



Vertretungsplan-App

Informatik-Projekt am St. Michael Gymnasium Ahlen

Inhaltsverzeichnis

1	Einführung	1
2	Aller Anfang ist mühsam	2
3	Das Vertretungsplan-Projekt	2
3.1	Die Vorbereitungen	2
3.2	Der Aufbau des Projektes	3
3.3	Eine neue Version veröffentlichen	4
4	Das Projekt weitergeben	6
5	Mitwirkende	6
5.1	Schüler	6
5.2	Lehrer	6

1 Einführung

Hey du. Wie ich höre hast du vor die Vertretungsplan-App zu begleiten. Lass dir gesagt sein, dass dieses Projekt andere Qualitäten abverlangt als der “normale” Informatik-Unterricht es tut. Aufgrund dessen sind auch einige wenige Voraussetzungen zu erfüllen, da du sonst keinen Spaß hast dieses Projekt zu begleiten. Du musst Spaß am Programmieren haben, so hat es bei mir angefangen. Jedoch musste ich schnell feststellen, dass eine App zu schreiben nicht nur das Programmieren beinhaltet, denn die Plattform, also das Android-Betriebssystem, bringt einige Voraussetzungen, in die man sich einlesen muss. Auch Kreativität ist wünschenswert, denn gelegentlich darf man auch mal ein Bild erstellen oder sich generell das Layout der App überlegen. Die Hardware-technischen Voraussetzungen sind leider auch gegeben. Ein Android-Handy ist von großem Vorteil, aber nicht zwingend notwendig, da man dieses auch auf dem Computer simulieren kann. So kommen wir zum nächsten, Zugriff zu einem Computer ist unabdingbar und nicht die Schulrechner, auf denen man nichts installieren kann, sondern am Besten ein eigener zu Hause. Wo ich gerade von zu Hause spreche, dies ist sicherlich eine perfekte Gruppenarbeit, denn das Einarbeiten in die Android-Programmierung und eventuell auch in (meine) Vertretungsplan-App dürfte in einer Gruppe deutlich unterhaltsamer und eventuell auch schneller von Statten gehen, wenn man sich gegenseitig dabei hilft.

Was bringt einem dieses Projekt denn eigentlich, magst du dich fragen. Zum einen hilfst du deinen Mitschülern, welche die App benutzen, zum anderen wirst du einiges über die Android-Programmierung erfahren, aber auch in einem gewissen Maße, wie ein Android-Handy funktioniert. Auch wirst du erfahren, wie man solch ein Projekt mit Git organisiert.

Nochmal: Gruppenarbeit ist sehr wünschenswert :)

2 Aller Anfang ist mühsam

Du möchtest dich jetzt erst mal langsam an Android antasten, was völlig verständlich ist. Dazu gibt es hier eine ausführliche Anleitung: <https://developer.android.com/training/basics/firstapp/index.html>. Als Programmierumgebung empfehle ich Android Studio, was seit 2015 die Standardumgebung für Android-Programmierung ist. In der Anleitung wird auch erklärt, wie man die Android SDK herunterlädt, außerdem wirst du beim Erstellen deiner ersten App begleitet. Bei Problemen einfach googlen oder auf <http://www.stackoverflow.com> suchen, welches eine sehr gute Hilfe ist. Damit hast du dann schon deine erste Android-App programmiert. Auch wenn du jetzt etwas erschlagen bist, so hast du doch bis jetzt sehr viel erreicht. Wenn du hierfür etwas länger brauchst ist das völlig normal, ich hab auch erst einmal eine kleine simple App programmiert, um mich heranzutasten.

3 Das Vertretungsplan-Projekt

3.1 Die Vorbereitungen

Du solltest jetzt dein erstes Android-Programm auf einem Emulator, oder noch besser auch auf einem Android-Smartphone, ans Laufen gebracht haben. Nun gilt es das Vertretungsplan-Projekt in Android Studio zu laden und dieses zu starten. Genutzt wird Git, es verwaltet Versionen von Dateien, in unserem Fall von den Programmdateien unseres Projektes. Zum Speichern der Daten im Internet benutzen wir GitHub. Um das Projekt herunterzuladen benötigst du keinen Account. Wenn du später die Änderungen und Verbesserungen, die du gemacht hast, wieder hochladen möchtest, musst du einen Account bei GitHub haben, welcher kostenlos ist. Ich empfehle dir also jetzt schon einen Account anzulegen und eventuell auch ein Testprojekt hochzuladen um die Features, die Git und GitHub bieten, auszuprobieren. Zu Git und GitHub gibt es hier eine sehr gute Anleitung: <https://help.github.com/>. Danach solltest du wissen, was z.B. ein Commit ist und auch, wie du deine Änderungen nach GitHub “pusht”. Mein Repository kannst du hier finden: <https://github.com/He-Ro/Vertretungsplan-App/>, die Webseite kannst du hier besuchen: <http://he-ro.github.com/Vertretungsplan-App>. Um nun Lese- und Schreib-Rechte auf meine Repository zu erhalten benötigst du einen Key, den ich zunächst erstellen muss, wie es hier beschrieben ist <https://help.github.com/articles/generating-ssh-keys>, dort steht auch, wie du den öffentlichen Key importieren kannst. Der Code kann in Android Studio direkt von GitHub heruntergeladen werden, gehe dazu auf *VCS* → *Checkout from Version Control* → *Git*. Wenn du *GitHub* benutzt musst du deinen GitHub Account mit Android Studio verknüpfen, was aber nicht notwendig sein sollte. Lass das Projekt jetzt laufen; bei Problemen googlen :)

3.2 Der Aufbau des Projektes

Den Aufbau eines Android Programms solltest du jetzt kennen, nämlich, dass im *src*-Ordner die Java-Dateien liegen. Im *res*-Ordner die Ressourcen, also vor allem die Layout-XML-Dateien, die Bild-Dateien, die statischen Strings, und so weiter. Und dass die *AndroidManifest.xml*-Datei einfach im Projektordner liegt. Zunächst einen Überblick über das Programm und die Klassen, die dazugehören:

Hauptscreen

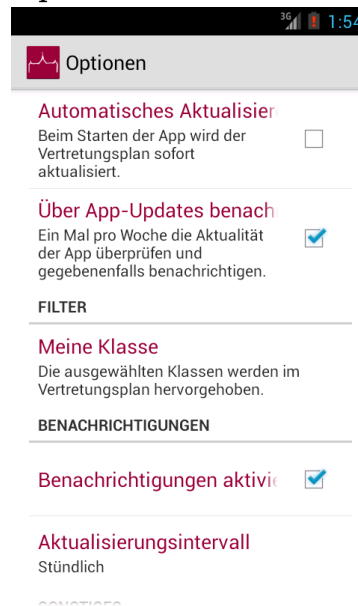


Diese Liste zeigt den Vertretungsplan an.

Die dazugehörigen Java Klassen:

- MainActivity
- MyListAdapter

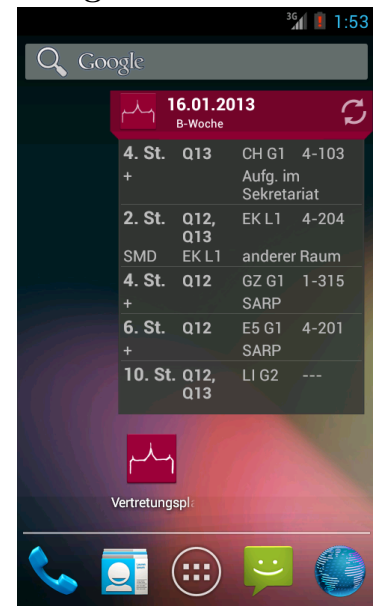
Optionen



In den Optionen kann der Benutzer die Einstellungen ändern.

- PrefsActivity

Widget



Das Widget zeigt den Vertretungsplan auf dem Homescreen an.

- WidgetProvider
- WidgetService
- WidgetUpdate
- WidgetViewsFactory

Dazugehörige wichtige Ressourcen:

- layout/main.xml
- layout/costum_row_view.xml
- menu/menu.xml
- xml/preferences.xml
- layout/ueber_dialog_layout.xml
- xml/widget_provider.xml
- layout/widget.xml
- layout/costum_row_view.xml

Klassen, die im Hintergrund laufen

Und deshalb nicht zu einer der drei Screens gehören.

- CheckForAppUpdate
- CheckForUpdates
- HtmlWork
- OnBoot

Detaillierte Informationen zu den einzelnen Klassen und Ressourcen sind als Kommentare in den einzelnen Dateien zu finden.

3.3 Eine neue Version veröffentlichen

Wir nehmen jetzt an, dass du ein paar Änderungen an der App vorgenommen hast und bist der Meinung, dass die neue Version stabiler/besser läuft als die Alte. Um diese Version zu veröffentlichen sind einige Maßnahmen zu treffen, diese liste ich hier auf:

1. Setzte den `versionCode` in der *build.gradle* Datei um genau einen hoch. Merke dir diese neue Versionsnummer.
2. Gib der Version einen sinnvollen Namen, dies hat nichts mit dem `versionCode` zu tun und ist sowohl in *build.gradle* unter `versionName` und in *res/values/strings.xml* unter `version_nr` zu finden. Wenn die aktuelle Version z.B 0.3.2 ist und du nur kleine Bug-Fixes veröffentlichst, solltest du die letzte Nummer um Eins erhöhen, sodass es 0.3.3 wird. Bei neuen Funktionen kannst du die zweite Nummer um Eins erhöhen und die letzte auf Null setzen, also 0.4.0. Bei einer GANZ neuen Funktion, z.B. das Laufband ist mit hinzugefügt, das Layout wurde grundlegend verändert o.ä. kannst du die erste Zahl um Eins erhöhen, dass es 1.0.0 wird. (Infos: [https://de.wikipedia.org/wiki/Version_\(Software\)](https://de.wikipedia.org/wiki/Version_(Software)))
3. Nun exportierst du die apk-Datei, dies ist in Android Studio sehr einfach. Gehe dazu auf *Build* → *Generate Signed APK...* Die keystore-Datei ist in dem Projektordner gespeichert, das Passwort dazu musst du dir von Frau Terveer holen. Weitere Infos zum Signieren gibt es hier: <https://developer.android.com/tools/publishing/app-signing.html>.

Falls du nicht Android Studio verwendest, kannst du unter Linux einfach `./gradle assembleRelease` ausführen.

Die APK kannst du dann mit Hilfe von `jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore <keystore> <APK> vertretungsplanandroidapp signieren`.

4. Diese apk-Datei kann nun auf dem Android-Handys installiert werden. Dies solltest du am Besten noch einmal testen. Benenne die Datei so um, dass der Dateiname den Versionsnamen aus 2. enthält, also z.B. **Vertretungsplan_0.3.apk**.
5. Die apk-Datei muss nun hoch geladen werden, damit deine Mitschüler Zugriff auf die neue Version haben. Kopiere die apk-Datei dazu in den Ordner **Veroeffentlicht** und füge die Datei zu deinem Git-Repository hinzu.
6. Danach änderst du die Versionsnummer in der Text-Datei `vertretungsplan_version.txt` zu der Nummer von 1. Dies bewirkt, dass die Klasse `CheckForAppUpdate` beim nächsten Check weiß, dass es eine neue Version gibt. Diese Änderungen werden nun “comitted”.
7. Bevor du die Änderungen nach GitHub “pusht”, solltest du die Version auch in Git vermerken. Dies wird mit sogenannten Tags erreicht. Führe dazu zum Beispiel `git tag v0.3.0` aus. So wird der jetzige Commit als Version 0.3.0 abgespeichert. Die Änderungen am Code und den neuen Tag pusht du mittels `git push --tags origin`.
8. Github bietet die Möglichkeit an diese Tags auch noch Dateien anzuhängen, das wollen wir uns zu Nutze machen und die APK Datei an den Release anhängen, siehe <https://help.github.com/articles/creating-releases/> für genauere Informationen.
9. Nun muss nur noch der Link auf der Webseite abgeändert werden. Der Link befindet sich natürlich in der `index.html`-Seite der Webseite, welche ein eigener Zweig(Branch) in GitHub ist. Diesen kannst du “auschecken” und dann lokal auf deinem Rechner bearbeiten, oder, weil es nur eine kleine Änderung ist auch in GitHub online. Den Link kannst du folgendermaßen erhalten. Gehe in GitHub im Zweig **master** auf **Veroeffentlicht**, dann auf die neue Version. Hier machst du einen Rechtsklick auf Raw und kopierst die URL-Adresse. Dies ist dann der Direktlink auf die neue Version, welche du in `index.html` einfügst. Falls du einen Tag erstellt hast und die APK an den Release angehängt hast, solltest du auch die URL von dieser APK nehmen. So ist der Ordner “Veröffentlicht” sogar obsolet.

Glückwunsch!! Du hast jetzt eine neue Version veröffentlicht. Wenn deine Mitschüler die Option zum Überprüfen der Aktualität, dürften diese innerhalb der nächsten Woche eine Meldung erhalten.

Falls du den Weg über den Tag und die angehängte APK Datei genommen hast, kannst du auch verfolgen, wie häufig die APK herunter geladen wurde. Dazu benutzen wir die API von GitHub, Informationen zum 0.5.1 Release kann man unter folgender URL einsehen <https://api.github.com/repos/He-Ro/Vertretungsplan-App/releases/tags/v0.5.1>. Mehr dazu kannst du hier finden: <https://developer.github.com/v3/repos/releases/>.

4 Das Projekt weitergeben

Wenn du dann schließlich auch das Abitur schaffst und von der Schule gehst musst du natürlich das Projekt so vorbereiten, dass es auch an jüngere Mitschüler weitergegeben werden kann. Hierzu solltest du diese Datei(L^AT_EX-Datei) anpassen, sodass es den jetzigen Stand widerspiegelt, hierzu gehört auch das Eintragen in die Autorenliste, sowie in der App um “Über”-Menü. Außerdem solltest du das Programm gut kommentieren, so wie man es immer macht!!

5 Mitwirkende

5.1 Schüler

- Hendrik Rosendahl (2012/13) `vertretungsplanapp-hero@yahoo.de`

5.2 Lehrer

- Frau Terveer (2012 - ...)