ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

KIV/NET

Aplikace pro správu rodinných financí

Autor: Antonín NEUMANN, A14N0139P

Akademický rok: 2014/2015

1 Zadání

Pamatuje si jednotlivé již zadané položky a dokáže je napovídat.
Umí automaticky započítat:

- pravidelné platby (elektrika, nájem, mobilní tarif, cigarety, ...)
    - roční, půlroční, čtvrtletní, měsíční
    - umí také pohlídat pravidelnou platbu bez specifické částky (např. mobil - výše účtu se může měnit, ale platit se musí každý měsíc)
- dočasné platby (splátky, hypotéka, …)

Umí rozdělit platby na:

- rodinné (elektřina, plyn, nájem …)
- osobní (mobilní tarif, oblečení)
- soukromé (tyto výdaje vidí pouze osoba, která je zadala)

Umožňuje přístup více lidí, každý má svoje přihlašovací údaje.
Umožňuje volbu typu platby/hashtag (dárky, jídlo, drogerie, …) → možnost filtrování plateb.
Uložení údajů v databázi, plně postačí DB integrovaná v C#.

1.1 Platba (jedna položka):

- ID
- částka
- datum
- uživatel
- účel (textový popis)
- skupina (rodinná, osobní, soukromá)
- rozsah (pravidelná, dočasná, pravidelná bez částky, jednorázová, plánovaná)
- tagy (#jidlo, …)
- zaplaceno [boolean] (kvůli pravidelným přip. plánovaným platbám - to že nájem mám platit každý měsíc ještě neznamemá, že jsem ho zaplatil)

2 Úvod

Rozhodl jsem se vytvořit program pro správu rodinných financí, kde by si každý člen rodiny mohl zaznamenávat svoje výdaje.

3 Implementace

Aplikace má 3 hlavní entity uživatel, platba, a plánovaná položka. K tvorbě entit jsem použil Entity Framework 6 a metodu „code first“, tedy že z hotových entit se vytváří mapování do databáze.

3.1 User

Tato entita slouží k přihlašování a oddělení jednotlivých plateb pro každého člena domácnosti.

Entita uživatele se skládá hlavně z jeho přihlašovacího jména a hesla, tyto položky jsou pro fungování aplikace nezbytné. Dále je součástí entity celé uživatelovo jméno a pohlaví.

3.2 Payment

Tato entita reprezentuje jednotlivé platby, např. za elektřinu, plyn, atp. Každá položka má nastavenou cenu a měnu, pole pro zapsání účelu platby, přepínač je-li již zaplacena, datum a nepovinnou poznámku (pro nějaké detailnější informace). Dále obsahuje položku Group, která určuje její viditelnost ostatními uživateli a zařazení (například hodnota Family říká, že platba se týká celé rodiny, naopak hodnota Private určuje, že položku nebude moci zobrazit nikdo další).

Poslední položku jsou tzv. tagy, které by měli sloužit k jednodušší identifikaci platby a též ke snadnějšímu filtrování.

3.3 ScheduledItem

Poslední důležitou entitou aplikace je plánování. Tato entita umožňuje každé platbě být naplánována, tzn. že taková platba se bude například periodicky opakovat (každý měsíc nebo rok) nebo se uskuteční jednorázově někdy v budoucnu.

Plánování může být také pouze dočasné, a po uplynutí doby platnosti se plánovaná platba přestane pravidelně zahrnovat mezi platby.

Důležitou položkou v této entitě je datum jejího posledního použití. Tento údaj bude využit hlavně při spuštění aplikace po delší době, aby došlo k započtení správného počtu opakování. Například pokud bude aplikace spuštěna po 3 měsících od předchozího spuštění, je nutné započíst pravidelné měsíční platby celkem 3krát.

4 Testování

Rozhodl jsem se pro první odevzdání, které bude obsahovat místo jednoduché konzolové aplikace sadu Unit testů. S unit testy jsem se nikdy předtím nesetkal, ale stále častěji jsem o nich slýchal.

Pomocí unit testů jsem otestoval jednotlivé entity a s použít knihovny „Moq“ potom i service jednotlivých entit.

U testování service pomocí Moqu jsem narazil na problém, který se mi zatím nepodařilo vyřešit. Konkrétně při testování operace smazání entity dochází k chybě „Operace výčtu nevrátila žádné výsledky.“ a testovací context je prázdný.