Univerzita Hradec Králové
Fakulta informatiky a managementu

Mobilní aplikace pro uchazeče a studenty FIM UHK

BAKALÁŘSKÁ PRÁCE

Jan Motyčka

Hradec Králové, Srpen 2016

Prohlášení

Prohlašuji, že tato bakalářská práce je mým vlastním dílem, které jsem vypracoval samostatně. Všechny zdroje, ze kterých jsem čerpal, jsou uvedeny včetně odkazu na příslušný zdroj.

Vedoucí práce: Ing. Pavel Kříž, Ph.D.

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce Ing. Pavlu Křížovi, Ph.D. za odborné vedení, za pomoc a rady při zpracování této práce.

Anotace

Práce se zaměřuje na popis aplikace vytvářené pro Fakultu informatiky a managementu a také pro případné další fakulty Univerzity Hradec Králové. V práci je popisován postup vývoje aplikace od prvotních uživatelských cílů přes analýzu a implementaci až k příručce používání, souhrn užitečných vlastností a uživatelských postupů, které aplikace nabízí a mimo jiné jsou popsány i platformy, pro které je aplikace vytvářena. V samotném závěru je zmiňován přínos aplikace pro studenty a uživatelská příručka.

Abstract

The work focuses on the description of applications created for the Faculty of Informatics and Management, and other faculty of the University of Hradec Kralove. This work describes the application development process, from initial user goals through analysis and implementation to guide use a summary of useful features and user procedures which applications and features among others are described and the platforms on which the application is created. In conclusion, it is mentioned benefits applications for students and a user manual.

Klíčová slova

Nette, Android, UHK, aplikace, vývoj, UML, diagram případu užití, diagram tříd, PHP

Obsah

1. Úvod

Dnešní doba by se jen těžko obešla bez informačních systémů. Člověk má čím dál větší potřebu si zjednodušovat a ulehčovat práci. Čím více roste chuť ulehčit si práci, tím více roste poptávka po vývoji softwaru, který v ideálních případech minimalizuje, či úplně nahradí lidskou práci. Právě proto jsou informační technologie jednou z nejvíce rostoucích oblastí.

Cílem mé bakalářské práce je analyzovat a navrhnout komplexní aplikaci pro uchazeče a studenty Fakulty informatiky a managementu Univerzity Hradec Králové. Hlavním přínosem pro uchazeče je fakt, že jim bude simulovat roli původce při dni otevřených dveří, kde se mimo jiné dozví o právě probíhajícím programu na fakultě. Pro studenty, kteří si podali přihlášku, aplikace nabízí informace o přípravných kurzech a veškeré podrobnosti o přijímacím řízení. Pro stávající studeny je připraven přehled všech možných událostí a na základě toho přehledu si student může sám dopředu naplánovat, kterou událost v nejbližší době navštíví.

Vývojem informačních systémů, zejména webových aplikací, se již zabývám několik let. Doposud jsem žádné zkušenosti s vývojem aplikací pro mobilní zařízení neměl, ale i tato oblast mě velice zaujala a chtěl jsem vyzkoušet, co vše nabízí. Všechny uvedené důvody tedy vedly k jednoznačnému vybrání tématu bakalářské práce.

2. Analýza a návrh systému

2.1. Systém a analýza požadavků

Systém obsahuje několik základních entit, které tvoří strukturu systému. Hlavním stavebním kamenem jsou fakulty, dle kterých se odvíjejí další cesty systémem a také zařazení jednotlivých spravujících uživatelů. Pod jednotlivými fakultami se nabízejí události, připravované ke dni otevřených dveří, přípravné kurzy, přijímací řízení. Aby ovšem byla tato evidence dostatečně informující, jsou evidovány jednotlivé obory, druhy studia a také formy studia. V neposlední řadě jsou zaznamenávány i budovy, podlaží a místnosti pro upřesňující informace a také jednotlivé uživatelské účty.

Uživatelské role v administračním rozhraní se rozdělují na dva druhy. Prvním je administrátor a druhým je účet s plnými právy s názvem super administrátor. Oba účty jsou určeny pro pracovníky univerzity.

2.1.1. Fungování systému

Systém, který je vytvořen v rámci bakalářské práce, bude sloužit zejména uchazečům o studium a současným studentům. Nicméně velice podstatnou roli sehrají i zaměstnanci univerzity, bez kterých by tento systém neměl žádný význam, a kteří budou tento systém plnit daty přes webové rozhraní.

Krok, který je pro všechny uživatele front-endové části aplikace nutný absolvovat, vede k vybrání fakulty ze seznamu. Tento krok je nutný vykonat pouze jednou, pokud si uživatel nebude později přát změnit fakultu. Poté se podle vybrané fakulty zobrazí data, která náleží dané fakultě. Následující kroky vedou k výběru dat, přidaných od správců systému. Uživatel si může zobrazit program dne otevřených dveří na fakultě, nebo informace o přípravných kurzech

k přijímacím zkouškám, či budoucímu studiu a v neposlední řadě jsou nabízeny informace také o přijímacím řízení.

Pokud je uživatelem osoba s přístupovými právy k administračnímu rozhraní, přihlásí se.

Je-li uživatel v roli administrátora, mohou jeho první kroky vézt k přidání události do programu dne otevřených dveří, administrátor vyplní formulář a v neposlední řadě se rozhodne, zda chce zveřejnit událost hned nebo až později, poté formulář odešle. Obdobný postup se opakuje i u přípravných kurzů a přijímacích řízení. Pokud se ovšem administrátor rozhodne přidat přijímací řízení, musí se rozhodnout, zda je přijímací řízení formou testu, či jiné formy. Pokud by bylo formou testu, musí uvést informace o datu a času v opačném případě je vyžadována poznámka, kde administrátor sdělí, jakou formou se přijímací řízení koná. Dále se může administrátor rozhodnout, že chce rozeslat zprávu např. o změně místnosti v události. Vyplní tedy formulář a odešle. Danou zprávu si poté přečtou jak uživatelé přistupující přes webovou stránku tak uživatelé využívající služeb mobilní aplikace. Poslední možnost, kterou administrátor smí vykonat je vytvářet užitečné odkazy. Administrátorovi je umožněno zapnout nebo vypnou kontrolu obsazenosti místnosti při práce s novými položkami.

Pokud se daný uživatel na začátku přihlásil jako super administrátor, je mu samozřejmě umožněno využití stejných služeb jako administrátorovi. Super administrátor se od uživatele s omezenými právy odlišuje tím, že může vytvářet ostatní uživatelské účty, dále může zaevidovat budovu a k ní příslušná podlaží a místnosti. Dále vytváří obory a druhy studia. Vše probíhá formou vyplnění formuláře a uložení.

Běžnou pracovní rutinu administrátora ale nelze vykonat, pokud super administrátor nevyužije svá vyšší práva k vyplnění základních dat, která byla zmiňována, a která jsou nutnou podmínkou pro fungování větší části aplikace.

2.1.2. Požadavky na systém

Požadavky patří mezi základní prvky při prvotním návrhu informačních systémů. Pomáhají nám definovat užitečné funkce systému a odfiltrovat je od těch nepodstatných. Odpovídají nám na otázku: „Co by měl systém dělat?“. Při nedůsledném výběru požadavků se mohou náklady na vývoj mnohonásobně zvýšit nebo může celý projekt ztroskotat. Požadavky systému také slouží jako podklad pro Use Case diagram (diagram případů užití), který si zmíníme později.

Požadavky na systém se dělí do dvou kategorií. První kategorie nese název funkční požadavky, které představují chování systému. Druhou kategorií jsou non-funkční (nefunkční) požadavky, které popisují technické specifikace zázemí aplikace. Například definuj dobu odezvy serveru, programovací jazyk nebo použitou technologii apod. [1].

Systém řeší evidenci událostí, přípravných kurzů a přijímací řízení. Jako doprovodné informace eviduje i jednotlivé budovy, podlaží, místnosti a také obory, formy a druhy studia. Dále jsou evidovány i jednotlivé fakulty, ale manipulace s nimi není již možná.

Předpokladem je, aby systém komunikoval i s externími systémy jako je například Facebook nebo IS/STAG. A to z důvodu, aby se obsáhlé rozšiřující informace nemusely udržovat v evidenci systému, ale přesto mohly být využívány. Pro potřeby mobilní aplikace je třeba využívat i služeb od společnosti Google a to konkrétně Google Cloud Messaging.

2.1.2.1. Funkční požadavky

- **Oprávnění a evidence uživatelů**

  V systému bude uchovávána evidence uživatelů. Role uživatelů budou celkem dvě. Každý uživatel smí mít nejvíce jednu roli. Role se základním oprávněním se nazývá administrátor. Je mu umožňováno pracovat s programem pro den otevřených dveří, přijímacími řízeními, přípravnými kurzy, vytvářet upozornění a užitečné odkazy. Nejvyšší oprávnění má role s názvem super administrátor, který je oprávněn ke stejným možnostem

jako administrátor a navíc muže pracovat s evidencí uživatelů. Dále je umožněno super administrátorovi spravovat evidenci budov, podlaží, místností, oborů a druhů studia. U každého uživatele bude veden údaj o přihlašovacím emailu, heslu, přiřazené roli a fakultě, do které spadá. Autentifikaci a autorizaci bude provádět systém během přihlášení uživatelů.

- **Evidence událostí ke dni otevřených dveří**

  Administrátor a také super administrátor budou mít možnost evidovat události ke dni otevřených dveří. Bude se evidovat název místnost, čas začátku události, čas konce události, datum, odkaz na případný obrázek nebo PDF[1] a poznámka. Uživatel si bude moci rozhodnout, zda si nechá kontrolovat obsazenost místnosti v danou chvíli a také zda danou událost zveřejní hned nebo až později.

- **Evidence přípravných kurzů**

  Administrátor a super administrátor budou evidovat přípravné kurzy. Položky, které se budou evidovat, jsou následující: odesílací email, místo konání, jiné umístění, název kurzu, zahájení, rozsah hodin, harmonogram kurzu, garant pracoviště, cena, garant kurzu, určení pro obory, kontaktní osoba a informace. Uživatel si bude moci rozhodnout, zda daný přípravný kurz zveřejní hned nebo až později.

- **Evidence přijímacích řízení**

  Administrátor a super administrátor budou evidovat přijímací řízení. U přijímacího řízení se bude evidovat datum a čas, místnost, forma studia, druh studia, obor a popis. Uživatel si musí rozmyslet, zda se přijímací řízení koná formou například testu nebo pouze vyhodnocením známek apod. Pokud platí první varianta je vyžadován povinně datum a čas, v opačném případě se vyžaduje popis. Opět si uživatel může rozmyslet, zda zveřejní dané přijímací řízení hned nebo až později.

[1] Portable Document Format. Je to přenosný formát dokumentů

- **Zasílání zpráv**

  Administrátor a super administrátor bude moci zasílat zprávy uživatelům front-endu a mobilní aplikace. Evidovat se bude titulek a obsah zprávy.

- **Evidence odkazů**

  Administrátor a super administrátor mohou spravovat užitečné odkazy, které se pak zobrazí uživatelům mobilní aplikace. Evidovány jsou název odkazu a samotný odkaz.

- **Evidence budov**

  Super administrátor bude mít možnost vést evidenci budov. U každé budovy se eviduje pouze adresa.

- **Evidence podlaží**

  Super administrátor vede evidenci podlaží. Podmínkou manipulace evidence je vytvořená budova. Při manipulaci s podlažími je od uživatele vyžadován pouze název, budova je automaticky doplněna systémem. Je evidován název podlaží.

- **Evidence místností**

  Super administrátor může evidovat místnosti. Podmínkou manipulace s touto evidencí je vytvořená budova a podlaží. U místnosti je evidován název a je jeho unikátní číslo.

- **Evidence oborů**

  Super administrátor může evidovat obory. U oboru je evidován název, odkaz pro PDF dokumentaci a druh studia.

- **Evidence druhů studia**

  Super administrátor může evidovat druhy studia. Evidován je pouze název daného druhu oboru.

2.1.2.2. Non-funkční požadavky

- Systém bude dostupný prostřednictvím webového prohlížeče
- Systém bude implementován pomocí PHP[2] a to za použití Nette Framework[3]
- Databáze bude implementována pomocí MySQL
- Design webové aplikace bude responzivní
- Heslo bude šifrováno 160 bitovým klíčem
- Mobilní aplikace bude dostupná i pro operační systém Android[4] 2.3 a vyšší

[2] Hypertext Preprocessor. Jde o skriptovací programovací jazyk
[3] Framework je softwarová struktura, která ulehčuje práci programování aplikace. Obsahuje připravené knihovny nebo doporučené postupy při programování
[4] Je operační systém, určený pro mobilní zařízení

2.2. Diagram případů užití

Diagram případů užití (Use Case Diagram) nám ukazuje, jak se systém chová z pohledu uživatele. Patří mezi nejzákladnější diagramy UML a vytváříme ho obvykle jako jeden z prvních na základě funkčních požadavků. Modelování se skládá z několika kroků. Nejprve je třeba nalézt hranice systému, poté vyhledat aktéry a nalézt případy užití, kde je třeba určit specifikace a alternativní scénář. Tento cyklus se opakuje do té doby, než se ustálí případy užití, aktéři a hranice systému. Obrázek 1 ukazuje diagram případů užití pro vyvíjenou aplikaci [2].

Případ užití (Use Case) symbolizuje definici jedné funkcionality, kterou by měl navrhovaný systém umět. Může v sobě obsahovat další akce, jako například přidání či smazání události, ovšem to v diagramu zachyceno nebude. Při modelování bychom neměli využívat funkční dekompozice a měli bychom využívat tzv. blackboxu (černé skříňce), ve které se skryje vnitřní logika, a my tedy pracujeme pouze s komponentami. Z čehož vyplívá, že by se v diagramu neměli vyskytovat tzv. CRUD[5] operace. [2]

2.2.1. Detailní popis vybraných případů užití

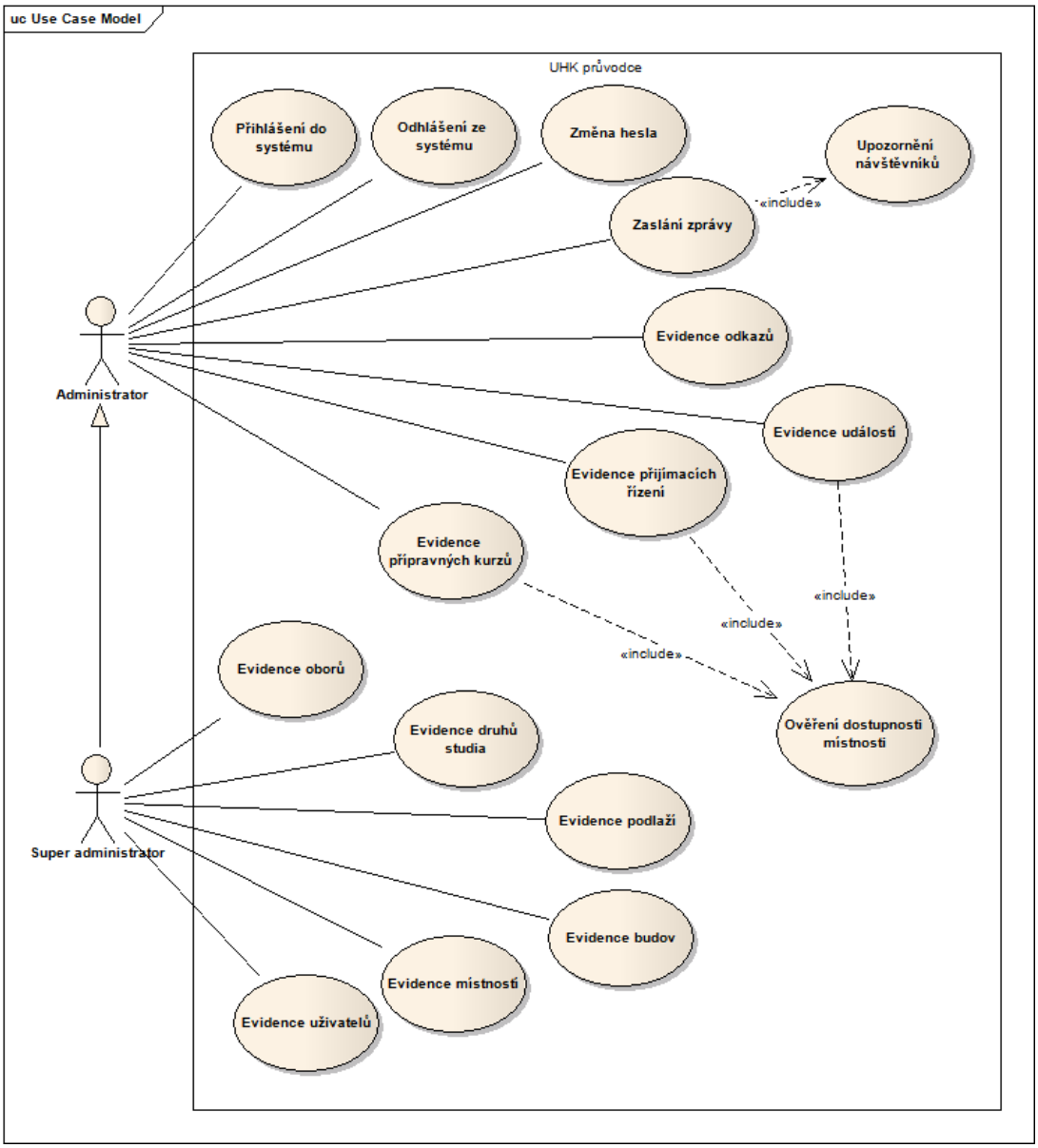
Případ užití: Evidence přípravných kurzů
ID: UC03
Stručný popis: Uživatel eviduje seznam přípravných kurzů
Hlavní aktéři: Administrátor, Super administrátor
Vstupní podmínky: Uživatel je přihlášen do systému
Hlavní scénář: 1. Aktér iniciuje evidenci přípravných kurzů 2. Systém vyhledá seznam přípravných kurzů
Výstupní podmínky: Seznam přípravných kurzů je zobrazen

Případ užití: Přihlášení do systému
ID: UC01
Stručný popis: Přihlášení uživatele do systému
Hlavní aktéři: Administrátor, Super administrátor
Vstupní podmínky: Uživatel není přihlášen do systému

[5] Create, Read, Update, Delete. Základní operace se záznamem v trvalém uložišti.

Hlavní scénář: 1. Uživatel iniciuje přihlášení do systému 2. Systém zobrazí formulář pro přihlášení 3. Uživatel vyplní a odešle formulář 4. Systém ověří přihlašovací údaje 5. JESTLIŽE přihlašovací údaje jsou správné: 5.1. Systém přihlásí uživatele 5.2. Systém přidělí práva uživateli 6. JINAK: 6.1. Systém zobrazí upozornění na chybné přihlašovací údaje
Výstupní podmínky: Uživatel je ověřen a přihlášen do systému

Případ užití: Zasílání zpráv
ID: UC05
Stručný popis: Přihlášený uživatel zasílá zprávu nepřihlášenému
Hlavní aktéři: Administrátor, Super administrátor
Vstupní podmínky: Uživatel je přihlášen do systému
Hlavní scénář: 1. Uživatel iniciuje zasílání zpráv 2. Systém zobrazí formulář pro zadání obsahu zprávy 3. Uživatel vyplní formulář a odešle 4. Systém upozorní nepřihlášeného uživatele na novou zprávu
Výstupní podmínky: Nepřihlášený uživatel obdržel zprávu



Obrázek 1: Diagram případů užití navrhovaného systému

2.3. Diagram tříd

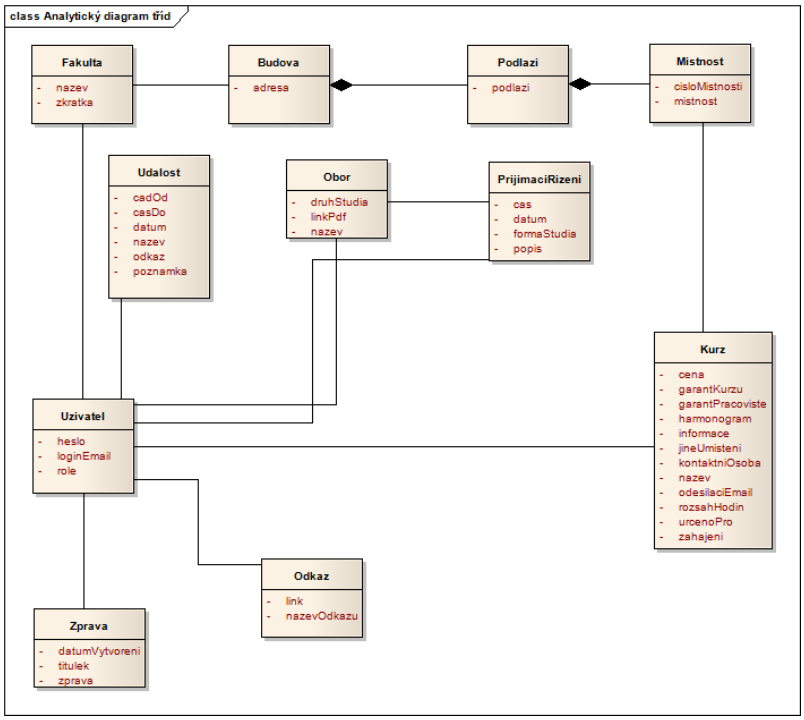
Diagram tříd představuje „statický pohled na modelovaný systém“ [3]. Diagram se dělí do tří úrovní: analytický model, návrhový model a implementační model.

Třída se skládá z atributů, operací a relací k jiným třídám. Lze jí chápat jako předpis pro konkrétní objekt. [4]

2.3.1. Analytický model

Analytický model představuje základní pohled na systém. Je modelován bez operací a případně i bez atributů. Tento model není závislý na konkrétním programovacím jazyku.

Doporučený postup jak hledat analytické třídy je pomocí tzv. CRC[6] štítků. Je to technika, okamžitých nápadů, během níž jsou důležité aspekty problémové domény zachycovány na samolepících štítcích [1].

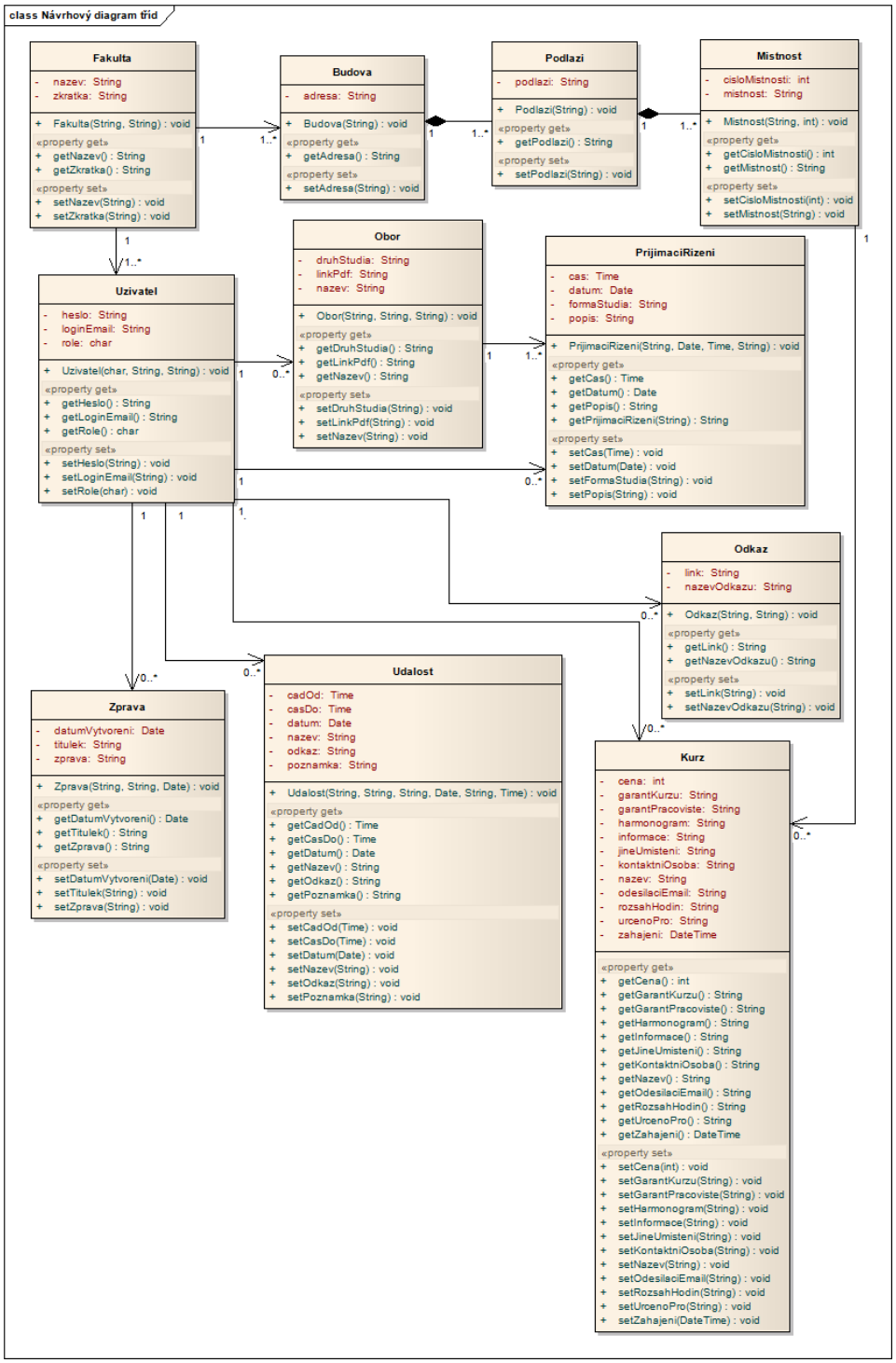


Obrázek 2: Analytický diagram tříd navrhovaného systému

[6] Class, Responsibilities & Collaborators. Třída, odpovědnosti a spolupracovníci

2.3.2. Návrhový model

Je založen na analytickém modelu, který je postupně upřesňován. Doplněny jsou datové typy, multiplicita, operace, do kterých patří mimo jiné i gettry, settry a konstruktor.



Obrázek 3: Návrhový model tříd navrhovaného systému

2.4. Datový model

2.4.1. E-R diagram

V dnešní době obsahují informační systémy nepřeberné množství dat a je tedy třeba dodržovat správnou strukturu, tak abychom mohli s těmito daty pracovat co nejefektivněji. Pro modelování databázové struktury se nejčastěji používá nástroj zvaný entitně-vztahový diagram.

ERD by měl být zcela nezávislý, tedy bez ohledu na to, jak budou data fyzicky uložena, zda se bude jednat od PostgreSQL[7], MySQL nebo textový soubor.

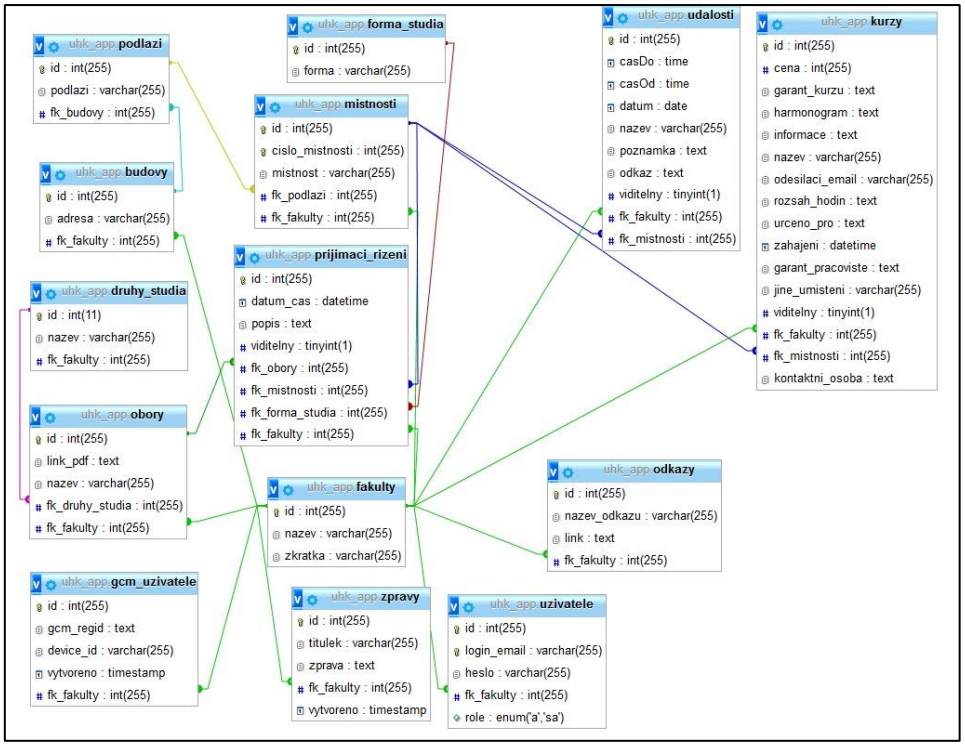
Pro tvorbu ERD neexistuje žádný mezinárodně uznávaný standard, proto se lze také setkat s různým grafickým vyjádřením prvků v diagramu.

Jak již z názvu vyplívá, tento diagram se skládá ze dvou základních stavebních kamenů. Jsou jimi entity a relace. Pod pojmem entita si lze představit objekt, který pochází z reálného světa, jako např. místnost, budova, podlaží nebo přijímací řízení. Entita se skládá z atributů. Atribut nebo také skupina atributů jednoznačně označující záznam se nazývá primární klíč.

Relace je pak vazba nebo také vztah mezi entitami. Takové vztahy lze popsat slovesy.

V souvislosti s ERD se lze také setkat s pojmy jako je volitelnost nebo kardinalita. Volitelnost popisuje, zda daný atribut je povinný, respektive jestli musí mít definovanou hodnotu nebo ne (může být NULL). Druhý pojem, tedy kardinalita vyjadřuje, v jakém množství se něco vyskytuje, zda tedy vyskytuje právě jednou nebo také vícekrát. Kardinalita se používá v souvislosti s relacemi.

[7] PostgreSQL je objektově-relační databázový systém (ORDMBS)

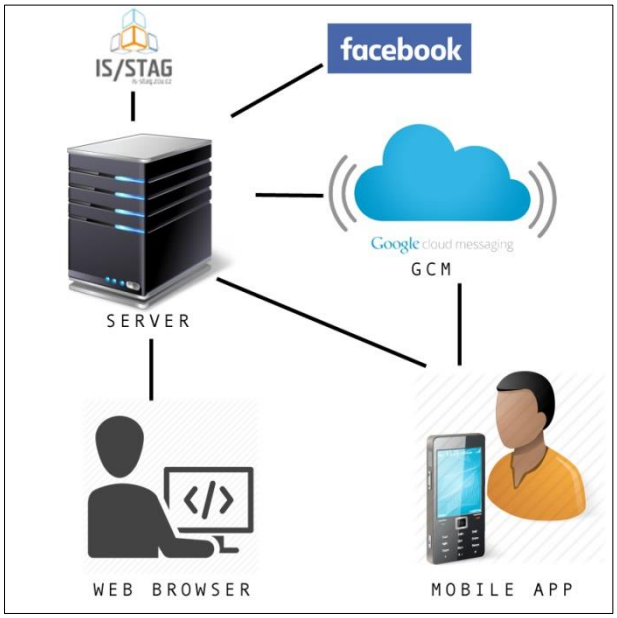


Obrázek 4: Datový model navrhovaného systému

3. Implementace systému

Důležitým prvkem systému je vzdálený server, který zpracovává PHP skripty a obsluhu databáze, dále je to Google Cloud Messaging, sloužící ke komunikaci s mobilním zařízením pro zasílání notifikací, v neposlední řadě je to i webový prohlížeč pro zobrazení stránky uživateli a mobilní aplikace komunikující se vzdáleným serverem. Základní přehled o implementační struktuře webové a mobilní aplikace poskytují kapitoly 3.4.1.2 a 3.6.4.

Server přijímá požadavky a vrací například HTML stránku nebo obrázky apod.



Obrázek 5: Struktura systému

Implementace spočívá v převodu návrhového modelu do spustitelného kódu. Z pohledu analytika nebo návrháře je smyslem implementace tvorba požadovaného implementačního modelu. [1]

Aplikace byla rozdělena dvou podob. První je webová aplikace, přístupná přes webový prohlížeč odkudkoli. Druhá je aplikace určená pro mobilní operační systém Android.

Webová i mobilní aplikace disponuje několika nástroji jako je například Google Cloud Messaging nebo Facebookové API zvané Graph. O všech nástrojích je podrobněji rozepsáno v následujících podkapitolách.

V souboru *config.neon* jsou uloženy klíče pro komunikaci s Facebookem, GCM a pro veřejné API.

Jestliže by se v budoucnu webová aplikace přesouvala na jiný webhosting, je třeba v mobilní aplikaci přepsat adresy, vedoucí k získání dat přes API nabízené webovou aplikací.

Pro tvorbu webové aplikace je použit Nette Framework, založený na programovacím jazyku PHP. Při implementaci bylo využito vývojového prostředí NetBeans 8.0.2., vlastněné a sponzorované firmou Oracle Corporation.

Při implementaci aplikace určené pro operační systém Android bylo využito služeb Android Studia, které je novým (psáno v roce 2016) doporučeným vývojovým prostředím pro tvorbu Android aplikací. Bylo vyvinuto společností JetBrains a nyní je provozováno společností Google. Dříve se k tvorbě mobilních aplikací pro Android využívalo prostředí Eclipse s pluginem Android Developer Tools.

Výhodou Android Studia oproti Eclipse je fakt, že obsahuje Android SDK Tools, kompilátor Android a základní emulátory a to vše bez nutnosti doinstalování [5]

3.1. Google Cloud Messaging

Správce systému může zaslat upozornění uživateli Android aplikace například na změnu místnosti apod., lze k tomu využít zasílání zpráv.

Jestliže má být nová zpráva zobrazena uživateli Android aplikace hned po tom co byla odeslána, tak aby si nemusel stále otevírat aplikaci a kontrolovat příchozí zprávy, pak lze využít služeb Google Cloud Messagingu (dále jen GCM).

GCM je bezplatná služba, která nám pomáhá odesílat zprávy ze serveru do aplikace pro systém Android. Může se jednat o zprávu obsahující až 4 kB údajů.

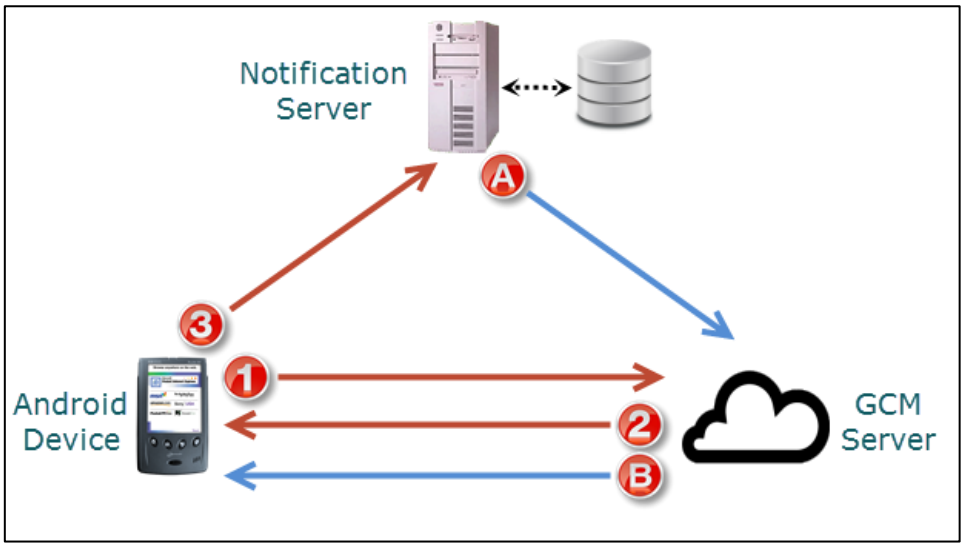
Dnes GCM nahrazuje starší službu C2DM (Android Cloud to Device Messaging).

Tato služba umožňuje trvalé připojení, odesílání zpráv a synchronizaci notifikací. Trvalé připojení umožňuje vývojářům zasílat velké množství dat na mnoho zařízení. Upstream messaging pak umožňuje zasílat data opačným

směrem, tedy z mobilní aplikace na server. Jestliže smaže uživatel danou notifikaci, pak se smaže i v ostatních zařízeních, tento proces označujeme jako synchronizace notifikací. Tyto tři body se vyskytly až jako novinka u GCM a tedy C2DM jej neobsahovaly.

Pro využívání této služby je nejprve nutné zaregistrovat aplikaci na stránkách www.developer.android.com. Poté nám je přiděleno Server Api Key a Sender ID.

Obrázek 6 ukazuje, jakým způsobem GCM funguje.



Obrázek 6: Princip fungování GCM a doručování notifikací (převzato z http://nkeegamedev.blogspot.cz/2013_01_01_archive.html)

Princip GCM je rozdělen na dvě části. V první části probíhá získání registračního ID a ve druhé části dochází již samotnému zasílání notifikace.

Jednotlivé kroky postupu GCM:

1. Mobilní zařízení odešle SENDER_ID na GCM Server pro potřebu registrace zařízení
2. Po úspěšné registraci odpoví GCM Server formou registračního ID

3. Pro potřeby identifikovat zařízení, kterým se zasílají notifikace, odešle mobilní zařízení, obdržené registrační ID na web server pro uložení do databáze

A) Pro zaslání notifikace webová aplikace získá z databáze registrační ID, které pak spolu s obsahem zprávy zašle na GCM server
B) Mezi GCM Server a mobilními zařízeními probíhá persistentní komunikace, proto je mobilní zařízení ihned upozorněno na novou notifikaci

3.2. Graph API

V aplikaci je umožněno sledovat příspěvky a události, které umisťují fakulty na svoji facebookovou stránku. To vše za pomoci Graph API[8].Slouží především pro obousměrnou komunikaci s Facebookem. Lze tedy získat například události, ale také odeslat příspěvek uživateli na zeď. Po navštívení URL[9] je odpovězeno ve formátu JSON[10], který lze dále zpracovat pro získání konkrétních dat.

Poskytován je i nástroj s názvem FQL (Facebook Query Language), který umožnuje dotazovat se na data pomocí syntaxe podobné SQL [6]. Některá data, jako například události z neveřejných facebookových skupin, nelze získat bez autorizace uživatele.

Facebook poskytuje nadstavbu nad svým API v podobě SDK[11], které usnadňuje autorizaci a práci s API. Nicméně, toto SDK není nutné vždy použít. SDK je poskytováno pro JavaScript, PHP, iOS[12] a Android.

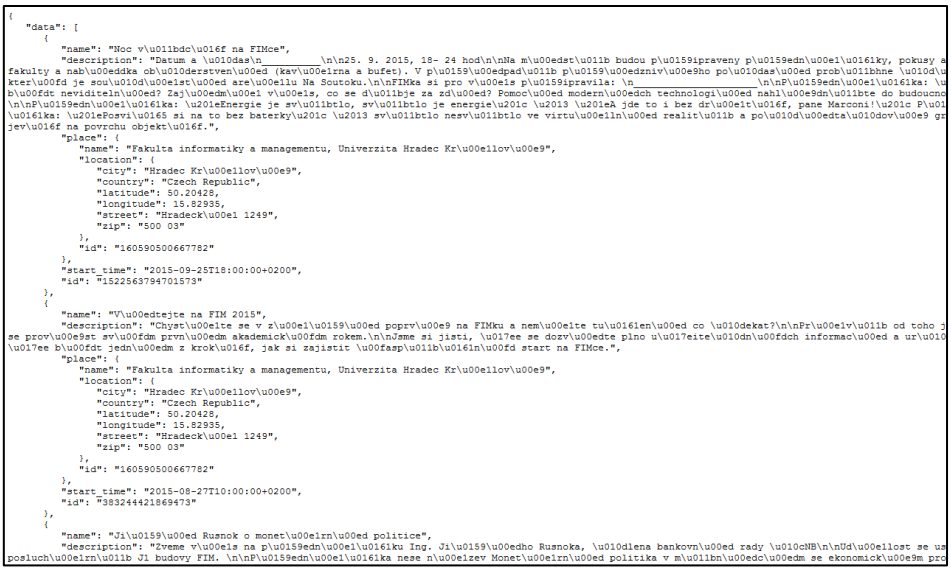
Pro využívání Graphu je nutné aplikaci zaregistrovat, jako tomu bylo u Google Cloud Messagingu. Poté je obdržen access_token, který se vloží do URL při získávání dat. Nabízí k distribuci spoustu drůhu facebookových informací. Jsou jimi například události, příspěvky, informace o profilu (po ověření přes

[8] Application Programming Interface. Rozhraní obsahující sbírku funkcí, tříd nebo knihoven
[9] Uniform Resource Locator. Slouží k přesné identifikaci dokumentů v prostředí internetu
[10] JavaScript Object Notation. Strukturovaný dokument napsaný v JavaScriptu
[11] Software Development Kit. Sada vývojových nástrojů
[12] Mobilní operační systém vyvíjený firmou Apple Inc.

OAuth 2.0) atd. Obrázek 7 představuje způsob složení URL, dle kterého se vyžádá o veřejně přístupné události z facebookové stránky FIM UHK. Obrázek 8 již pak znázorňuje výsledek předchozího volání, který je třeba parsovat a zobrazit uživateli ve smysluplném výstupu.

```
sprintf("https://graph.facebook.com/%s/posts/?access_token=%s&since=%s&fields=created_time,message,link",
$pageId, $fbAccessToken, $sinceUnixTimestamp);
```

Obrázek 7: Ukázka volání URL Graph API pro získání událostí k navrhovanému systému



```
{
   "data": [
      {
         "name": "Noc v\u011bdc\u016f na FIMce",
         "description": "Datum a \u010das\n_________\n\n25. 9. 2015, 18- 24 hod\n\nNa m\u00edst\u011b budou p\u0159ipraveny p\u0159edn\u00e1\u0161ky, pokusy a
fakulty a nab\u00eddka ob\u010derstven\u00ed (kav\u00e1rna a bufet). V p\u0159\u00edpad\u011b p\u0159\u00edzniv\u00e9ho po\u010das\u00ed prob\u011bhne \u010d\u
kter\u00fd je sou\u010d\u00e1st\u00ed are\u00e1lu Na Soutoku.\n\nFIMka si pro v\u00e1s p\u0159ipravila: \n_____\n\nP\u0159edn\u00e1\u0161ka: \u
b\u00fdt neviditeln\u00ed? Zaj\u00edmav\u00e1 v\u00e1s, co se d\u011bje za zd\u00ed? Pomoc\u00ed modern\u00edch technologi\u00ed nahl\u00e9dn\u011bte do budoucnos
\n\nP\u0159edn\u00e1\u0161ka: \u201eEnergie je sv\u011btlo, sv\u011btlo je energie\u201c \u2013 \u201eA jde to i bez dr\u00e1t\u016f, pane Marconi!\u201c P\u01
\u0161ka: \u201ePosv\u0165 si na to bez baterky\u201c \u2013 sv\u011btlo nesv\u011btlo ve virtu\u00e1ln\u00ed realit\u011b a po\u010d\u00edta\u010dov\u00e9 gr
jev\u016f na povrchu objekt\u016f.",
         "place": {
            "name": "Fakulta informatiky a managementu, Univerzita Hradec Kr\u00e1lov\u00e9",
            "location": {
               "city": "Hradec Kr\u00e1lov\u00e9",
               "country": "Czech Republic",
               "latitude": 50.20428,
               "longitude": 15.82935,
               "street": "Hradeck\u00e1 1249",
               "zip": "500 03"
            },
            "id": "160590500667782"
         },
         "start_time": "2015-09-25T18:00:00+0200",
         "id": "1522563794701573"
      },
      {
         "name": "V\u00edtejte na FIM 2015",
         "description": "Chyst\u00e1te se v z\u00e1\u0159\u00ed poprv\u00e9 na FIMku a nem\u00e1te tu\u0161en\u00ed co \u010dekat?\n\nPr\u00e1v\u011b od toho j
se prov\u00e9st sv\u00fdm prvn\u00edm akademick\u00fdm rokem.\n\nJsme si jisti, \u017ee se dozv\u00edte plno u\u017eite\u010dn\u00fdch informac\u00ed a ur\u010
\u017ee b\u00fdt jedn\u00edm z krok\u016f, jak si zajistit \u00fasp\u011b\u0161n\u00fd start na FIMce.",
         "place": {
            "name": "Fakulta informatiky a managementu, Univerzita Hradec Kr\u00e1lov\u00e9",
            "location": {
               "city": "Hradec Kr\u00e1lov\u00e9",
               "country": "Czech Republic",
               "latitude": 50.20428,
               "longitude": 15.82935,
               "street": "Hradeck\u00e1 1249",
               "zip": "500 03"
            },
            "id": "160590500667782"
         },
         "start_time": "2015-08-27T10:00:00+0200",
         "id": "383244421869473"
      },
      {
         "name": "Ji\u0159\u00ed Rusnok o monet\u00e1rn\u00ed politice",
         "description": "Zveme v\u00e1s na p\u0159edn\u00e1\u0161ku Ing. Ji\u0159\u00edho Rusnoka, \u010dlena bankovn\u00ed rady \u010cNB\n\nUd\u00e1lost se us
posluch\u00e1rn\u011b J1 budovy FIM. \n\nP\u0159edn\u00e1\u0161ka nese n\u00e1zev Monet\u00e1rn\u00ed politika v m\u011bn\u00edc\u00edm se ekonomick\u00e9m pro
```

Obrázek 8: Ukázka JSON dokumentu z facebookové stránky FIM UHK

3.3. Stag WS

Stag je systém pro správu studia studentů, postavený na databázi Oracle. Obhospodařuje známky, zkouškové termíny, kompletní správu předmětů apod. Kromě portálového rozhraní nabízí i rozhraní webových služeb.

Stag WS jsou webové služby studijní agendy IS/STAG, které umožňují jiným aplikacím komunikovat s tímto systémem.

Webová služba je místo na webu, které má svoji adresu URL. Operace s těmito službami se pak provádí pomocí volání těchto URL a každá taková operace čte nebo zapisuje data do IS/STAG.

Jsou dva druhy standardu webových služeb:

- **SOAP** – protokol pro zasílání zpráv XML[13],
- **REST** – je oproti SOAP orientován datově, nikoli procedurálně

Obě zmiňované služby fungují tak, že po přístupu na URL je získán soubor s výstupem služby. Může jim být např. soubor typu XML, JSON, CSV[14], XLS[15] apod.

Aplikace, které je věnována tato celá práce, využívá hned několik volání URL ze Stag WS. Důvodem volání je snaha získat seznam předmětů vyučovaných na dané fakultě. Pokud nejsme ve Stagu přihlášeni, pak nastává složitější cesta, jak tyto předměty získat. Je třeba využít nejprve funkce *getStudijniProgramy*, která nám vrátí studijní programy, ale především tzv. stprIdno, což je identifikátor pro

studijní program. Poté je volána funkce *getOboryStudijnihoProgramu*, na základě, které je získán tzv. oborIdno, tedy identifikátor pro daný obor. Nakonec již stačí zavolat funkci *getPredmetyByObor*, případně *getPredmetyByOborFullInfo* a tím je tedy dokončen proces získání předmětů. Obrázek 9 ukazuje zavolání adresy respektive funkce *getPredmetyByOborFullInfo*, která nám vrátí seznam předmětů v XML.

Obrázek 9: Ukázka předmětů v XML ze STAG/IS

[13] Extensible Markup Language. Obecný značkovací jazyk vyvinut konsorciem W3C
[14] Comma-separated-values. Souborový formát určený pro výměnu tabulkových dat
[15] XLS je formát souboru určený pro aplikaci Microsoft Excel

3.4. Webová aplikace

Technické specifikace webové aplikace byly brány s ohledem na non-funkční požadavky. Aplikace je tedy implementována pomocí Nette Framework, značkovacím jazykem HTML5, kaskádovými styly CSS 3. Jako uložiště byla vybrána databáze MySQL. Všechny tyto nástroje jsou hojně podporovány řadou webhostingů, což byl hlavní důvod, proč byly zvoleny.

Pro tvorbu front-endu se využilo služeb knihovny jQuery 1.11.2., která je dnešním standardem mezi vývojáři front-endu. Dalším front-end nástrojem je Twiter Bootstrap 3, který je opět masivně využíván, především pro svoji responzivní všestrannost.

3.4.1. Server-side

3.4.1.1. Nette Framework

Nette Framework patří do rodiny frameworku založených na programovacím jazyku PHP. Při programování webových aplikací velmi usnadňuje práci. Autorem je český vývojář David Grudl. Framework je v Čechách velmi oblíbený a hojně používaný. Využívají ho projekty jako např. ČSFD, Slevomat, ale také webové stránky Václava Klause.

Jako abstraktní databázovou vrstvu využívá PDO, které využívá objektově orientovaných vlastností PHP, což vede k vyspělejšímu a efektivnějšímu přístupu k databázové komunikaci.[7]

Celý Framework je postaven na architektuře MVP, která obsahuje 3 druhy komponent. Jsou jimi modely, pohledy a presentery.

Presentery

Starají se o řízení mezi pohledem a modelem. Při zavolání presenteru mu jsou předány parametry a poté je vrácena HTML stránka. V praxi je funkčnost taková, že presenter předá data modelu, Ten data zpracuje (obvykle pracuje s databází) a pošle zpět presenteru, který je předá do šablony (pohledu), kde je

kolem oněch dat vytvořený i HTML kód. Lze tedy definovat, že presenter je prostředníkem komunikace mezi modelem a pohledem.

Modely

Model se skládá z různých datových entit (přípravný kurz, událost, přijímací řízení, uživatel apod.). Jednou z hlavních činností modelu je komunikace s datovým uložištěm, ve většině případů s databází. Pro shrnutí lze říci, že model obstarává hlavní logiku aplikace.

Pohledy

Pohled nebo také šablona tvoří výstup aplikace. Surová data, která jsou obdržena od presenteru obdrží v šabloně grafické formátování. V šabloně se setkat s velmi užitečným nástrojem v podobě Latte, což je šablonovací jazyk, který umožňuje vkládat data z PHP za pomoci speciálních značek. Po zkompilování šablony je výsledný soubor kombinací HTML a PHP nicméně zdrojový soubor je vždy kombinací Latte jazyka a HTML. Vývojář je tady chráněn před tvorbou tzv. špagetového kódu, tedy míchání HTML a PHP dohromady a velmi rychle může zeditovat pro případné úpravy.

Základní prvky Nette a řešení zabezpečení

Mezi silné zbraně Nette patří formuláře, podpora Ajaxu[16], práce s databází a ladící nástroje. Postupným vývojem se Nette rozštěpil na několik částí, tak, že jsou použitelné i bez vzájemné spolupráce. Součástí projektu Nette je tedy samotný framework, Tracy („laděnka“ pro hledání chyb v kódu), Latte (šablonovací systém) a Tester (testovací nástroj).

Mezí hlavní důvody vybrání Nette Frameworku pro implementaci webové aplikace, se vedle uživatelské přívětivosti, výborné optimalizace řadí i zabezpečení.

[16] Asynchronous JavaScript and XML. Technologie měnící obsah stránek bez nutnosti znovunačtení

Mezi nejčastější útoky potencionálních útočníků patří Cross-Site Scripting (XSS), který spočívá v neošetřených vstupech a útočník pak může uživateli podstrčit javascriptový kód a získat tak citlivé údaje nebo poškodit vzhled stránky. Nette využívá technologii Context-Aware Escaping, která všechny výstupy automaticky ošetří.

Další útokem může být CSRF (Cross-Site Request Forgery), který vykoná útok na aplikaci, kde je uživatel právě přihlášen. Ochranou je pouhé zavolání funkce *addProtection()* na formuláři v administrační fázi.

Samozřejmostí je i ochrana před takovými útoky jako je URL attack, control codes, invalid UTF-8[17]. V neposlední řadě Nette zabezpečuje i před session hijacking apod., tedy druhy útoků, která jsou schopné zcizit nebo podstrčit session ID a tím získat přístup do webové aplikace.

3.4.1.2. Struktura aplikace

Adresářová struktura je dodržována dle oficiálního doporučení Nette frameworku.

app	adresář obsahující celou aplikaci
AdminModule	modul pro správu administrace frontendu
Config	konfigurační soubory NEON[18]
FrontModule	modul obsahující veřejnou část aplikace
Model	třídy s aplikační logikou
Presenters	třídy s prezentační logikou
Router	továrna spravující URL adresy
Templates	Latte[19] šablony definující desing webu
Bootstrap.php	zaváděcí soubor aplikace
Libs	knihovny aplikace
Nette	komprimovaná knihovna Nette

[17] UCS Transformation Format. Způsob kódování řetězců znaků Unicode
[18] Nette Object Notation. Formát souboru obsahující dokument se serializací PHP pole či objektu, který využívá velice podobnou syntax jako JSON.
[19] Latte je šablonovací systém pro definování designovanu struktury webu

Log	chybová hlášení
Temp	dočasné soubory
www	adresář se soubory pro správu designu
css	kaskádové styly
fonts	fonty
images	obrázky
js	klientské skripty
index.php	inicializace aplikace
.htaccess	soubor konfigurace webového serveru

Presenters/AdminModule

- AdministratorPresenter – komponenty a předávání dat pro události, přípravné kurzy, přijímací řízení, odkazů, změnu hesla, odeslání upozornění
- SuperAdministratorPresenter – komponenty a předávání dat pro přehled budov, podlaží, místností, oborů, druhů studia
- SignPresenter **–** komponenty pro přihlášení a resetování hesla

Model

- AdministratorInfo – komunikace s databází pro události, přípravné kurzy, přijímací řízení, odkazů, změnu hesla, odeslání upozornění
- SuperAdministratorPresenter – komunikace s databází pro práci s budovami, podlažími, místnostmi, obory, druhy studia
- GCM – zasílání notifikací pro Google Cloud Messaging

3.4.2. Client-side

Důležitou součást aplikace tvoří i klientské skripty, které jsou posílané serverem a zpracovávány na straně klienta v internetovém prohlížeči. Do

takových to skriptů patří HTML stránka, JavaScipt soubory, obrázky, CSS[20], JSON nebo XML dokumenty apod.

Nezbytnou složkou celé webové aplikace je i její vzhled, u kterého je třeba intuitivnost a vyhnutí se zbytečným problémům, které by mohly uživatele odradit od využívání aplikace.

Pro tvorbu této klientské části aplikace bylo využito služeb nástrojů jQuery, Twiter Bootstrap 3, které jsou dnešním trendem a jsou tedy hojně využívány i webovými kodéry a návrháři uživatelského rozhraní.

Design byl přizpůsoben barevnému schématu, které využívá Univerzita Hradec Králové. Po přihlášení uživatele do administrace nebo po výběru fakulty u veřejné části pro nepřihlášené uživatele, se nastaví schéma dle fakultního zařazení daného uživatele. Nutnou podmínkou dnešní tvorby webdesignu je plná podpora mobilních zařízení, respektive přizpůsobení se jejich malým displejům, což spolehlivě právě řeší Twiter Bootstrap 3.

3.4.2.1. jQuery

jQuery je nejrozšířenější JavaScriptová knihovna. Je jednoduchá a intuitivní. K práci s ní stačí znalost CSS selektorů, které se běžně využívají při tvorbě CSS. Sama velice dobře řeší nekompatibilitu mezi prohlížeči.

Ve světě webového vývoje se bohužel setkáváme s tím, že každý prohlížeč má nějakou svou skupinu odchylek, jimiž se liší od vydaných standardů. Kvůli tomu musíme věnovat podstatnou část své webové aplikace tomu, abychom používali funkce jinými způsoby na různých platformách. Ačkoliv napříč webovými prohlížeči není možné dosáhnout stoprocentní shody, knihovna jQuery přináší abstraktní vrstvu, jež sjednocuje běžné úkoly a tím zmenšuje velikost zdrojového kódu a zjednodušuje ho.

Pracuje s DOM (objektový model dokumentu), který sleduje a podle toho volá různé části kódu. Umožňuje pracovat s animacemi, CSS, HTML a podporuje interakci s AJAX. Oproti JavaScriptu snižuje množství psaného kódu.

[20] Cascading Style Sheets. Jsou to kaskádové styly, které určují, jak mají být zobrazeny HTML elementy na stránce.

jQuery se dělí na několik samostatných podčástí. Prvním je jQuery UI, který je zaměřený na uživatelské rozhraní, umožňuje pracovat například s interakcemi (draggable, droppable, resizable apod.), widgety (autocomplete, datepicker, progressbar apod.) nebo efekty (show, hide, toggle, add class apod.). Druhou částí je jQuery Mobile, která je optimalizována pro dotyková zařízení. Dalšími částmi jsou Sizzle a QUnit.

Obrázek 10 ukazuje jak se na prvek s id *save-admission* napojí handler, který čeká, až bude na element kliknuto a poté provede kód uvnitř funkce.

```
$('#save-admission').on("click", function () {
    if ($("#studyfield").val() === null) {
        alert("Vyberte prosím obor z nabídky");
        return false;
    }
    else
        return true;

});
```

Obrázek 10: Ukázka použití knihovny jQuery v navrhovaném systému

jQuery umí komunikovat i s PHP a obstarávat tak překreslení stránky. Této technologii se říká AJAX a je jí věnována kapitola 3.5.

3.4.2.2. Twitter Bootstrap 3

Bootstrap 3 je velice jednoduchý framework pro tvorbu moderních webů nebo webových aplikací. Pracuje s technologiemi jako je HTML, CSS nebo JavaScript. Proto, aby mohl vývojář využívat tohoto nástroje, je třeba znalost HTML, CSS ale také jQuery.

Bootstrap vyvinuli Mark Otto a Jacob Thornton z firmy Twitter. Vydán byl pak v roce 2011.

Proto, aby mohl být na webu zprovozněn, je zapotřebí nalinkovat knihovnu jQuery a poté samozřejmě samotné knihovny Bootstrapu. Je vydáván jak v plné verzi, tak v kompilované (bootstrap.min.*).

Komponenty

- Button groups
- Button dropdown
- Navigational tabs, pills, lists
- Navbar
- Labels
- Badges
- Alerts
- Progress bars
- Modals
- Dropdowns
- Tooltips
- Popovers
- Carousel
- a další

Ukázka HTML kódů pro využití CSS třídy *btn* jejíž výsledek znázorňuje Obrázek 11:

```
<a class="btn btn-default" href="/admin/administrátor/preparatory-course"> Nový přípravný kurz</a>
```

Obrázek 11: Tlačítko s CSS třídami btn a btn-default

Obrázek 12 ukazuje akci, která se provede při události *hidden.bs.modal*, tedy při zavření modálních okna. Ukázka demonstruje, jak Bootstrap provádí interakci mezi HTML a uživatelem za pomocí jQuery.

```
$(document).ready(function () {
    $("#modal").on('hidden.bs.modal', function (e) {
        $("#frm-sendNotifForm input, #frm-sendNotifForm textarea").val(null);
        $("#sendmsg").val("Odeslat");
    });
});
```

Obrázek 12: Ukázka jQuery kódu, který obsluhuje událost modálního okna uvnitř systému

3.5. Ajax

Pod zkratkou Ajax je skrývá název Asynchronous JavaScript and XML. Je to technologie, která využívá zdrojů jak na straně klienta, tak serveru.

Pod pojmem Ajax si lze vybavit technologii, která asynchronně, tedy bez nutnosti znovunačtení, změní obsah webové stránky. Při hlubším pohledu na problematiku tato technologie využívá hned několik dalších technologií:

- **HTML, CSS** pro smysluplné zobrazení dat
- **DOM a JavaScript** pro zpracování dynamické změny zobrazených informací
- **XMLHttpRequest** pro samotnou výměnu dat s webovým serverem. Pro přenos dat je využito typicky XML, ale lze také využít JSON nebo HTML

Mezi výhody patří zajisté odstranění nutnosti znovunačtení stránky. Vykreslení nového obsahu na stránce může být způsobeno i interakcí uživatele a to, že například klikne na tlačítko. Poté server zašle pouze, tu část stránky, která se změnila. V šablonovacím systému Latte se takové to části stránky nazývají snippety a jsou obsluhovány metodou s prefixem *handle* v názvu.

Nevýhodou je fakt, že se díky této technologii může zvýšit počet HTTP[21] požadavků. Dalším neduhem může být i to, že jakmile na pozadí probíhá asynchronní operace a uživateli taková to obsluha není signalizována, může jí přerušit, neboť se domnívá, že systém na jeho požadavek nereaguje.

[21] Hypertext Transfer Protocol je internetový protokol, sloužící pro výměnu HTML dokumentů

Obrázek 13 znázorňuje javascriptový kód jako výňatek z latte šablony. Za využití makra {link} se vygeneruje odkaz na handle metodu, která interaguje se serverem a zpětně vrátí JSON dokument, podle kterého javascript překreslí daný snippet na stránce.

```
function fillMsgDlg(id) {
    $("#ajax-spinner").ajaxStart().show(0, function () {
    });

    $.nette.ajax({
        url: {link fillMessageDlg!},
        data: {
            msgId: id
        },
        type: 'GET',
        complete: function () {
            $("#ajax-spinner").ajaxStart().hide(0, function () {
            });

            $("#sendmsg").val("Uložit");
            $("#modal").modal("show");
        }
    });
}
```

Obrázek 13: Ukázka asynchronního naplnění formuláře za pomocí komunikace jQuery s PHP v systému

3.6. Mobilní aplikace pro Android

Android je operační systém určený pro mobilní zařízení. Je založený na jádře Linuxu a je distribuován jako tzv. open source, tedy software s otevřeným zdrojovým kódem.

V srpnu 2013 měl Android bezmála 80% podíl na trhu s chytrými telefony, což je důkaz jeho populárnosti a tento fakt také dává za důvod proč aplikaci pro mobilní zařazení směrovat pro operační systém Android.

Vývoj aplikací pro Android vyžaduje znalost jazyka Java, který je standardem mezi vývojáři. Jelikož pro Android se využívá programovacího jazyka Java, je tedy potřeba mít nainstalovaný JDK (Java Development Kit) a SDK (Software

Development Kit) pro Android, tedy balíčky s vývojovými nástroji. Původním doporučovaným vývojovým prostředím pro vývoj byl Eclipse. V roce 2014 Google odkoupil Android studio a začal ho propagovat jako oficiální a tedy hlavní vývojové prostředí.

Android Software Development Kit (SDK)

SDK nabízí nástroje pro vývoj aplikací pro platformu Android. Tato sada je rozdělena do několika druhů:

Základní konfigurace

- **SDK Tools** – nástroje pro testování aplikace, správu Android Virtual Devices, Android emulátor, apod.
- **Android SDK platforms** – knihovny, systémový obraz, ukázky kódů, atd.

Doporučená konfigurace

- **USB Driver –** nástroj, umožňující ladění a testování na živém zařízení

Plné konfigurace SDK

- **Google API –** knihovny pro využití Google Maps

Emulátor

Emulátor má za úkol simulovat spuštěný Android. Umožňuje testovat aplikace bez fyzického zařízení.

3.6.1. Popis aplikace

Android aplikace byla vyvíjena jako alternativa k webové aplikaci a nabízí pohodlnou práci při zjišťování informaci, pokud by uživatelé nechtěli využít služeb webové aplikace z prostředí mobilního prohlížeče.

Funkční možnosti nabízené mobilní aplikace jsou totožné s webovým front-endem. Nicméně mobilní aplikace oproti webové aplikaci, upozorňuje na zprávy

krátce po tom, co jsou webovým administrátorem odeslány a to vše díky službě Google Cloud Messaging. Taková to zpráva se objeví v upozorňovací části mobilního zařízení a poté navede přímo do aplikace k přečtení. Tato aplikace nabízí ještě jednu výhodu oproti webové aplikaci, a to možnost přímého uložení událostí do interního Google kalendáře.

Nutným požadavkem pro používání aplikace je minimální verze Android 2.3. nesoucí označení Gingerbread.

3.6.2. Základní součásti aplikace

Mezi základní prvky Androidu patří komponenty *aktivity* představující obrazovku, *service* zprostředkující provádění akcí na pozadí, *content providers* pro přístup k datům, *broadcast reciever* reagující na příchozí oznámení. Komponenty mohou mezi sebou interagovat pomocí *intentů*.

Activity

Aktivita je třída, která odpovídá právě zobrazené obrazovce. Aktivita interaguje s uživatelem prostřednictvím GUI (grafického uživatelského rozhraní). Slouží například pro vyplnění formuláře nebo vybrání položky ze seznamu apod.

Na displejích s větším prostorem, například tablety apod., lze realizovat komplexnější úkony, než na menších displejích. Typickým příkladem je master-detail seznam, který na menších displejích zobrazuje seznam položek na dvou obrazovkách, zatím co na větších displejích se zobrazí na jedné obrazovce současně seznam a vedle něho detailní informace o dané položce.

Z hlediska fungování operačního systému je aktivita náročná záležitost. Při spuštění aktivity se musí vytvořit nový proces a alokovat paměť pro komponenty uživatelského rozhraní. O ušetření výpočetních prostředků, například během vzniku, zániku nebo opětovného vzniku aktivity, se stará tzv. Activity Manager, který pracuje se správou životního cyklu aktivity. Activity Manager pracuje se zásobníkem, kde jsou uloženy právě spuštěné aktivity. Vrchol zásobníku tvoří aktivita, která je aktuálně zobrazena.

Aktivita se může nacházet ve čtyřech stavech, během kterých se volají příslušné metody v předem definovaném pořadí.

Stavy aktivity

- **Activity starts**

  Inicializace aktivity. Metody, které se volají:

  1. onCreate()
  2. onStart()
  3. onResume()

- **Acitivity is running**

  Aktivita je právě zobrazována uživateli a může docházet k interakci s uživatelem. Během tohoto stavu může dojít k požadavku zobrazení jiné aktivity do popředí. Původní aktivita pak vyvolá metodu onPause(). Pokud dojde k požadavku na opětovné probuzení aktivity, je vyvolána metoda onResume(). Může ovšem nastat situace, že dochází k nedostatku paměti, poté aktivita přechází do stavu *process is killed.* V další možné situaci může dojít, k tomu, že aktivita není viditelná. V tom případě se volá metoda onStop(). Při této mětodě pak může dojít k opětovnému probuzení a přes metodu onRestart() dochází k zavolání onStart(). Může se ale opět stát, že dojde k nedostatku paměti a místo onRestart() přejde aktivita do režimu *process is killed.* Pokud během metody onStop() nedojde k ničemu zásadnímu, aktivita přechází k metodě onDestroy() a ta pak do stavu *activity is shut down.*

- **Process is killed**

  K tomuto stavu se přechází pokud, dojde k nedostatku paměti. Uživatel může vyslat signál na znovuspuštění aktivity, což vede k zavolání metody onCreate() [8].

- **Activity is shut down**
  Activity Managerem je aktivita ukončena a ta již nevyužívá žádnou paměť [9].

Služby (Services)

Služby zprostředkovávají operace, běžící na pozadí, které mohou trvat i delší dobu. Na rozdíl od aktivit nevyužívají uživatelské rozhraní.

Pro potřeby vyvíjené aplikace bylo těchto služeb využito a to v podobě asynchronního zpracování dat, během které dojde pomocí třídy *AsyncTask* a webového API k zisku dat z databáze uložené na webovém serveru.

Content provider

Content provider nebo také poskytovatel obsahu umožňuje ukládat data nebo sdílet mezi více aplikacemi nebo procesy. Možnosti jak ukládat data je více. Lze využít například souborů, SQLite databáze nebo webu. Content provider nabízí pro práci standardní metody insert, update, delete a query.

Broadcast recevier

Broadcast recevier slouží jako „naslouchač“ oznámení. Podobně jako service ani broadcast recevier nedisponuje uživatelským rozhraním. Objekty na vysílání a přijímání, které pracují na pozadí, reagují na události, které se odehrávají v zařízení. Za události jsou považovány objekty typu Intent, tedy záměr.

Příkladem může být zpracování push notifikací zprostředkovávané službou Google Cloud Messaging, která byla využita při vývoji této mobilní a webové aplikace. V okamžiku, kdy přijde zpráva do telefonu, systém tuto skutečnost oznámí uživateli. Systém ale nemůže nijak vědět, kdy zpráva dojde, proto je vybaven softwarovou službou, která zprávu očekává. V okamžiku, že je zpráva obdržena, služba vyšle záměr. V telefonu je poté přijímač, který záměr spustí. Systém už poté nabídne uživateli upozornění na zprávu.

3.6.3. Některé další použité třídy

DefaultHttpClient

Důležitou součástí aplikace je třída DefaultHttpClient, která umožňuje volat HTTP požadavky. Důležitou součástí je proto, že se stará o přenos dat mezi webovým serverem a mobilní aplikací. Taková to data jsou poté přenášena ve formátu JSON. Tato třída je využívána například pro výpis událostí, přijímacích řízení, přípravných kurzů apod. Tedy při všech akcích spojených s komunikací s databází na serveru.

AsyncTask

Tato třída je určena pro asynchronní úlohy, které běží v samostatném vlákně. Po vykonání úlohy může mít vliv na modifikaci prvků uživatelského rozhraní. V aplikaci je využívána především na načtení dat ze serverové databáze.

Nabízí čtyři metody, které lze implementovat:

- **onPreExecute()**
  Do této metody se zapisuje kód, který se má provést před spuštěním úlohy na pozadí. Využívá se například na zobrazení signalizace průběhu (Progress bar).

- **doInBackground(Params...)**
  Metoda obsahuje příkazy, které se mají provést na pozadí mimo hlavní vlákno. Provede se v okamžiku, jakmile se dokončí metoda *onPreExecute().*

- **onProgressUpdate(Progress...)**
  Tato metoda slouží pro zobrazení stavu průběhu příkazů, které se zpracovávají na pozadí.

- **onPostExecute(Long result)**
  Zde se vykonají příkazy poté, co se dokončí výpočty běžící na pozadí.

SharedPreferences

Tato třída slouží pro ukládání a správu malého množství jednoduše strukturovaných údajů. Údaje se ukládají jako párové hodnoty typu klíč-hodnota. Mezi podporované datové typy patří *string*, *boolean*, *int*, *long* a *float*.

Typicky je tato třída využívána k dlouhodobému uložení dat aplikace, jelikož jsou data ukládána perzistentně a budou dostupné i po ukončení nebo restartu aplikace. V této vyvíjené aplikaci je třída SharedPreferences využita k uložení identifikátoru k dané fakultě, aby si uživatel nemusel vybírat fakultu a ihned přistoupil k požadovaným datům.

Je-li potřeba získat objekt SharedPreferences spojený s aktivitou, je dostupný skrz metodu aktivity *getPreferences(int mode).* Pokud je ovšem potřeba objekt, který není vázaný k aktivitě, pak je dostupný skrz metodu *Context.getSharedPreferences(String name, int mode).*

Zápis se provádí pomocí objektu *SharedPreferences.Editor*, který lze získat přes *SharedPreferences.edit()*. Hodnoty se pak mění pomocí metod [10]:

- putInt(String, int value)
- putString(String key, String value)
- remove (String key)

Následně se změny potvrdí podobně jako v databázové terminologii příkazem commit a to konkrétně *SharedPreferences.Editor.commit()*.

Ke čtení lze využít metody:

- getAll()
- getBoolean(String key, …)
- getString(String key, …)

3.6.4. Balíčky aplikace

Balíčky aplikace jsou rozděleny na aktivity, komponenty a služby.

cz.uhk.app.activities

- **ProtectedActivity** – nadřazená aktivita pro všechny aktivity. Pracuje s interním uložištěm mobilu pro získání id fakulty, zjišťuje, zda je síťové připojení online
- **SubjectsActivity** – zobrazení předmětů ze systému IS/STAG
- **TakerActivity** – zobrazení programu dne otevřených dveří
- **ApplicantActivity** – Rozcestník k přípravným kurzům nebo přijímacím řízením
- **MainActivity** – Zobrazení a zvolení fakulty. Vybranou fakulty uloží interního uložiště
- **RoleActivity** – Rozcestník k roli zájemce, uchazeč, student. Nabídka dalších možností (užitečné odkazy, zprávy, předměty, facebook)
- **FacebookEventsActivity** – zobrazení facebookových událostí a příspěvků
- **NotificationActivity** – přehled upozornění
- **StudyFieldsActivity –** přehled studijních předmětů
- **AdmissionActivity** – zobrazení konkrétního přijímacího řízení
- **PreparatoryCoursesActivity** – přehled přípravných kurzů
- **StudentActivity** – přehled odkazů
- **PreparatoryCourseActivity –** zobrazení konkrétního přípravného kurzu
- **AdmissionsActivity** – přehled přijímacích řízení

cz.uhk.app.model.components

- **GcmMessageHandler** – přečte a zobrazí upozornění

cz.uhk.app.model.services

- **DataSource** – hlavní zdroj dat, který čerpá z webového API
- **GcmReceiver** – hlídá příchozí upozornění
- **JSONParser** – Parser, zpracovávající dokument v JSON formátů, který obdrží od webového API.

4. Příručka uživatele

4.1. Obsluha mobilní aplikace

Mobilní aplikace je navrhnuta jako průvodce uchazeče, či studenta. Oproti webové aplikaci umožňuje zvolit si roli, která ho již navede k požadovaným datům. Webové aplikace je v tomto směru přímočařejší, kde si uživatel vybere přímo požadovaná data. Kromě role lze nahlížet i na upozornění, která jsou v Androidu oznamována v reálnem čase. Lze také ukládat významné události do interního Google kalendáře.

Stejně jako ve webové aplikaci uživatel může shlédnout přijímací řízení, přípravné kurzy, užitečné odkazy, ale také události a příspěvky z facebookových stránek fakult. Uživateli je nabízeno také filtrování dle zadaných kritérií ve vyhledávání předmětů ze systému IS/STAG.

4.1.1. Výběr fakulty

Jak již bylo mnohokrát řečeno, celá aplikace je koncipována pro celou Univerzitu Hradec Králové a tedy i pro všechny její fakulty.

Uživatel si tedy může zvolit mezi Fakultou informatiky a managementu, Přírodovědeckou fakultou, Filozofickou fakultou, Pedagogickou fakultou nebo ústavem Sociální práce.

Tato volba se poté uloží do interní paměti mobilního telefonu a při další návštěvě se již nasměruje na předchozí vybranou fakultu. Na podobném principu funguje webová aplikace.

Obrázek 14: Náhled výběru fakulty

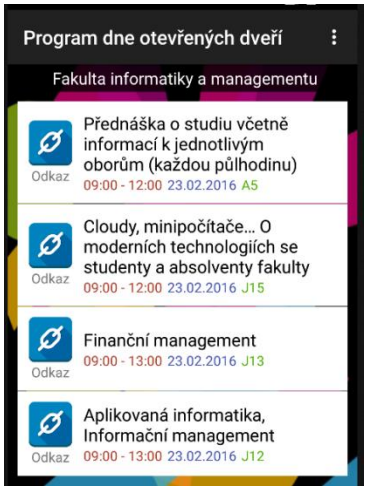
4.1.2. Výběr role

Obrazovka s názvem výběr role umožňuje vybrat ze tří možných rolí, kterými jsou zájemce, uchazeč a student. Všechny tyto možnosti by měli být dostatečně intuitivní, aby si z nich uživatel mohl vybrat.

Kromě rolí nabízí toto okno i ostatní nabídku, která umožňuje vstoupit například do přehledu předmětů, facebookových událostí nebo upozornění.

4.1.3. Program dne otevřených dveří

Toto okno přináší uživateli podstatné informace o jednotlivých položkách programu ke dni otevřených dveří. Mezi parametry, které popisují jednotlivé položky, patří datum a čas konání, místnost a v neposlední řadě i odkaz na akci.



Obrázek 15: Náhled na program dne otevřených dveří

4.1.4. Přijímací řízení

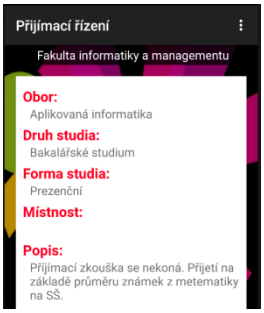
Přijímací řízení sděluje uživateli, kdy a jakou formou je konáno. Aplikace nám přináší informace o přijímacím řízení doplněné o datum, čas a případný odkaz. Může se ovšem stát, že u takové to položky nebude datum ani čas. V tom případě je pak podstatný parametr popis, kde je sděleno, jakou konkrétní formou se dané řízení koná.

Obrázek 16: Náhled na obrazovku s konkrétním přijímacím řízením

4.1.5. Přípravné kurzy

Z okna přípravné kurzy, se mimo jiné dovíte, kdy a v kolik hodin se konají. Po otevření položky, lze dohledat i další upřesňující informace jako jsou například

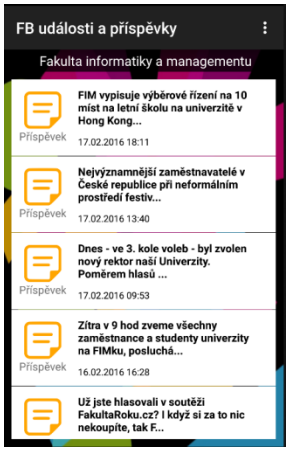
rozsah hodin, harmonogram kurzu, odesílací mail, garant kurzu apod. Tuto událost si lze, pomocí kliknutí na tlačítko, přidat do interního kalendáře.



Obrázek 17: Náhled na otevřený detail přijímacího řízení

4.1.6. FB události a příspěvky

Díky komunikaci s externími systémy, jako je například Facebook, lze sledovat přímo v aplikace události nebo příspěvky, které jsou publikované na facebookových stránkách fakult. Konkrétní příspěvek nebo událost lze otevřít, což navede přímo na danou facebookovou stránkou.

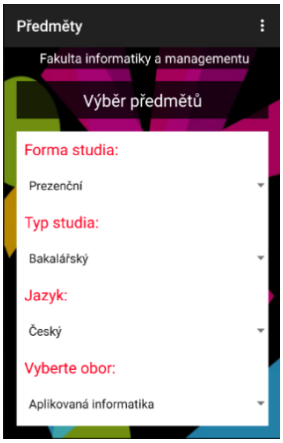


Obrázek 18: Náhled na FB události a příspěvky

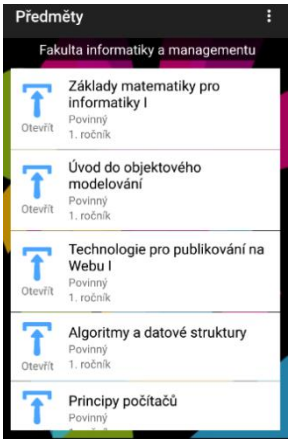
4.1.7. Předměty

Mnohdy se stává, že uchazeči o studium se chtějí dozvědět více o předmětech, vyučovaných na jednotlivých oborech. I tuto možnost nabízí jak webová, tak mobilní aplikace.

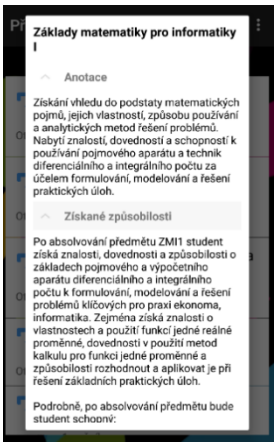
Přehledné zobrazení předmětů je realizováno skrz API IS/STAG. Uživatel nejprve zvolí možnosti ve filtru, tedy formu studia, typ studia, jazyk a obor. Poté přes tlačítko hledat systém nejde vhodné předměty. U jednotlivých předmětů je možnost detailního náhledu, kde uživatel zjistí anotaci, získané způsobilosti apod.



Obrázek 21: Náhled na filtr předmětů



Obrázek 20: Seznam předmětů



Obrázek 19: Detail předmětu

4.1.8. Upozornění

Tzv. push notifikace upozorňují na důležitou informaci, kterou by uživatel neměl přehlédnout. Taková to informace se objeví v oznamovací oblasti mimo aplikaci na displeji mobilního zařízení, na kterou lze poté kliknout a přejít tak na danou informaci skrz mobilní aplikaci.

4.1.9. Přidání do Google kalendáře

Jednoduchým způsobem, přes tlačítko přidat do kalendáře, se vloží události při dni otevřených dveří, případně přípravný kurz nebo jiná důležitá událost, do interního kalendáře na mobilním zařízení. Výhodou je i fakt, že taková to událost se poté spojí i s webovým Google kalendářem, který je pod daným uživatelským účtem.

4.2. Správa systému

Vkládání samotných dat probíhá přes webové rozhraní. Každý uživatel má svůj přiřazený design, který odpovídá barevnému schématu fakulty, ke které je přiřazen. Kromě barevného schématu, je uživateli přiřazena role, která mu určuje práva pro správu systému.

První role nese název administrátor a umožňuje uživateli pracovat s kompletní evidencí přípravných kurzů, přijímacích řízení, zasílat zprávy uživatelům, vytvářet program pro den otevřených dveří a přidávat užitečné odkazy.

Druhá role (super administrátor) má nejvyšší práva a kromě toho, že umožňuje pracovat se stejnými funkcemi jako role administrátora, dále nabízí možnosti jako správa uživatelů, správa budov (podlaží, místnosti), evidence oborů a druhů studia.

4.2.1. Přihlášení, odhlášení a změna hesla

Pro přihlášení je třeba uživatelské jméno v podobě emailu a heslo. Tyto údaje lze obdržet poté, co super administrátor vytvoří požadovaný účet. Na zvolený email přijde uživatelské jméno a vygenerované heslo.

Odhlášení se provádí přes tlačítko odhlásit se v pravém horním rohu.

Pokud žádá uživatel změnu hesla, opět stačí kliknout na tlačítko v pravém horním rohu.

4.2.2. Události, přípravné kurzy, přijímací řízení

Události, přípravné kurzy a přijímací řízení jsou nejdůležitějšími částmi celé aplikace. Samotné události se objeví vždy, při dni otevřených dveří na fakultě. Vkládání údajů je velmi snadné a intuitivní. Všechny tyto záložky umožňují funkci skrytí/zobrazení položky na veřejné části webu a v mobilní aplikaci.

U přijímacích řízení je nutné rozlišovat, zda se jedná o řízení formou přijímacího testu nebo jinou formou. Pokud by se jednalo o přijímací řízení formou přijímacího testu, pak uživatel je nucen vyplnit datum, čas a místnost konání přijímací řízení, v opačném případě je třeba vyplnit popis a tak sdělit uživatelům, jakou formou se bude dané řízení konat.

Systém disponuje volitelně zapínatelnou funkcí pro kontrolu obsazenosti místnosti při vytváření nové položky. Uživatel se tak nemusí starat, zda v daný čas a datum je místnost již obsazena.

4.2.3. Správa uživatelů

Do záložky správa uživatelů má přístup pouze uživatel s právy super administrátora. Takový to uživatel může přidávat administrátory, super administrátory a editovat pouze administrátory.

4.2.4. Budovy, obory, druhy studia

Do záložek pro správu budov, oborů a druhů studia má opět přístup pouze uživatel s oprávněním super administrátor. Uživatel má zde možnost pracovat s evidencí oborů a druhy studia. Pod záložkou budovy se skrývají ještě další podkategorie, jako jsou např. evidence podlaží a jednotlivých místností.

5. Testování

Pro testovací účely byla webová aplikace přesunuta na veřejný hosting www.php5.cz, kde lze bezplatně publikovat kromě statických stránek také dynamické webové aplikace a to i s použitím databáze MySQL. Tento webhosting je provozován společnosti Czechia.

Prvky, které byly při testování sledovány, jsou správné chování aplikační logiky webového systému a také responzivní design pro mobilní webové prohlížeče. Vše probíhalo uspokojivě bez žádných znatelných potíží. Aplikace je dostupná na adrese www.uhkapp.php5.cz a webová administrace poté www.uhkapp.php5.cz/admin. Spíše než chybný chod aplikace se projevili později nedostatečné funkční možnosti, které uživatelé vyžadovali. Například při prvotním záměru obsazovat místnost pouze jednou akcí v daný čas a datum, se ukázalo jako ne vždy vyhovující. Proto bylo přistoupeno k ručně volitelné možnosti, zda povolit či naopak zamítnout kontrolu obsazenosti místnosti.

Mobilní aplikace byla testována na dvou druzích zařízení. První byl emulátor, tedy virtuální systém spuštěný přes Android Studio, na kterém nebyly pozorovány žádné potíže. Na emulátoru byl spuštěn Android 5.0 Lollipop. Druhou kategorií byla skutečná mobilní zařízení. Testováno bylo na zařízení s Android 2.3 Gingerbread a 5.0 Lollipop. Starší verze 2.3 nevykazovala správnou funkčnost chodu notifikací přes Google Cloud Messaging. Aplikace byla umístěna panem proděkanem doc. Mgr. Tomášem Kozlem, Ph.D. na obchod Google Play a je dostupná z adresy https://play.google.com/store/apps/details?id=cz.uhk.app.

Později v průběhu testování mezi uživateli byla vyžadována integrace se systémem IS/STAG a to konkrétně prohlížení studijních předmětů pod jednotlivými obory. Na základě tohoto požadavku byl vytvořen filtr, který se dotazuje školního systému IS/STAG.

6. Závěr

Základním cílem této práce bylo vytvořit komplexní aplikaci pro správu aktivit uchazeče o studium na Univerzitě Hradec Králové.

Vývoj prošel několika fázemi, jako například stanovení funkčních a nefunkčních požadavků, analytickým návrhem, implementací až k samotnému nasazení do provozu.

Výsledkem je aplikace, která má dvě základní podoby. První verzí je mobilní aplikace pro operační systém Android, která je snadno dostupná přes obchod Google Play. Obsahuje několik funkčních výhod oproti verzi webové, a to například živé zasílání notifikací, nebo přímé ukládání do Google kalendáře. Druhou verzí je webová podoba aplikace, která obsahuje i samotnou administraci, přístupnou pro správce systému. Celá webová aplikace je responzivní, což je tedy dnešním standardem.

Aplikace bude bezesporu velkým přínosem pro uchazeče o studium, poněvadž shromažďuje informace, které jsou nejčastěji samotnými uživateli vyžadovány. Uživatel zde najde například informace o přípravných kurzech, přijímacích řízeních, informace o vyučovaných předmětech a spoustu dalšího.

Seznam obrázků

Literatura

[1] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007, 567 s. ISBN 978-80-251-1503-9.

[2] ČÁPKA, David. *2. díl – Use Case Diagram* [online]. Dostupné z: http://www.itnetwork.cz/navrhove-vzory/uml/uml-use-case-diagram, 2013 [cit. 2016-01-05]

[3] KRÁLOVÁ, Iveta. *Diagram tříd* [online]. Dostupné z: http://java.vse.cz/modelyUML/diagramtrid.html [cit. 2016-01-07]

[4] GILMORE, W. *Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály*. Nové, 3. vyd. Překlad Jan Pokorný. Brno: Zoner Press, 2011. Encyklopedie Zoner Press. ISBN 978-80-7413-163-9.

[5] Android Developers Blog. *Android Studio: An IDE bulit for Android.* [online]. 15.5.2016 [cit. 2016-06-28]. Dostupné z: http://android-developers.blogspot.cz/2013/05/android-studio-ide-built-for-android.html

[6] MÁJSKÝ Michal, DOHNAL Václav. *Aplikace pro Facebook od základů – díl I.* [online]. Dostupné z: https://www.zdrojak.cz/clanky/aplikace-pro-facebook-od-zakladu-dil-i/, 2011

[7] CHAFFER, Jonathan a Karl SWEDBERG. *Mistrovství v jQuery: [kompletní průvodce vývojáře]*. Brno: Computer Press, 2013. Mistrovství. ISBN 978-80-251-4103-8.

[8] *Installing the SDK* [online]. Android developers, 2010 [cit. 2016-06-28]. Dostupné z: https://developer.android.com/studio/index.html

[9] Lenka Hodlová. ZCU. *Vývoj aplikací.* [online]. 28.6.2016 [cit. 2016-06-28]. Dostupné z: http://home.zcu.cz/~hodlova/html/aplikace.html

[10] LACKO, Ľuboslav. *Vývoj aplikací pro Android.* 1. Vyd. Brno: Computer Press, 2015, 472 s. ISBN 978-80-251-4347-6.

Přílohy

Obsah přiloženého CD-ROM

- Zdrojové kódy webové aplikace (webapp.zip)
- Zdrojové kódy mobilní aplikace (mobileapp.zip)
- Instalační soubor mobilní aplikace (uhkapp.apk)
- Vyexportovaná databáze (databaze.sql)