

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Teze diplomové práce

**Výpůjční informační systém malého vědeckého
pracoviště**

Vedoucí práce: doc. Ing. Vojtěch Merunka, Ph.D.

Autor: Bc. Zdeno Dubnička

© 2016 ČZU v Praze

Souhrn

Tato práce pojednává o vývoji výpůjčního informačního systému pro katedru na FJFI ČVUT. Nejprve je popsáno pracoviště, na kterém bude systém využíván a hardware, který jej bude obsluhovat. Poté jsou shrnuty všechny požadavky, které bude uživatelka na systém mít. Po jejich shrnutí byl vybrán programovací jazyk vývoje. Vzhledem k tomu, že server je spuštěn na operačním systému Mac OSX od společnosti Apple, byly vybrány jazyky Objective-C a Swift. Dále bylo potřeba zvolit databázové řešení, které se postará o veškerou práci s daty systému. Vzhledem k jejich dnešní popularitě a možnostem rozšíření do budoucna byl zvolen cloudový databázový poskytovatel PARSE. Poté byla vybrána metoda samotného vývoje aplikace, kterou se stala česká metoda BORM, a mohlo být přistoupeno k samotnému vývoji. Byla provedena podrobná analýza problému a vypracován návrh jeho řešení. Ten byl následně rozpracován pro implementaci v konkrétním programovacím jazyce a bylo vytvořeno grafické uživatelské rozhraní, přes které bude aplikace ovládána. Závěr práce je věnován způsobům, jak je možné aplikaci rozšířit v budoucnu.

Klíčová slova: Výpůjční systém, Mac OSX, Objective-C, Swift, Xcode, vývoj software, cloud, databáze, App Store, grafické uživatelské rozhraní

1 Cíl práce a metodika

1.1 Cíl práce

Záměrem práce bylo navrhnout a naprogramovat řešení výpůjčního informačního systému pro vědecko-výzkumné pracoviště FJFI ČVUT. Cílem práce je plně využít server, který je již na pracovišti instalován a navrhnout pro něj optimálně řešenou aplikaci. Snahou autora bylo vytvořit aplikaci s jednoduchým a přehledným uživatelským rozhraním, kterou bude uživatel schopen ovládat intuitivně a zároveň také aplikaci optimalizovat pro konkrétní řešení databáze, aby bylo možné nakládat s daty co nejefektivněji. Posledním cílem autora bylo umožnit aplikaci správnou volbou komponentů její budoucí rozvoj a možné rozšiřování.

1.2 Metodika

Použité postupy a obecné informace obsažené v této práci vychází z převážné části ze zkušeností autora. Autor pracoval čtyři roky jako člen IT oddělení renomované nadnárodní pojišťovny, kde se podílel na uživatelské a technické podpoře produktů společnosti. Zde nasbíral zkušenosti, jak by mělo ideálně vypadat uživatelské rozhraní, aby byla práce s aplikací přímočará. Také působí dva roky jako vývojář cloudového ERP systému NetSuite, což mu poskytlo náhled na proces vývoje aplikací a programátorské zázemí. Autor se také aktivně věnuje platformě Apple od roku 2008 a o jazyk Swift se zajímá již od uvedení jeho beta verze. Veškeré ostatní informace byly získány studiem literatury, která je v práci beze zbytku citována. Klíčovým zdrojem informací byla oficiální dokumentace k programovacímu jazyku Swift vydaná přímo společností Apple: *The Swift Programming Language, Swift 2 Edition*.

Pro samotný vývoj bylo využito české metodiky vývoje softwaru BORM, která je v této práci také popsána. Nejprve byly sebrány požadavky uživatele na systém. Po jejich podrobné analýze bylo navrženo konečné řešení a vytvořen design samotného Výpůjčního Systému. Tento design byl následně implementován v konkrétním programovacím jazyku.

2 Přehled řešené problematiky

Stolní platforma Mac OSX od společnosti Apple se stává celosvětově stále úspěšnější. Operační systém Windows má stále náskok v celkovém počtu uživatelů, což je dáno především jeho firemním nasazením a také faktem, že jsou na něj uživatelé zvyklí a většinou to bývá první operační systém, se kterým přijdou do styku. Nicméně i přesto lze najít využití, kde je operační systém Mac OSX vhodnějším kandidátem, než právě Windows nebo Linux. To byl důvod, proč byl pro vědecké pracoviště na FJFI ČVUT vybrán jako malý server právě tento systém.

Jeho konfigurací se zabýval autor ve své bakalářské práci. Na tu také navazuje zadání této práce, kdy byl požadován Výpůjční Systém pro stejné pracoviště, který by mohl být obsluhován právě na tomto serveru. Z požadavků na systém totiž vyplynulo, že nebude třeba neobvykle vysoký výpočetní výkon ani nebudou kladeny jiné specifické požadavky na server a na stávajícím hardwaru nebude problém systém provozovat.

Práce je rozdělena do tří hlavních logických celků. První část popisuje obecná specifika vývoje aplikace pro Mac OSX. Jsou zde rozebrány možnosti, které vývojáři dnes mají, pokud se rozhodnou svou aplikaci vydat právě pro tento operační systém, podmínky pro zapojení se do celého ekosystému Apple a také možnosti, pokud by se tomuto zapojení chtěl vývojář vyhnout.

Proběhlo vyjmenování programovacích jazyků, pomocí kterých je možné požadovanou aplikaci vytvořit. Každý jmenovaný jazyk byl vybrán z určitého důvodu a tyto důvody jsou také spolu s výhodami a nevýhodami každého jazyka v textu rozebrány. Jazyky, nad kterými autor uvažoval byly: Java, Smalltalk, Objective-C a Swift. V závěru této části byly vybrány konkrétní jazyky použité pro vývoj, kterými jsou hlavně nejnovější oficiální programovací jazyk Swift a také Objective-C, ze kterého však byly použity pouze dílčí knihovny. Každý z těchto jazyků je následně podrobněji popsán a charakterizován.

Z výběru jazyka bylo jednoznačné, že pro vývoj bude použito oficiální vývojové prostředí společnosti Apple, Xcode. Díky tomu se dostane aplikaci plné podpory systémových knihoven a také je toto prostředí pro vývoj aplikací pro Mac OSX plně optimalizováno.

Navíc poskytuje komfort vývoje kompletní aplikace pouze v jediném okně, kdy jsou všechny prvky (zdrojový kód, design uživatelského rozhraní, výstupy z testování i nastavení samotné distribuce) přístupny na jednom místě.

3 Vlastní řešení

Druhá část obsahuje počátek samotného vývoje Výpůjčního Systému. Nejprve je proveden sběr požadavků na systém a z něj je následně vypracován úvodní návrh aplikace. Po jeho akceptaci je následně zpracován podrobný návrh aplikace včetně jejího datového modelu a způsobu naprogramování. Také je zde popsán výběr databáze, kterou se stalo cloudové databázové řešení PARSE. To neomezí výkon aplikace, ale poskytne jí potenciál růst v budoucnu z pohledu dalších služeb, které může poskytnout. Také je poskytováno bezplatně, pokud počet požadavků nepřekročí 30 dotazů za vteřinu, což při softwaru pro jednoho uživatele nemůže nastat.

Následně bylo navrženo uživatelské rozhraní aplikace a bylo vytvořeno pomocí prostředí Xcode spolu s veškerým chováním oken a jejich polí. Toto uživatelské rozhraní bylo následně propojeno se zdrojovým kódem, který obsluhuje pozadí aplikace, tedy zápis do a čtení z databáze.

Pro názorně předvedení výsledného chování aplikace byla věnována část práce také popisu vzorové výpůjčky. Každý krok je rozebrán a doplněn otiskem obrazovky daného okna aplikace, aby bylo zřejmé, jaké informace se v každou chvíli zobrazí uživateli. Zde je jasně vidět přínos zvolení designu aplikace, ve které uživatel pracuje vždy pouze s jedním aktivním oknem. Nedochozí k nejasnostem, které okno je aktivní a odkud uživatel na dané místo aplikace přišel.

Na konci práce bylo nastíněno několik možností, kam by se Výpůjční Systém mohl ubírat v budoucnu. Jsou mezi nimi například: integrace na fakultní Active Directory server, možnost vývoje mobilní a webové verze aplikace anebo například také její zpřístupnění zbytku fakulty, pokud by o to byl zájem a bylo by svoleno k částečnému přepracování systému, aby toho byl schopen. Prostředky použité při vývoji i schopnosti použitých programovacích jazyků umožňují plně využít nejmodernějších metod a řešení.

4 Závěr

Z výsledné aplikace vyplývá, že se vytyčené cíle podařilo naplnit. Aplikace má velmi jednoduché a přímočaré uživatelské rozhraní, které uživatele nepřetěžuje množstvím oken, ale zároveň umožňuje provést veškerou nutnou práci. Díky použití programovacího jazyka Swift a vývojového prostředí Xcode bylo možné jednoduše využít všech systémových knihoven a nástrojů. To umožnilo nejen jednoduché testování a řešení případných problémů při vývoji, ale také to do budoucna otevírá cestu k hlubšímu propojení aplikace s operačním systémem a tím také k rozšíření její funkcionality.

Seznam vybraných použitých zdrojů

Knižní publikace

MERUNKA, Vojtěch. *Objektové modelování*. 1. Vyd. Praha: Alfa Nakladatelství, 2008. 184 s. ISBN 978-80-87197-04-2

KOCHAN, Stephen G. *Objective-C 2.0, Výukový kurz programování pro Mac OSX a iPhone*. Dotisk 1.Vyd. Brno: Computer Press, 2011. 550 s. ISBN 978-80-251-2654-7

ČADA, Ondřej. *Cocoa: úvod do programování počítačů Apple*. 1. vyd. Praha: Grada, 2009. 199 s. ISBN 978-80-247-2778-3

ČADA, Ondřej. *Objektové programování: naučte se pravidla objektového myšlení*. 1. vyd. Praha: Grada, 2009. 200 s. ISBN 978-80-247-2745-5

Internetové zdroje

Cocoa and Objective-C cookbook: move beyond basic Cocoa development using over 70 simple and effective recipes for Mac OS X development [online]. Birmingham, U.K.: Packt Pub., 2011 [cit. 2016-03-30]. HAWKINS, Jeff. Dostupné na WWW: <http://site.ebrary.com/lib/natl/Doc?id=10477262>

The Swift Programming Language, Swift 2 Edition [online]. Cupertino, California USA: Apple Inc., 2016. Kolektiv autorů. Dostupné na WWW: https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/index.html#//apple_ref/doc/uid/TP40014097-CH3-ID0