

KIV/MKZ

To Do List

Obsah

1	Zadání	2
2	Analýza problému a návrh řešení	2
2.1	Úlohy	2
2.2	Data	3
2.3	Notifikace	3
3	Implementace (popis řešení)	4
3.1	MainActivity	4
3.2	TaskAktivity	4
3.3	TaskList	4
3.4	Task	5
3.5	DBHelper	5
3.6	NotificationService	5
3.7	NotificationServiceReceiver	5
4	Uživatelská příručka	6
4.1	Hlavní okno	6
4.2	Okno editace/přidání úlohy	6
5	Testování	6
6	Závěr	7

1 Zadání

Vytvořte aplikaci pro mobilní zařízení na platformě Android. Aplikace bude typu "To Do List". Měla by poskytovat možnost zadání, úpravy, řazení a dokončení úlohy.

2 Analýza problému a návrh řešení

Úkolem je vytvořit praktickou aplikaci, která přehledně zobrazuje úlohy zadané uživatelem. Důraz by měl být kladen na jednoduchost aplikace v takovém smyslu, aby uživatel nebyl nucen ke zbytečným rutinním úkonům, jako je tomu zvykem u aplikací podobného typu. Pokud je činnost s aplikací otravná nebo zdoluhavá uživatel ztrácí zájem o její používání.

2.1 Úlohy

Konkrétní úlohou je myšlena jakákoliv aktivita (odeslání emailu, seznam nákupů, schůzka, ...), proto nemá smysl zabývat se výběrem typu jednotlivých úloh. Každá úloha by měla obsahovat časový údaj o jejím datu vypršení.

Seznam úloh bude řazen podle jejich data. Úlohy, které slouží jako poznámka nepotřebují datum prodlení, nicméně přiřazením aktuálního data můžeme zajistit jejich pozici na začátku seznamu s informací o jejich času vytvoření. Předpokládáme že poznámky nemají dlouhodobý význam a budou dříve odstraňovány.

Především kvůli poznámkám je třeba udržovat informaci s nepovinným obsáhlejším popisem daného úkolu. Podrobné informace s možnostmi změn data a času by měli být v samostatném okně kvůli minimalizaci prvků na obrazovce.

Editační okno by mělo být přehledné a zajistit výběr data a času pomocí dostupných podpůrných prvků (kalendář, hodiny). Z vlastní zkušenosti s podobnými aplikacemi, aby nedocházelo k ukončování jednotlivých úloh omylem, v hlavním okně, tlačítko pro ukončení by mělo být součástí okna pro editaci.

Z hlediska neobtěžování uživatele ovládáním, potvrzovacími dialogy, nastavováním jednotlivých úkolů, atp. omezíme nastavení pouze na tyto jednoduché atributy, které se budou bez vyplnění inicializovat na validní hodnoty (aktuální datum a čas). Samotné potvrzení ukončení úlohy lze také zjedno-

dušit na potvrzení dvojklikem bez dialogu. Podobně není nutné vyžadovat potvrzování editace úlohy.

2.2 Data

Uložení seznamu úloh musí být perzistentní nezávisle na běhu aplikace. Nabízí se ukládání do souboru a databáze. Předpokládáme práci s poměrně malým počtem dat nicméně časté čtení a upravování souboru považují za zbytečně náročné a použití tlačítka pro uložení je proti našemu hlavnímu cíli jednoduchosti ovládání a použijeme tedy perzistentní databázi. Bude potřeba automaticky propagovat změny dat z vnitřních datových struktur do databáze, aby nedošlo ke ztrátě. Pro založení databáze bude vhodné využít SQLite podporovaný platformou.

Jako vnitřní datovou strukturu bude vhodné použít mapu vzhledem k snadnému přístupu k jednotlivým úlohám podle jejich ID. Protože aplikace využívá více oken bude výhodné navrhnout datovou strukturu s využitím návrhového vzoru *Design Pattern Singleton*, která bude zajišťovat jednotný přístup k datům a může se starat o vlastní propagaci do databáze.

2.3 Notifikace

Upozornění by měla být volitelná služba, která zobrazí upozornění o nesplněných úlohách pro daný den. Protože aplikace není myšlena jako plánovač akcí a měla by sloužit jako nenásilné upozornění na čekající úlohy. Notifikace bude tedy probíhat pouze jednou denně, kdy poskytne zběžný přehled o čekajících úlohách. Z logiky upozorňování služba tvořící notifikace by měla být spouštěna i když aplikace běží pouze na pozadí. Což půjde zařídit nastavením plánovače úloh.

3 Implementace (popis řešení)

V této sekci budou popsány hlavní moduly programu a jejich vzájemná funkce.

3.1 MainActivity

Zajišťuje funkčnost hlavního okna aplikace. Jedná se tedy o seřazený seznam jednotlivých úloh s tlačítkem pro zapnutí (vypnutí) notifikací a přidání nové úlohy.

`onCreate()` Naplní list úlohami a nastaví `Listener` na jeho prvky pro umožnění editace prvku spuštěním aktivity `TaskAktivita` v módu editace (atribut `editing = true`).

`btAdd_onClick()` Umožní přidání nové úlohy spuštěním `TaskAktivita` s atributem `editing = false`.

`scheduleNotifications()` Registruje opakované volání služby pro tvoření notifikací `NotificationService` pomocí `AlarmManager`. Vypnutí notifikací zajišťuje `cancelNotification()`.

3.2 TaskAktivita

Třída zajišťující funkčnost okna pro editaci nebo přidání úlohy.

`onCreate()` Naplní prvky okna příslušnými hodnotami (hodnoty vybrané úlohy pro editaci, základní hodnoty jinak).

`addTask()` Z hodnot z jednotlivých elementů okna vytvoří novou instanci úlohy, kterou předá do datové struktury.

3.3 TaskList

Představuje vnitřní datovou strukturu, poskytuje přístup k datům a stará se o propagaci změn do databáze.

`getInstance()` Vrátil jedinou instanci třídy a při prvním volání provede naplnění datových struktur z databáze.

`put()`, `remove()` Přidání a odebrání úlohy.

`update()` Zajistí propagaci změn dané úlohy do databáze.

`getSortedList()` Získá seřazený list všech úloh.

3.4 Task

Obalová třída reprezentující jednu úlohu s jejími atributy.

3.5 DBHelper

Poskytuje rozhraní pro snadný přístup k databázi je využívána pouze z vnitřní datové struktury (`TaskList`).

3.6 NotificationService

Služba zajišťující vytvoření notifikace.

3.7 NotificationServiceReceiver

Zajišťuje volání služby `NotificationService` na podnět plánovače úloh (`AlarmManager`).

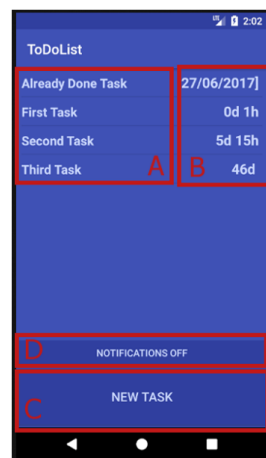
4 Uživatelská příručka

Pro spuštění programu je třeba minimální platforma Android verze JellyBean.

4.1 Hlavní okno

Hlavní okno obsahuje seznam seřazených úkolů podle data. Editaci jednotlivých úloh lze vyvolat kliknutím na požadovaný prvek. Nový úkol přidáme pomocí tlačítka **NEW TASK**.

- A Seznam jednotlivých úloh.
- B Zbývajíc čas do vypršení úlohy.
- C Tlačítko pro přidání nové úlohy.
- D Vypnutí/zapnutí notifikací.

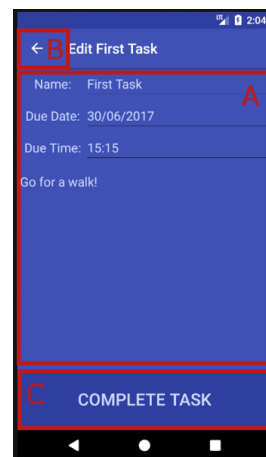


Obrázek 1: Hlavní okno.

4.2 Okno editace/přidání úlohy

Okno editace obsahuje detailní informace zvoleného nebo nového prvku. V případě editace prvku **dvojklikem** na tlačítko **COMPLETE TASK** úlohu ukončíme. Kliknutím na datum/čas vyvoláme komponentu pro výběr data/času.

- A Informace o zvolené/nové úloze.
- B Tlačítko pro návrat na hlavní obrazovku.
- C Ukončení úlohy/přidání nové úlohy.



Obrázek 2: Okno editace.

5 Testování

Testování proběhlo ve vývojovém prostředí Android Studio na virtuálních strojích Nexus 5 API 25 a Nexus 5X API 25. Dále na vlastním zařízení Xperia L CZ105 s verzí Android 4.2.2 JellyBean.

Při testování bylo odhaleno několik nedostatků týkajících se zobrazení, které byli odstraněny a funkční chyby tlačítka pro zapnutí notifikací.

6 Závěr

Program bylo překvapivě náročné zprovoznit přesto že programově náročný není. Většina problémů při vývoji vznikla v souvislosti s komplexností vývojového prostředí a následně jeho konfigurace. Ohledně problému při kódování jde pouze o jiný pohled na běh aplikace co se týče života aktivity a služeb.

Samotná aplikace vyhovuje požadavku na jednoduchost ovládání a je mnohem pohodlnější než předchozí alternativa, kterou jsem používal, kde se často stávalo že se úloha ukončila nechtěným stisknutím tlačítka.