



ŽILINSKÁ UNIVERZITA V ŽILINE

Fakulta riadenia
a informatiky

Semestrálna práca z predmetu
vývoj aplikácií pre mobilné zariadenia

***DAILYSTEPS – OSOBNÝ DENNÍK CIEĽOV A
POKROKU***

Vypracoval: [Anatolii Lopatiuk](#)

Študijná skupina: 5ZYI36

Akademický rok: [2024/2025](#)

V Žiline dňa [06.04.2025](#)



Obsah

Úvod	2
Prehľad podobných aplikácií	2
1. Todoist	2
2. Habitica	3
3. Daylio	4
Zhodnotenie	4
Analýza navrhovanej aplikácie	4
Funkcie:	4
Use Case diagram:	4
Mimofunkčné požiadavky:	5
Návrh architektúry aplikácie	5
Konceptný návrh	5
Sekvenčný diagram – Hlavný denný cyklus	6
Komponenty zapojené do cyklu:	6
Priebeh interakcie:	6
Návrh vzhľadu obrazoviek	8
1. Denný plán	8
2. Večerné vyhodnotenie	9
3. História	10
4. Štatistiky	11
Zoznam zdrojov	14



Úvod

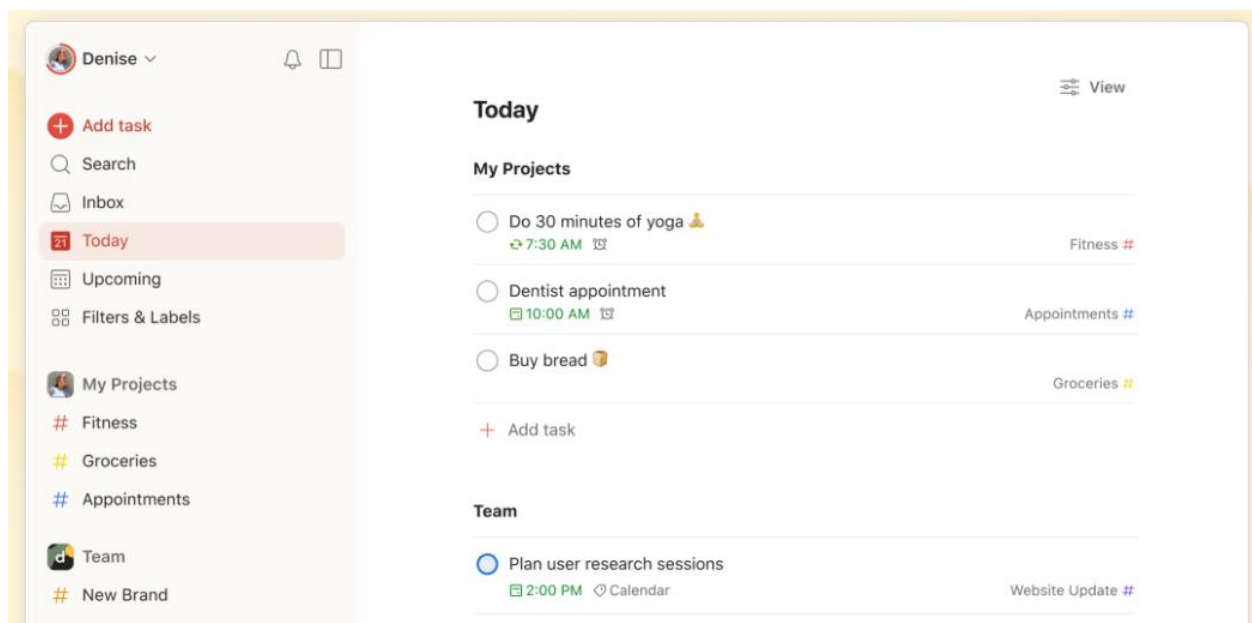
Cieľom aplikácie *DailySteps* je pomôcť používateľom organizovať každodenné úlohy v rôznych oblastiach života: práca, škola, šport, domácnosť, osobný rozvoj a iné. Aplikácia podporuje **dvojfázový režim**: ráno si používateľ naplánuje, čo chce dosiahnuť, a večer dostane pripomienku na vyhodnotenie dňa. Nesplnené úlohy sa automaticky presunú na ďalší deň.

Hlavnou myšlienkou je kombinácia minimalizmu, funkčnosti a efektívneho sledovania pokroku pomocou intuitívneho rozhrania.

Prehľad podobných aplikácií

1. Todoist

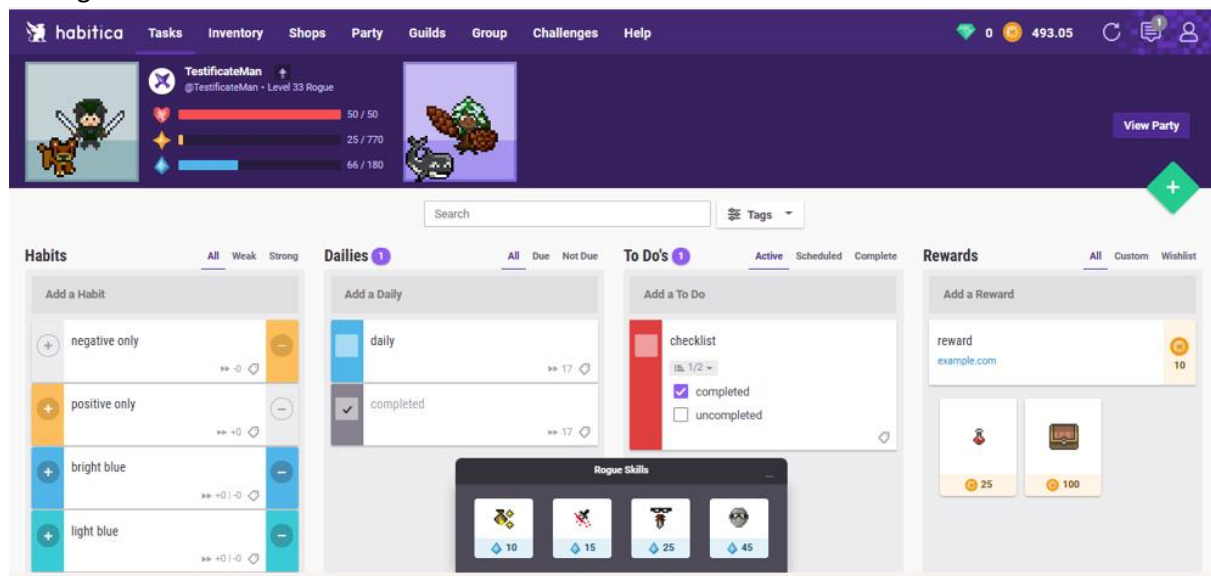
populárna aplikácia na správu úloh. Podporuje kategórie, prioritu a notifikácie. Neponúka však denný cyklus s automatickým vyhodnotením.





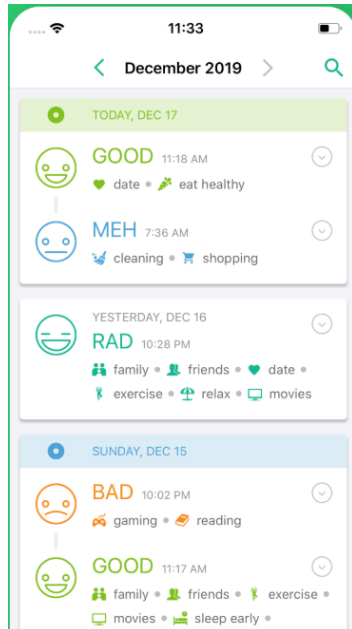
2. Habitica

aplikácia pre sledovanie zvykov s hernými prvkami. Zameraná skôr na motiváciu ako na denný time management.



3. Daylio

aplikácia na sledovanie nálady a činností. Užitočná pre záznam dňa, ale neponúka prehľad plánovaných cieľov.

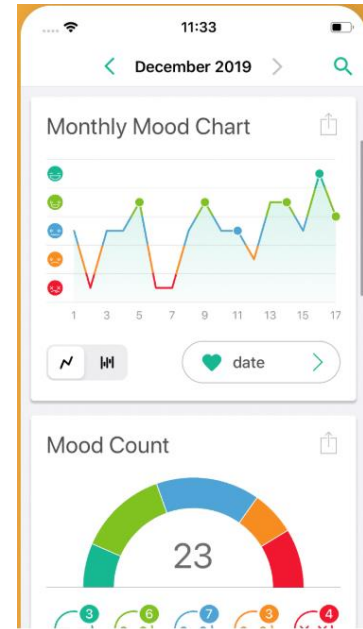


Zhodnotenie

DailySteps kombinuje elementy z týchto troch aplikácií do jedného elegantného denno-denného procesu: plán → vykonanie → vyhodnotenie.

Analýza navrhovanej aplikácie

Používatelia: bežný jednotlivci bez nutnosti registrácie, cieľom je jednoduchá offline funkcionálna.



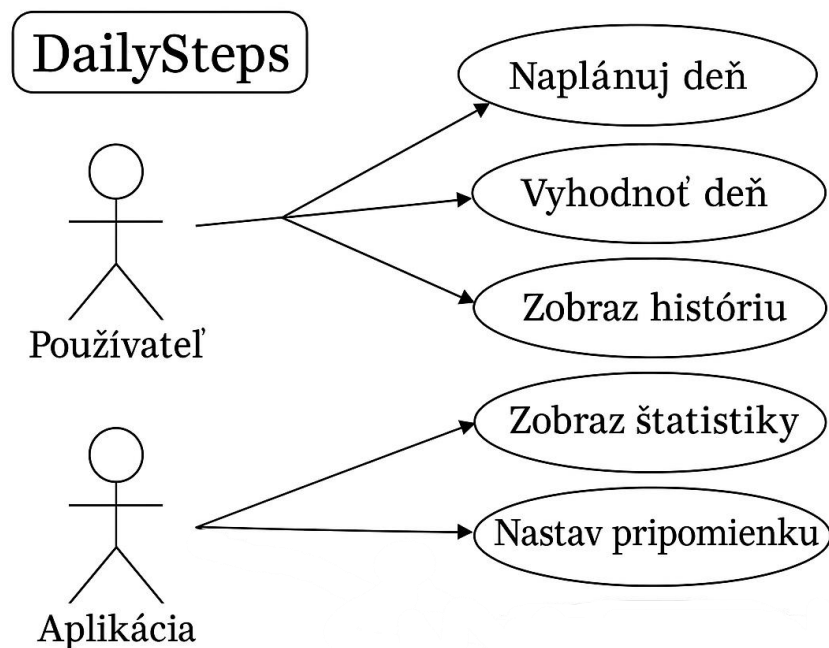
Používateľské role: len „User“

Funkcie:

- ranné naplánovanie úloh v preddefinovaných kategóriách
- večerné vyhodnotenie toho, čo sa podarilo
- automatické presunutie nespĺnených úloh
- notifikácie (WorkManager)
- zobrazenie histórie a štatistík

Use Case diagram:

Use Case	Aktér	Popis
Naplánuj deň	Používateľ	Používateľ vytvorí plán dňa z predvolených kategórií
Vyhodnoť deň	Používateľ	Používateľ označí splnené úlohy a pridá poznámky
Zobraz históriu	Používateľ	Používateľ si pozrie zoznam predchádzajúcich denných záznamov
Zobraz štatistiky	Používateľ	Aplikácia zobrazí prehľad pokroku (napr. splnené dni za sebou, úspešnosť)
Pripomienka	Aplikácia	Aplikácia automaticky zobrazí ranné a večerné upozornenia
Presuň nespĺnené úlohy	Aplikácia	Aplikácia automaticky prenesie nespĺnené úlohy na ďalší deň



Use Case Diagram

Mimofunkčné požiadavky:

- aplikácia funguje offline
- jednoduché rozhranie pre každodenné používanie

V tejto časti popíšte, aké funkcie (prípady použitia) bude navrhovaná aplikácia obsahovať z pohľadu používateľa. Predstavte jednotlivé role, ktoré sa v aplikácii budú nachádzať (napr. admin, hráč, ...). Ak viete vytvoriť UML diagramy prípadov použitia (*use cases*), môžete ich v tejto časti uviesť. Takisto uveďte mimofunkčné požiadavky (ak nejaké sú).

Návrh architektúry aplikácie

Koncepčný návrh

Aplikácia **DailySteps** je navrhnutá ako moderná Android aplikácia s využitím architektonického vzoru **MVVM (Model – View – ViewModel)** a Jetpack komponentov. Cieľom je vytvoriť **udržiavateľný, rozšíriteľný a testovateľný kód**, ktorý umožní jednoduchú prácu s dennými údajmi a automatickými notifikáciami.

Architektúra podporuje oddelenie zodpovednosti medzi UI, logikou vyhodnocovania a databázovým ukladacím priestorom.

Vrstva	Komponenty	Popis
--------	------------	-------



UI (View)	Jetpack Compose obrazovky (DailyPlanScreen, StatsScreen)	Zobrazenie denného plánu, histórie a štatistík, vstup používateľa
ViewModel	MainViewModel, StatsViewModel	Správa stavu obrazoviek, logika plánovania/odškrtavania úloh
Repository	TaskRepository	Abstrakcia pre databázu a pozadie (Room, WorkManager)
Model	DailyTask, TaskCategory, UserPreferences	Dátové triedy, entita Room, kategorizácia úloh
Data	DailyTaskDao, AppDatabase, WorkManager	Trvalé uloženie záznamov, pripomienky pomocou WorkManager

Sekvenčný diagram – Hlavný denný cyklus

Sekvenčný diagram zobrazuje **časový priebeh interakcií medzi komponentmi aplikácie** počas bežného dňa – teda od okamihu, keď používateľ vytvorí denný plán, až po večerné vyhodnotenie a automatický prenos nesplnených úloh.

Komponenty zapojené do cyklu:

- **Používateľ** – plánuje a vyhodnocuje deň
- **DailyPlanScreen (UI)** – vstup úloh, zobrazenie formulára
- **MainViewModel** – uchová stav dňa, reaguje na interakcie
- **TaskRepository** – sprostredkováva ukladanie a načítanie údajov
- **Room DB** – ukladá úlohy podľa dátumu
- **WorkManager** – zabezpečuje večerné pripomenutie + spúšťa prenos úloh

Priebeh interakcie:

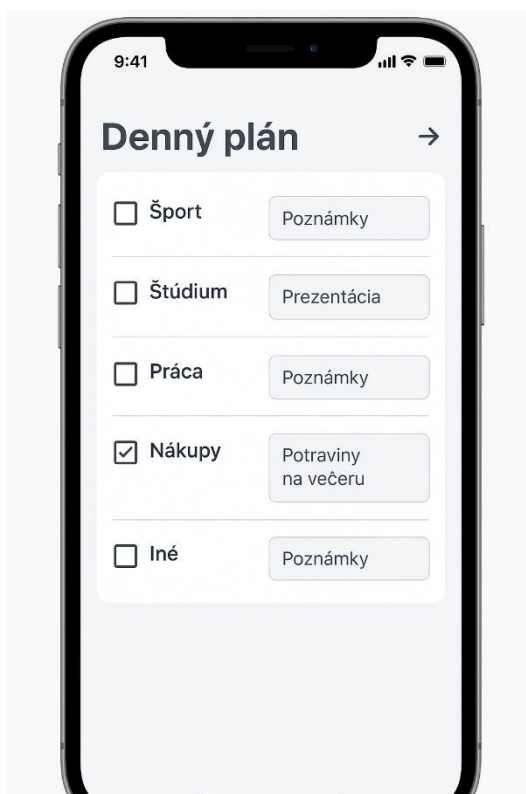
1. Používateľ otvorí aplikáciu ráno → vytvorí plán dňa cez **DailyPlanScreen**
2. Údaje (úlohy, kategórie) sú odoslané do **MainViewModel**
3. **MainViewModel** ich posiela cez **TaskRepository** do **Room DB**
4. Večer WorkManager zobrazí pripomienku → Používateľ otvorí **DailyReviewScreen**
5. Používateľ označí splnené úlohy
6. **MainViewModel** uloží aktualizovaný stav dňa
7. **Repository** zistí, ktoré úlohy neboli splnené
8. Nesplnené úlohy sa automaticky prenású do plánu na ďalší deň



Návrh vzhľadu obrazoviek

1. Denný plán

zoznam kategórií s checkboxami a poznámkami



Komponenty UI: Checkbox, OutlinedTextField, LazyColumn, Button



2. Večerné vyhodnotenie

formulár s možnosťou označiť, čo bolo vykonané

Vyhodnotenie dňa

Označte, čo ste vykonali:

- ☒ Fitness
- ☐ Štúdium
- ☐ Práca
- ☐ Nákup

Poznámky

Pridať poznámky

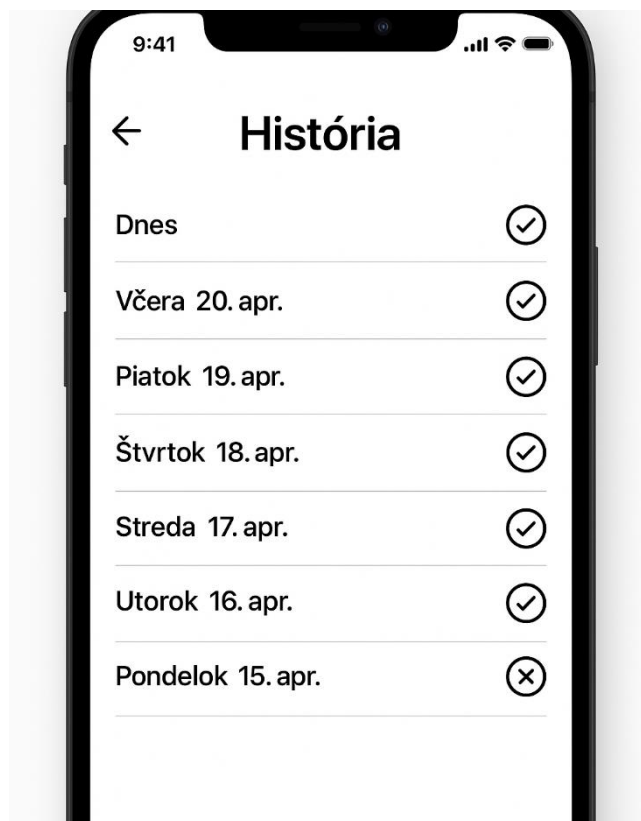
Hotovo

Komponenty UI: Checkbox, Text, ElevatedButton, Card, Spacer



3. História

zoznam uplynulých dní a ich stavov

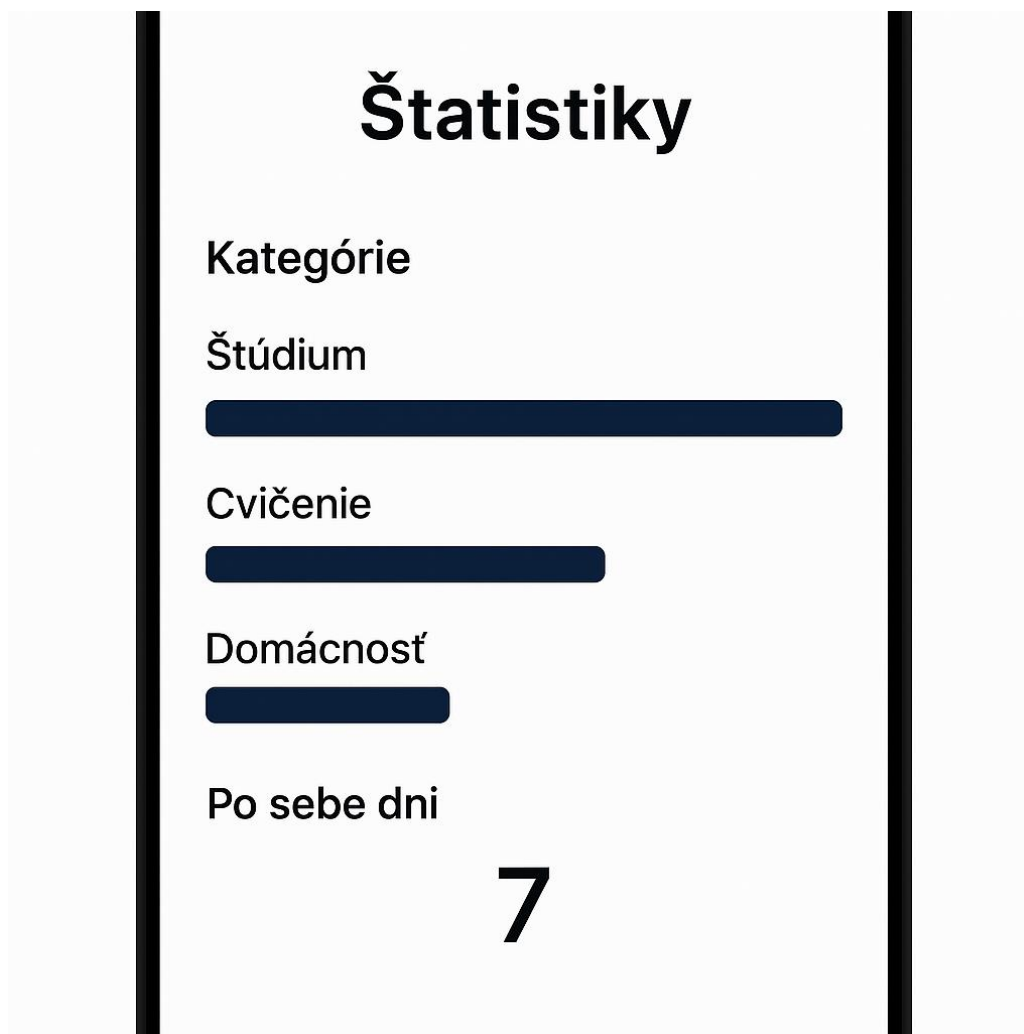


Komponenty UI: LazyColumn, Row, Icon, Text, Divider



4. Štatistiky

prehľad najčastejšie plnených kategórií, počet dní za sebou



Komponenty UI: ProgressBar, Text, Column, Row, Card, Canvas chart

Skutočný návrh riešenia problému

Krátka analýza

Východiskom bol denný cyklus používateľa, ktorý ráno plánuje úlohy, počas dňa ich plní a večer vyhodnocuje. Z hľadiska používateľských prípadov (Use Cases) sme v aplikácii pokryli:

1. **Naplánuj deň** – používateľ pridá nové úlohy (DailyPlanScreen + PlanViewModel + AddTaskUseCase).
2. **Vyhodnoť deň** – používateľ označí splnené úlohy a uloží poznámku (DailyReviewScreen + ReviewViewModel + ToggleDoneUseCase, SaveDayNoteUseCase).
3. **Zobraz históriu** – používateľ prehliada predošlé dni s percentuálnymi ukazovateľmi (HistoryScreen + HistoryViewModel + GetHistoryDatesUseCase, GetCompletionRatesUseCase).
4. **Zobraz štatistiky** – prehľad progresu a streakov (StatsScreen + StatsViewModel + GetStreakUseCase, GetTaskStreaksUseCase, GetWeeklyCompletionUseCase).
5. **Pripomienka** – ranné, denné a večerné notifikácie spúšťané cez WorkManager (ReminderWorker).
6. **Presuň nesplnené úlohy** – každodenný rollover nesplnených úloh na ďalší deň (DailyRolloverWorker).

Návrh riešenia

- **Architektonický vzor:** MVVM + Repository + Use Case.
- **Komponenty a vzťahy:**
 - **UI (Compose):** DailyPlanScreen, DailyReviewScreen, HistoryScreen, StatsScreen, SettingsDialog.
 - **ViewModely:** PlanViewModel, ReviewViewModel, HistoryViewModel, StatsViewModel, MainMenuViewModel.
 - **Use Case:**
 - **Tasks:** GetTasksUseCase, AddTaskUseCase, ToggleDoneUseCase, UpdateTaskUseCase, DeleteTaskUseCase
 - **DayNote:** GetDayNoteUseCase, SaveDayNoteUseCase
 - **History & Stats:** GetHistoryDatesUseCase, GetCompletionRatesUseCase, GetStreakUseCase, GetTaskStreaksUseCase, GetWeeklyCompletionUseCase
 - **Repository:** TaskRepositoryImpl, DefaultTaskRepositoryImpl, DailyDayNoteRepositoryImpl, StepEntryRepositoryImpl.
 - **Dátová vrstva:** Room (AppDatabase, DailyTaskDao, DailyDayNoteDao, DefaultTaskDao, StepEntryDao) + DataStore (PreferencesManager).
 - **WorkManager:** DailyRolloverWorker, ReminderWorker.
 - **ServiceLocator:** jednoduchý singleton na poskytovanie závislostí.



Popis implementácie

1. Persistencia dát

- **Room** pre entitu DailyTask, DailyDayNote, DefaultTask, StepEntry.
- **DataStore** (PreferencesManager) na ukladanie témy, lokality, lastDate a cieľa krokov.

2. Stavový manažment

- **StateFlow/SharedFlow** v ViewModel na prenos zoznamov úloh, poznámok, chýb a stavov lokality.
- **rememberSaveable + LazyListState.Saver** pre obnovu stavu scrollu a rozbalenia zoznamov pri rotácii obrazovky.

3. Lokalizácia a téma

- Výber jazyka cez SettingsDialog s použitím AppCompatDelegate.setApplicationLocales a Context.updateLocale pre Compose.
- Svetlý/tmavý režim cez PreferencesManager a DailyStepsTheme.

4. Notifikácie

- Vytvorenie kanála (NotificationHelper.createNotificationChannel).
- Tri periodické WorkManager úlohy (ReminderWorker) s filtrovaním obsahu a kontrolou povolenia POST_NOTIFICATIONS.
- Ranná, denné a večerné pripomienky spravované zo DailyStepsApp.scheduleDailyReminders().

5. Kľúčové hodnotené časti

- **Automatický rollover** (nesplnené úlohy → ďalší deň).
- **Pripomienky** (WorkManager + NotificationCompat).
- **Lokalizácia** (viacjazyčné zdroje, prepínanie jazyka za behu).
- **MVVM + Use Case** – čisté oddelenie zodpovedností.
- **Obnova stavu UI** (rotácie, scrollovanie, rozbalenie zoznamov).
- **Chytenie chýb** pomocou MutableSharedFlow<String> a SnackbarHost.

Tento návrh a implementácia zabezpečujú, že aplikácia je modulárna, testovateľná, rozšíriteľná a spĺňa všetky požadované funkčné aj mimofunkčné požiadavky.



Zoznam zdrojov

Android Developers. [Room Persistence Library](#). Google.

Android Developers. [WorkManager](#). Google.

Android Developers. [ViewModel](#). Google.

Android Developers. [Jetpack Compose](#). Google.

Material Design. [Design System](#). Google.

Habit Coach Blog. [How to Track Habits Effectively](#).

Nielsen Norman Group. [10 Usability Heuristics for User Interface Design](#).

[OpenAI](#). *Diagramy a návrhové nástroje použité na ilustráciu (AI generované vizualizácie UI a UML)*

<https://daylio.net/>

<https://habitica.com/static/home>

<https://www.todoist.com/>