesserung der Software werden die Tests anhand vorgegebener Testkriterien durchgeführt, um standardisierte Testergebnisse zu erhalten. Die Testergebnisse werden zur Auswertung an die entsprechende Projektgruppen zurück gespiegelt.

4.6.3 QUALITÄTSNACHWEISE

Während des Projektzeitraumes werden sämtliche Meeting, Tests und Kontrollen protokolliert und zur Dokumentation in einem dafür vorgesehen Bereich abgelegt. Die gesetzten Standards und Vorgaben werden eingehalten. Zum Projektabschluss ist der geplante Umfang erreicht und alle Muss-Kriterien sind enthalten. Die APP ist funktionstüchtig.

4.6.4 OFFENLEGUNG DER QUALITÄTSKONTROLLPLÄNE

Nr.	Anforderung	Verantwortliche
1	Projektrahmen festlegen	gesamtes Team
1.1	Projektrahmen festlegen	gesamtes Team
1.2	Architektur	gesamtes Team
1.3	Vorgehensmodell	gesamtes Team
2	Vollständige Dokumentation	gesamtes Team
2.1	UML Diagramme erstellen	AD, CM
2.2	Installationsanleitung und Benutzerhandbuch erstellen	GE, JS
2.3	Lasten- Pflichtenheft erstellen	SK, SC
2.4	Protokolle	ML, PK
2.5	Dokumentation zusammenstellen und gestalten	JS, SC
3	App erstellen	gesamtes Team
3.1	Design nach HfTL-Vorgaben erstellen	ML, SC
3.2	Programmierung der APP	GE
4	Qualitätskontrolle	gesamtes Team
4.1	Testkriterien festlegen	PK
4.2	Testprotokollelayout erstellen	ML
4.3	Softwaretests	ML, PK
4.4	Kontrolle der Dokumentation	PK
5	Anfertigen der Präsentation	gesamte Team
5.1	Präsentation halten	SC, ML, SK, JS

4.6.5 BERICHTE, PROTOKOLLE ZUM NACHWEIS DES VORGEHENS GEMÄSS DER QUALITÄTSKONTROLLPLÄNE

Hier sollen Sitzungsprotokolle und Testprotokolle rein, schonmal angefangen was einzufügen
 siehe Anhang Testprotokolle"5.7

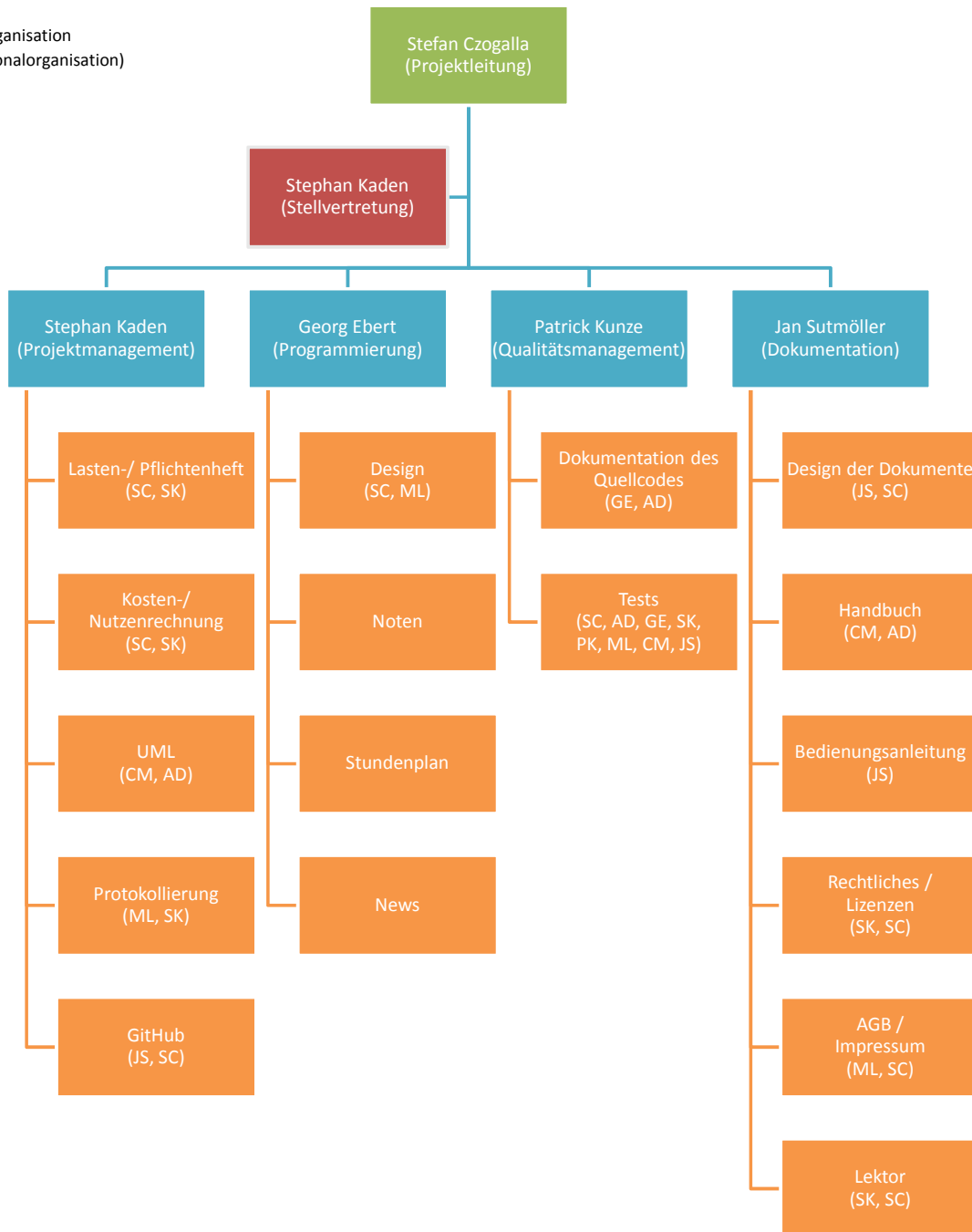
4.7 ANFORDERUNG AN DIE ENTWICKLUNG

4.7.1 ENTWICKLUNGS-UMGEBUNG

Für die Entwicklung wird Android Studio inkl. Gradle in der Version 1.x genutzt. Für die Dokumentation und Projektkoordination wird GitHub verwendet. Die Dokumentation wird mittels \LaTeX erstellt.

4.7.2 PROJEKT-ORGANISATION

Projektorganisation
(& Funktionalorganisation)



4.7.3 PROJEKT-PLANUNG

Verweis auf Projektstrukturplan im Anhang

4.7.4 ÄNDERUNGSMANAGEMENT

siehe Anhang 5.5

4.7.5 TESTANFORDERUNGEN

siehe Anhang 5.6

5 ANHANG

- Dialogmasken
- Dokumente
- Liste der Softwarelieferungen
- Projektorganigramm
- Projektstrukturplan
- Haupt-Terminaten



Benutzerhandbuch - HfTL-APP -

5.1 BENUTZERHANDBUCH

5.1.1 FUNKTIONSUMFANG

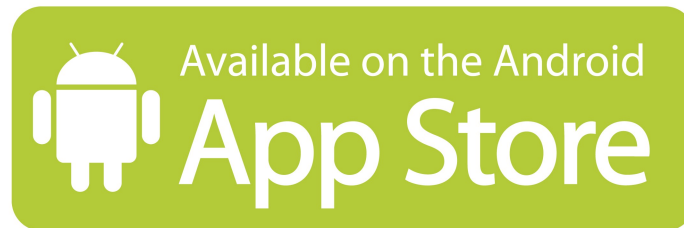
In diesem Dokument wird die Benutzerfunktionen von der HfTL-APP für Android-Geräte beschrieben. Es dient als Benutzerhandbuch für die unterschiedlichen Funktionen der Anwendung und soll Ihnen beim Ausführen von häufigen Aktionen innerhalb der Anwendung Hilfe bieten. Die Installation und Konfiguration von der HfTL-APP wird in einem separaten Dokument behandelt.

Die HfTL-APP ist eine mobile Informationslösung für Android Geräte. Die App kann kostenlos über das Rechenzentrum der Hochschule für Kommunikation-Leipzig bezogen werden.

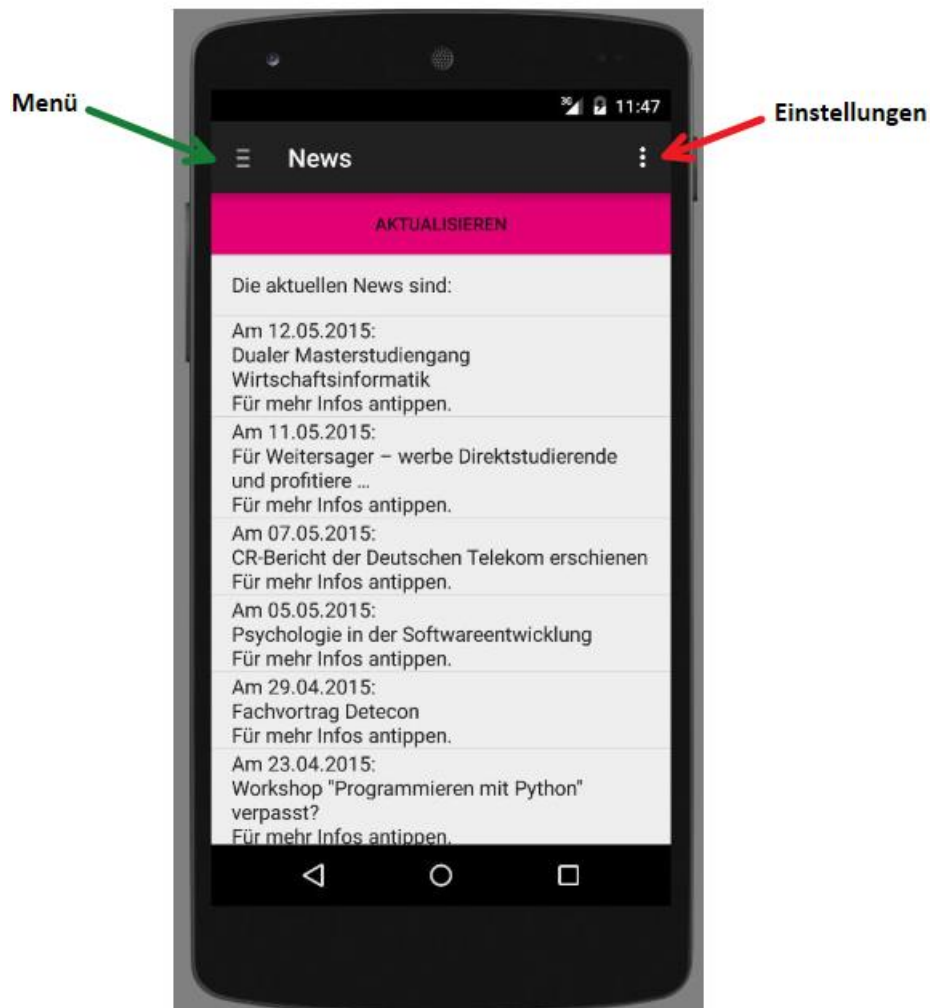
Die HfTL-APP bietet folgende Funktionen:

- Abfrage der News von der HfTL-Homepage
- Abfrage der Noten aus QIS/HIS nach erfolgreicher Anmeldung an dem betreffenden System
- Abfrage des zu einem Studenten passenden Stundenplans

HINWEIS: In diesem Dokument wird vorausgesetzt, dass die HfTL-APP bereits installiert ist.



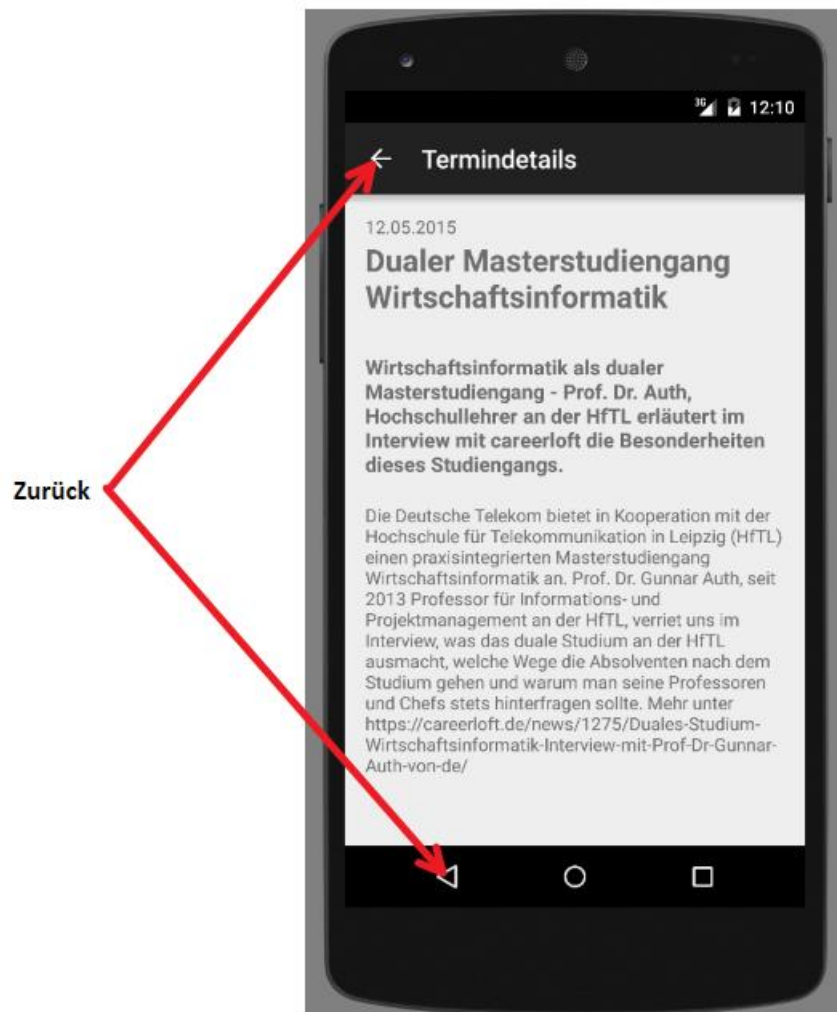
5.1.2 STARTBILDSCHIRM



Nach Starten der APP erscheint zunächst die News-Seite. Die News werden bei bestehender Internetverbindung automatisch aktualisiert. Mit klicken auf den Aktualisierungsbutton kann eine manuelle Aktualisierung durch den Nutzer angestoßen werden.

Über den Menü-Button gelangt der Nutzer in das Programm-Menü. Über den Einstellungs-Button gelangt man in das Einstellungs-Menü. Mit Auswählen der einzelnen News gelangt man in deren Detailansicht.

5.1.3 NEWSANSICHT



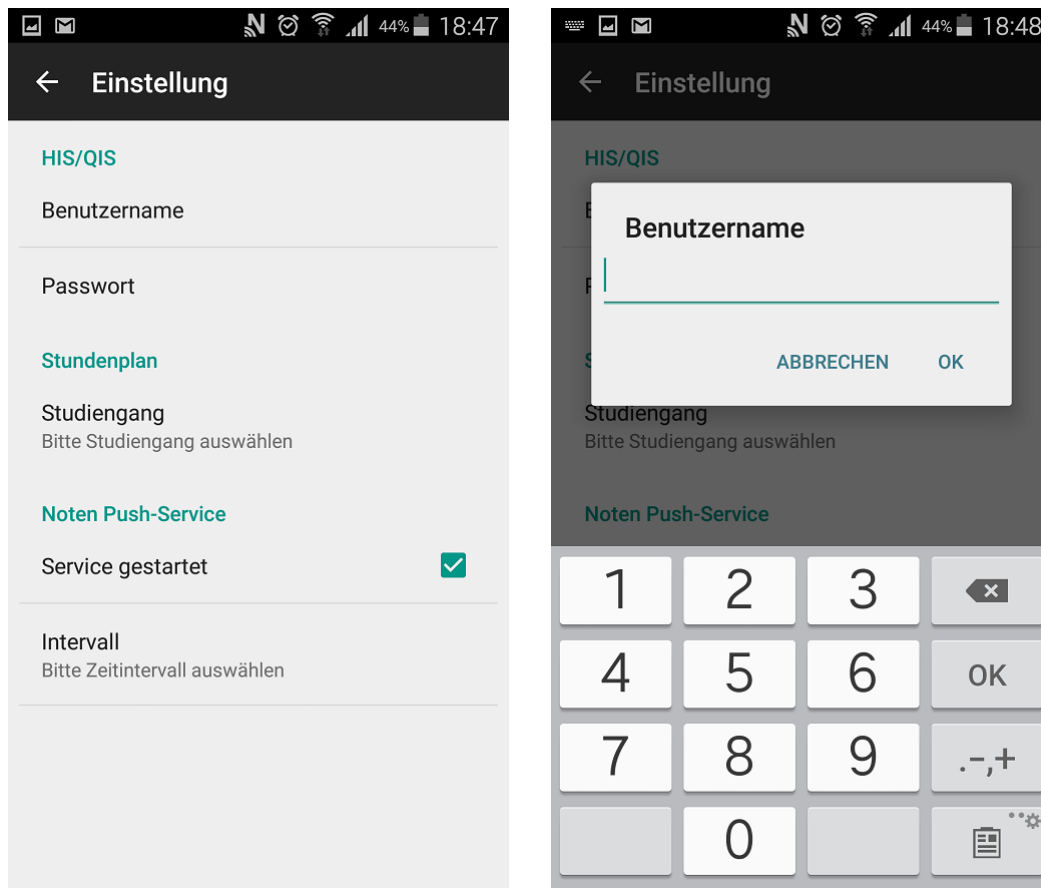
Mit den Zurück-Buttons gelangt man in die vorherige Ansicht.

5.1.4 NOTEN

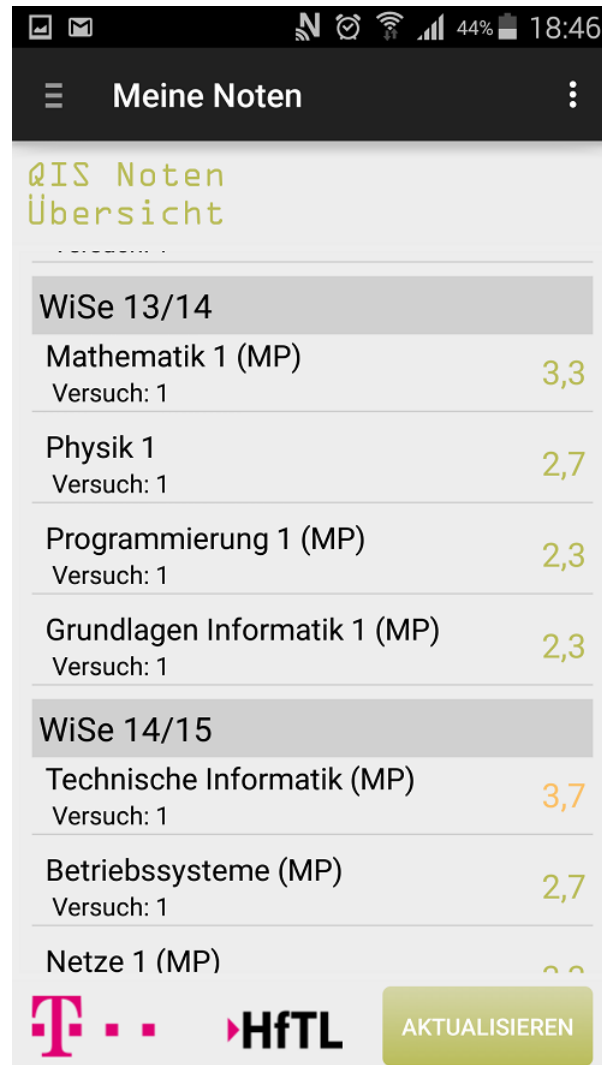
Um die Noten abzurufen geht man zunächst in den Einstellungskontext. Dort kann der Nutzer und das jeweilige Passwort eingegeben werden.

Mit einem Klick auf Benutzername bzw. Passwort öffnet sich ein neuer Kontext welcher zum eingeben des Benutzernamens bzw. Passwortes auffordert.

Die Eingabe wird mit "OK" gespeichert und mit "Abbrechen" verworfen. Bei beiden Aktionen schließt sich der Kontext.



Nach erfolgreichem Eintragen des Benutzeraccounts und verlassen der Einstellungen kann über den Menü-Button der Punkt Noten ausgewählt werden. Hier werden die aktuellen Noten aus QIS/HIS geladen und angezeigt. Sollte es beim Anmelden an QIS/HIS zu einem Fehler kommen, erscheint eine Fehlermeldung und die vorher vorgenommenen Einstellungen sollten noch einmal kontrolliert werden.



5.1.5 STUNDENPLAN

Den Stundenplan erreicht man über das Menü. Hat man in den Einstellungen vorher seinen Studiengang und das Matrikeljahr angegeben, erscheint dazu passende Stundenplan.



5.2 ENTWICKLERHANDBUCH



Entwicklerhandbuch - HfTL-APP -

Stand: 30. August 2015

INHALTSVERZEICHNIS

5.2.1 ALLGEMEINES

Dieses Dokument dient lediglich als Hilfswerkzeug zur (Weiter-)Entwicklung und Wartung der HfTL-App. In diesem Dokument ist der grobe Aufbau, sowie die wichtigsten verwendeten Funktion mitsamt zugehörigen Quelltext erklärt.

Dieses Dokument ist keine Programmieranleitung.

Es empfiehlt sich, gewisse Grundkenntnisse in objektorientierter Programmierung im Allgemeinen und in JAVA, XML und AndroidStudio im Speziellen mitzubringen, um dieses Dokument effizient nutzen zu können.

Standard-Methoden und Klassen sind nicht im Detail erklärt, da das den Rahmen dieses Dokuments sprengen würde. Für tiefer greifende Informationen wird die [Android-API](#) empfohlen.

5.2.2 VERWENDETE SOFTWARE

AndroidStudio

AndroidStudio ist die Standard-Entwicklungsumgebung für Android. Es bietet bereits ein fertiges Gerüst für eine funktionsfähige App an. Das Programm bietet Klassenbibliotheken, Debugger und selbst ein Emulator mit dessen Hilfe Android-Endgeräte auf dem PC virtuell dargestellt werden können.

Das Programm kann kostenlos unter developer.android.com heruntergeladen werden.

GitHub

GitHub ist ein onlinebasierter Dienst, der es einem ermöglicht, Dateien zu hosten und parallel im Team zu bearbeiten. Zudem bietet GitHub eine einfache und übersichtliche Form der Versionierung.

Durch des Erstellen unterschiedlicher Repositories kann an mehreren Stellen eines Projekts gleichzeitig gearbeitet und getestet werden, ohne Datenverluste befürchten zu müssen.

Gimp

Gimp ist ein Open-Source Bildbearbeitungsprogramm. Es wurde in dem konkreten Fall genutzt, um Grafiken für die App zu erstellen.

Das Programm kann kostenlos auf der Seite des Herstellers (www.gimp.org) herunter geladen werden.

Microsoft Office

MS Office ist das Office Paket von Microsoft, welches bei der Entwicklung der App nur eine beiläufige Rolle spielte. Genutzt wurde insbesondere MS Word zur schnellen Erstellung von Fließtext bzw. der Dokumentation. Die Eigentliche Aufbereitung des Textes erfolgte dann in LaTeX.

Die Erstellung der Organigramme bzw. Pläne erfolgte durch MS Excel.

LateX

ist ein Softwarepaket, das die Benutzung des Textsatzsystems TeX mit Hilfe von Makros vereinfacht. Die Projektdokumentation wurde mit TeX entwickelt

5.2.3 AUFBAU DES PROJEKTS

5.2.3.1 Manifest.XML

Diese Datei ist für jede Android-App zwingend notwendig. Hier werden grundsätzliche Dinge definiert, z.B. Welche Berechtigung diese App benötigt und auf welche Hardware im laufenden Zustand zugegriffen werden muss.

Des Weiteren wird hier auch das package für den Javaquellcode definiert:

Listing 1: AndroidManifest.XML

```
<manifest xmlns:android="..." package="bkmi.de.hftl_app" >
```

Die Zugriffsberechtigungen sind wie folgt definiert:

Listing 2: AndroidManifest.XML

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

Bei der Installation der App wird der Nutzer entsprechend informiert, dass die App Zugriff auf die jeweiligen Funktionen des Endgerätes zugreift.

In der manifest.xml ist ebenfalls eine Übersicht über die verwendeten Verzeichnisse und Komponenten hinterlegt, z.B. für die Activities.

Bei der HfTL-App ist der Verweis für die MainActivity (die Activity, mit der die App startet) für die NewsActivity gesetzt.

Die folgenden Tags wurden bei der App nicht verwendet, sind aber theoretisch für Erweiterungen möglich:

Listing 3: Zugriffsbeispiel

```
<uses-feature android:name="android.hardware.camera" />
```

Das wäre ein Tag, um im laufenden Betrieb auf die Kamera des Telefons zuzugreifen.

5.2.3.2 Ordnerstruktur

Database

- beinhaltet die NotenDB.java – Inhalt ist der Connector und die Kernfunktionen um Inhalte der eigentlichen Datenbank zu aktualisieren und zu modifizieren
- NotenTabelle.java – das ist die eigentliche Datenbank, bzw. die eigentliche Definition vom Aufbau der Datenbank

Fragmente

- beinhaltet die Fragmente, die von den Activities verwendet werden.

Service

- beinhaltet die Datei NotenService.java, die zum Erzeugen von Push-Nachrichten dient, sobald es in der Notenübersicht neue Noten für den jeweiligen Studenten gibt.

Help

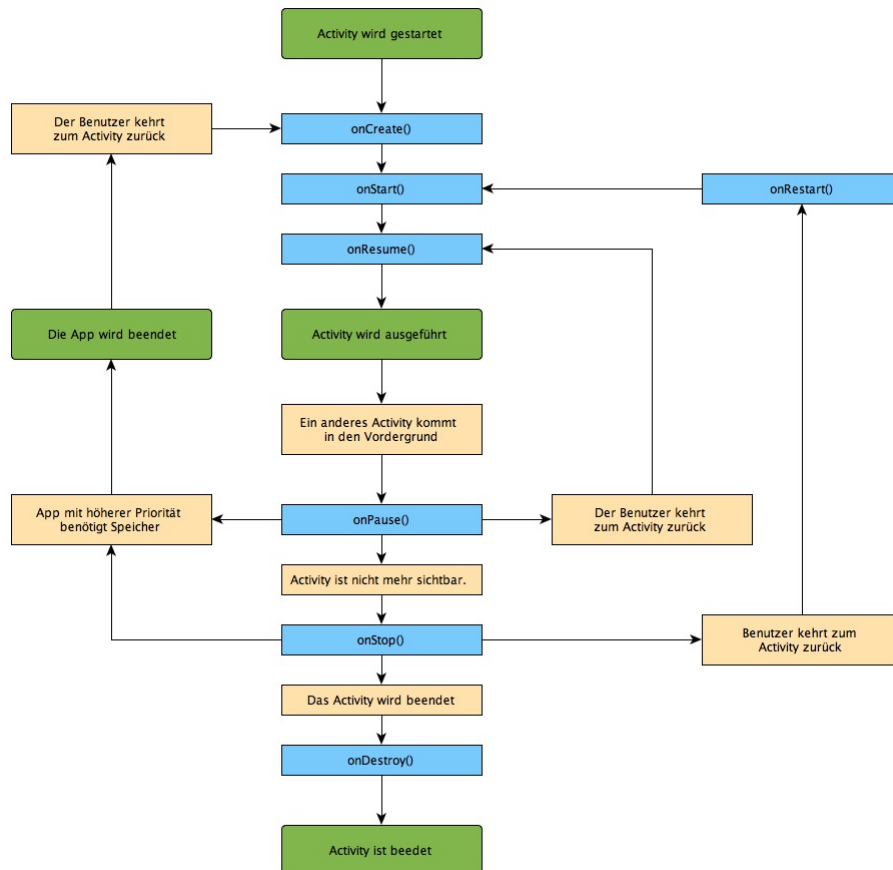
- beinhaltet diverse Hilfsfunktionen, die unter anderem zum Ausführen von Threads dienen.

Ressourcen

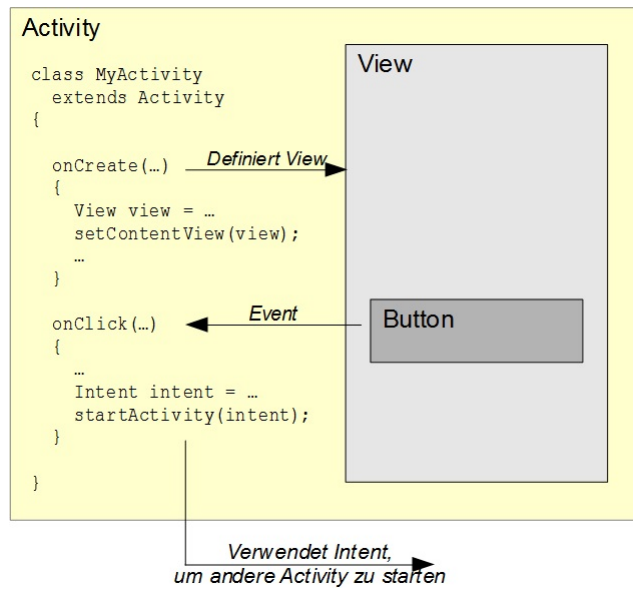
- unter /src/main/res befindet sich eine Ordnerstruktur, welche XML- und Bild-dateien für verschiedenste Anwendungszwecke beinhaltet. Diese werden beim Kompilieren des Projekts in die Ressourcendatei R.java geschrieben. Über diese Datei, wird dann auf die Ressourcen zugegriffen.
Es wird im Weiteren nicht detailliert auf alle Dateien eingegangen. Es wird nur auf jene Dateien eingegangen, deren Inhalt zur Schilderung von wichtigen Kernfunktionen dienlich ist.

Activities

- Eine Activity stellt ein sichtbares Fenster dar, welches die eigentliche Interaktion mit dem Nutzer ermöglicht. Diese Interaktionen werden als Events bezeichnet bzw. behandelt. Durch sogenannte Intents ist es möglich andere Activitys zu starten. Der sichtbare Teil der Activity wird durch eine View definiert. Dort befinden sich dann die Eventauslöser, wie zum Beispiel **Buttons** etc.
- Folgende Grafik dient zur Übersicht:

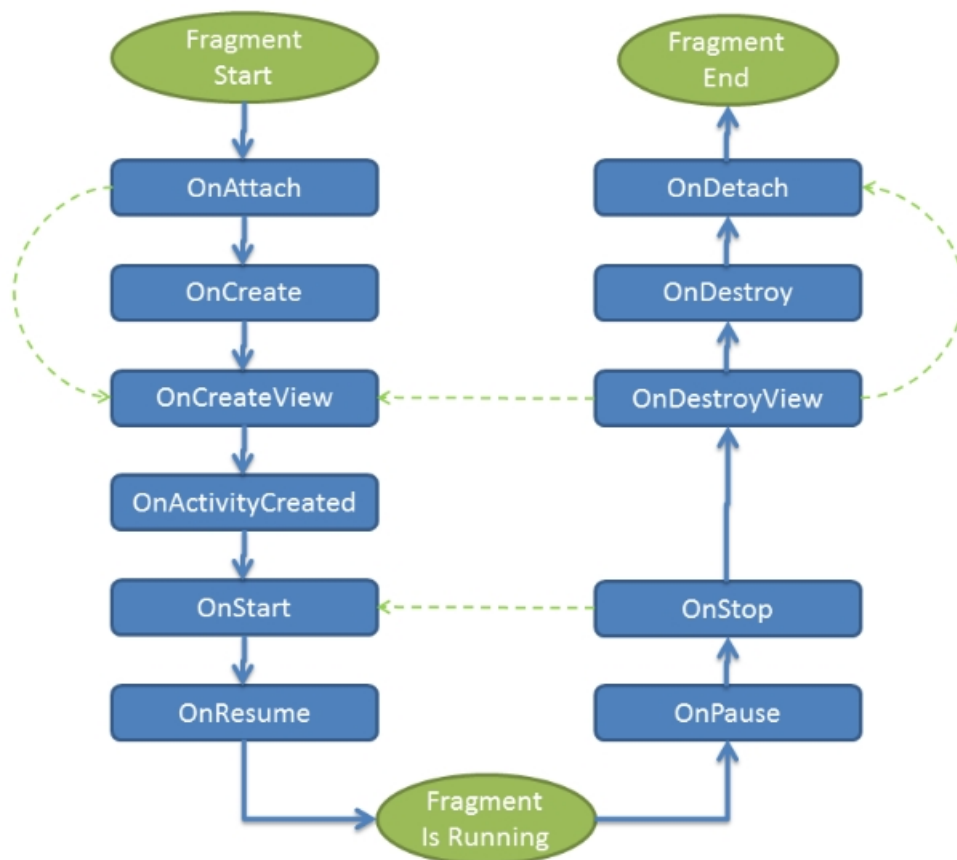


- Jede Activity hat einen sogenannten “Lifecycle“ (=Lebenszyklus), der von dem Betriebssystem gesteuert und verwaltet wird. Dem Entwickler ist es nun überlassen, welche Ressourcen er entsprechend sichert, sollten die Ressourcen auf dem Mobiltelefon knapp werden. Das passiert in den Events:
 - `onPause()`: die App ist noch sichtbar, aber nicht mehr im Vordergrund
 - `onStop()`: die App ist nicht mehr sichtbar
- Lifecycle einer Activity:



Fragmente

- In Android ist es nicht möglich, mehrere Activities gleichzeitig anzuzeigen, wenn es der Platz auf dem Display theoretisch erlauben würde. Um das dem Entwickler dennoch zu ermöglichen, bietet Android den Objekttypen "Fragment" an.
- Fragmente sind UI-Container, die über laufende Activities gelegt werden. Dies macht besonders dann Sinn, wenn es um die Fragmentierung der App geht (verschiedene Displaygrößen) bzw. wenn man zwischen den Ansichten wechselt (Haltung des Telefons)
- Ebenso wie eine Activity hat ein Fragment ein Lifecycle, welcher in direkter Kommunikation mit dem Lifecycle der Activity steht. So werden zum Beispiel alle Fragmente einer Activity pausiert, wenn die Activity selbst pausiert wird. Folgendes Diagramm soll das veranschaulichen:



5.2.4 ACTIVITIES

5.2.4.1 NewsActivity.java

- Klasse wird mit Fragment extended
- [NavigationDrawerFragment](#) wird benötigt, um die Interaktion mit Fragmenten der Anzeige zu ermöglichen
- Intent wird benötigt, um andere Activities neben der NewsActivity zu handeln.
- mTitle ist der Titel des zuletzt geladenen Screens

onCreate()

- Hier wird das Layout geladen, welches in der zugehörigen xml definiert wurde
- Laden des mNavigationDrawerFragment, zum Abbilden des Menüs am linken Seitenrand
- Laden des Layouts für das Fragment

Durch das Laden des Layouts für das mNavigationDrawerFragment (durch setUp) wird die Funktion onNavigationDrawerItemSelected gerufen, weil in der Layoutdefinition Item 1 der Listview ausgewählt wird.

onNavigationDrawerItemSelected()

- Durch das Aufrufen der Funktion wird das eigentliche Fragment für den FragmentManager ausgewählt (durch die switch-case-Anweisung) und abschließend durch das commit aktiv geladen.

onSectionAttached()

- Je nach Auswahl wird hier mTitle aktualisiert.

restoreActionBar()

- Dient zum Aktualisieren der Titelleiste (durch setTitle und mTitle als Argument)

onCreateOptionsMenu()

- Erstellt das Menü oben rechts (drei Punkte) und befüllt es mit den Daten aus der /res/menu/news.xml

onOptionsItemSelected()

- Überprüft welches Element aus dem Menü ausgewählt wurde

5.2.4.2 EinstellungsActivity.java

onCreate()

- Einbinden der "einstellung.xml" (beinhaltet Definitionen für Stringvariablen, Listen und Checkboxes)
- Setzen von Startwerten für shared, check und list
- Ausführen von registerPreferenceListener()

registerPreferenceListener()

- Es wird ein anonymer Listener erstellt, der auf Änderungen in der SharedPreferences.xml reagiert
- In der Methode onSharedPreferenceChanged(SharedPreferences prefs, String key) werden die Änderungen abgefangen
- Listener wird am SharedPreferences Objekt registriert

testeBenutzerdaten()

- Funktion zum Überprüfen von Anmeldedaten
- via TextSecure ts wird Ver- und Entschlüsselung gewährleistet

keineBenutzerdaten()

- Funktion zum Ausgeben, dass die Anmeldeinformationen falsch eingegeben wurden

5.2.5 NEWSCLICKEDACTIVITY.JAVA

5.2.5.1 Allgemein

Diese Activity wird geladen, sobald in der News-Übersicht ([NewsActivity.java](#)) ein Eintrag geöffnet wird.

Der Inhalt wird durch NewsResolver und dessen Funktion getDetailsStringArray in das String-Array "s" geladen.

onCreate()

- Laden des in der zugehörigen xml definierten Layouts (activity_news_clicked.xml)
- Einbinden der "Extras" (Übergebene Variablen)
- Zuweisung der URL aus dem rufenden NewsFragment
- Aufruf des DetailHelpers durch

```
new Detailhelper().execute();
```

5.2.5.2 Klasse DetailHelper

onPreExecute()

- Anzeige eines ProgressDialogs (Ladebalken mit Hinweistext), um den Anwender zu informieren.

```
ladebalken = ProgressDialog.show(NewsClickedActivity.this, "
    Bitte warten", "Nachricht wird geladen", true, false);
```

onPostExecute()

- Ressourcen des ProgressDialogs werden freigegeben

```
ladebalken.dismiss();
```

- TextViews 1 bis 3 werden mit den Inhalten (Strings) befüllt.

```
public void run() {
    tv1=(TextView)findViewById(R.id.tv_news_date);
    ...
    tv1.setText(s[2]);
}
```

- Der User kann den Inhalt der TextViews kopieren und in die Zwischenablage des Endgeräts speichern.

```
tv1.setTextIsSelectable(true);
```

- Im TextView 4(*Inhalt der News*) werden einige besondere Anweisungen benötigt:
 - Darstellung und Verfolgung von (Hyper-)links.

```
tv4.setClickable(true);  
tv4.setMovementMethod(LinkMovementMethod.getInstance());
```

- Verarbeitung des vom NewsResolver übergebenen HTML-Strings

```
tv4.setText(Html.fromHtml(s[3]));
```

doInBackground()

- Neuinstanzierung eines NewsResolvers mit übergebener Url, um durch die Funktion `getDetailsStringArray` das Array `s` mit Daten zu füllen

```
s[0]=elements.get(0).child(1).text()+"\n";    //Ueberschrift  
s[1]=elements.get(0).child(2).text()+"\n";    //Subhead  
s[2]=elements.get(0).child(0).text();          //Zeit  
s[3]=elements.get(0).child(0).outerHtml().replaceAll("<p>&  
    nbsp;</p>", "");                            //Text  
return s;
```

5.2.6 FRAGMENTE

5.2.6.1 NewsFragment

Initialisierung erfolgt durch „newInstance“, die aus der NewsActivity heraus gerufen wird. (siehe Funktionsaufruf -> onNavigationDrawerItemSelected) Durch das .commit wird diese neue Instanz der Klasse dann geladen.

Überschriebene Funktionen

OnAttach()

- Verknüpfung des NewsFragments mit der MainActivity

OnCreateView()

- Hier wird lediglich das Layout der View des Fragments geladen und angewendet

onActivityCreated()

- es wird geprüft, ob es SavedInstances (bereitsgeladene Inhalte) gibt
- wenn JA:
 - die Informationen werden aus dem ARRAYSPEICHER geholt und angewandt
- wenn NEIN:
 - die Funktion „zeigeNews“ wird gerufen, welche die aktuellsten News vom Server lädt

onStart()

- in dieser Funktion wird nur noch der Listener für den Aktualisierbutton mit der Schaltfläche verknüpft. Als onClick-Event wird dann lediglich die Funktion „zeigeNews“ gerufen.

standAllone-Funktionen**isOnline()**

- Diese Methode prüft durch einen Connectivity Manager, ob eine Verbindung zum Internet besteht

zeigeNews()

- Zunächst wird durch „isOnline“ geprüft, ob eine aktive Netzverbindung besteht
- wenn NEIN:
 - Es wird ein Hinweis an den Nutzer ausgegeben
- wenn JA:
 - Es wird eine Instanz der Klasse [NewsHelper](#) erstellt, welche dann als Hintergrund-Task ausgeführt wird, um ein Einfrieren der App zu verhindern.

5.2.6.2 Klasse NewsHelper

- Klasse mit asynchroner Task-Ausführung
- Nach dem Aufruf der Methode *zeigenews* wird durch den Unterfunktionsaufruf *.execute* zunächst die Funktion *doInBackground* ausgeführt, wo eine neue Instanz des NewsResolvers erstellt wird.

```
private void zeigeNews(){
    NewsHelper nh = new NewsHelper();
    nh.execute();
}
```

- Durch *getTermineStringArray* des NewsResolvers wird ein String-Array zurückgegeben.
- Danach wird die Methode *onPostExecute* gerufen.

onListItemClick()

- Es wird ein intent verwendet um die *NewsClickedActivity* zu starten, als “Übergabeparameter“ wird *putExtra* verwendet, im Falle der App die URL zu dem angeklickten Event.

```
intent = new Intent(getActivity(), NewsClickedActivity.class)
;
intent.putExtra(TERMINDetail, newsResolver.getURLasString(
    position));
startActivity(intent);
```

onPostExecute()

- Überprüfung ob das Fragment noch aktiv ist

```
if(getActivity()==null) return;
```

- einbinden des **CustomAdapterNews** in die ListView

```
setListAdapter(new CustomAdapterNews(getActivity(), ...));
```

5.2.6.3 NavigationDrawerFragment

- Dieses Fragment bildet das Menü auf dem linken Rand der App ab und aktiviert den Button mit dem man in die Einstellungen gelangt.

onCreate()

- In dieser Methode werden die Einstellungen der Activity übernommen und der Drawer ausgewählt.
- Außerdem wird geprüft ob eine gesicherte Instanz vorhanden ist, aus der dann das zuletzt ausgewählte Fragment ermittelt wird und über die Funktion *selectItem()* aufgerufen wird.
- Falls keine gespeicherte Instanz existiert, wird die "0" (*NewsFragment*) als Standardwert übergeben.

onActivityCreated()

- Hier wird das Menü für das aktuelle Fragment aktiviert indem an *setHasOptionsMenu()* "true" übergeben wird.

onCreateView()

- In dieser Funktion wird das Design für die ActionBar (als Listview) und die dazugehörigen Menüpunkte festgelegt.
- Zudem wird hier der ClickListener initialisiert, der dann den ausgewählten Eintrag an *selectItem()* übergibt.

isDrawerOpen()

- Diese Methode prüft ob der Drawer bereits offen ist und liefert einen Wahrheitswert zurück.

setUp()

- Es werden hier folgende Einstellungen vorgenommen:
 - Einstellungen zum Design
 - Aktivierung des HomeButtons und dessen Animation beim Draufklicken
 - Zusammenführung der *ActionBar* und des *NavigationDrawers*
 - weitere Einstellungen zum Drawer
- Dann wird die Konfiguration in *mDrawerToggle* abgespeichert

selectItem()

- Hier wird die Animation auf den angeklickten Menüpunkt ausgeführt.
- Zudem wird der Drawer geschlossen und die Position des angeklickten Punktes an *mCallbacks.onNavigationDrawerItemSelected()* übergeben.

onAttach()

- Diese Methode setzt den Zeiger `mCallbacks` auf die Activity.

onDetach()

- Hier wird der Zeiger `mCallbacks` auf `“null“` gesetzt.

onSaveInstanceState()

- Diese Funktion sichert die aktuelle Instanz.

onConfigurationChanged()

- Bei einer Änderung in den Einstellungen konfiguriert diese Methode `mDrawerToggle` um.

onCreateOptionsMenu()

- Diese Funktion legt das Design, aus einer XML-Datei für das Menü Einstellungen, fest.

onOptionsItemSelected()

- Hier wird der Button, über den man zu den Einstellungen gelangt, aktiviert und mit dessen Klasse verknüpft.

NavigationDrawerCallbacks()

- Hier wird die ausgewählte Menüpunkt-ID an die Activity übergeben.

5.2.6.4 Notenfragment

onCreateView()

- Laden des entsprechenden XML-Layouts *fragment_noten.xml*
- Laden und Zuweisung der Schriftart des TextViews für die Überschrift des Fragments

onAttach()

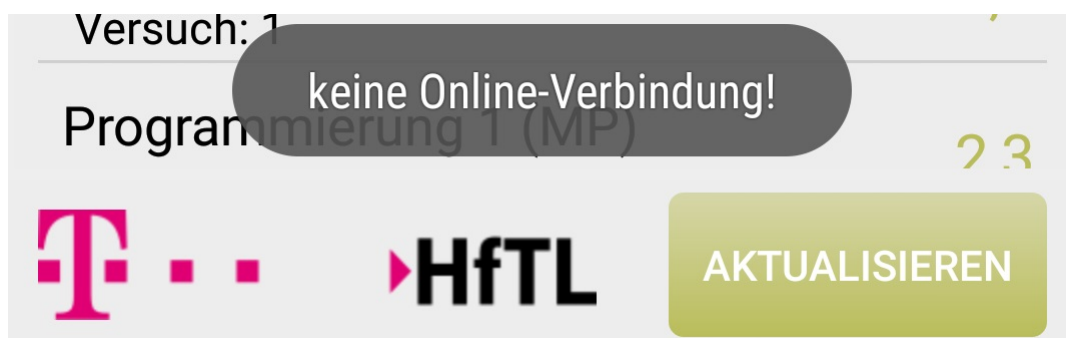
- Aufruf der Datenbank, in der die Noteneinträge abgelegt werden

onDetach()

- Schließen der Datenbank

onStart()

- Erstellen des Buttons zum Aktualisieren
- Mittels *OnClickListener* für den Button, wird über die Methode *testeBenutzerdaten()* überprüft, ob die Benutzerdaten für QiS eingetragen wurden. Andernfalls erfolgt die Ausgabe mittels der Methode *keineBenutzerdaten()*, dass diese nicht eingetragen wurden.
- Sofern die Benutzerdaten eingetragen wurden, und eine Onlineverbindung zu QiS besteht, werden die Noten erneut abgerufen.
- Ist keine Verbindung zu QiS vorhanden, wird der Nutzer mittels Toast-Benachrichtigung darüber informiert

**getNoten()**

- Liest in der Notendatenbank alle Werte in der Spalte **NOTENABFRAGE** aus und schreibt diese in das String-Array *s*, welches auch übergeben wird.

getSemester()

- Liest in der Notendatenbank alle Werte in der Spalte **SEMESTERABFRAGE** aus und schreibt diese in ein String-Array *s*, welches auch übergeben wird.

getFach()

- Liest in der Notendatenbank alle Werte in der Spalte **FACHABFRAGE** aus und schreibt diese in das String-Array `s`, welches auch übergeben wird.

getVersuche()

- Liest in der Notendatenbank alle Werte in der Spalte **VERSUCHABFRAGE** aus und schreibt diese in das String-Array `s`, welches auch übergeben wird.

Hinweis:

Die String-Arrays sind nur lokal in den jeweiligen Methoden definiert, weshalb diese - der Einfachheit halber - alle den Namen "s" erhielten. Tatsächlich werden hier vier unterschiedliche String-Arrays befüllt.

setzeListView()

- Hier werden zunächst entsprechende String-Arrays durch die jeweiligen Methoden zur Abfrage in der Datenbank befüllt:

```
notenList = getNoten();  
semesterList = getSemester();  
fachList = getFach();  
versuchList = getVersuche();
```

- Die Strings stehen entsprechend der alphabetischen Reihenfolge der Semesterbeschreibung in den Arrays.
- Bsp: SoSe 12, SoSe 13, WiSe 11/12, WiSe 12/13, WiSe 13/14
- Um die Übersichtlichkeit zu wahren, werden die Werte in den Arrays nun sortiert, wobei die Werte vom aktuellsten Semester am Anfang stehen und anschließend chronologisch geordnet werden.
- Um dies zu realisieren wird zunächst mit Hilfe eines regulären Ausdrucks ein Suchpattern festgelegt, nach dem die zu sortierenden Elemente gefunden werden.
- Entscheidend für die Sortierung sind die letzten beiden Ziffern für die Jahreszahl.

```
Pattern p = Pattern.compile("\\d{2}$");
```

- Anschließend wird der Einfachheit halber ein Bubblesort verwendet (kann ggf. geändert werden), der durch das String-Array `semesterList` geht und bei benachbarten Einträgen den Suchpattern anwendet.

```
Matcher m = p.matcher(semesterList[j]);  
Matcher n = p.matcher(semesterList[j+1]);
```

- Sofern die Pattern in den beiden Einträgen vorhanden sind (das ist eine zwingende Bedingung), werden diese jeweils in Hilfsvariablen geschrieben.

```

if((m.find())&&(n.find())){
    int x = Integer.parseInt(m.group());
    int y = Integer.parseInt(n.group());
    ...}

```

- Danach werden diese Hilfsvariablen verglichen und ggf sortiert.
- Sofern die Einträge in (**semesterList**) sortiert werden, müssen auch die zugehörigen Werte in den anderen Arrays (**notenList**, **fachList**, (**versuchList**) entsprechend sortiert werden.

```

if(x<y){
    temps = semesterList[j];
    semesterList[j] = semesterList[j+1];
    semesterList[j+1] = temps;
    temps = notenList[j];
    notenList[j] = notenList[j+1];
    notenList[j+1] = temps;
    temps = fachList[j];
    ...
}

```

- Sobald die Sortierung erfolgt ist, müssen noch redundante Werte im Array (**semesterList**) gefunden und durch (**NULL**) ersetzt werden.
 - Dadurch wird eine Auflistung der Noten, Fächer und Versuche, aufgeschlüsselt nach Semester realisiert, ohne dass über jedem Eintrag die Semesterbeschreibung steht. (s. **CustomAdapterNoten.java**)

```

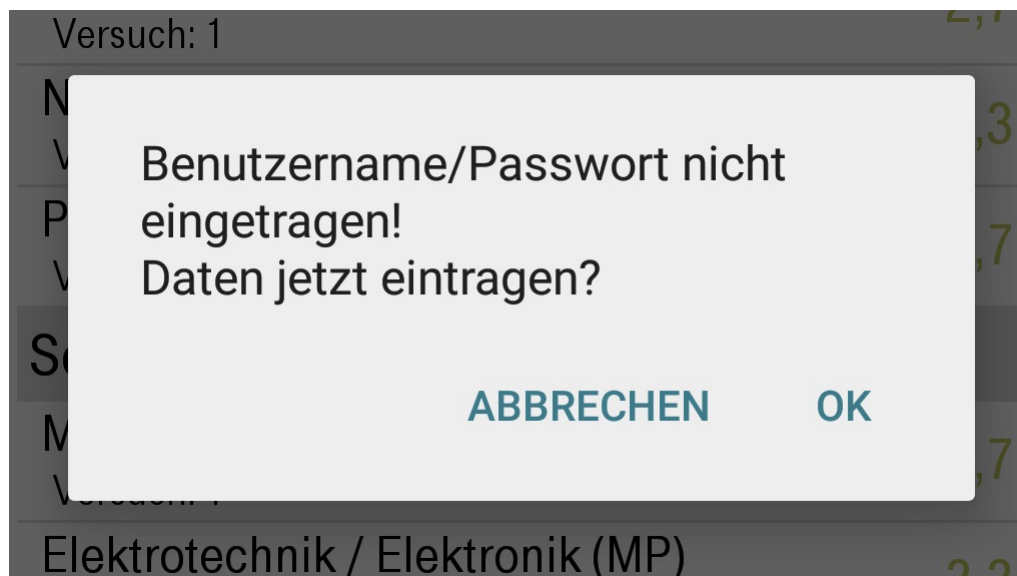
String[] vgl = new String[1];
vgl[0]="";
for(int k=0; k<semesterList.length;k++){
    if (semesterList[k].equals(vgl[0])) {
        vgl[0] = semesterList[k];
        semesterList[k] = null;
    }
    else{
        vgl[0] = semesterList[k];
    }
}

```

- Anschließend werden diese String-Arrays an den **CustomAdapterNoten** übergeben. Damit wird eine individuelle Befüllung und Formatierung der Liste mit den ausgelesenen Werten aus der Datenbank realisiert.

keineBenutzerdaten()

- Diese Methode prüft, ob in den Einstellungen Login und Passwort für QiS eingetragen wurden. Andernfalls wird eine Fehlermeldung ausgegeben:



– NotenHelper (Class)

5.2.6.5 StundenplanFragment

newInstance()

- Erstellt ein StundenplanFragment und “steckt” die aktive Position (aus dem Navigation Drawer) in das “Bundle args” welches als Argument im Fragment übergeben wird.

onCreateView()

- Laden des entsprechenden XML-Layouts fragment_noten.xml Laden und Zuweisung der Schriftart für das TextView für die Überschrift des Fragments

onViewCreated()

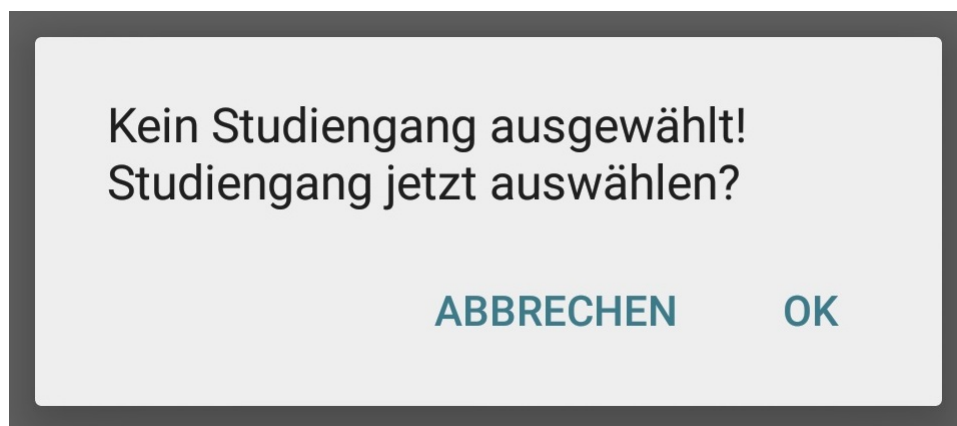
- Falls im “Stundenplanspeicher” Daten vorhanden sind, werden diese geladen
- Methode für das Dropdownmenü wird gerufen und Array für “events” erstellt
- anonyme Listener für die Buttons werden erstellt

erzeugeDropdown()

- Dropdown aus xml einen Objekt zuweisen
- Listener für Dropdown (als anonymer Listener) wird erzeugt und registriert
- Beim registrieren wird die Methode *onItemSelected* aufgerufen und ein StundenplanHelper ausgeführt
- Mittels eines “Calendar”, “Date” und zwei “SimpleDateFormat” wird das Dropdownmenü befüllt, indem die Daten in eine String-List eingefügt werden (*list.add(temp)*)
- Dropdown mit Liste verknüpfen

keinStudiengang()

- Prüft ob in den Einstellungen der Studiengang eingetragen wurde, ansonsten wird eine Fehlermeldung ausgegeben:



erstelleStundenplan()

- Erzeugt die Ausgabe des Stundenplans und fügt sie in den ListView ein
- Falls keine Daten vorhanden sind, wird “keine Date“ ausgegeben

– StundenplanHelper (class)

onPostExecute()

- erzeugt einen Ladebalken

onPostExecute

- falls das Fragment noch aktiv ist wird die Methode `erstelleStundenplan()` gerufen
- Ladebalken wird entfernt

doInBackground

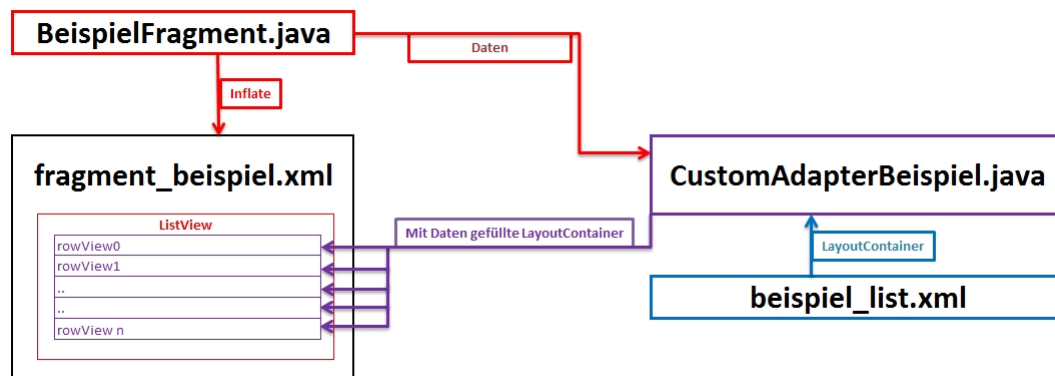
- Die Methode erzeugt einen *StundenplanResolver*, befüllt das “StundenplanEvents-Array Events“ mit Daten durch die Methode *erzeugeStundenplan(String woche)* des *StundenplanResolvers*.
- Falls ein Fehler auftritt wird ein Event erstellt in dem keine Daten sind.

5.2.7 CUSTOMADAPTER

5.2.7.1 Allgemein

Die CustomAdapter kommen da zum Einsatz, wo eine ListView genutzt und individuell befüllt werden muss.

- 1 Fragment.java wird mittels inflate der fragment.xml zugeordnet
- 2 Fragment.java übergibt die zuvor ermittelten Daten (mit Strings befüllte Arrays) an den CustomAdapter
- 3 Entsprechend der Länge x der Arrays wird x-mal die zugehörige _list.xml als LayoutContainer aufgerufen und ihre TextViews mit den Inhalten der Arrays befüllt.
- 4 Die befüllten LayoutContainer werden zeilenweise an die ListView in der fragment.xml übergeben.



class Holder

Dies ist ein Container mit TextViews

```
TextView tv_name;
```

Dieser Container wird dann als jeweilige Zeile in der ListView der zugehörigen Activity dargestellt.

Die Grundlage der Ausgabe bilden spezielle XML-Dateien. Diese dienen definieren die Output jeder einzelnen Zeile der ListViews. Hier sind Name, Position, Dimensionen, Farben, **Schriftarten**, Schriftgröße etc festlegt.

Für jeden Customadapter gibt es eine entsprechende _list.xml:

- news_list.xml
- noten_list.xml
- stundenplan_list.xml

public View getView()

erstellt einen neuen Container

```
Holder holder = new Holder;
```

erstellt eine View rowView

```
View rowView;
```

Das Layout der View und damit jeder Zeile in der ListView wird durch das Laden der entsprechenden XML namelist formatiert:

```
rowView = inflater.inflate(R.layout.name_list, null);
```

Die TextViews des Containers werden den entsprechenden TextViews in der XML zugeordnet:

```
holder.tv_name=(TextView)  
rowView.findViewById(R.id.namelist_name.xml);
```

Das TextView wird mit dem entsprechenden Wert des zugehörigen String-Arrays befüllt:

```
holder.tv_name.setText(name[position]);
```

Der nun befüllte Container wird nun als rowView übergeben.

```
return rowView;
```


5.2.7.2 CustomAdapterNews.java

Dies ist der simpelste CustomAdapter der HfTL-App. Er wird verwendet um die News-Liste (also die Übersicht der News aus der HfTL-Homepage) zu formatieren.

Dabei werden drei String-Arrays (date, headline, content) in das zugehörige TextView der rowView übergeben. Der Inhalt dieser Arrays wird mittels der Klasse NewsHelper, die wiederum die Methoden der Klasse NewsResolver aufruft, befüllt.

5.2.7.3 CustomAdapterNoten.java

Dieser CustomAdapter wird für die Befüllung der ListView des Notenfragments benutzt.

Hier werden vier String-Arrays (**subject**, **trys**, **mark**, **semester**) mit den Daten aus der Notendatenbank befüllt.

Entsprechend des Inhalts, werden die zugehörigen TextViews noch gesondert formatiert.

Ist der Inhalt an der Position des String-Arrays "NULL", wird das zugehörige TextView auf "GONE" gesetzt. So wird realisiert, dass mehrere Fächer unter dem selben Semester gelistet sind, ohne dass ein leeres (dunkelgraues) TextView erscheint.

```
if(semester[position]==null){
    holder.tv_semester.setVisibility(View.GONE);
}
```

Auch für die jeweiligen Noten gibt es eine gesonderte Formatierung:

- Note schlechter als 5.0: **magenta**
- Note schlechter als 3,4: **gelb**
- Note besser als 3,5: **grün**

```
if (mark[position].equals("5,0" ))
    holder.tv_mark.setTextColor
        (context.getResources().getColor(R.color.magenta));
else if (mark[position].equals("4,0" ) |
        mark[position].equals("3,9" ) |
        mark[position].equals("3,8" ) |
        mark[position].equals("3,7" ) |
        mark[position].equals("3,6" ) |
        mark[position].equals("3,5" ) )
    holder.tv_mark.setTextColor
        (context.getResources().getColor(R.color.gelb));
else
    holder.tv_mark.setTextColor
        (context.getResources().getColor(R.color.gruen));
```

5.2.7.4 CustomAdapterStundenplan

Dieser CustomAdapter befüllt die ListView des StundenPlanfragments.

Hier werden fünf StringArrays(**datum**,**fach**,**zeit**,**raum**,**kategorie**) übergeben, die zuvor mittels der Methode *erstelleStundenplan()* des StundenplanFragments aus dem HTML-Code der QiS/HiS-Seite ausgelesen wurden.

Auch hier gibt es ein einige spezifische Formatierungen, abhängig vom übergebenen Inhalt.

Ist der Inhalt des StringArrays **datum** an einer Stelle "NULL", so wird die Sichtbarkeit des zugehörigen TextViews auf "GONE" gesetzt.

Analog zum **CustomAdapterNoten** werden so die einzelnen Fächer unter dem selben Datum gelistet. Andernfalls würde über jedem Fach ein dunkelgraues TextView stehen.

```
if(datum[position]!=null)
    holder.tv_date.setText(datum[position]);
else
    holder.tv_date.setVisibility(View.GONE);
```

Das StringArray **kategorie** wird - neben der reinen Ausgabe - dazu verwendet, wichtige Ereignisse farblich kenntlich zu machen. Dabei wird der Inhalt an der jeweiligen Position des StringArrays geprüft und eine entsprechende (CI/CD-)Farbe für das Rechteck auf der rechten Seite gesetzt:

- Prüfung: **magenta**
- Praktikum: **dunkelblau**
- Rest: **grau01**

- Diskrete Mathematik
Uhr, Virtuell - TT Raum 1



Indikator-Feld für die entsprechenden Kategorien

```
if (kategorie[position].equals("Pruefung"))
    holder.tv_category.setBackgroundColor
        (context.getResources().getColor(R.color.magenta));
else if ((kategorie[position].equals("Praktikum"))
    holder.tv_category.setBackgroundColor
        (context.getResources().getColor(R.color.dunkelblau));
else
    holder.tv_category.setBackgroundColor
        (context.getResources().getColor(R.color.grau01));
```

5.2.8 XML-DATEIEN

5.2.8.1 fragment.xml

Die fragment.xml bilden das visuelle Rückgrat der App. Hier wird das Layout festgelegt und damit wo welcher Inhalt steht bzw. zu stehen hat. Statische Inhalte und Formatierungen (Farben, Dimensionen, Elementposition, Schriftarten etc.) werden bereits hier festgelegt.

Variable Formatierungen und Inhalte werden entweder direkt durch die zugehörige *Fragment.java* oder - wenn sie die ListViews betreffen - durch den zugehörigen CustomAdapter befüllt.

Auf den detaillierten Aufbau wird nicht näher eingegangen, da dieser mit einigen XML-Kenntnissen selbsterklärend sein dürfte.

5.2.8.2 _list.xml

Hier gibt es derzeit drei XML-Dateien:

- news_list.xml
- noten_list.xml
- stundenplan_list.xml

Diese dienen als LayoutContainer für die ListViews. Sie werden durch den entsprechenden CustomAdapter mit den Daten aus der *Fragment.java* befüllt und zeilenweise an das zugehörige ListView der *Fragment.xml* übergeben.

5.2.8.3 activity_news_clicked.xml

Dient als Designvorlage für eine aufgerufene News aus der Newsübersicht. Der Inhalt wird durch die Klasse [NewsClickedActivity.java](#) befüllt.

5.2.8.4 settings_toolbar

Definiert die Kopfzeile der *einstellung.xml*.

5.2.8.5 impressum.xml & activity_impresum

Hier steht das Impressum. Während das Layout in der *activity_impresum.xml* festgelegt wird, wird der Inhalt (*Ähnlich wie bei strings.xml*) zentral abgelegt.

5.2.8.6 colors.xml

Um den Aufruf der Farben nach CI/CD zu vereinfachen, wurden die RGB-Werte (in hexadezimal) zentral in der Datei main\res \values \colors.xml gespeichert.

```
<item name="magenta" type="color">#E20074</item>
<item name="grau01" type="color">#A4A4A4</item>
...
<integer-array name="telekomcolors">
    <item>@color/magenta</item>
    <item>@color/grau01</item>
    ...
</integer-array>
```

Die Farben können dann sowohl mittels JAVA als auch mittels XML aufgerufen werden:

- XML:

```
android:color="@color/magenta"
```

- JAVA:

```
getResources().getColor(R.color.magenta)
```

5.2.8.7 array.xml

Hier stehen alle aktuellen Studiengänge mit Matrikel und ihre zugehörigen Links zu QiS.

Ein Studiengang muss in den Einstellungen ausgewählt werden, damit der Stundenplan angezeigt werden kann.

Ferner sind hier die Werte für das Intervall der Notenabfrage abgelegt.

Die Pflege der Datei erfolgt derzeit manuell. (s. [Pflege der array.xml](#))

5.2.8.8 einstellung.xml

Layout-Definition des Einstellungsmenüs.

5.2.8.9 strings.xml

Statische Strings, wie sie beispielsweise für die Fragmentüberschriften verwendet werden, werden nicht in den XML-Tag geschrieben.

```
<TextView
...
android:text="News aus der HFT-Leipzig"
.../>
```

Vielmehr werden diese zentral in der *strings.xml* abgelegt.

Listing 4: Ablage in strings.xml

```
<string name="News_headline">News aus der HFT-Leipzig</string>
```

Anschließend werden diese Strings dort abgerufen, wo sie benötigt werden.

Listing 5: Aufruf im fragment.xml

```
<TextView  
...  
android:text="@string/News_headline"  
.../>
```

5.2.9 LAYOUT

5.2.9.1 Allgemeines

Das Layout wird zum Großteil über die XML-Dateien realisiert. Um die Handhabung zu realisieren, wurden einige Elemente bzw. Bezeichnungen standardisiert.

Im Folgenden werden die wichtigsten Details rund um die Darstellung des Layouts erläutert. Mit einigen XML-Kenntnissen sollte der Großteil selbsterklärend sein.

5.2.9.2 Schriftarten

Da die Standardbibliothek von Android nur sehr wenige Schriftarten liefert, mussten die Schriftarten nach CI/CD nachträglich eingefügt werden. Diese befinden sich unter `main/assets/fonts`. Der Aufruf der Schriftarten erfolgt im Allgemeinen über folgende JAVA-Anweisung:

Listing 6: Beispiel: Einbinden der Schriftart OcrA im Notenfragment

```
View notenView = inflater.inflate(R.layout.fragment_noten,
    container, false);

TextView hl = (TextView) notenView.findViewById(R.id.hl_Noten);
Typeface headline = Typeface.createFromAsset(getActivity().
    getAssets(), "fonts/OCRA.TTF");
hl.setTypeface(headline);

return notenView;
```

In der Praxis hat sich jedoch herausgestellt, dass diese Art und Weise, Schriftarten einzubinden auf zwei Probleme trifft:

- Über das Fragment können die Schriftarten nicht den ListViews zugewiesen werden, da bei Erstellung der Fragmente die ListViews bzw. deren Inhalt (die durch die jeweiligen CustomAdapter befüllt werden) noch nicht existieren.
- Es wäre möglich die Schriftarten im jeweiligen CustomAdapter zuzuweisen. Jedoch hat die Praxis gezeigt, dass dadurch der Speicherverbrauch extrem ansteigt (200 bis 300 Prozent) und damit die Performance der App merkbar sinkt. Der Grund hierfür liegt daran, dass bei jedem TextView, das angezeigt wird, die Schriftarten aus den Assets geladen und im Cache abgelegt werden.

Das Problem wurde durch einige Anpassungen umgangen:

- Erstellen der Klasse FontCache
 - Diese Klasse prüft ob die geforderten Schriftarten bereits im Cache hinterlegt sind.

- Sollten die Schriftarten bereits im Cache hinterlegt sein, dann werden diese genommen. Andernfalls werden sie aus den Assets geladen und im Cache abgelegt.
- Einbinden der Schriftarten in jeweilige Klassen
 - Die Klassen werden direkt durch die TextViews in den XML-Dateien aufgerufen.
 - Beim Aufruf wird die o.g. Klasse *FontCache* aufgerufen.
- Setzen der Schriftarten in den XML-Dateien
 - Um die benötigten Schriftarten zu laden, werden die XML-Tags geändert.
 - * Ursprüngliche (native) XML-Tags:

Listing 7: Normales TextView

```
<TextView
    android:id="@+id/newslist_date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    ... />
```

- * Pfadangabe zur jeweiligen Schriftart als neues XML-Tag:

Listing 8: Einbinden von TeleGrotNorm

```
<bkmi.de.hftl_app.help.Typefaces.TeleGrotNorm
    android:id="@+id/newslist_date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    ... />
```

5.2.9.3 Buttons

Grundlegend werden vier Standardbuttons in der App verwendet. Sie alle wurden über eine eigene XML-Datei definiert und durch die XML-Dateien der jeweiligen Fragmente aufgerufen. Die Methoden, die bei Betätigung der Buttons aufgerufen werden, befinden sich in der JAVA-Datei des jeweiligen Fragments.

Durch die Auslagerung der Definitionen für die Buttons ist es u.a. möglich, Farbverläufe, abgerundete Ecken und interaktives Verhalten (bspw. Farbänderungen bei Betätigung) festzulegen. Im Grunde genommen wird nur ein *Item* generiert, das als Vorlage für einen Button herhalten muss.

Exemplarischer Button

- Es gibt drei Zustände:

- gedrückt

```
<item android:state_pressed="true"> ... </item>
```

- fokussiert

```
<item android:state_focused="true"> ... </item>
```

- standard

```
<item> ... </item>
```

- Der Status *focused* wird derzeit nur als Dummy angesehen, da er derzeit nicht Verwendung ist.
- Die Definitionen innerhalb der Items findet jeweils zwischen diesen Tags statt:

```
<item>  
  <shape> ... </shape>  
</item>
```

- Farbverlauf mit Start- und Endfarbe, sowie dem Winkel des Farbverlaufs.

```
<gradient  
  android:startColor="#77E20074"  
  android:endColor="#E20074"  
  android:angle="270"/>
```

- Umrandung

```
<stroke  
  android:width="3dp"  
  android:color="@color/hellblau"/>
```

- Rundung der Ecken


```
<corners  
    android:radius="5dp" />
```

Folgende Standardbuttons werden benutzt:

custom_button

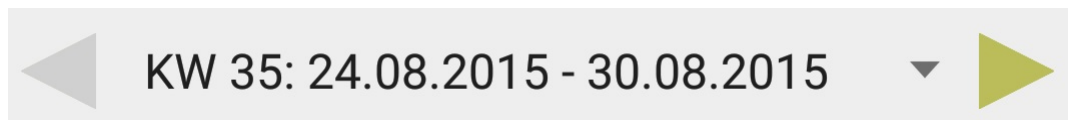
- Dies ist der Standard-Button.

custom_newsclick

- Hier werden die News-Einträge der News-Übersicht als Button verwendet, um ein optisches Feedback zu erhalten, wenn eine News ausgewählt wurde.

custom_vorbutton & custom_zurbutton

- Diese beiden Buttons werden beim Stundenplan verwendet. (Grüne Dreiecke um zwischen den Wochen vor- und zurückzublättern.)



- als Vorlage dienen jpg-Images
- Je nach Status werden die entsprechenden Images geladen:
 - inaktiv: grau
 - aktiv: grün
 - gedrückt (optisches Feedback): magenta

5.2.10 ERWEITERUNGEN UND VERBESSERUNGEN FÜR KOMMENDE VERSIONEN

Da es aufgrund der Projektrahmenbedingungen nicht möglich gewesen ist, alle gewünschten Features zu implementieren, folgt hier eine Auflistung weiterer Funktionen oder Verbesserungen, die für die App denkbar wären.

5.2.10.1 Pflege der array.xml

Die Datei array.xml dient als Grundlage für das Einstellungsmenü.

Hier werden alle Studiengänge, mit den entsprechenden Matrikeln aufgelistet. Das Kernproblem ist, dass diese Liste manuell auf dem aktuellen Stand gehalten werden muss.

Denkbar wäre hier eine Methode, die zu gewissen Zeitpunkten (*Ende der Semesterferien*) die neuen Matrikel hinzufügt. Problematisch dabei sind jedoch mehrere Aspekte, die beachtet werden müssen:

- Festlegung der Zeitpunkte zur Aktualisierung der Daten
- Wegfall der alten Matrikel (Vermeidung von Datenmüll)
- Umbenennung von Studiengängen
- Hinzufügen neuer Studiengänge
- Wegfall alter Studiengänge

Die Problemstellung ist damit für das Projekt zu komplex und kann wahrscheinlich besser in Absprache und Zusammenarbeit mit dem Hochschul- und Prüfungsamt der HfTL gelöst werden.

5.2.10.2 Raumplan

Ein Kernfeature, das aber von der Komplexität her wohl ein eigenes Projektteam beschäftigen kann, ist die Raumplanung.

Hierbei soll es dem Nutzer bspw. möglich sein, freie Räume für selbst organisiertes Lernen zu finden.

Wichtig dabei ist auch zu wissen, wann welcher Raum durch wen belegt ist. Insofern müsste es abseits von der zentralen Raumplanung durch die HfTL möglich sein, dass der Nutzer sich einen freien Raum buchen kann und dieser Buchungswunsch dann auch für andere Anwesende zu sehen ist.

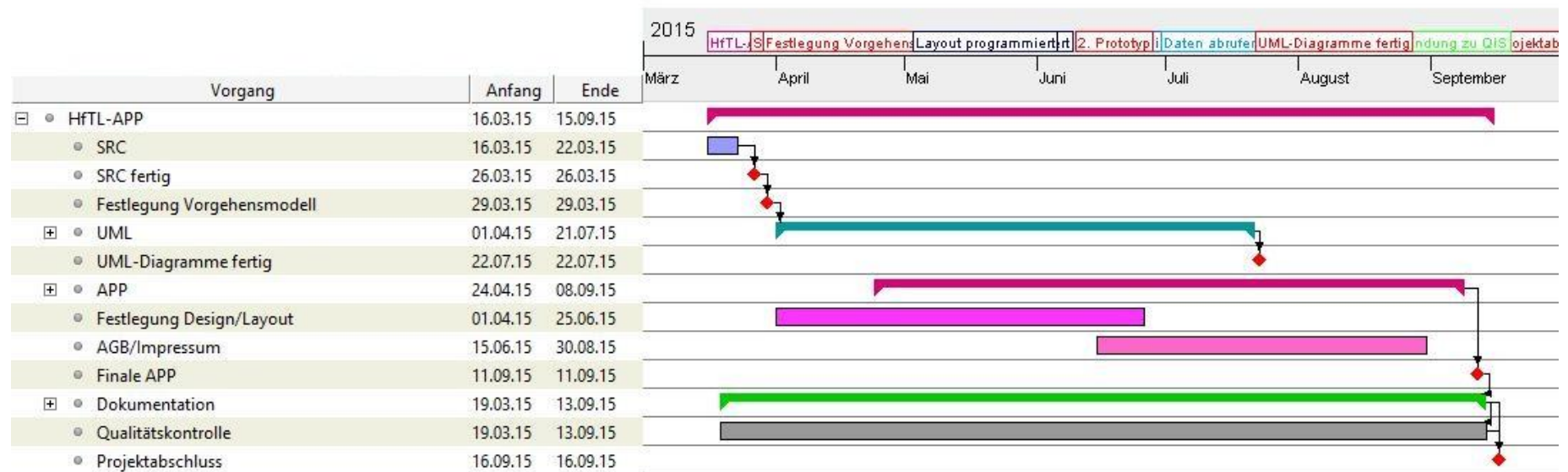
Idealerweise gäbe es auch einen Lageplan, wo welcher Raum zu finden ist.

5.2.10.3 Portierung auf andere Systeme

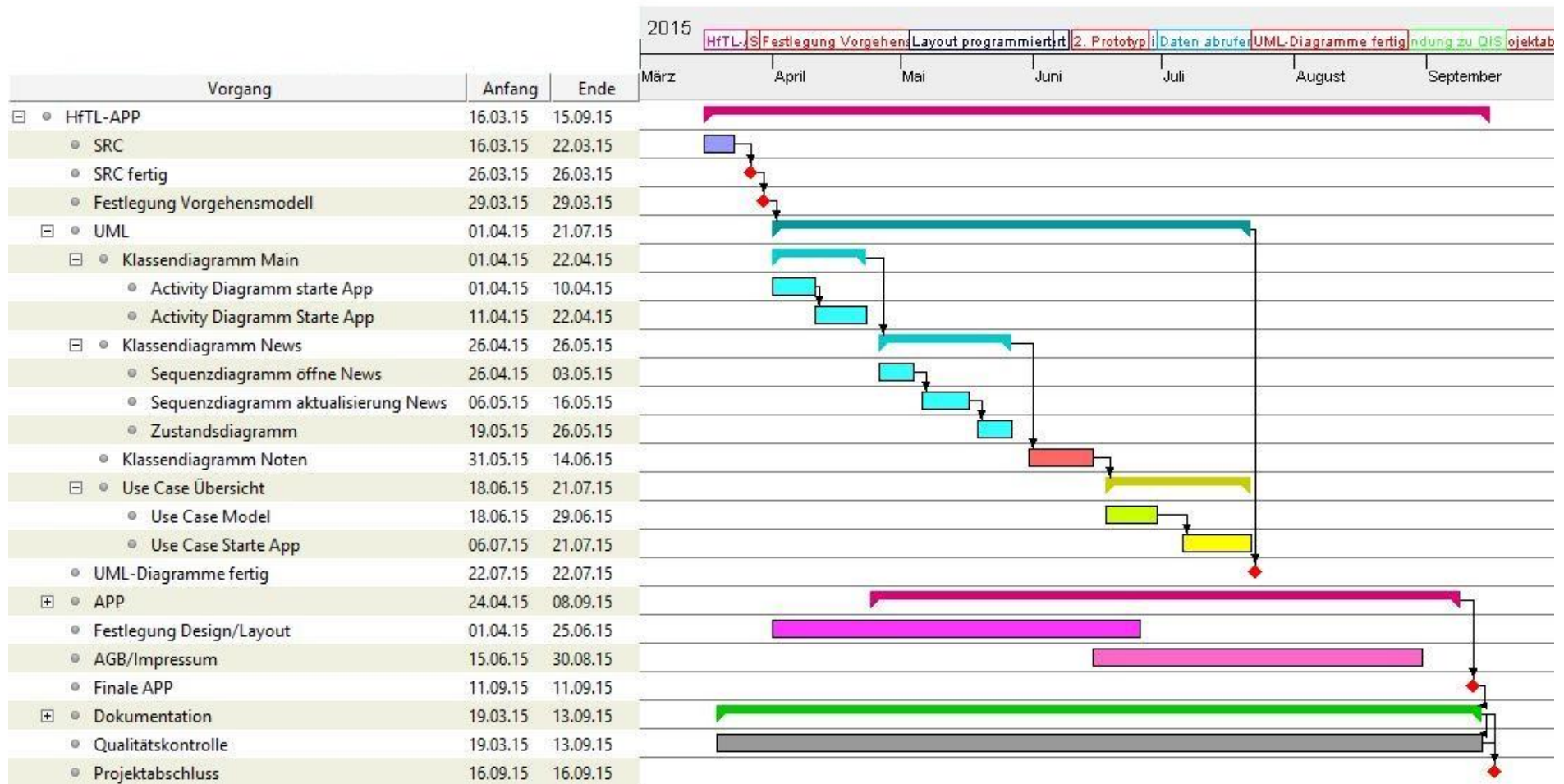
Die hier beschriebene App ist eine reine Android-App. Sofern es eine Nachfrage gibt, wäre eine Portierung auf andere Betriebssysteme (iOS, Windows Mobile, Ubuntu Touch und weitere) sicherlich sinnvoll.

5.3 GANTT

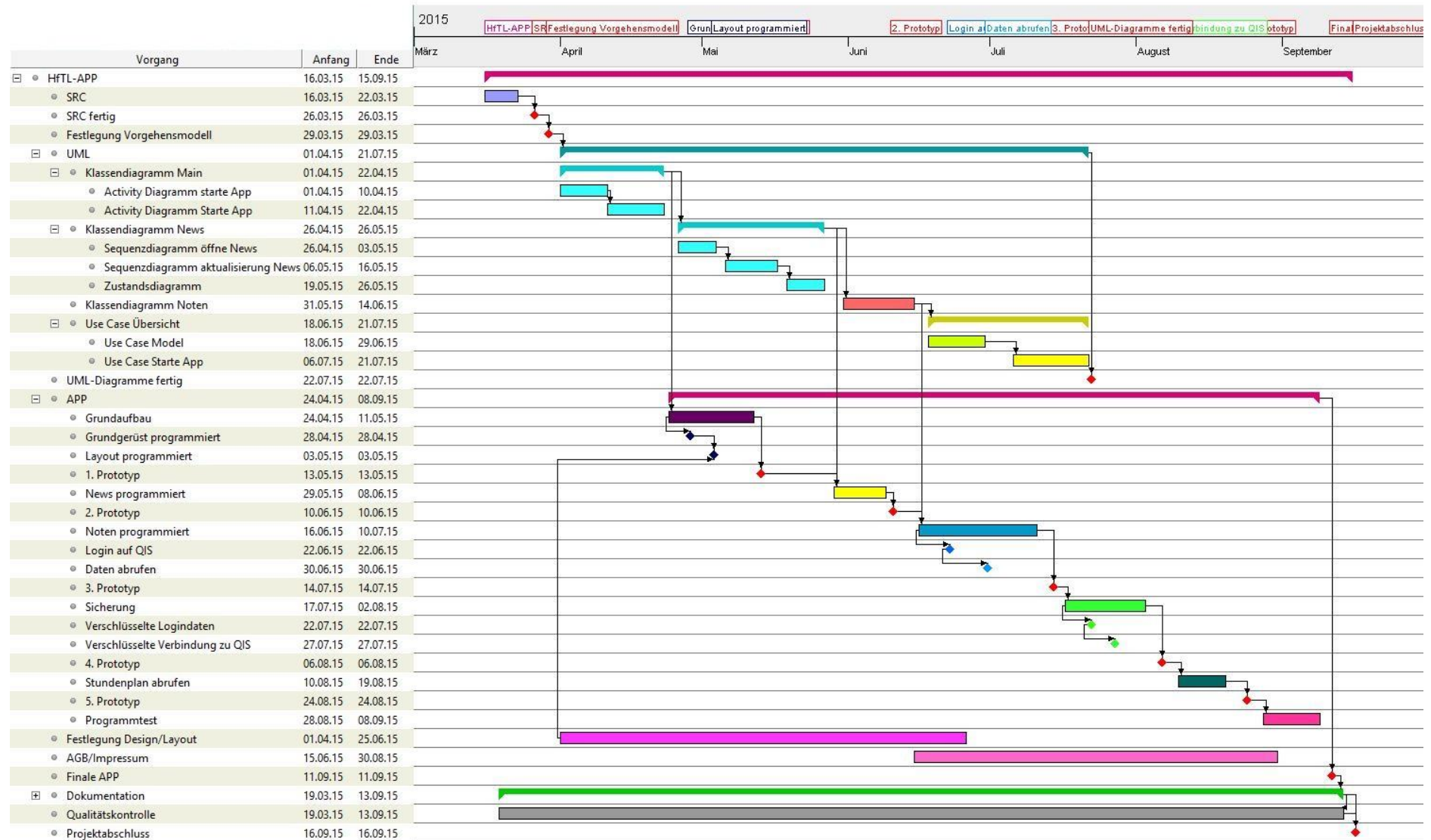
Allgemeine Übersicht



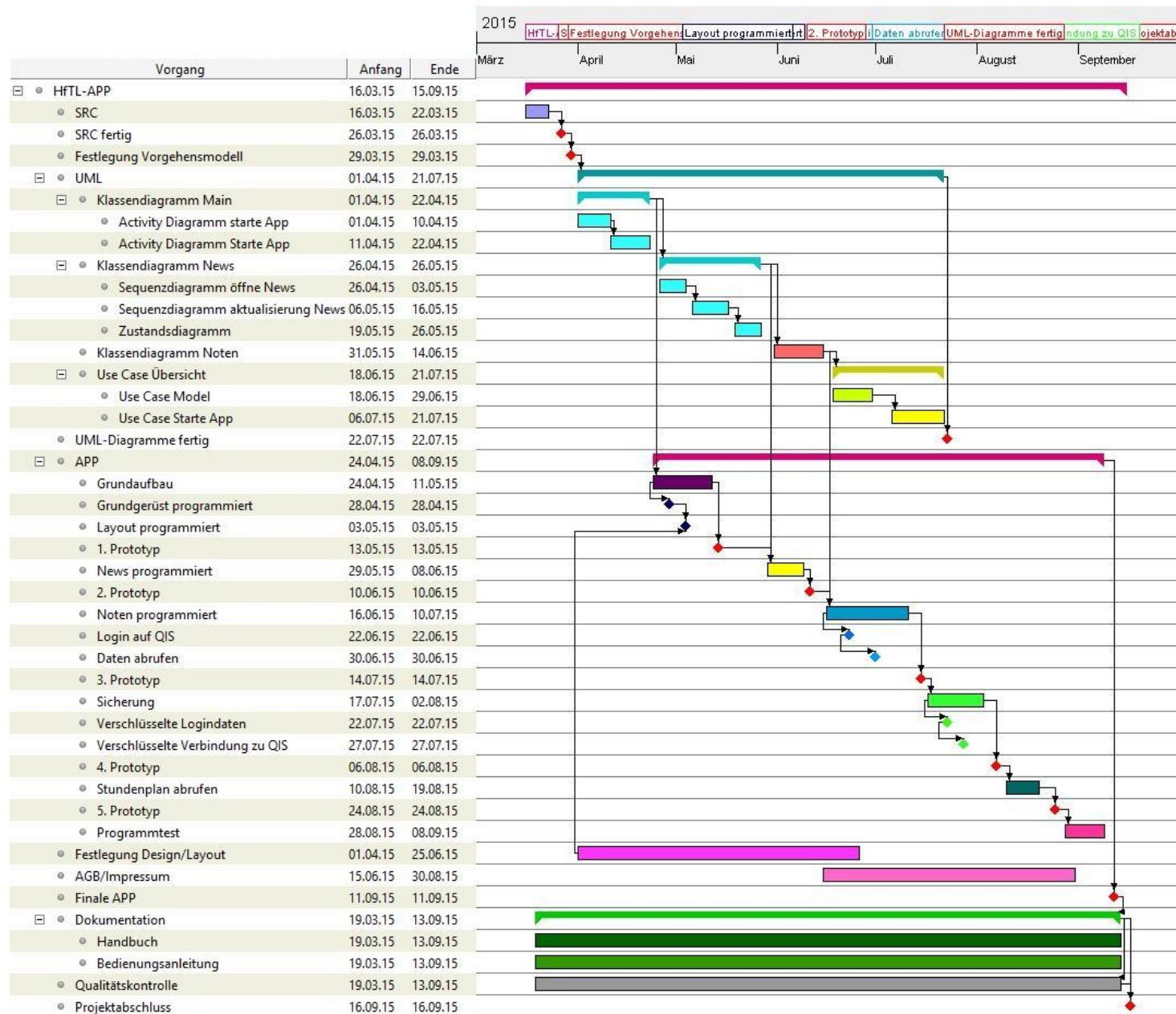
Detailsicht UML



Detailsicht UML & APP



Gesamtübersicht



5.4 APP-LAYOUT



Seite: App-Start
Hintergrund: Grafik_Stift.eps
Logo: HFTL_4C_P.eps
Auflösung: 720 x 1280 px

Content: Padding (global): 8dp;
Hintergrundfarbe (global): Grau04;

Allgemeine Angaben:

- Die angegebenen Werte beziehen sich auf eine Punktdichte von 320dpi
- sollte die verwendete Schriftart nicht verfügbar sein gilt 'Arial' als Rückfalloption
- Der Seitenabstand von 8dp stellt einen Kompromiss zwischen dem CD der Hochschule und den Vorgaben von Google-Developers dar





Seite: News

A - Header: Menü-Button-Grafik: ic_drawer.png
Headline-Schrift: wird vom Smartphone gesetzt

B - Überschrift: Schrift: OCR A Extended; 24sp; Grün;

C - Footer

C1 - Logo: HfTL_OB_4C_P.eps

C2 - Button: Schrift: Tele-GroteskHal; 18dp; Weiß;
Verlauf: #E4E96E, #9A9C44

D - Content: Hintergrund: Grau06;
Grafik: Bulletpoint.png; Größe: 15dp;
Linear Layout Orientation: Vertical;
Padding: 2dp;
Schrift: Datum - Tele-GroteskNor; 15sp;
Überschrift - Tele-GroteskHal; 20sp;
Introtext - Tele-GroteskNor; 15sp;
Margin Left: Überschrift - 15dp;
Introtext - 15dp;



E - Abstand: Padding Top + Bottom: 8dp;
Divider: Grau06;

Seite: Artikel

F - Content: Linear Layout Orientation: Vertical;

Schrift: Datum - Tele-GroteskNor; 16sp;
Einleser - Tele-GroteskFet; 16sp;
Lesetext - Tele-GroteskNor; 16sp;
URL - Tele-GroteskHal; 16sp;



Seite: Noten (mit Benutzerdaten)

G - Semester: Hintergrund: Grau06;
Schrift: Tele-GroteskHal; 20sp; Schwarz;
Padding: 4dp;

H - Modulnote: Linear Layout Orientation: Vertical;

H1 - Modul: Schrift: Tele-GroteskHal; 18sp; Schwarz;
Margin Left: 8dp;

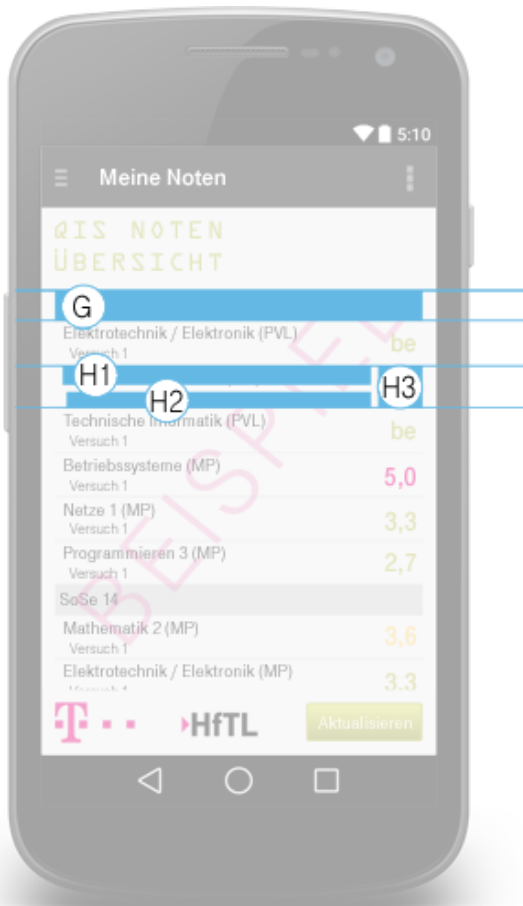
H2 - Versuchsanzahl:
Schrift: Tele-GroteskNor; 15sp; Schwarz;
Margin Left: 12dp;

H3 - Note: Schrift: Tele-GroteskHal; 20sp;
Schriftfarbe: Grün - bei Noten $\leq 3,4$
Gelb - bei Noten $\leq 4,0$
Magenta - bei Note 5,0

Layout Gravity: Center;

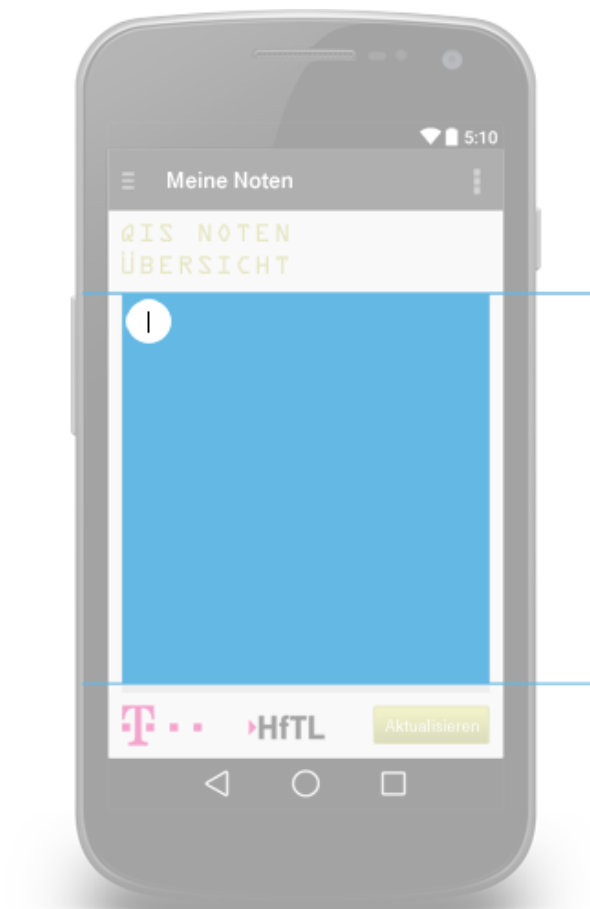
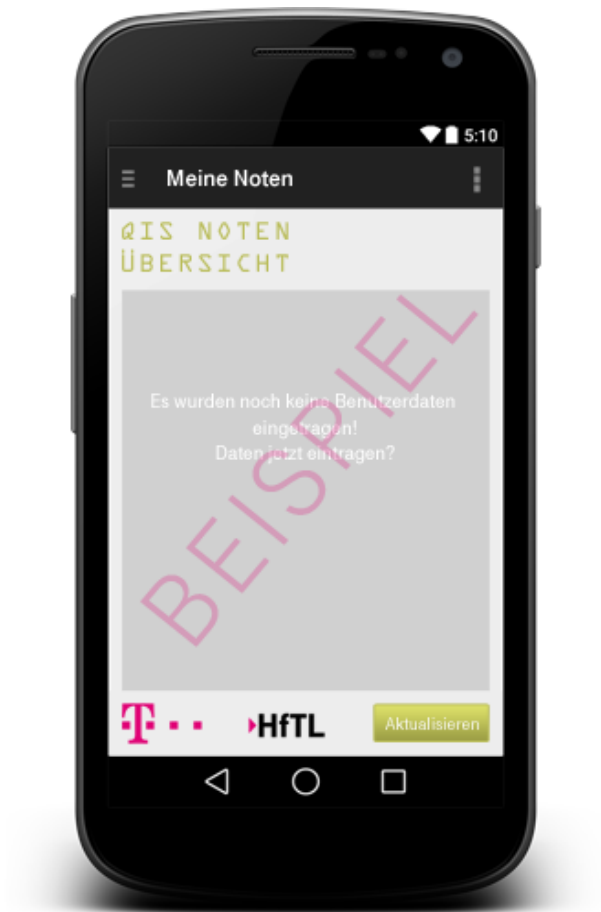
Margin Left: 4dp;

Padding: 4dp;



Seite: Noten (ohne Benutzerdaten)

I - Content: Hintergrund: Grau06;
Schrift: Tele-GroteskHal; 20sp; Weiß;



Seite: Stundenplan

J - Datumwahl: Linear Layout Orientation: Horizontal;

J1 - Button-Zurück:

Grafik: Bulletpoint_rev.png

Grafik onClick: Bulletpoint_rev_magenta.png

Grafik inaktiv: Bulletpoint_rev_grau.png

Größe: 25dp;

Margin: 8dp;

J2 - Spinner: Schrift: wird vom Smartphone gesetzt

Layout Gravity: Center;

Schriftfarbe: Grün;

J3 - Button-Vor:

Grafik: Bulletpoint.png

Grafik onClick: Bulletpoint_magenta.png

Grafik inaktiv: Bulletpoint_grau.png

Größe: 25dp;

Margin: 8dp;

K - Wochentag: Hintergrund: Grau06;

Schrift: Tele-GroteskHal; 20sp; Schwarz;

Padding: 4dp;

L - Modul: Linear Layout Orientation: Vertical;

L1 - Kategorie:

Schrift: Tele-GroteskHal; 18sp; Schwarz;

Margin Left: 8dp;

L2 - Modul: Schrift: Tele-GroteskHal; 18sp; Schwarz;

Margin Left: 8dp;

L3 - Zeit: Schrift: Tele-GroteskNor; 15sp; Schwarz;

Margin Left: 10dp;

L4 - Kategoriefarbe:

Prüfung - Magenta;

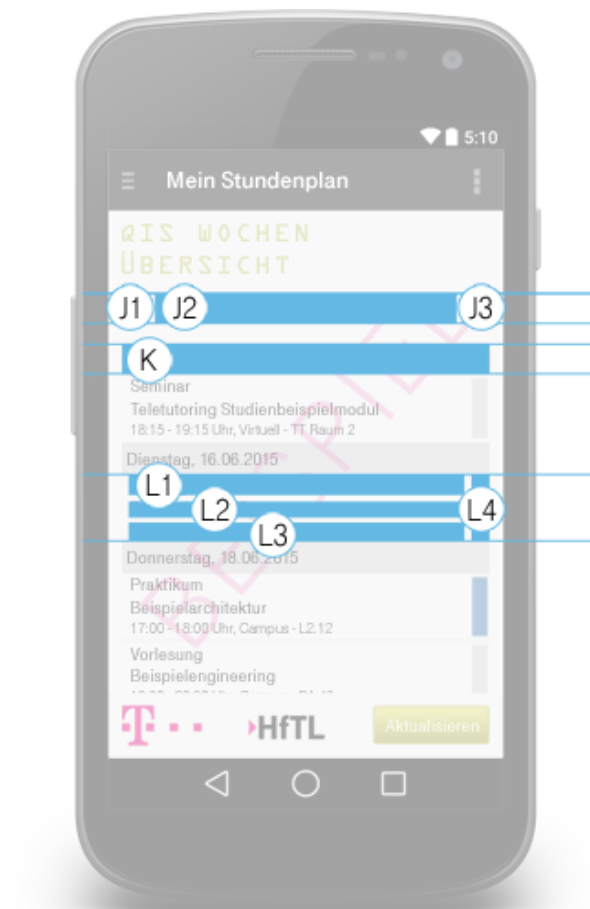
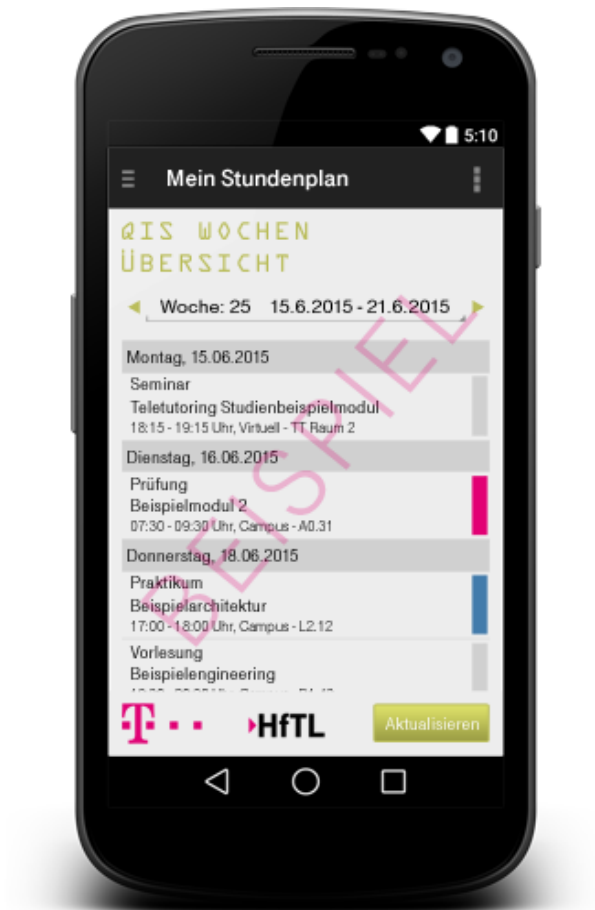
Praktikum - Dunkelblau;

Vorlesung, Seminar - Grau06;

Margin Left: 4dp;

Padding: 4dp;

Layout Gravity: Center;



5.5 RELEASE-HISTORIE

Pre-release

- v0.1

News und Notenabfrage funktionstüchtig

Ein erster Testrelease. News- und Notenabfrage sind möglich. Design noch nicht komplett implementiert. Stundenplanabfrage noch nicht implementiert.

Pre-release

- v0.2

Newsmodul fertiggestellt

- News optisch angepasst
- Notenabfrage funktionstüchtig, aber optisch noch nicht abgeschlossen.
- Stundenplan noch nicht implementiert

Pre-release

- v0.3

Stundenplan Funktion implementiert

Änderungen:

- Stundenplan Funktion implementiert
- in den Einstellungen kann der Studiengang und das Matrikel gewählt werden
- Design wird noch angepasst

FEHLER:

- wenn der Studiengang nicht vor öffnen des Stundenplans gewählt wird, stürzt die APP ab

Angehängt:

- aktuelle .apk
- aktueller Quelltext

Noten-Design implementiert

JAVA:

NotenTabelle.java:

- TYPO von "SEMSETER" auf "SEMESTER" korrogiert
- Die Spalte "VERSUCH" hinzugefügt

CustomAdapter.java:

- Umbenannt in "CustomAdapterNews"

NewsFragment.java:

- Einbinden des umbenannten CustomAdapters

CustomAdapterNoten

- neuer Adapter für die Darstellung der Noten
- Befüllung von 4 TextViews in der noten_list.xml
- wenn redundante Semesterbeschreibung (Wert: null), dann wird das entsprechende TextView auf "GONE" gesetzt
- farbliche Darstellung der Noten (5 | 4,0 -3,5 | 3,4-1,0)

NotenFragment.java:

- Einbinden des CustomAdaptersNoten.java
- Methode getVersuche() hinzugefügt
- Methode getSemester() angepasst sodass redundante Semesterbezeichnungen als "null" in das StringArray eingetragen werden

NewsResolver.java

- Löschen von auskommentiertem Code, der nicht mehr benötigt wird

NotenResolver.java

- Auslesen des Felds "Versuch" aus QiS

XML:

noten_list.xml

- neu erstellt
- gibt die die 3 bzw 4 TextViews als Reihe im ListViews der NotenView zurück

BUGS:

Design Stundenplan und PUSH-Service implementiert

funktionelle Änderungen

- Notenservice eingefügt,
- kleine Änderung am Exceptionhandling
- Anpassung Optionsmenü
- Push-Nachrichten-Funktion implementiert

Probleme:

- Push-Nachrichten starten die Newsaktivität, aber das Fragment wechselt nicht auf NotenFragment.

Todo:

- Layout: Icon für Push einfügen (hat bei mir nicht geklappt -- siehe Todo im Quelltext).
- Intervalleinstellung: "einmal pro Stunde" anstatt "einmal Stunde"

Änderungen am Design:

JAVA:

CustomAdapterNoten:

- aufgrund von Leistungsproblemen wurden die Telekom-Schriftarten für das Notenfragment entfernt(auskommentiert).
- Überschrift des Noten-Fragments bleibt davon unbetroffen.

StundenplanFragment.java:

- Typeface und Schriftart OCRA für die Überschrift des Layouts eingefügt
- Bei der Auswahl der Kalenderwoche im Spinner wird diese nun direkt geladen. Damit ist der Go-Button obsolet. (Wird in einer kommenden Version entfernt)
- Anzeige im Spinner der Woche von "Woche: " auf "KW: " geändert (*)
- Anzeige des Wochentags, neben dem Datum in der ListView

StundenplanResolver.java:

- (*) Änderung des Vergleichspatterns, damit Einträge mit "KW: " verglichen werden

NewsResolver.java:

- Änderung der Newsabfragen (Überschrift, Datum, Link und Text [s. FIXES])

XML:

fragment_stundenplan.xml:

- Design des Fragments eingefügt

strings.xml:

- Überschrift für Fragment Stundenplan hinzugefügt

SONSTIGES:

Grafik:

- bulletpoint_rev eingefügt, als "zurück-Button" für den Spinner im Stundenplanfragment

FIXES:

Da aufgrund der Umstellung am 10.07.2015 der Homepage der HfTL die Parser für die News teilw. ins Leere gelaufen sind, wurde die Newsliste teilw. falsch befüllt oder gar nicht befüllt. Aufgrund dessen kam es zu Crashes der App.

-> Fehler wurden durch Anpassung der Parser behoben.

Kalenderübertragung implementiert

Optimierungen & Einbinden der Schriftarten nach CI/CD

ALLGEMEIN

Neues Package "Typefaces" unter Package "help" erstellt.
Die Schriftarten nach CI/CD konnten nun ohne Probleme eingebunden werden. Durch die Speicheroptimierung konnte der allgemeine Cache-Verbrauch reduziert werden. (ca. 10%)

JAVA:

Neue Klassen erstellt, die via Aufruf in XML die Schriftart für das jeweilige TextView festlegen:

- OcrA.java.java: -TeleGrotFett.java: -TeleGrotHalb.java: -TeleGrotNorm.java:

FontCache.java:

Neue Klasse, die beim Laden der Schriftarten zunächst prüft, ob diese bereits in den Cache eingelagert wurden. Wenn ja, werden diese aus dem Cache verwendet.

Die Individuellenn Formatierungen in den Java-Files der einzelnen Fragmente wurden entfernt

- NewsFragment.java
- Notenfragment.java
- StundenplanFragment.java
- NewsClickedActivity.java

CustomAdapterStundenplna.java:

- Variable Zeilenanzahl in Abhängigkeit der Länge der Fachbezeichnung
 - Entfernen von Kommentare
 - Standardtext, falls keine Raumnummern übergeben werden.
-

XML:

Einbinden der Schriftarten über die Java-Klassen:

- fragment_nes.xml
- fragment_noten.xml
- fragment_stundenplan.xml
- activity_news_clicked.xml
- stundenplan_list.xml
- noten_list.xml
- news_list.xml

weiteres

Sortierung der Noten nach Semester (Absteigend nach Aktualität)

- Sortierung (BubbleSort) in Notenfragment.java eingefügt
- Dass "null"-Setzen der Semester in eine andere Methode verschoben.
- kleinere Designanpassungen

5.6 TESTPROTOKOLLENTWURF

Testprotokoll

Projekt: HFTL-APP

Testobjekt: *Dateiname/Pfad (inkl. Versionsnummer)*

Testumgebung:

Datum:

Tester:

Testfall 1:	<i>Kurze Beschreibung was getestet wird</i>
Erwartetes Ergebnis:	<i>Beschreibung welches Ergebnis erwartet wird</i>
Tatsächliches Ergebnis:	<i>Beschreibung welches Ergebnis tatsächlich erhalten wurde</i>
Testergebnis:	<i>Stimmt das tatsächliche Ergebnis mit dem erwarteten Ergebnis überein?</i>
Aufgetretene Probleme:	<i>Welche Komplikationen sind aufgetreten?</i>
Bewertung:	<i>Aussage, ob der Test als erfolgreich gewertet werden kann. Wenn nicht, dann muss dies detailliert dargelegt werden. Insbesondere die signifikanten Abweichungen sind hervorzuheben.</i>

Testfall 2:	<i>Kurze Beschreibung was getestet wird</i>
Erwartetes Ergebnis:	<i>Beschreibung welches Ergebnis erwartet wird</i>
Tatsächliches Ergebnis:	<i>Beschreibung welches Ergebnis tatsächlich erhalten wurde</i>
Testergebnis:	<i>Stimmt das tatsächliche Ergebnis mit dem erwarteten Ergebnis überein?</i>

Aufgetretene Probleme:

Welche Komplikationen sind aufgetreten?

Bewertung:

Aussage, ob der Test als erfolgreich gewertet werden kann. Wenn nicht, dann muss dies detailliert dargelegt werden. Insbesondere die signifikanten Abweichungen sind hervorzuheben.

Protokollempfänger: Alle Projektteilnehmer

Datum: xx.xx.xxxx

5.7 TESTPROTOKOLLE

Gesamtprotokolle

Testprotokoll

Projekt: HFTL-APP
Testobjekt: *HfTL-App (Stand: 27.05.2015, 20:00Uhr)*
Testumgebung: HTC One Android 5.0.2
Datum: 27.05.2015
Tester: Maik Lorenz

Testfall 1:	<i>Test der App auf einem Smartphone im Echtzeitbetrieb</i>
Erwartetes Ergebnis:	<i>App startet und die Funktionen können getestet werden</i>
Tatsächliches Ergebnis:	<i>App konnte nicht gestartet werden, es war kein Funktionstest möglich</i>
Testergebnis:	<i>Es fand kein erfolgreicher Test statt</i>
Aufgetretene Probleme:	Execution failed for task ':app:mergeDebugResources'. > C:\..\App\HfTL-App\app\src\main\res\values\strings.xml:15:1: Error: Der Content von Elementen muss aus ordnungsgemäß formatierten Zeichendaten oder Markups bestehen. BUILD FAILED
Bewertung:	<i>Der Test der App im Echtzeitbetrieb konnte nicht durchgeführt werden. Fehler in der Programmierung erzeugen Fehler in der Erstellung einer lauffähigen App.</i>

Protokollempfänger: Alle Projektteilnehmer
Datum: 31.05.2015

Testprotokoll

Projekt: HFTL-APP
Testobjekt: HfTL-App (Stand: 31.05.2015, 14:30 Uhr)
Testumgebung: HTC One Android 5.0.2
Datum: 31.05.2015
Tester: Maik Lorenz

Testfall 1:	<i>Funktionstest News-Fragment und Formatierungsprüfung</i>
Erwartetes Ergebnis:	<i>Ausgabe der News als ListView nach Formatierungsvorgaben, Abruf der News und lesen der Inhalte</i>
Tatsächliches Ergebnis:	<i>Ausgabe als ListView nach Vorlage, öffnen der einzelnen News nach anklicken</i>
Testergebnis:	<i>Erhofftes und eingetretenes Ergebnis sind weitestgehend gleich</i>
Aufgetretene Probleme:	<i>Anhänge der News werden nicht mit abgerufen, Links in einer News nicht anklickbar</i>
Bewertung:	<p><i>Der Test verlief weitestgehend erfolgreich. Formatierungsvorgaben wurden umgesetzt. Es gibt nur noch kleine Auffälligkeiten, welche die grundlegende Funktion jedoch nicht beeinflussen. Die Abfrage nach neuen News kann nicht bewertet werden, da die Funktion des Aktualisierungsbuttons nicht „sichtbar“ ist.</i></p> <p><i>Aufgefallene Punkte:</i></p> <ul style="list-style-type: none"> - Es werden nicht alle News geladen - Links in einer News nicht anklickbar - Bilder und weiterführende Links einer News werden nicht mit ausgegeben - „Überschrift“ in der News ist „← Termindetails“ - Aktualisierungsbutton könnte Hinweistext ausgeben

Testfall 2:	<i>Funktionstest Noten-Fragment ohne Benutzerdaten</i>
Erwartetes Ergebnis:	<i>Fehlermeldung, da keine Benutzerdaten eingetragen</i>
Tatsächliches Ergebnis:	<i>Ausgabe der Fehlermeldung mit Option die Daten nachzutragen.</i>
Testergebnis:	<i>Erwartetes Ergebnis und tatsächliches Ergebnis stimmen überein. Das tatsächliche Ergebnis übertrifft die Erwartungen.</i>
Aufgetretene Probleme:	<i>Keine.</i>
Bewertung:	<p><i>Erwartet wurde eine einfache Fehlermeldung, dass keine Daten eingetragen sind. Der Nutzer erhält eine Auswahloption die Daten sofort oder später einzutragen. Beide Optionen wurden getestet und funktionieren problemlos. Bricht man die Abfrage ab, kommt eine leere Seite, doch mit einem Klick auf „Aktualisieren“ kann die Abfrage neu „gestartet“ werden.</i></p> <p><i>Verbesserungsvorschläge:</i></p> <ul style="list-style-type: none"> - <i>Nach abbrechen der Abfrage erscheint eine leere Seite. Zusätzlich zum Aktualisierungsbutton könnte noch ein Hinweistext oder ein Button ähnlich der Formatierungsvorlage erscheinen.</i>

Testfall 3:	<i>Funktionstest Noten-Fragment mit Benutzerdaten</i>
Erwartetes Ergebnis:	<i>Gelistete Ausgabe der Noten, angepasst an Formatierungsvorgaben</i>
Tatsächliches Ergebnis:	<i>Gelistete Ausgabe der Fächer und Noten ohne Sortierung und Formatierungen</i>
Testergebnis:	<i>Das erwartete Ergebnis wurde zu Hälfte erfüllt.</i>
Aufgetretene Probleme:	<i>Keine.</i>
Bewertung:	<i>Nach Eingabe der Nutzerdaten lädt die App die Notenübersicht aus QIS problemlos. Formatierungen müssen noch angepasst werden, die Funktion ist jedoch erkennbar.</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 31.05.2015

Testprotokolle zum News Modul

Testprotokoll News Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_2.apk
Prototyp-Version: 2
Protokoll-ID: 11
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 10.06.2015
Tester: PK

Funktionstest:	<i>Funktion des News Modul testen</i>
Vorschau News:	<i>Wird eine News-Vorschau angezeigt und ist die Formatierung usw. OK?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige News:	<i>Wird die News vollständig angezeigt und ist die Formatierung usw. OK?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Aktualisierung:	<i>Lassen sich die News aktualisieren?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anhänge und Links:	<i>Werden anhänge und Links innerhalb der News angezeigt und funktionieren diese auch?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Kein Feedback bei aktualisieren der News.</i> <i>Überschriften der News werden in der Vorschau nicht angezeigt, nur das Datum ist groß als Überschrift zu sehen.</i>

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Wenn die Datenverbindung getrennt ist stürzt die App ab.</i>
Fehler reproduzierbar:	<i>Lässt sich zu 100% reproduzieren</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 10.06.2015

Testprotokoll News Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_3.apk
Prototyp-Version: 3
Protokoll-ID: 15
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 23.06.2015
Tester: PK

Funktionstest:	<i>Funktion des News Modul testen</i>
Vorschau News:	<i>Wird eine News-Vorschau angezeigt und ist die Formatierung usw. OK?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige News:	<i>Wird die News vollständig angezeigt und ist die Formatierung usw. OK?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Aktualisierung:	<i>Lassen sich die News aktualisieren?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anhänge und Links:	<i>Werden anhänge und Links innerhalb der News angezeigt und funktionieren diese auch?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Kein Feedback bei aktualisieren der News.</i> <i>Überschriften der News werden in der Vorschau nicht angezeigt, nur das Datum ist groß als Überschrift zu sehen</i>

Keine Änderungen zum letzten Test.

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Wenn die Datenverbindung getrennt ist stürzt die App ab.</i>
Fehler reproduzierbar:	<i>Lässt sich zu 100% reproduzieren</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 23.06.2015

Testprotokoll News Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_4.apk
Prototyp-Version: 4
Protokoll-ID: 20
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 05.07.2015
Tester: PK

Funktionstest:	<i>Funktion des News Modul testen</i>
Vorschau News:	<i>Wird eine News-Vorschau angezeigt und ist die Formatierung usw. OK?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige News:	<i>Wird die News vollständig angezeigt und ist die Formatierung usw. OK?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Aktualisierung:	<i>Lassen sich die News aktualisieren?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anhänge und Links:	<i>Werden anhänge und Links innerhalb der News angezeigt und funktionieren diese auch?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Überschriften werden noch immer nicht angezeigt. Im Fenster in dem die Komplette News angezeigt wird steht oben Termindetails, dass muss geändert werden.</i>

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Die App stürzt beim laden der vollständigen News ohne Fehlermeldung ab. Die News Vorschau wird angezeigt. Gründe sind möglicherweise Änderungen an der HFTL Homepage</i>
Fehler reproduzierbar:	<i>Lässt sich zu 100% reproduzieren</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 05.07.2015

Testprotokoll News Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_5.apk
Prototyp-Version: 5
Protokoll-ID: 23
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 14.07.2015
Tester: PK

Funktionstest:

Funktion des News Modul testen

Vorschau News:

Wird eine News-Vorschau angezeigt und ist die Formatierung usw. OK?

☐ ja ☒ nein ☐ nicht bewertet

Anzeige News:

Wird die News vollständig angezeigt und ist die Formatierung usw. OK?

☐ ja ☒ nein ☐ nicht bewertet

Aktualisierung:

Lassen sich die News aktualisieren?

☐ ja ☒ nein ☐ nicht bewertet

Anhänge und Links:

Werden anhänge und Links innerhalb der News angezeigt und funktionieren diese auch?

☐ ja ☒ nein ☐ nicht bewertet

Sonstiges:

Anmerkungen zum Modul

Hinweis:

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>News werden nicht geladen oder nicht dargestellt</i>
Fehlerursache:	<i>unbekannt</i>
Fehler reproduzierbar:	<i>Lässt sich zu 100% reproduzieren</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 14.07.2015

Testprotokoll News Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_6.apk
Prototyp-Version: 6
Protokoll-ID: 42
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio.
 Samsung Galaxie S3 mit Android 4.3
Datum: 19.08.2015
Tester: PK

Funktionstest:

Funktion des News Modul testen

Vorschau News:

Wird eine News-Vorschau angezeigt und ist die Formatierung usw. OK?

☒ ja ☐ nein ☐ nicht bewertet

Anzeige News:

Wird die News vollständig angezeigt und ist die Formatierung usw. OK?

☒ ja ☐ nein ☐ nicht bewertet

Aktualisierung:

Lassen sich die News aktualisieren?

☒ ja ☐ nein ☐ nicht bewertet

Anhänge und Links:

Werden anhänge und Links innerhalb der News angezeigt und funktionieren diese auch?

☐ ja ☒ nein ☐ nicht bewertet

Sonstiges:

Anmerkungen zum Modul

Hinweis:

Werden z.B. gerade andere Aktionen der App ausgeführt wie z.B. das Aktualisieren der Noten so dauert das laden der News mitunter recht lange. Es gibt dabei kein Feedback und der Nutzer könnte denken die App sei eingefroren. Eventuell sollte man wie beim Stundenplan einen

Ladebildschirm einbauen, damit keine ungewollten eingaben oder Bildschirm Wechsel erfolgen, die die App zum Absturz bringen können. Das laden ist nach ein paar Sekunden erledigt und somit sollte dies keine Einschränkung auf die Benutzerfreundlichkeit haben.

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Bei anstoßen des Ladens der Noten und dem sofortigen Wechsel in die News. Siehe Logcat Auszug</i>
Fehler reproduzierbar:	<i>Lässt sich sporadisch reproduzieren.</i>
Kritikalität:	<i>5/10 mittlerer Fehler</i>

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Beim Laden der vollständigen News. Ursache ist nicht erkennbar</i>
Fehler reproduzierbar:	<i>Fehler tritt selten auf.</i>
Kritikalität:	<i>5/10 mittlerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 19.08.2015

Testprotokoll News Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_6.apk
Prototyp-Version: 0.6
Protokoll-ID: 40
Testumgebung: Sony Xperia Z3 Android 5.1.1
Datum: 24.08.2015
Tester: JS

Funktionstest:	<i>Funktion des News Modul testen</i>
Vorschau News:	<i>Wird eine News-Vorschau angezeigt und ist die Formatierung usw. OK?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige News:	<i>Wird die News vollständig angezeigt und ist die Formatierung usw. OK?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Aktualisierung:	<i>Lassen sich die News aktualisieren?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anhänge und Links:	<i>Werden anhänge und Links innerhalb der News angezeigt und funktionieren diese auch?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Sonstiges:	<i>Links können nicht direkt aufgerufen werden</i>
Hinweis:	<i>Ist meiner Meinung nach aber zu vernachlässigen</i>

Fehlermeldungen:

Fehler: *Links können nicht direkt aufgerufen werden*

Fehlerursache: *Links werden beim Parsen wahrscheinlich nicht richtig deklariert*

Fehler reproduzierbar: *ja*

Kritikalität: *2*

Fehlermeldungen: *Fehlermeldungen protokollieren*

Fehler: *Beschreibung des Fehlers*

Fehlerursache: *Bei welchen eingaben /Bedienung ist der Fehler aufgetreten*

Fehler reproduzierbar: *Lässt sich der Fehler zuverlässig reproduzieren?*

Kritikalität: *Wie schwer ist der Fehler auf einer Skala von 1-10*

Protokollempfänger: Alle Projektteilnehmer

Datum: 24.08.2015

Testprotokolle zum Noten Modul

Testprotokoll Noten Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_2.apk
Prototyp-Version: 2
Protokoll-ID: 12
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 10.06.2015
Tester: PK

Funktionstest:*Funktion des Noten Modul testen***Abfrage:***Funktioniert die Abfrage der Noten fehlerfrei?*☒ ja ☐ nein ☐ nicht bewertet**Intervallabfrage:***Funktioniert das Abfragen der Noten mit einem voreigestellten Intervall?*☐ ja ☒ nein ☐ nicht bewertet**Speicherung der Daten:***Werden die Noten auf dem Gerät gespeichert?*☐ ja ☒ nein ☐ nicht bewertet**Benachrichtigung:***Funktionieren die Benachrichtigungen bei Änderungen?*☐ ja ☒ nein ☐ nicht bewertet**Datenübertragung:***Werden die Benutzerdaten verschlüsselt übertragen oder im Klartext?*☐ ja ☒ nein ☐ nicht bewertet**Formatierung:***Ist das Layout in Ordnung? (wird alles Lesbar dargestellt)*☐ ja ☐ nein ☒ nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Eingaben:	<p><i>Können Benutzername und Passwort eingegeben werden ohne Fehler zu erzeugen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Intervallabfrage:	<p><i>Lässt sich ein Abfrageintervall einstellen?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Benachrichtigungen:	<p><i>Lässt sich einstellen ob man bei Änderungen benachrichtigt werden möchte?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Länge Eingabefelder:	<p><i>Ist die Länge der Eingabefelder ausreichend?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Erlaubter Zeichensatz:	<p><i>Sind die erlaubten Zeichen ausreichend um z.B. ein sicheres Passwort zu garantieren?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Sonderzeichen:	<p><i>Sind Sonderzeichen erlaubt? Wenn ja welche und werden diese fehlerfrei akzeptiert?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p>Wenn ja: Es werden alle Sonderzeichen akzeptiert, unabhängig davon ob diese den Passwortrichtlinien von QIS entsprechen</p>
Zu kurze Eingaben:	<p><i>Was ist bei kurzen bis sehr kurzen Eingaben? (z.B. nur ein Zeichen oder keines)</i></p> <p>Bemerkung: Kein Zeichen ist nicht möglich. Eingaben mit nur einem Zeichen werden akzeptiert.</p>
Zu lange Eingaben:	<p><i>Was ist bei sehr langen Eingaben? (größer als der Variablenbereich)</i></p> <p>Bemerkung: Es gibt keine Begrenzung der Eingabefelder</p>
Nicht unterstützte Zeichen / Sonderzeichen:	<p><i>Was passiert bei der Verwendung von nicht unterstützten Zeichen und Sonderzeichen?</i></p> <p>Bemerkung: soweit getestet werden alle Zeichen und Sonderzeichen akzeptiert</p>
Speicherung:	<i>Funktioniert das Speichern der Benutzerdaten fehlerfrei?</i>

Verschlüsselung:	<input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige Benutzerdaten:	<p><i>Werden Benutzername und Passwort verschlüsselt gespeichert?</i></p> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Funktioniert die Verschlüsselung?</i></p> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Wird das Passwort in den Einstellungen im Klartext angezeigt?</i></p> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Wird der Benutzername in den Einstellungen im Klartext angezeigt?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Das Layout ist noch nicht final, deswegen auch nicht bewertet. Formatierung stimmt noch nicht und auch die Reihenfolge der Noten ist weder geordnet noch lässt sie sich ordnen</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab.</i>
Fehlerursache:	<i>Wenn keine datenverbindung vorhanden ist stürzt die app ab</i>
Fehler reproduzierbar:	<i>Zu 100% reproduzierbar</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Nicht zu erkennen. Ausgelöst vermutlich beim Wechsel zwischen den Seiten.</i>
Fehler reproduzierbar:	<i>Fehler tritt unregelmäßig beim aktualisieren oder anzeigen der Noten auf.</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 10.06.2015

Testprotokoll Noten Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_3.apk
Prototyp-Version: 3
Protokoll-ID: 16
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 23.06.2015
Tester: PK

Funktionstest:

Funktion des Noten Modul testen

Abfrage:

Funktioniert die Abfrage der Noten fehlerfrei?

☒ ja ☐ nein ☐ nicht bewertet

Intervallabfrage:

Funktioniert das Abfragen der Noten mit einem voreigestellten Intervall?

☐ ja ☒ nein ☐ nicht bewertet

Speicherung der Daten:

Werden die Noten auf dem Gerät gespeichert?

☒ ja ☐ nein ☐ nicht bewertet

Benachrichtigung:

Funktionieren die Benachrichtigungen bei Änderungen?

☐ ja ☒ nein ☐ nicht bewertet

Datenübertragung:

Werden die Benutzerdaten verschlüsselt übertragen?

☒ ja ☐ nein ☐ nicht bewertet

Formatierung:

Ist das Layout in Ordnung? (wird alles Lesbar dargestellt)

☐ ja ☒ nein ☐ nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Eingaben:	<p><i>Können Benutzername und Passwort eingegeben werden ohne Fehler zu erzeugen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Intervallabfrage:	<p><i>Lässt sich ein Abfrageintervall einstellen?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Benachrichtigungen:	<p><i>Lässt sich einstellen ob man bei Änderungen benachrichtigt werden möchte?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Länge Eingabefelder:	<p><i>Ist die Länge der Eingabefelder ausreichend?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Erlaubter Zeichensatz:	<p><i>Sind die erlaubten Zeichen ausreichend um z.B. ein sicheres Passwort zu garantieren?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Sonderzeichen:	<p><i>Sind Sonderzeichen erlaubt? Wenn ja welche und werden diese fehlerfrei akzeptiert?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p>Wenn ja: Es werden alle Sonderzeichen akzeptiert, unabhängig davon ob diese den Passwortrichtlinien von QIS entsprechen</p>
Zu kurze Eingaben:	<p><i>Was ist bei kurzen bis sehr kurzen Eingaben? (z.B. nur ein Zeichen oder keines)</i></p> <p>Bemerkung: Kein Zeichen ist nicht möglich. Eingaben mit nur einem Zeichen werden akzeptiert.</p>
Zu lange Eingaben:	<p><i>Was ist bei sehr langen Eingaben? (größer als der Variablenbereich)</i></p> <p>Bemerkung: Es gibt keine Begrenzung der Eingabefelder</p>
Nicht unterstützte Zeichen / Sonderzeichen:	<p><i>Was passiert bei der Verwendung von nicht unterstützten Zeichen und Sonderzeichen?</i></p> <p>Bemerkung: soweit getestet werden alle Zeichen und Sonderzeichen akzeptiert</p>
Speicherung:	<i>Funktioniert das Speichern der Benutzerdaten fehlerfrei?</i>

	<input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Verschlüsselung:	<p><i>Werden Benutzername und Passwort verschlüsselt gespeichert?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Funktioniert die Verschlüsselung?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige Benutzerdaten:	<p><i>Wird das Passwort in den Einstellungen im Klartext angezeigt?</i></p> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Wird der Benutzername in den Einstellungen im Klartext angezeigt?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Ausgabe der Noten unsortiert als Liste. Sortierung der Noten nicht möglich.</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Nicht zu erkennen. Ausgelöst vermutlich beim Wechsel zwischen den Seiten.</i>
Fehler reproduzierbar:	<i>Fehler tritt unregelmäßig beim aktualisieren oder anzeigen der Noten auf.</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Fehlermeldungen:	Fehlermeldungen protokollieren
Fehler:	App stürzt ab oder gibt falsche Fehlermeldung aus
Fehlerursache:	Bricht die Datenverbindung beim Anzeigen oder aktualisieren der Noten ab stürzt die App entweder ab oder es kommt die Fehlermeldung „Benutzername und Passwort falsch“
Fehler reproduzierbar:	Fehler zu 100% reproduzierbar
Kritikalität:	10/10 schwerer Fehler

Protokollempfänger: Alle Projektteilnehmer

Datum: 23.06.2015

Testprotokoll Noten Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_4.apk
Prototyp-Version: 4
Protokoll-ID: 21
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 05.07.2015
Tester: PK

Funktionstest:

Funktion des Noten Modul testen

Abfrage:

Funktioniert die Abfrage der Noten fehlerfrei?

☒ ja ☐ nein ☐ nicht bewertet

Intervallabfrage:

Funktioniert das Abfragen der Noten mit einem voreigestellten Intervall?

☐ ja ☒ nein ☐ nicht bewertet

Speicherung der Daten:

Werden die Noten auf dem Gerät gespeichert?

☒ ja ☐ nein ☐ nicht bewertet

Benachrichtigung:

Funktionieren die Benachrichtigungen bei Änderungen?

☐ ja ☒ nein ☐ nicht bewertet

Datenübertragung:

Werden die Benutzerdaten verschlüsselt übertragen oder im Klartext?

☒ ja ☐ nein ☐ nicht bewertet

Formatierung:

Ist das Layout in Ordnung? (wird alles Lesbar dargestellt)

☒ ja ☐ nein ☐ nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Eingaben:	<p><i>Können Benutzername und Passwort eingegeben werden ohne Fehler zu erzeugen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Intervallabfrage:	<p><i>Lässt sich ein Abfrageintervall einstellen?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Benachrichtigungen:	<p><i>Lässt sich einstellen ob man bei Änderungen benachrichtigt werden möchte?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Länge Eingabefelder:	<p><i>Ist die Länge der Eingabefelder ausreichend?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Erlaubter Zeichensatz:	<p><i>Sind die erlaubten Zeichen ausreichend um z.B. ein sicheres Passwort zu garantieren?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Sonderzeichen:	<p><i>Sind Sonderzeichen erlaubt? Wenn ja welche und werden diese fehlerfrei akzeptiert?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p>Wenn ja: Es werden alle Sonderzeichen akzeptiert, unabhängig davon ob diese den Passwortrichtlinien von QIS entsprechen</p>
Zu kurze Eingaben:	<p><i>Was ist bei kurzen bis sehr kurzen Eingaben? (z.B. nur ein Zeichen oder keines)</i></p> <p>Bemerkung: Kein Zeichen ist nicht möglich. Eingaben mit nur einem Zeichen werden akzeptiert.</p>
Zu lange Eingaben:	<p><i>Was ist bei sehr langen Eingaben? (größer als der Variablenbereich)</i></p> <p>Bemerkung: Es gibt keine Begrenzung der Eingabefelder</p>
Nicht unterstützte Zeichen / Sonderzeichen:	<p><i>Was passiert bei der Verwendung von nicht unterstützten Zeichen und Sonderzeichen?</i></p> <p>Bemerkung: soweit getestet werden alle Zeichen und Sonderzeichen akzeptiert</p>
Speicherung:	<i>Funktioniert das Speichern der Benutzerdaten fehlerfrei?</i>

<p>Verschlüsselung:</p>	<p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
<p>Anzeige Benutzerdaten:</p>	<p><i>Werden Benutzername und Passwort verschlüsselt gespeichert?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p><i>Funktioniert die Verschlüsselung?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p><i>Wird das Passwort in den Einstellungen im Klartext angezeigt?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p><i>Wird der Benutzername in den Einstellungen im Klartext angezeigt?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
<p>Sonstiges:</p> <p>Hinweis:</p>	<p><i>Anmerkungen zum Modul</i></p> <p><i>Die Reihenfolge der Semester ist ungeordnet, was die Übersichtlichkeit stört. Formatierungen sind noch nicht Final aber werden korrekt dargestellt.</i></p>
<p>Fehlermeldungen:</p> <p>Fehler:</p> <p>Fehlerursache:</p> <p>Fehler reproduzierbar:</p> <p>Kritikalität:</p>	<p><i>Fehlermeldungen protokollieren</i></p> <p><i>Die App gibt eine falsche Fehlermeldung aus</i></p> <p><i>Wenn die Datenverbindung beim laden der Noten abbricht wird die Fehlermeldung „Benutzername/Passwort falsch!“ angezeigt.</i></p> <p><i>Zu 100% reproduzierbar</i></p> <p><i>5/10 mittlerer Fehler</i></p>

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Nicht zu erkennen. Ausgelöst vermutlich beim Wechsel zwischen den Seiten.</i>
Fehler reproduzierbar:	<i>Fehler tritt unregelmäßig beim aktualisieren oder anzeigen der Noten auf.</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 05.07.2015

Testprotokoll Noten Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_5.apk
Prototyp-Version: 5
Protokoll-ID: 24
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 14.07.2015
Tester: PK

Funktionstest:

Funktion des Noten Modul testen

Abfrage:

Funktioniert die Abfrage der Noten fehlerfrei?

☒ ja ☐ nein ☐ nicht bewertet

Intervallabfrage:

Funktioniert das Abfragen der Noten mit einem voreigestellten Intervall?

☐ ja ☐ nein ☒ nicht bewertet

Speicherung der Daten:

Werden die Noten auf dem Gerät gespeichert?

☒ ja ☐ nein ☐ nicht bewertet

Benachrichtigung:

Funktionieren die Benachrichtigungen bei Änderungen?

☐ ja ☐ nein ☒ nicht bewertet

Datenübertragung:

Werden die Benutzerdaten verschlüsselt übertragen oder im Klartext?

☒ ja ☐ nein ☐ nicht bewertet

Formatierung:

Ist das Layout in Ordnung? (wird alles Lesbar dargestellt)

☒ ja ☐ nein ☐ nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Eingaben:	<p><i>Können Benutzername und Passwort eingegeben werden ohne Fehler zu erzeugen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Intervallabfrage:	<p><i>Lässt sich ein Abfrageintervall einstellen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Benachrichtigungen:	<p><i>Lässt sich einstellen ob man bei Änderungen benachrichtigt werden möchte?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Länge Eingabefelder:	<p><i>Ist die Länge der Eingabefelder ausreichend?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Erlaubter Zeichensatz:	<p><i>Sind die erlaubten Zeichen ausreichend um z.B. ein sicheres Passwort zu garantieren?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Sonderzeichen:	<p><i>Sind Sonderzeichen erlaubt? Wenn ja welche und werden diese fehlerfrei akzeptiert?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p>Wenn ja: Es werden alle Sonderzeichen akzeptiert, unabhängig davon ob diese den Passwortrichtlinien von QIS entsprechen</p>
Zu kurze Eingaben:	<p><i>Was ist bei kurzen bis sehr kurzen Eingaben? (z.B. nur ein Zeichen oder keines)</i></p> <p>Bemerkung: Kein Zeichen ist nicht möglich. Eingaben mit nur einem Zeichen werden akzeptiert.</p>
Zu lange Eingaben:	<p><i>Was ist bei sehr langen Eingaben? (größer als der Variablenbereich)</i></p> <p>Bemerkung: Es gibt keine Begrenzung der Eingabefelder</p>
Nicht unterstützte Zeichen / Sonderzeichen:	<p><i>Was passiert bei der Verwendung von nicht unterstützten Zeichen und Sonderzeichen?</i></p> <p>Bemerkung: soweit getestet werden alle Zeichen und Sonderzeichen akzeptiert</p> <p><i>Funktioniert das Speichern der Benutzerdaten fehlerfrei?</i></p>

Speicherung:	<input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Verschlüsselung:	<p><i>Werden Benutzername und Passwort verschlüsselt gespeichert?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Funktioniert die Verschlüsselung?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige Benutzerdaten:	<p><i>Wird das Passwort in den Einstellungen im Klartext angezeigt?</i></p> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Wird der Benutzername in den Einstellungen im Klartext angezeigt?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<p><i>Die Benachrichtigung bei Änderungen der Noten konnte nicht getestet werden da keine Möglichkeit besteht diese in QIS anzupassen.</i></p> <p><i>Die Noten könnten noch nach Semester sortierbar gemacht werden</i></p>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App zeigt falsche Fehlermeldung an</i>
Fehlerursache:	<i>Wenn die Datenverbindung beim laden der Noten abbricht wird als Fehlermeldung „Benutzername/Passwort falsch“ angezeigt.</i>
Fehler reproduzierbar:	<i>Zu 100% reproduzierbar</i>
Kritikalität:	<i>5/10 mittlerer Fehler</i>

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Nicht zu erkennen. Ausgelöst vermutlich beim Wechsel zwischen den Seiten.</i>
Fehler reproduzierbar:	<i>Fehler tritt unregelmäßig beim aktualisieren oder anzeigen der Noten auf.</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 10.06.2015

Testprotokoll Noten Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_6.apk
Prototyp-Version: 6
Protokoll-ID: 43
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio.
 Samsung Galaxie S3 mit Android 4.3
Datum: 19.08.2015
Tester: PK

Funktionstest:

Funktion des Noten Modul testen

Abfrage:

Funktioniert die Abfrage der Noten fehlerfrei?

☒ ja ☐ nein ☐ nicht bewertet

Intervallabfrage:

Funktioniert das Abfragen der Noten mit einem voreigestellten Intervall?

☒ ja ☐ nein ☐ nicht bewertet

Speicherung der Daten:

Werden die Noten auf dem Gerät gespeichert?

☒ ja ☐ nein ☐ nicht bewertet

Benachrichtigung:

Funktionieren die Benachrichtigungen bei Änderungen?

☐ ja ☐ nein ☒ nicht bewertet

Datenübertragung:

Werden die Benutzerdaten verschlüsselt übertragen oder im Klartext?

☒ ja ☐ nein ☐ nicht bewertet

Formatierung:

Ist das Layout in Ordnung? (wird alles Lesbar dargestellt)

☒ ja ☐ nein ☐ nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Eingaben:	<p><i>Können Benutzername und Passwort eingegeben werden ohne Fehler zu erzeugen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Intervallabfrage:	<p><i>Lässt sich ein Abfrageintervall einstellen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Benachrichtigungen:	<p><i>Lässt sich einstellen ob man bei Änderungen benachrichtigt werden möchte?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Länge Eingabefelder:	<p><i>Ist die Länge der Eingabefelder ausreichend?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Erlaubter Zeichensatz:	<p><i>Sind die erlaubten Zeichen ausreichend um z.B. ein sicheres Passwort zu garantieren?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Sonderzeichen:	<p><i>Sind Sonderzeichen erlaubt? Wenn ja welche und werden diese fehlerfrei akzeptiert?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p>Wenn ja: Es werden alle Sonderzeichen akzeptiert, unabhängig davon ob diese den Passwortrichtlinien von QIS entsprechen</p>
Zu kurze Eingaben:	<p><i>Was ist bei kurzen bis sehr kurzen Eingaben? (z.B. nur ein Zeichen oder keines)</i></p> <p>Bemerkung: Kein Zeichen ist nicht möglich. Eingaben mit nur einem Zeichen werden akzeptiert.</p>
Zu lange Eingaben:	<p><i>Was ist bei sehr langen Eingaben? (größer als der Variablenbereich)</i></p> <p>Bemerkung: Es gibt keine Begrenzung der Eingabefelder</p>
Nicht unterstützte Zeichen / Sonderzeichen:	<p><i>Was passiert bei der Verwendung von nicht unterstützten Zeichen und Sonderzeichen?</i></p> <p>Bemerkung: soweit getestet werden alle Zeichen und Sonderzeichen akzeptiert</p> <p><i>Funktioniert das Speichern der Benutzerdaten fehlerfrei?</i></p>

Speicherung:	<input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Verschlüsselung:	<p><i>Werden Benutzername und Passwort verschlüsselt gespeichert?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Anzeige Benutzerdaten:	<p><i>Funktioniert die Verschlüsselung?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Wird das Passwort in den Einstellungen im Klartext angezeigt?</i></p> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
	<p><i>Wird der Benutzername in den Einstellungen im Klartext angezeigt?</i></p> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Die Benachrichtigung bei Änderungen der Noten konnte nicht getestet werden da keine Möglichkeit besteht diese in QIS anzupassen.</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App zeigt falsche Fehlermeldung an</i>
Fehlerursache:	<i>Wenn die Datenverbindung beim laden der Noten abbricht wird als Fehlermeldung „Benutzername/Passwort falsch“ angezeigt.</i>
Fehler reproduzierbar:	<i>Zu 100% reproduzierbar wenn die Datenverbindung beim übertragen der Benutzerdaten zu QIS abbricht.</i>
Kritikalität:	<i>3/10 leichter Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 19.08.2015

Testprotokoll Noten Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_6.apk
Prototyp-Version: 0.6
Protokoll-ID: 41
Testumgebung: Sony Xperia Z3 Android 5.1.1
Datum: 24.08.2015
Tester: JS

Funktionstest:	<i>Funktion des Noten Modul testen</i>
Abfrage:	<i>Funktioniert die Abfrage der Noten fehlerfrei?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Funktioniert das Abfragen der Noten mit einem voreigestellten Intervall?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Speicherung der Daten:	<i>Werden die Noten auf dem Gerät gespeichert?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Benachrichtigung:	<i>Funktionieren die Benachrichtigungen bei Änderungen?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Datenübertragung:	<i>Werden die Benutzerdaten verschlüsselt übertragen oder im Klartext?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Formatierung:	<i>Ist das Layout in Ordnung? (wird alles Lesbar dargestellt)</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Eingaben:	<p><i>Können Benutzername und Passwort eingegeben werden ohne Fehler zu erzeugen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Intervallabfrage:	<p><i>Lässt sich ein Abfrageintervall einstellen?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Benachrichtigungen:	<p><i>Lässt sich einstellen ob man bei Änderungen benachrichtigt werden möchte?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Länge Eingabefelder:	<p><i>Ist die Länge der Eingabefelder ausreichend?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Erlaubter Zeichensatz:	<p><i>Sind die erlaubten Zeichen ausreichend um z.B. ein sicheres Passwort zu garantieren?</i></p> <p><input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet</p>
Sonderzeichen:	<p><i>Sind Sonderzeichen erlaubt? Wenn ja welche und werden diese fehlerfrei akzeptiert?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p>Wenn ja: #</p>
Zu kurze Eingaben:	<p><i>Was ist bei kurzen bis sehr kurzen Eingaben? (z.B. nur ein Zeichen oder keines)</i></p> <p>Bemerkung:</p>
Zu lange Eingaben:	<p><i>Was ist bei sehr langen Eingaben? (größer als der Variablenbereich)</i></p> <p>Bemerkung:</p>
Nicht unterstützte Zeichen / Sonderzeichen:	<p><i>Was passiert bei der Verwendung von nicht unterstützten Zeichen und Sonderzeichen?</i></p> <p>Bemerkung:</p>
Speicherung:	<p><i>Funktioniert das Speichern der Benutzerdaten fehlerfrei?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>

Verschlüsselung:	<p><i>Werden Benutzername und Passwort verschlüsselt gespeichert?</i></p> <p><input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet</p> <p><i>Funktioniert die Verschlüsselung?</i></p> <p><input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet</p>
Anzeige Benutzerdaten:	<p><i>Wird das Passworts in den Einstellungen im Klartext angezeigt?</i></p> <p><input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p> <p><i>Wird der Benutzername in den Einstellungen im Klartext angezeigt?</i></p> <p><input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet</p>
Sonstiges:	%
Hinweis:	%
Fehlermeldungen:	%
Fehler:	%
Fehlerursache:	%
Fehler reproduzierbar:	%
Kritikalität:	%
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Beschreibung des Fehlers</i>
Fehlerursache:	<i>Bei welchen eingaben /Bedienung ist der Fehler aufgetreten</i>
Fehler reproduzierbar:	<i>Lässt sich der Fehler zuverlässig reproduzieren?</i>

Kritikalität:

Wie schwer ist der Fehler auf einer Skala von 1-10

Protokollempfänger: Alle Projektteilnehmer

Datum: 24.08.2015

Testprotokolle zum Stundenplan Modul

Testprotokoll Stundenplan Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_2.apk
Prototyp-Version: 2
Protokoll-ID: 13
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 10.06.2015
Tester: PK

Funktionstest:	<i>Funktion des Stundenplan Modul testen</i>
Abfrage:	<i>Funktioniert die Abfrage des Stundenplans fehlerfrei?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Funktioniert das Abfragen des Stundenplans mit einem voreigestellten Intervall?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Synchronisierung:	<i>Funktioniert die Synchronisierung mit dem Kalender des Smartphones?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Studiengang:	<i>Werden nur zum Nutzer passende Stundenpläne angezeigt?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Formatierung:	<i>Ist das Layout in Ordnung? (wird alles lesbar dargestellt)</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Wochenwahl:	<i>Funktioniert die Anzeige verschiedener Wochen?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Lässt sich ein Abfrageintervall einstellen?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Benachrichtigungen:	<i>Funktionieren die Benachrichtigungen bei sich ändernden Stundenplänen usw.?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Modul noch nicht implementiert. Test deswegen nicht möglich</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Beschreibung des Fehlers</i>
Fehlerursache:	<i>Bei welchen eingaben /Bedienung ist der Fehler aufgetreten</i>
Fehler reproduzierbar:	<i>Lässt sich der Fehler zuverlässig reproduzieren?</i>
Kritikalität:	<i>Wie schwer ist der Fehler auf einer Skala von 1-10</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Beschreibung des Fehlers</i>
Fehlerursache:	<i>Bei welchen eingaben /Bedienung ist der Fehler aufgetreten</i>
Fehler reproduzierbar:	<i>Lässt sich der Fehler zuverlässig reproduzieren?</i>
Kritikalität:	<i>Wie schwer ist der Fehler auf einer Skala von 1-10</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 10.06.2015

Testprotokoll Stundenplan Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_3.apk
Prototyp-Version: 3
Protokoll-ID: 17
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 23.06.2015
Tester: PK

Funktionstest:	<i>Funktion des Stundenplan Modul testen</i>
Abfrage:	<i>Funktioniert die Abfrage des Stundenplans fehlerfrei?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Funktioniert das Abfragen des Stundenplans mit einem voreigestellten Intervall?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Synchronisierung:	<i>Funktioniert die Synchronisierung mit dem Kalender des Smartphones?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Studiengang:	<i>Werden nur zum Nutzer passende Stundenpläne angezeigt?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Formatierung:	<i>Ist das Layout in Ordnung? (wird alles lesbar dargestellt)</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Wochenwahl:	<i>Funktioniert die Anzeige verschiedener Wochen?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Lässt sich ein Abfrageintervall einstellen?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Benachrichtigungen:	<i>Funktionieren die Benachrichtigungen bei sich ändernden Stundenplänen usw.?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Stundenplan wird als unsortierte Liste ausgegeben.</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Beschreibung des Fehlers</i>
Fehlerursache:	<i>Bei welchen eingaben /Bedienung ist der Fehler aufgetreten</i>
Fehler reproduzierbar:	<i>Lässt sich der Fehler zuverlässig reproduzieren?</i>
Kritikalität:	<i>Wie schwer ist der Fehler auf einer Skala von 1-10</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Beschreibung des Fehlers</i>
Fehlerursache:	<i>Bei welchen eingaben /Bedienung ist der Fehler aufgetreten</i>
Fehler reproduzierbar:	<i>Lässt sich der Fehler zuverlässig reproduzieren?</i>
Kritikalität:	<i>Wie schwer ist der Fehler auf einer Skala von 1-10</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 23.06.2015

Testprotokoll Stundenplan Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_4.apk
Prototyp-Version: 4
Protokoll-ID: 22
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 05.07.2015
Tester: PK

Funktionstest:	<i>Funktion des Stundenplan Modul testen</i>
Abfrage:	<i>Funktioniert die Abfrage des Stundenplans fehlerfrei?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Funktioniert das Abfragen des Stundenplans mit einem voreigestellten Intervall?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Synchronisierung:	<i>Funktioniert die Synchronisierung mit dem Kalender des Smartphones?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Studiengang:	<i>Werden nur zum Nutzer passende Stundenpläne angezeigt?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Formatierung:	<i>Ist das Layout in Ordnung? (wird alles lesbar dargestellt)</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Wochenwahl:	<i>Funktioniert die Anzeige verschiedener Wochen?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Lässt sich ein Abfrageintervall einstellen?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Benachrichtigungen:	<i>Funktionieren die Benachrichtigungen bei sich ändernden Stundenplänen usw.?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>Das Layout ist noch nicht final aber die Formatierung ist ok Wird eine bestimmte Woche ausgewählt wird der entsprechende Stundenplan nicht automatisch geladen, sondern muss noch mit dem GO Knopf gestartet werden. Das automatische laden wäre hier sinnvoll</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Besteht keine Datenverbindung stürzt die App beim laden der Stundenpläne sofort ohne Fehlermeldung ab.</i>
Fehler reproduzierbar:	<i>Lässt sich zu 100% reproduzieren</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Bricht die Datenverbindung während des ladens von Stundenplänen ab so stürzt die App sofort ohne Fehlermeldung ab</i>
Fehler reproduzierbar:	<i>Lässt sich zu 100% reproduzieren</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 05.07.2015

Testprotokoll Stundenplan Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_5.apk
Prototyp-Version: 5
Protokoll-ID: 25
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio
Datum: 14.07.2015
Tester: PK

Funktionstest:	<i>Funktion des Stundenplan Modul testen</i>
Abfrage:	<i>Funktioniert die Abfrage des Stundenplans fehlerfrei?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Funktioniert das Abfragen des Stundenplans mit einem voreigestellten Intervall?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Synchronisierung:	<i>Funktioniert die Synchronisierung mit dem Kalender des Smartphones?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Studiengang:	<i>Werden nur zum Nutzer passende Stundenpläne angezeigt?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Formatierung:	<i>Ist das Layout in Ordnung? (wird alles lesbar dargestellt)</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Wochenwahl:	<i>Funktioniert die Anzeige verschiedener Wochen?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Lässt sich ein Abfrageintervall einstellen?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Benachrichtigungen:	<i>Funktionieren die Benachrichtigungen bei sich ändernden Stundenplänen usw.?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Bricht die Datenverbindung beim laden des Stundenplans ab stürzt die App ohne Fehlermeldung ab</i>
Fehler reproduzierbar:	<i>Zu 100% reproduzierbar</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Absturz der App</i>
Fehlerursache:	<i>Die App stürzt ab wenn in den Einstellungen kein Studiengang ausgewählt ist</i>
Fehler reproduzierbar:	<i>Fehler lässt sich zu 100% reproduzieren</i>
Kritikalität:	<i>Sehr schwerer Fehler 10/10</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 14.07.2015

Testprotokoll Stundenplan Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_6.apk
Prototyp-Version: 6
Protokoll-ID: 44
Testumgebung: Emuliertes Gerät Nexus 5 API 22 mit Android Studio.
 Samsung Galaxie S3 mit Android 4.3
Datum: 19.08.2015
Tester: PK

Funktionstest:

Funktion des Stundenplan Modul testen

Abfrage:

Funktioniert die Abfrage des Stundenplans fehlerfrei?

☒ ja ☐ nein ☐ nicht bewertet

Intervallabfrage:

Funktioniert das Abfragen des Stundenplans mit einem voreigestellten Intervall?

☐ ja ☒ nein ☐ nicht bewertet

Synchronisierung:

Funktioniert die Synchronisierung mit dem Kalender des Smartphones?

☒ ja ☐ nein ☐ nicht bewertet

Studiengang:

Werden nur zum Nutzer passende Stundenpläne angezeigt?

☒ ja ☐ nein ☐ nicht bewertet

Formatierung:

Ist das Layout in Ordnung? (wird alles lesbar dargestellt)

☒ ja ☐ nein ☐ nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Wochenwahl:	<i>Funktioniert die Anzeige verschiedener Wochen?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Lässt sich ein Abfrageintervall einstellen?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Benachrichtigungen:	<i>Funktionieren die Benachrichtigungen bei sich ändernden Stundenplänen usw.?</i> <input type="checkbox"/> ja <input checked="" type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	...
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>App stürzt ab</i>
Fehlerursache:	<i>Besteht bereits am Anfang keine oder bricht die Datenverbindung beim laden des Stundenplans ab stürzt die App ohne Fehlermeldung ab</i>
Fehler reproduzierbar:	<i>Zu 100% reproduzierbar</i>
Kritikalität:	<i>10/10 schwerer Fehler</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 19.08.2015

Testprotokoll Stundenplan Modul

Projekt: HFTL-APP
Testobjekt: HfTL-App\app\Versionen\HFTL-APP_V0_6.apk
Prototyp-Version: 0.6
Protokoll-ID: 39
Testumgebung: Sony Xperia Z3 Android 5.1.1
Datum: 24.08.15
Tester: JS

Funktionstest:	<i>Funktion des Stundenplan Modul testen</i>
Abfrage:	<i>Funktioniert die Abfrage des Stundenplans fehlerfrei?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Funktioniert das Abfragen des Stundenplans mit einem voreingestellten Intervall?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Synchronisierung:	<i>Funktioniert die Synchronisierung mit dem Kalender des Smartphones?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Studiengang:	<i>Werden nur zum Nutzer passende Stundenpläne angezeigt?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Formatierung:	<i>Ist das Layout in Ordnung? (wird alles lesbar dargestellt)</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet

Einstellungen:	<i>Test der Einstellungen des Moduls</i>
Wochenwahl:	<i>Funktioniert die Anzeige verschiedener Wochen?</i> <input checked="" type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht bewertet
Intervallabfrage:	<i>Lässt sich ein Abfrageintervall einstellen?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Benachrichtigungen:	<i>Funktionieren die Benachrichtigungen bei sich ändernden Stundenplänen usw.?</i> <input type="checkbox"/> ja <input type="checkbox"/> nein <input checked="" type="checkbox"/> nicht bewertet
Sonstiges:	<i>Anmerkungen zum Modul</i>
Hinweis:	<i>z.B. langsames Gerät beim nutzen des Moduls oder dergleichen</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Beschreibung des Fehlers</i>
Fehlerursache:	<i>Bei welchen eingaben /Bedienung ist der Fehler aufgetreten</i>
Fehler reproduzierbar:	<i>Lässt sich der Fehler zuverlässig reproduzieren?</i>
Kritikalität:	<i>Wie schwer ist der Fehler auf einer Skala von 1-10</i>
Fehlermeldungen:	<i>Fehlermeldungen protokollieren</i>
Fehler:	<i>Beschreibung des Fehlers</i>
Fehlerursache:	<i>Bei welchen eingaben /Bedienung ist der Fehler aufgetreten</i>
Fehler reproduzierbar:	<i>Lässt sich der Fehler zuverlässig reproduzieren?</i>
Kritikalität:	<i>Wie schwer ist der Fehler auf einer Skala von 1-10</i>

Protokollempfänger: Alle Projektteilnehmer

Datum: 24.08.2015