



Zápočtová úloha z předmětu KIV/MKZ

Počasí pro iPhone

5. května 2016

Lukáš Černý

1 Úvod

Úkolem mé semestrální práce je vytvořit mobilní aplikaci zobrazující počasí. Operační systém, na kterém aplikace poběží, jsem zvolil iOS. Vývoj a testování budu provádět na iPhone 4S. S tím souvisí i vzhledová optimalizace pro tohle zařízení (tj. uhlopříčka obrazovky 3.5”).

2 Teorie

Operační systém iOS běží pouze na zařízeních od společnosti Apple. Konkrétně se jedná o iPhony a iPady. Omezení této platformy je použití pouze operačního systému OS X. Vývoj lze provádět i na jiných operačních systémech, ovšem spuštění emulátoru a nahrání do fyzického zařízení vyžaduje OS X. Nativní programovací jazyky jsou Objective-C a Swift. Oba programovací jazyky jsou objektově orientované.

3 Možnosti řešení

API

Aplikace má zobrazovat počasí aktuálního dne (případně i v aktuální hodinu) a předpověď na dalších několik dní. Data je potřeba někde získat. Jediným řešením, jak lze data získat, je pomocí API¹. Dostupných API pro počasí je mnoho, ale nejsou k dispozici zdarma bez omezení. Nejnižším omezením je založení účtu a pomocí vygenerovaného identifikátoru je možné služby poskytované přes API využívat. Takové služby nabízí například Google, Yahoo nebo OpenWeatherMap. Data na předpověď počasí jsou velice ceněná. Chtěli bychom mít co nejpřesnější data, museli bychom si za nemalé peníze zakoupit přístupy přímo od společností, které se předpovědi zabývají.

Databáze měst a obcí

Stejně důležité jako odkud brát aktuální data pro počasí, je i zdroj dat, pro které město či obci se data zobrazí. Zde je několik možností výběru.

Najít službu, která poskytuje veřejné API. Tento úkol může být velice obtížný a já jsem žádnou takovou službu, která by mi poskytovala to co potřebuji, nenašel. Problém by mohl nastat i s výkonem při použití autocomplete².

Databáze dat uložená v zařízení. Najít tyto data není takový problém, jelikož jsou tyto informace veřejné. I tohle řešení má nevýhodu. Chceme-li mít data pro celý svět, bude se jednat o velký soubor a načítat takto velká data na mobilním zařízení vyžaduje dobré skrytí, kdy se data budou načítat a dobrý návrh struktury. Využití autocomplete je výkonově přátelnější.

Poslední možností je nepoužívat žádná data a spolehnout se pouze na situaci, že uživatele bude zajímat pouze předpověď pro aktuální polohu. Poté nejsou potřeba žádná data a aktuální polohu nám sdělí GPS senzor v telefonu.

¹Application Programming Interface označuje v informatice rozhraní pro programování aplikací. Jde o sbírku procedur, funkcí, tříd či protokolů nějaké knihovny, které může programátor využívat.

²Automatické doplňování zbytku slova při psaní.

4 Navržené řešení

Grafické rozložení

Aplikaci jsem navrhl tak, aby největší část obrazovky zabírala aktuální předpověď pro daný den. V horní část obrazovky se nachází grafické znázornění předpovědi, datum a poloha. Na středu obrazovky je velkým fontem zobrazená teplota. Těsně pod ní se pak nachází rychlost větru a vlhkost. V dolní části obrazovky je předpověď pro následující čtyři dny. U každého dne je název dne v týdnu, grafické znázornění předpovědi a teplota.

Data

Pro zobrazení předpovědi jsou zapotřebí aktuální data. Rozhodl jsem se využít API poskytované OpenWeatherMap. Služby jsou nabízeny i zdarma, ale jsou zde omezení. V rámci semestrální práce neovlivňují omezení běhu aplikace. Problémy by mohly nastat při zveřejnění aplikace na App Store. Jednou z podmínek využívání API od OpenWeatherMap je 60 dotazů za minutu a předpověď na následujících pět dní po třech hodinách.

Aby bylo možné zobrazit předpověď v určitém městě, je důležité mít databázi měst a obcí. Zvolil jsem textový soubor se seznamem měst celého světa. Tento seznam je možné také najít na OpenWeatherMap a obrahuje identifikátor města, název, zeměpisnou šířku a délku a zkratku státu. Dále jsem dal možnost využít aktuální polohy pomocí GPS senzoru.

5 Implementace programu

Program jsem napsal v programovacím jazyku Objective-C. Obsahuje celkem tři graficky odlišné obrazovky:

- uvítací obrazovka
- obrazovka s počasím
- nastavení

Struktura programu

- **main** - spouští aplikaci
- **AppDelegate** - spouští grafickou část aplikace
- **DataHandler** - uchovává a připravuje data
- **LocationHandler** - obsluhuje lokalizaci
- **NetworkHandler** - obsluhuje komunikaci po síti
- **WeatherController** - obsluhuje grafickou část hlavní obrazovky
- **SettingsController** - obsluhuje grafickou část obrazovky s nastavením
- **WeatherCellView** - rozšiřuje UIView pro rozobrazení dalších dnů
- **CityTableCell** - definuje řádku pro tabulku při vyhledávání města
- **Utils** - pomocné metody pro formátování, ukládání nastavení a práci s daty

Naskytnuté problémy

- **Observer a observable:**

Je nutné reagovat na události, ale nevěděl jsem, kdy daná událost přijde. Od verze iOS 9.0 je většina metod asynchronních a dříve používané synchronní metody jsou deprecated³. Jednou z těchto metod je metoda pro komunikaci se sítí. Vytvoří se požadavek a pošle se na server. Výsledek lze získat v bloku metody. Do tohoto bloku ovšem nelze jinak zasahovat a přistoupit. Data si lze vzít například pomocí třídy návrhového vzoru jedináček. Mít všechny třídy jedináčky nelze. Tento problém jsem vyřešil využitím třídy *NSNotificationCenter*. Na daných místech vytvořím posluchače na událost. A až v asynchronním bloku přijdou požadovaná data, pošlu je ke zpracování *DataHandleru* a následně vyšlu zprávu, že se změnila data a grafika může překreslit na aktuální data.

- **Výběr města nebo použití lokalizace**

Dal jsem uživateli možnost zvolit si počasí podle aktuální polohy nebo výběrem města. Tyto dvě možnosti jinak začínají, ale končí stejně. Jiné jsou až do chvíle, než proběhne požadavek na server. Rozdíl je ten, že při použití lokalizace se vytváří jiný dotaz, než při použití názvu města. Problém jsem vyřešil pomocí výše zmiňovaného problému, a to pomocí notifikací. Vybere-li uživatel město ručně, vytvoří notifikaci o změně města ručním výběrem a pokud použije lokalizaci, vytvoří se notifikace o použití lokalizace. Na obě tyto notifikace naslouchá *NetworkHandler*, který si danou zprávu převezme, vytvoří dotaz z poskytnutých dat a odešle na server.

- **Databáze měst a obcí**

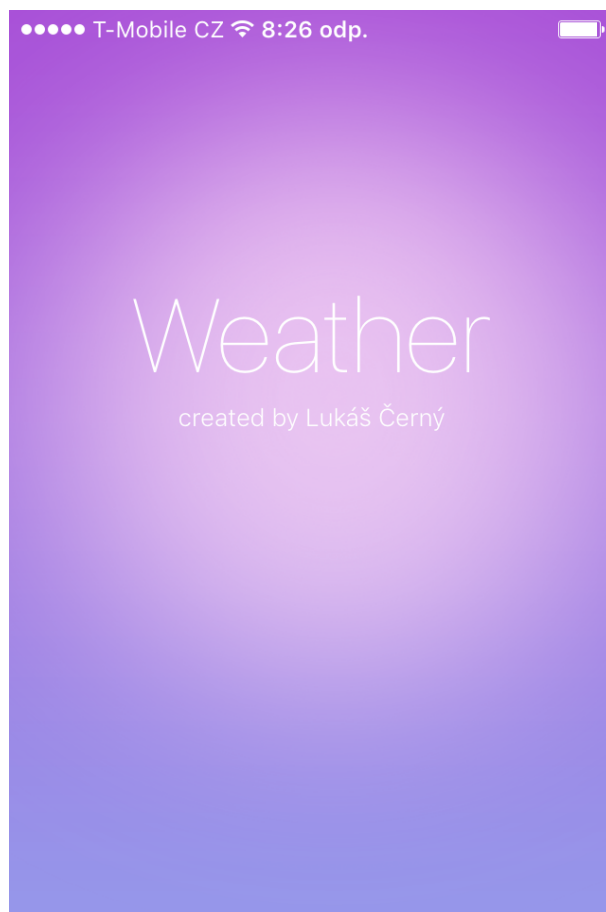
Rozhodl jsem se využít textový soubor s názvy měst, ID města, GPS souřadnicemi a zkratkou státu. Tyto data jsem načel a uložil do datové struktury. Všechny údaje jsem chtěl využít. ID města pro vytváření požadavku (byl by pouze jeden), GPS souřadnice při hledání města, který je nejbližší aktuální poloze, a jméno jměsta pro zobrazení na hlavní obrazovce. Jelikož se jedná o velký soubor (pro načítání na telefonu), trvalo jeho načtení 5s. Takto dlouhou dobu načtení jsem schoval při zobrazování úvodní obrazovky. Ovšem i tak to je dlouhá doba. Měřit dobu hledání nejbližšího města z GPS souřadnic jsem nezkoušel a raději jsem toto řešení rovnou zavrhl. Problém s GPS souřadnicemi jsem vyřešil viz. problém výše.

Textový soubor jsem si upravil tak, že jsem poté potřeboval pouze ID města a jméno. Změnil jsem i strukturu dat. Načítání se zrychlilo o 1s (na 4s). Pokud jsem vybíral město ručně, fungovalo vše správně. Problém nastal při použití aktuální polohy. V odpovědi ze serveru jsem dostával ID města, které není v textovém souboru. Výsledkem bylo, že se nezobrazil název města. Provedl jsem další úpravu vstupních dat. Vynechal jsem ID města a nechal pouze název. Rychlost načítání se velice zrychlilo. Soubor se načte na 0.5s a rychlost vyhledávání pomocí autocomplete je také rychlejší.

³Metody, které se nedoporučuje používat.

6 Obsluha programu

Při spuštění aplikace se zobrazí úvodní obrazovka (viz obr. 1).



Obrázek 1: Úvodní obrazovka.

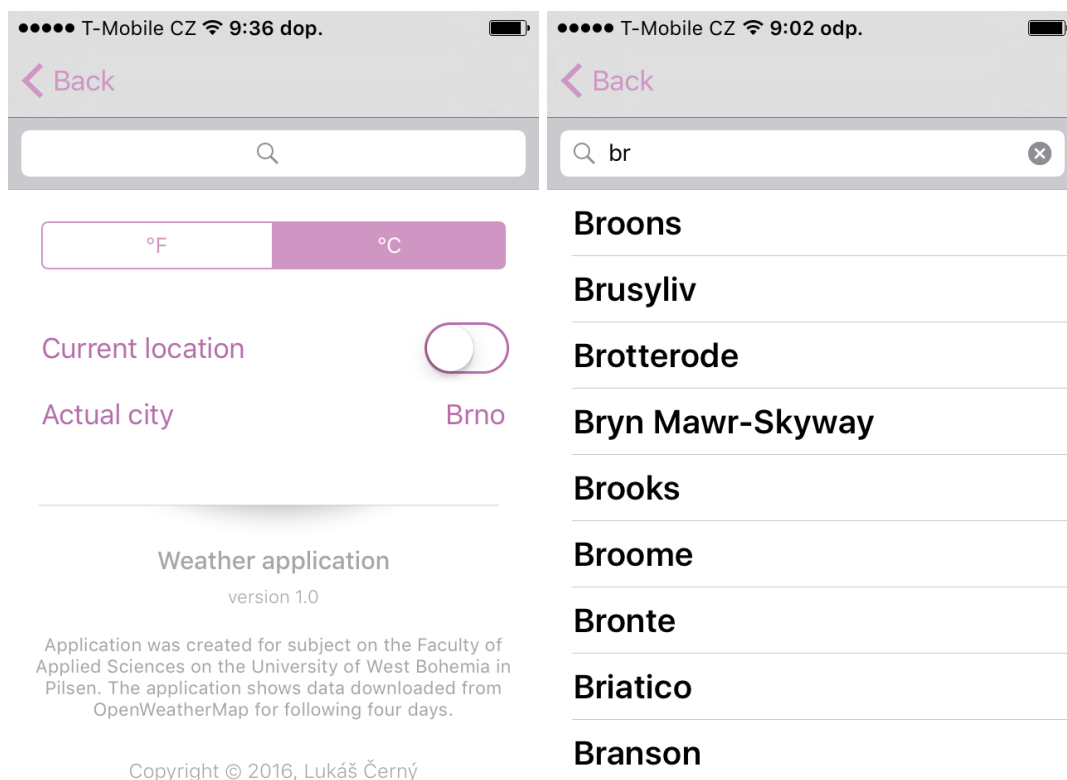
Úvodní obrazovka po několika sekundách zmizí a zobrazí se hlavní obrazovka s aktuální předpovědí (viz obr. 2). V horní části se nachází po levé straně tlačítko na aktualizaci počasí a na pravé straně je tlačítko pro vstup do nastavení. Mezi těmito tlačítky je grafické zobrazení aktuálního počasí, název města a aktuální datum. Uprostřed je velkým fontem aktuální teplota. Pod teplotou se nachází hodnota rychlosti větru a vlhkosti. V dolní části je předpověď na další čtyři dny, kde se zobrazuje název dne v týdnu, grafické znázornění předpovědi a teplota.



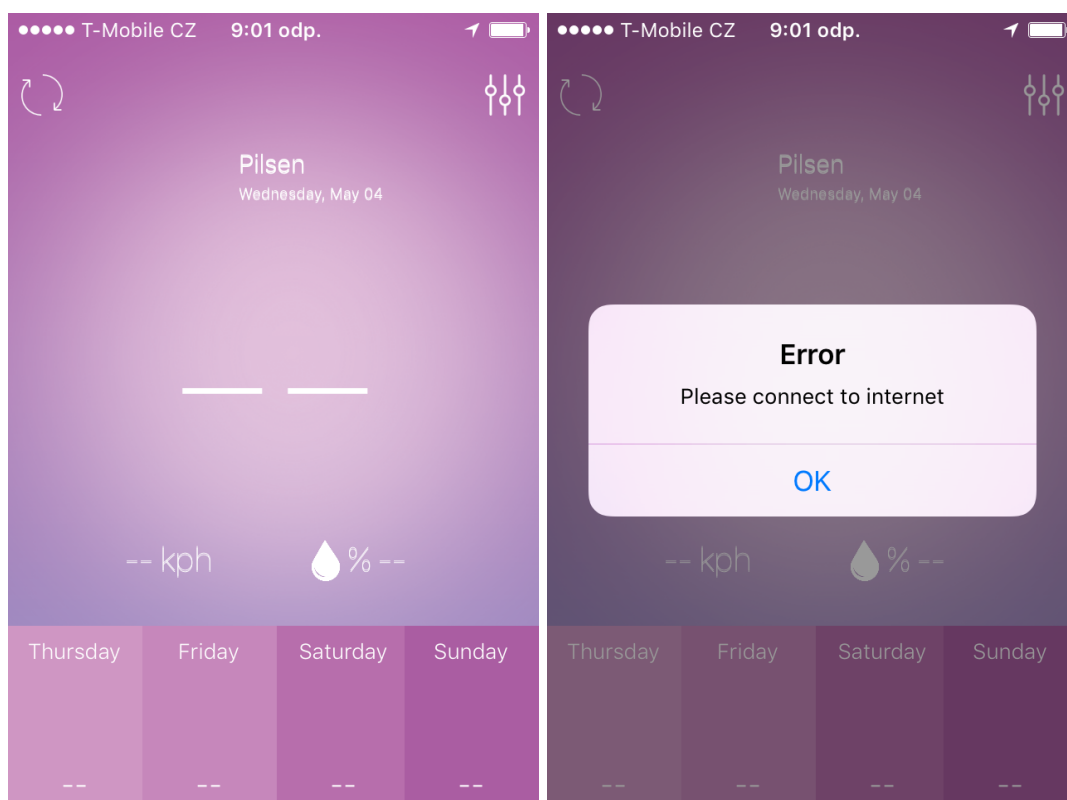
Obrázek 2: Hlavní obrazovka ve stupních celsia a ve stupních fahrenheitu.

Po vstupu do nastavení (viz obr. 3) se v horní části nachází vyhledávací pole měst. Ve vyhledávání funguje automatické vyhledávání měst podle psaného textu (autocomplete). Po kliknutí na dané město se opět vrátíte na obrazovku nastavení. Pod vyhledávacím polem se nachází přepínač s volbou jednotek zobrazované teploty. Pokud nechcete vyhledávat město ručně, můžete využít přepínač, který najde aktuální polohu. Pokud vypnete v nastavení iOS možnost lokalizování, zobrazí se chybová hláška (viz obr. 6). V dolní část jsou informace o aplikaci a autor.

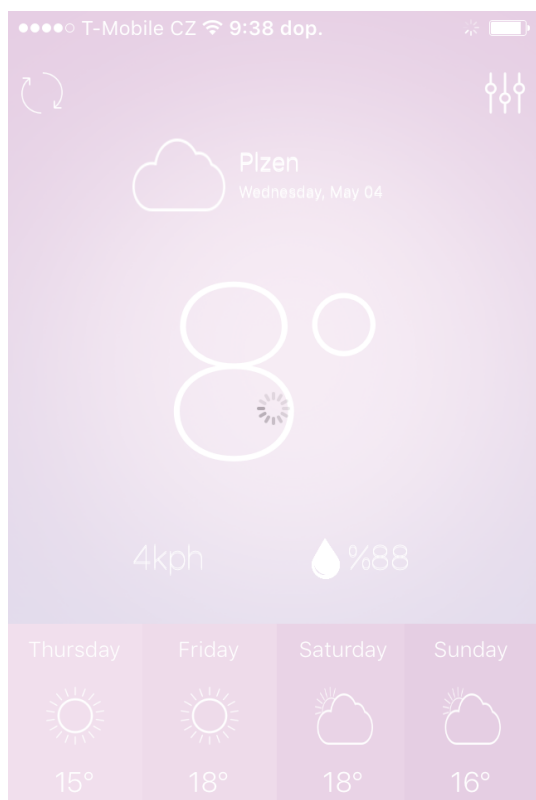
Při vstupu zpět na hlavní obrazovku se může zobrazit otáčející se kolečko (viz obr. 5), které se zobrazuje při načítání nových dat z internetu. V horním pravém rohu, u indikátoru stavu baterie, se také nachází otáčející se kolečko, které značí komunikaci přes internet. Pokud není k dispozici internet, zobrazí se chybová hláška a nezobrazí se žádná data (viz obr. 4).



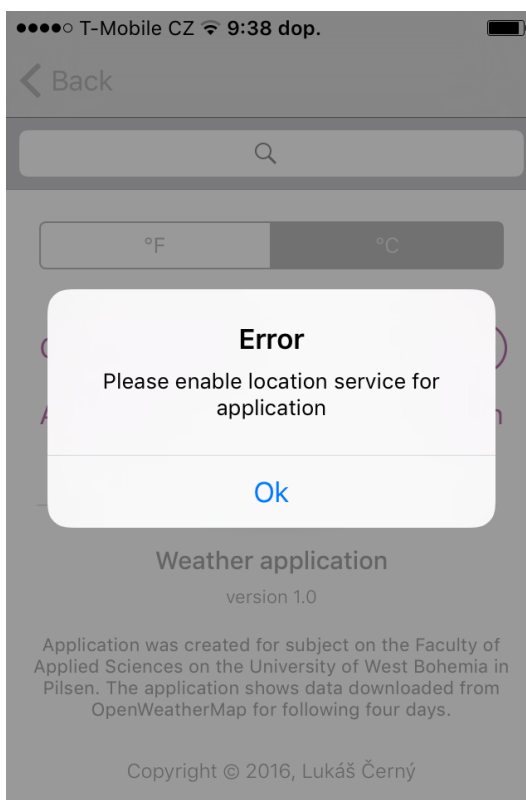
Obrázek 3: Nastavení a výběr města.



Obrázek 4: Výpadek internetu.



Obrázek 5: Načítání nových dat.



Obrázek 6: Chybová hláška lokalizace.

7 Závěr

Aplikace funguje tak, jak bylo prvotně zamýšleno. I na starším telefonu iPhone 4S běží plynule a neseká se. Dlouhodobější testování jsem zatím neprováděl, ale při krátkodobém používání se mi nestalo, že by při běhu aplikace došlo k pádu. Relevantnost zobrazovaných dat je těžké posoudit, jelikož tyto údaje lze vypořizovat až při dlouhodobějším testování. Pokud by data neodpovídala realitě, je jednoduché změnit API a převodník dat a aplikace by fungovala dál.