RoommateApp

Aplikace pro správu výdajů spolubydlících

Tadeáš Pohlídal

29. května 2025

Obsah

1 Přehled aplikace

RoommateApp je mobilní aplikace vytvořená v .NET MAUI pro správu společných výdajů mezi spolubydlícími. Umožňuje uživatelům vytvářet skupiny, přidávat výdaje, automaticky počítat dluhy a spravovat jejich splacení s notifikačním systémem.

1.1 Klíčové funkcionality

- Správa uživatelských účtů s bezpečnou autentizací
- Vytváření a správa skupin spolubydlících
- Přidávání výdajů
- Automatický výpočet dluhů mezi členy
- Systém notifikací o splacených dluzích
- Přehledné zobrazení výdajů a dluhů

2 Architektura aplikace

2.1 Technologie

- **.NET 8**
- **.NET MAUI**
- **Entity Framework Core**
- **SQLite**

2.2 Vrstvová architektura

Aplikace je navržena ve čtyřvrstvové architektuře:

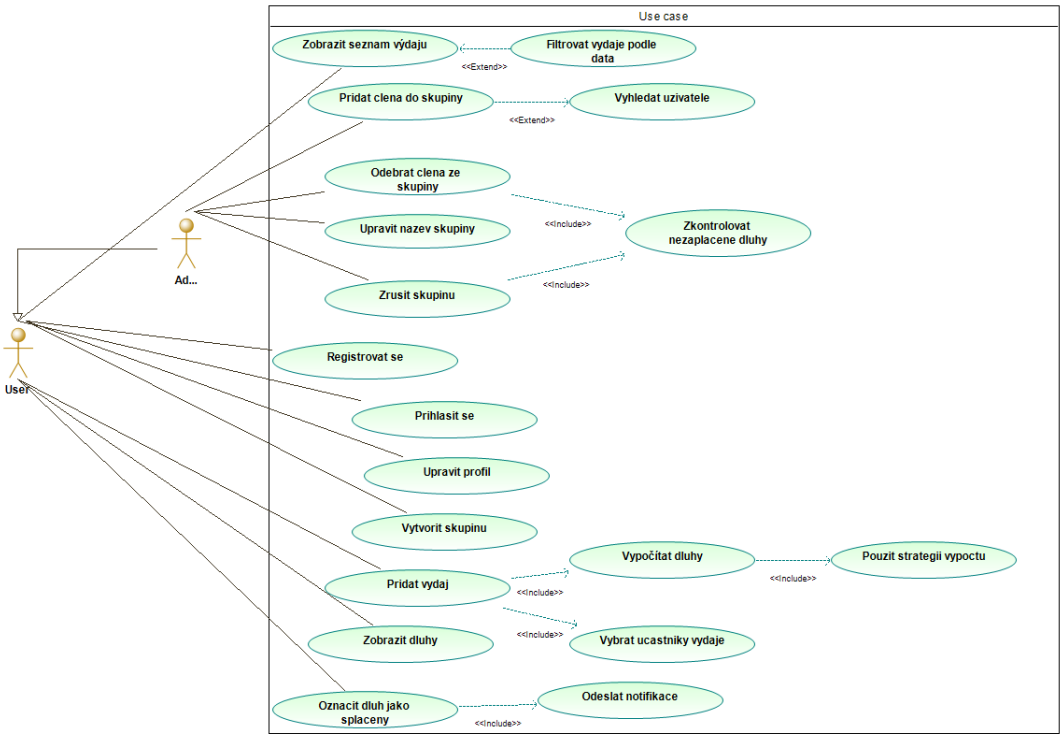
1. **Presentation Layer (Prezentační vrstva)**
   - Views – XAML stránky aplikace
   - ViewModels – business logika pro UI
2. **Application Layer (Aplikační vrstva)**
   - Services – aplikační služby (AuthService, NotificationService)
   - Factories – továrny pro vytváření objektů (Factory pattern)
   - Auth – logika pro autentizaci a autorizaci
3. **Domain Layer (Doménová vrstva)**
   - Models – doménové entity (Uzivatel, Skupina, Vydaj, Dluh)
   - Strategies – Strategy pattern

- Observers – Observer pattern

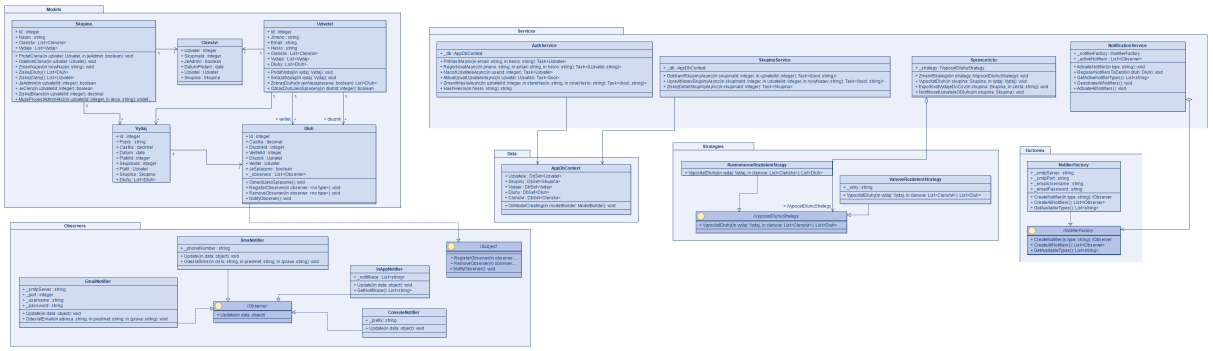
4. **Infrastructure Layer (Infrastrukturní vrstva)**
   - Data – AppDbContext a databázové konfigurace
   - DbContext – Entity Framework context

3 UML diagramy

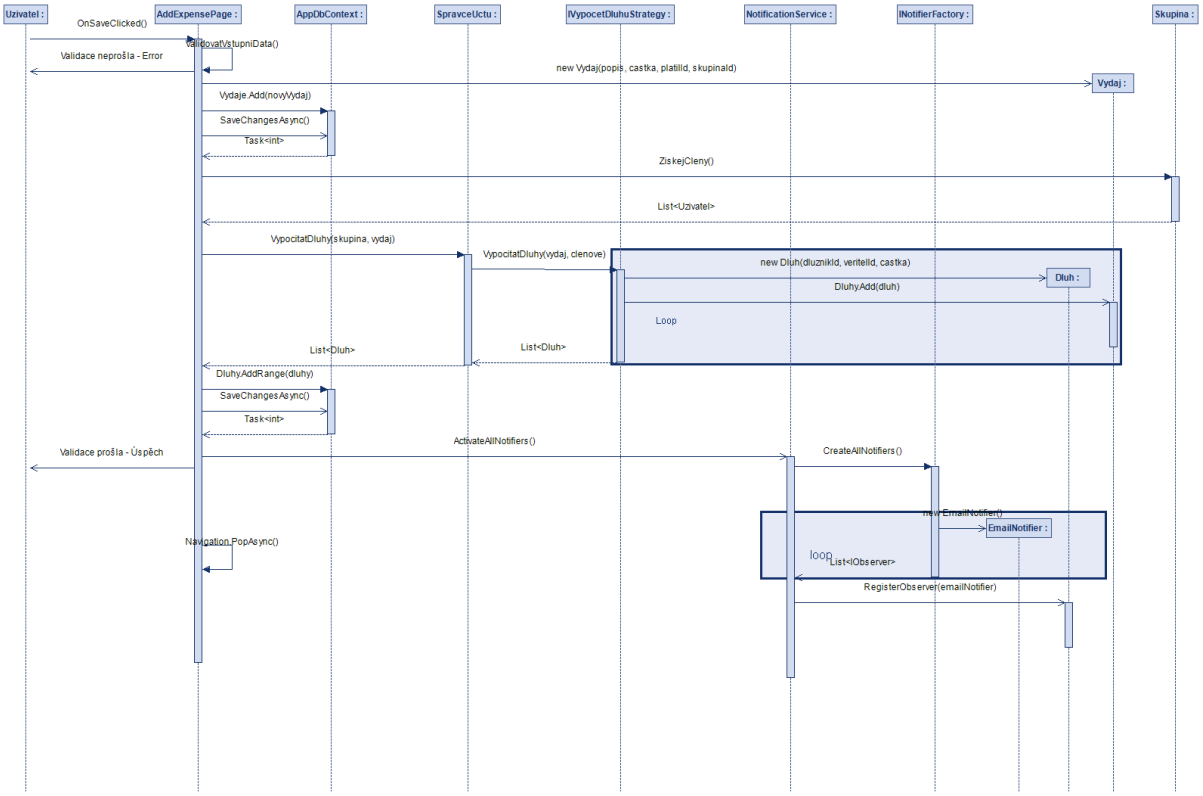
3.1 Use Case diagram



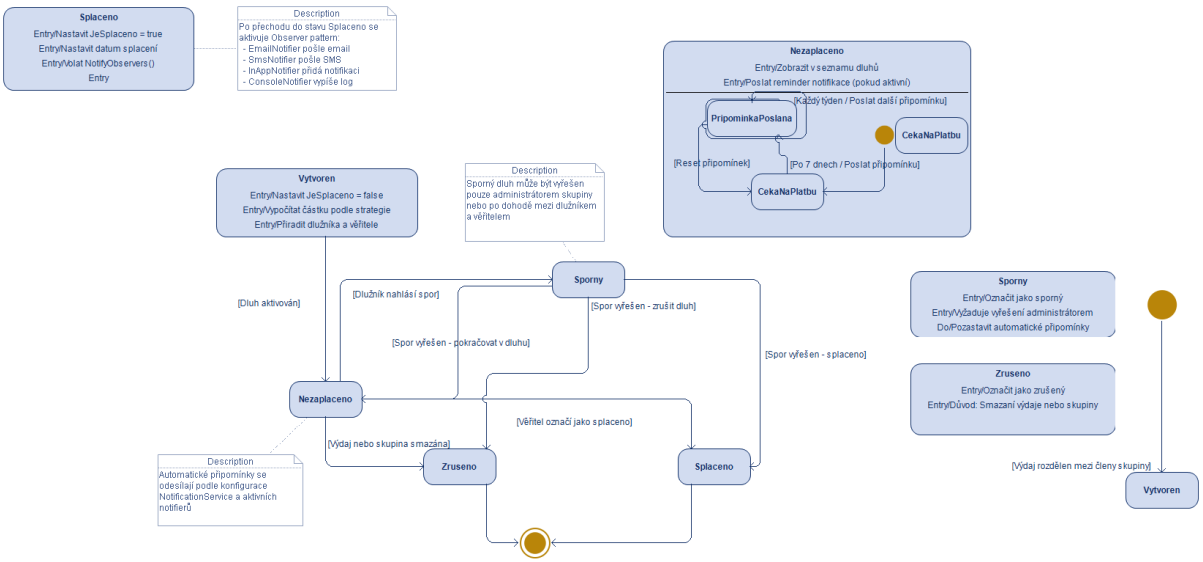
3.2 Class diagram



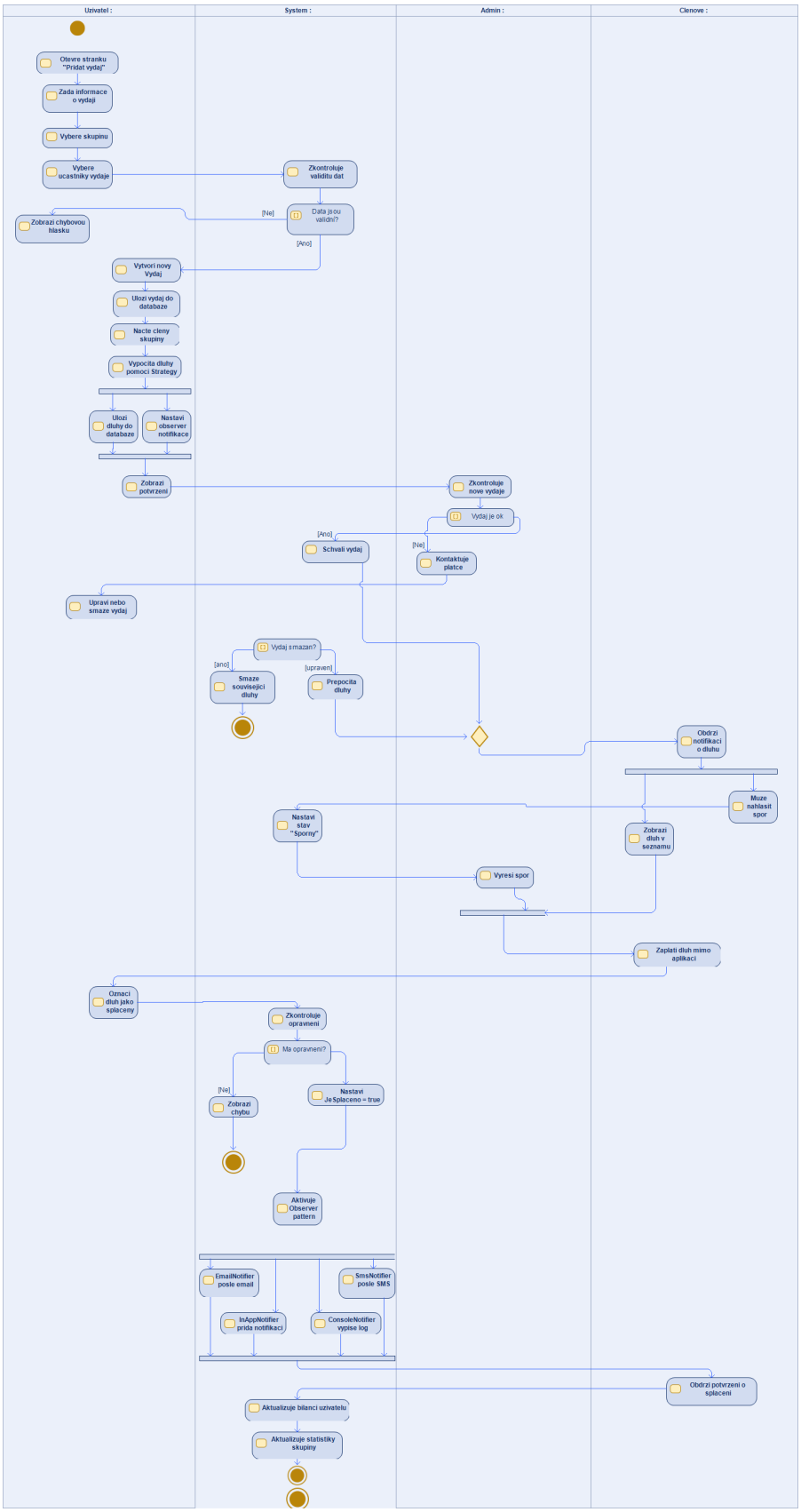
3.3 Sequence diagram



3.4 State Machine diagram



3.5 Activity diagram



4 Návrhové vzory

4.1 Strategy Pattern

Účel: Flexibilní výpočet dluhů podle různých strategií

- `IVypocetDluhuStrategy` – rozhraní strategie
- `RovnomerneRozdeleniStrategy` – rovnoměrné dělení částky
- `VahoveRozdeleniStrategy` – dělení podle vah členů

Použití: `SpravceUctu` může za běhu měnit strategii výpočtu

4.2 Observer Pattern

Účel: Notifikace o změnách stavu dluhů

- `ISubject` / `IObserver` – základní rozhraní
- `Dluh` implementuje `ISubject`
- Konkrétní observeři: `EmailNotifier`, `SmsNotifier`, `InAppNotifier`, `ConsoleNotifier`

Použití: Při označení dluhu jako splacený se automaticky rozešlou notifikace

4.3 Factory Pattern

Účel: Vytváření instancí notifikačních služeb

- `INotifierFactory` – rozhraní factory
- `NotifierFactory` – konkrétní implementace
- Dynamické vytváření notifierů podle typu

Použití: `NotificationService` používá factory pro vytváření aktivních notifierů

4.4 MVVM Pattern

Účel: Oddělení prezentační a business logiky (Jenom částečně implementované)

- `BaseViewModel` – základní třída pro view modely
- `RelayCommand` / `AsyncRelayCommand` – implementace ICommand
- Data binding mezi Views a ViewModels

5 Databázový model

5.1 Hlavní entity

- **Uzivatel** – uživatelské účty s hashem hesel
- **Skupina** – skupiny spolubydlících
- **Clenstvi** – M:N vazba mezi uživateli a skupinami
- **Vydaj** – společné výdaje ve skupinách
- **Dluh** – automaticky generované dluhy z výdajů

5.2 Klíčové vztahy

- Uživatel <-> Skupina (M:N přes Clenstvi)
- Skupina -> Vydaj (1:N)
- Vydaj -> Dluh (1:N)
- Uživatel -> Dluh (1:N jako dlužník i věřitel)

6 Bezpečnost

6.1 Přihlašování

- Hesla jsou šifrovaná pomocí BCryptu
- Aplikace si pamatuje přihlášeného uživatele
- Při prvním spuštění se automaticky vytvoří testovací účty (V Homepage je zakomentované tlačítko pro reset databáze pro testovací purposes)

6.2 Oprávnění uživatelů

- Každá skupina má svého administrátora
- Pouze administrátor může mazat skupinu nebo upravovat členy
- Dluh může označit jako splacený pouze věřitel
- Všechna zadávaná data se kontrolují před uložením

7 Testování a kvalita

7.1 Implementované kontroly

- **Validace dat** na všech vrstvách
- **Exception handling** s uživatelsky přívětivými hláškami

- **Asynchronní operace** pro databázové volání
- **Memory management** s properly disposed objekty

7.2 Potenciální rozšíření

- Dodělat přidávání přátel
- Unit testy pro business logiku
- Refactoring
- Integration testy pro databázové operace
- UI testy pro kritické user journeys

8 Závěr

Aplikace RoommateApp je funkční mobilní aplikace pro správu výdajů spolubydlících (a teoreticky jakýchkoliv výdajů). Úspěšně implementuje všechny požadované návrhové vzory a splňuje zadání projektu. Pro další rozšíření by bylo vhodné dokončit MVVM implementaci a přidat testování.