

MyStudiez – Órarend alkalmazás diákoknak

MilanCodes

Herke Milán

Felkészítő tanár: Vargáné Magusics Erika

EJSz Széchenyi István Gimnázium, Pécs.

1. Bevezetés

Régóta a fejemben volt egy alkalmazás ötlete, melynek segítségével könnyebbé tehetem saját magam és mások iskolai életét is.

Miután középiskolába kerültem, rögtön szembesültem egy személyes problémával – nagyon szétszórt voltam. Elegem volt a papír órarendekből, amiket folyamatosan elhagytam, a házfeladatokból, amiket sajnos sokszor elfelejtettem és a dolgozatokból, amikre mindig az előző éjszaka kezdtem el készülni. Úgy gondoltam, hogy nem én vagyok az egyetlen diák, aki ilyen problémákkal küzd. Gimnáziumi vagy akár egyetemi szinten sem.

Az alkalmazás tulajdonképpen e problémákra ad megoldást. Segítségével teljes mértékben digitalizálható egy diák, vagy akár egy tanár órarendje is. Amellett, hogy a felhasználó mindennap láthat egy részletes áttekintést a napjáról, felírhatja a megoldásra váró házfeladatokat, teendőket, illetve a közelgő dolgozatokat, eseményeket is. Továbbá a felhasználó, amennyiben kívánja, minden hozzáadott feladat/dolgozat előtt egy általa választott időpontban értesítve lesz.

2. Probléma megoldásának menete

2019 decemberében kezdtem el dolgozni az alkalmazáson. Megfelelő körülmények hiánya miatt, mivel az Apple csak MacOS-ról teszi lehetővé a rendszereikre való fejlesztést, úgy döntöttem, hogy Android operációs rendszerre fogom megírni az applikációt.

2.1 Megfelelő nyelv és Android verzió kiválasztása

Habár a fejemben volt, hogy Flutter-t fogok használni a fejlesztéshez, ami megoldást jelentett volna még az iOS/Android problémára is, hiszen a Google által kibocsátott SDK lehetővé teszi a cross-platform programok írását, de végül teljesen elvettem az ötletet, mert nem akartam, hogy egy több platformon futó alkalmazásnak egy közös kódbázisa legyen. Többek között teljesítménybeli gondokhoz is vezethetett volna. A natív Android fejlesztés egyszerűen szimpatikusabb volt és úgy gondoltam, hogy ahogy egyre nagyobb és nagyobb lesz a projekt, egyszerűbb is lesz átlátni.

Androidon belül is egy kisebb problémába ütköztem, hiszen el kellett döntennem, hogy Java-t vagy Kotlin-t fogok használni. Végül a Kotlin mellett döntöttem, hiszen a Google 2019 nyarán bejelentette, hogy az lesz az „első-nyelvük” és az Androidra történő fejlesztés „Kotlin-first” lesz. A következő feladat a minimum API-szint kiválasztása volt, amiben javarészt az alábbi táblázat segített:

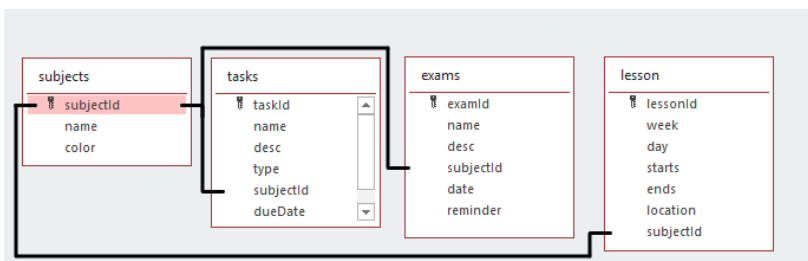
Verzió	Kódnév	API	Felhasználók
<6.0	-	<23	25.2
6.0	Marshmallow	23	16.9
7.0	Nougat	24	11.4
7.1		25	7.8
8.0	Oreo	26	12.9
8.1		27	15.4
9	Pie	28	10.4

1. táblázat: OS verziók megoszlása az összes Androidos készüléken 2019 májusában

Mivel az egyik legtöbb újdonságot bevezető API az Android 6.0-ás verziójával jött be, és az Android felhasználók több, mint 74%-a Marshmallow-t vagy újabb Android verziót használ, ezért a döntésem az Android 6.0 (API-level 23) mellé esett.

2.2 Az adatbázis tervezése

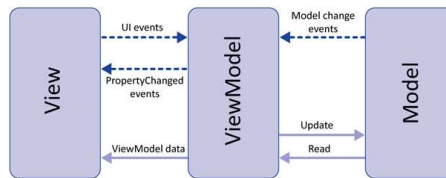
Az alkalmazás egy SQLite3 adatbázist használ, ami egy relációs, gyors, illetve nyílt forráskódú adatbázis, továbbiakban támogatja az SQL szintaxist. Alapvetően 4 táblát használ az applikáció. Egyet a tantárgyak tárolására (subjects), egyet a tanórák tárolására (lesson), egyet a feladatok tárolására (tasks) és egyet a dolgozatok tárolására (exams). Az 1. ábrán megtekinthető az applikáció adatbázissémája:



1. ábra: Adatbázisséma az alkalmazás 0.3.3-as verziója után

2.3 Az alkalmazás architektúrája

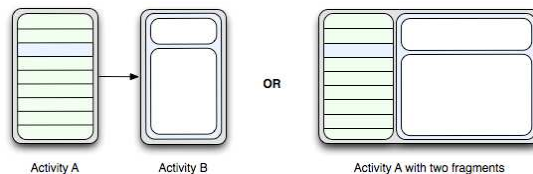
Még a fejlesztés megkezdése előtt el kellett döntenem, hogy milyen architektúráis mintát fog követni az alkalmazásom. A választásom végül a MVVM-re (Model-View-ViewModel, vagy magyarul Modell-nézet-nézetmodell) esett. Tulajdonképpen kettéválasztja a felhasználói felületre és az üzleti logikára vonatkozó kódot. Így minden UI-ra vonatkozó osztály mellé (Pl.: UiExampleFragment.kt) készült egy ViewModel osztály is (Pl.: UiExampleFragmentViewModel.kt), ahol például meghatározásra kerül, hogy mit csináljon egy gomb, vagy ahol lekérdezzük, frissítjük az adatokat az adatbázisból. Az architektúra használata miatt sokkal könnyebben átlátható lett a projekt. A 2. ábra demonstrálja a működésének lényegét:



2. Ábra: MVVM architektúra működése

2.4 Az alkalmazás UI részei

Az alkalmazás egy Activity-vel (MainActivity.kt) dolgozik, amibe számos fragment-et tölt be. Egyrészt azért döntöttem a fragment-ek használatára mellett, mert a fejlesztések későbbi szakaszában az egyes fragmentek könnyedén újra felhasználhatóak, másrészt, mert így sokkal egyszerűbb lesz táblagépekre optimalizálni a layout-ot (ahogy azt a 3. ábra is mutatja). Továbbá minden fragment-nek van egy ViewModel párja, a MVVM architektúráis mintának megfelelően.



3. ábra: Fragment-ek és Activity-k használata különböző méretű eszközökön

2.5 Tesztelés

Úgy vélem, hogy még az ilyen, kisebb méretű projektek fejlesztése közben is fontos a tesztek írása, hiszen segítségükkel sokkal egyszerűbben

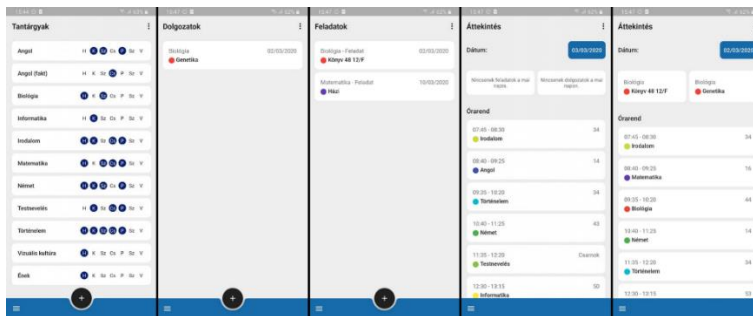
megtalálunk egy bugot a kódban, sokkal egyszerűbben észrevesszük, ha valami nem úgy működik az alkalmazásunkban, ahogy azt mi szeretnénk. A tesztek megírásához a JUnit4 keretrendszert használtam. Idő hiánya miatt jelenleg csak az adatbázisra vonatkozó egységteszteket tartalmazza a projekt.

2.6 Versioning

A szoftver lefejllesztése előtt egy SCRUM-táblát készítettem, ahol 4 részre osztottam a teendőket: „Közelgő verziók”, „Jelenleg fejlesztés alatt”, „Teszt írásra vár” és végül az „Implementált” kategória. Így nem csak sokkal átláthatóbbak lettek a teendőim, de ha esetleg a későbbiekben valaki be akar szállni a fejlesztésbe, sokkal könnyebben fel tudja venni a fonalat és egyszerűbben el lehet osztani a feladatokat majd a fejlesztők között.

3. Elért eredmények

Jómagam sokat tanultam a projekt elkészítése során, hiszen előtte még nem programoztam Kotlin nyelven és az XML Layoutok elkészítésében sem voltam túlzottan jártas, de végül sikerült minden tervezett funkciót implementálni a programba. Az alkalmazás jelenlegi állapotáról több képernyőkép is látható a 4. ábrán.



4. Ábra: „Áttekintés”, „Tantárgyak”, „Feladatok”, „Dolgozatok” Képernyőképek

Közel 3 havi munka után, használható lett az alkalmazás. A projektet teljesen nyíltforráskódúvá tettem, hiszen, ha már diákoknak készült, akkor minden diáknak joga van hozzá, hogy teljesen ingyen használhassa, illetve, ha valaki úgy érzi, hogy segíteni szeretne a fejlesztésbe, arra is lehetősége van. A projekt megtekinthető a [GitHub \(https://github.com/Studician/android-client\)](https://github.com/Studician/android-client) oldalán, ahol részletesebb információk találhatóak a telepítésre, code style-ra vonatkozóan. Így, ha bárki bármilyen problémát észlel, nyithat egy új issue-t és akár el is kezdhet dolgozni a megoldásán.