



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Univerzální aplikace pro výstavy a konference

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 1802T007 – Informační technologie

*Autor práce:* **Bc. Jaroslav Jakoubě**

*Vedoucí práce:* Mgr. Jiří Vraný, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC  
Faculty of Mechatronics, Informatics  
and Interdisciplinary Studies ■

# Universal application for exhibitions and conferences

## Diploma thesis

*Study programme:* N2612 – Electrical Engineering and Informatics

*Study branch:* 1802T007 – Information Technology

*Author:* **Bc. Jaroslav Jakoubě**

*Supervisor:* Mgr. Jiří Vraný, Ph.D.



## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jaroslav Jakoubě**  
Osobní číslo: **M13000010**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Informační technologie**  
Název tématu: **Univerzální aplikace pro výstavy a konference**  
Zadávající katedra: **Ústav nových technologií a aplikované informatiky**

### Z á s a d y p r o v y p r a c o v á n í :

- 1) Seznamte se s problematikou programování webových služeb s podporou REST API a s problematikou programování multiplatformních mobilních aplikací.
- 2) Vytvořte návrh aplikace pro podporu výstav a konferencí. Mělo by jít o multiplatformní mobilní aplikaci poskytující návštěvníkům potřebné informace i v offline režimu. Dále navrhnete administrační systém, pro vytváření a úpravy informací o konferenci. V neposlední řadě navrhnete princip vzájemné komunikace mezi mobilní a serverovou částí aplikace, umožňující propagaci změn v datech do již nainstalovaných mobilních aplikací.
- 3) Návrhy prakticky implementujte a vytvořte aplikaci pro podporu konferencí s multiplatformním mobilním klientem.

Rozsah grafických prací: **dle potřeby**  
Rozsah pracovní zprávy: **cca 60 stran**  
Forma zpracování diplomové práce: **tištěná/elektronická**  
Seznam odborné literatury:

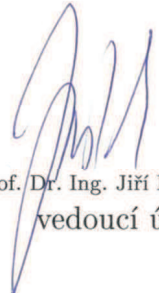
- [1] FIELDING, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures [online]. University of California, 2000 [cit. 2013-10-08]. Dostupné z: <http://www.ics.uci.edu/fielding/pubs/dissertation/top.htm>
- [2] ALLAMARAJU, Subrahmanyam. RESTful Web services cookbook. 1st ed. Sebastopol, CA.: O'Reilly, c2010, xv, 293 p. ISBN 05968501688.
- [3] BOCKLAGEŠRYANNEL, Juergen a Johan THELIN. A Book about Qt5 [online]. 2014 [cit. 2014-10-10]. Dostupné z: <http://qmlbook.org/>
- [4] QTŠPROJECT. Qt 5.3 documentation online [online]. 2014. vyd. [cit. 2014-10-10]. Dostupné z: <http://qtproject.org/doc/qt5/index.html>

Vedoucí diplomové práce: **Mgr. Jiří Vraný, Ph.D.**  
Ústav nových technologií a aplikované informatiky

Datum zadání diplomové práce: **20. října 2015**  
Termín odevzdání diplomové práce: **16. května 2016**

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
prof. Dr. Ing. Jiří Maryška, CSc.  
vedoucí ústavu

V Liberci dne 20. října 2015

## Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 20. 5. 2016

Podpis:



## **Poděkování**

Chtěl bych poděkovat svému vedoucímu diplomové práce Mgr. Jiřímu Vranému, Ph.D. za odborné vedení. Dále bych chtěl poděkovat společnosti Heureka Shopping s. r. o. za umožnění realizace projektu s možností využití firemního hardwaru.



## **Abstrakt**

Česká verze:

Tématem této práce je vytvoření univerzální mobilní aplikace určené pro potřeby výstav a konferencí. Aplikace byla vytvořena pomocí prostředí Qt Creator a pro jádro aplikace byly použity programovací jazyky C++, JavaScript. Komponenty grafického uživatelského rozhraní jsou vytvořeny pomocí QML. Cílovými platformami jsou Android a iOS. Aplikace je tvořena konfigurovatelnými objekty, které lze upravit pomocí webové administrace. Výsledné nastavení se uloží do databáze a po jeho stažení se aplikace vykreslí přesně dle definovaných parametrů, nezávisle na platformě či velikosti cílového zařízení. Uživatelé aplikace jsou tímto způsobem zpřístupněny přehledné komponenty obsahující informace o dané konferenci. K dispozici je zde mapa jednotlivých sálů, kalendář přednášek dělený dle sálů a stránka obsahující pouze dynamické prvky. Veškerá data jsou ukládána přímo do zařízení a díky tomu jsou k dispozici i bez internetového připojení.

Klíčová slova:

Mobilní aplikace, Qt Creator, Android, iOS, konference

English version:

The main focus of this diploma thesis is to create universal mobile application for exhibition and conference purposes. This application has been created in Qt Creator studio and its core has been programmed in C++, JavaScript. Graphical user interface components have been made with QML. Target platforms are Android and iOS. The whole application is made from objects which can be configured via administration web page. Final settings are saved to database and can be downloaded to target device. Once it has been downloaded than it is saved into app settings and the whole GUI is generated exactly by defined parameters, independently by platform or size of device screen. Organized components with information about given conference are accessible to the application user. A user has an open access to map of each hall, scheduler of events organized into columns by a specific hall and a page with dynamic components only. All

data are saved into target device and they are available without internet connection as well.

Keywords:

Mobile application, Qt Creator, Android, iOS, conference



## Obsah

Prohlášení .....	3
1 Úvod .....	12
2 Podrobnější představení problematiky .....	13
2.1 Co by měla aplikace poskytovat .....	13
2.2 Program přednášek .....	13
2.2.1 Plán místa a sálů .....	13
2.2.2 Doplnující funkce .....	14
2.3 Způsoby zpracování .....	14
2.3.1 Webová aplikace .....	14
2.3.2 Mobilní aplikace .....	14
2.4 Volba aplikace .....	15
2.4.1 Existující projekt .....	15
2.4.2 Vlastní aplikace .....	15
2.5 Tvorba multiplatformní aplikace .....	15
2.5.1 Průzkum využití .....	16
2.5.2 Omezení vývoje .....	16
2.6 Využívané nástroje a přístupy .....	17
2.7 Vývoj v Qt .....	19
3 Cíle práce .....	21
3.1 Upřesnění zadání .....	21
4 Návrh řešení .....	22
4.1 Inspirace .....	22
4.1.1 Snappii .....	22
4.1.2 Aplikace na konference .....	23
4.2 Požadované chování aplikace .....	26

4.3	Mobilní aplikace .....	27
4.3.1	Uživatelská část .....	27
4.3.2	Mobilní administrace .....	28
4.4	Webová administrace .....	30
4.5	System, jazyk, IDE .....	31
5	Realizace řešení .....	32
5.1	Komunikace mezi jednotlivými vrstvami .....	32
5.2	Mobilní část .....	33
5.2.1	Struktura komponent aplikace .....	34
5.2.2	Výběr konference .....	35
5.2.3	Struktura stránek .....	36
5.2.4	Hlavní stránka .....	36
5.2.5	Mapy a obrázky .....	37
5.2.6	Stránka s programem .....	39
5.2.7	Stránka s informacemi .....	40
5.2.8	Hlavička .....	41
5.2.9	Patička .....	42
5.2.10	Notifikační panel .....	43
5.3	Webová část .....	44
5.3.1	Přidání konference .....	44
5.3.2	Úprava konference .....	45
5.3.3	Editace přednášek .....	46
6	Vyhodnocení aplikace .....	48
6.1	Výhody .....	48
6.1.1	Upravitelnost .....	48
6.1.2	Multijazyčnost .....	48

6.1.3	Využití off-line .....	49
6.1.4	Rychlost .....	49
6.2	Nevýhody .....	49
7	Shrnutí práce .....	50
7.1	Hlavní body jednotlivých kapitol .....	50
7.2	Splnění cílů .....	50
7.3	Plány budoucího rozšíření .....	51
8	Seznam použité literatury .....	52

## Seznam obrázků

Obr. 1:	Zastoupení mobilních platforem na trhu 2015 .....	16
Obr. 2:	Architektura PhoneGap aplikace .....	18
Obr. 3:	Rozdělení částí v Appcelerator Titanium .....	18
Obr. 4:	Výstřižek kódu z třídy FileDownloader .....	19
Obr. 5:	Ukázka kódu QML .....	20
Obr. 6:	Tvorba aplikace ve Snappii .....	22
Obr. 7:	Ukázka aplikace Conference Companion .....	23
Obr. 8:	Náhled Noodle Live aplikace .....	24
Obr. 9:	Prostředí aplikace EventPilot .....	25
Obr. 10:	Příklad využití EventMobi .....	25
Obr. 11:	Přidání dynamických parametrů .....	26
Obr. 12:	Schéma komunikace uživatelské části aplikace .....	28
Obr. 13:	Stránka se zvýrazněnými objekty .....	29
Obr. 14:	Mobilní verze administrace .....	29
Obr. 15:	Náhled webové administrace .....	30
Obr. 16:	Schéma komunikace součástí aplikace .....	32

Obr. 17: Výstřižek kódu pro generování pole stránek .....	33
Obr. 18: Základní struktura komponent mobilní aplikace .....	34
Obr. 19: Ukázka kódu na rekurzivní procházení objektem .....	35
Obr. 20: Základní struktura stránky aplikace .....	36
Obr. 21: Výstřižek kódu generující tlačítka .....	37
Obr. 22: Výstřižek kódu pro optimalizaci přibližování obrázku .....	38
Obr. 23: Výstřižek kódu vykreslující objekty reprezentující přednášky .....	39
Obr. 24: Ukázka kódu dynamického prvku .....	40
Obr. 25: Náhled hlavičky aplikace .....	41
Obr. 26: Kód zajišťující návrat na předchozí stránku .....	41
Obr. 27: Náhled patičky aplikace .....	42
Obr. 28: Část kódu pro řízení notifikačního panelu .....	43
Obr. 29: Funkce pro vytvoření nového nastavení konference .....	45
Obr. 30: Funkce pro tvorbu komponenty titulku .....	45

## Seznam zkratk

- PHP – „Hypertext Preprocessor“, programovací skriptovací jazyk určený pro tvorbu webových aplikací
- GPS – „Global Positioning System“, vojenský družicový systém pro určení přesné zeměpisné polohy
- REST – „Representational state transfer“, architektura určená k jednoduchému vytvoření, čtení, editaci nebo smazání informace ze serveru pomocí HTTP volání
- CRUD – „Create Read Update Delete“, označení základních operací pro práci s databází
- HTTP – „Hypertext Transfer Protocol“, internetový protokol určený pro výměnu hypertextových dokumentů
- POST – dotazovací metoda HTTP protokolu umožňující předat určitá data
- API – „Application Programming Interface“, rozhraní pro programování aplikací

- MSVC – „Microsoft Visual C++“, integrované vývojové prostředí pro jazyky C a C++
- GCC – „GNU Compiler Collection“, sada překladačů vytvořených v rámci projektu GNU
- Projekt GNU – projekt vyvíjející svobodný software inspirovaná Unixovými systémy
- QML – „Qt Markup/Modeling Language“, značkovací jazyk využívaný pro tvorbu uživatelského rozhraní
- IDE – „Integrated Development Environment“, software určený pro programátory většinou zaměřený na konkrétní programovací jazyk
- SDK – „Software development kit“, sada vývojářských balíčků umožňující tvorbu aplikací
- JDK – „Java Development Kit“, balíček nástrojů určený pro tvorbu aplikací v jazyce Java
- NDK – „Native Development Kit“, nástroj umožňující vývoj aplikací pro systém Android v jazyce C nebo C++
- GUI – „Graphical User Interface“, uživatelské rozhraní obsahující interaktivní ovládací prvky
- WYSIWYG – „What you see is what you get“, česky „co vidíš, to dostaneš“
- JSON – „JavaScript Object Notation“, struktura pro zápis dat nezávislý na využívané platformě
- URL – „Uniform Resource Locator“ – řetězec znaků s předepsanou strukturou určující přesné umístění zdroje informací
- URI – „Uniform Resource Identifier“ – řetězec znaků s předepsanou strukturou určující přesnou specifikaci zdroje informací
- ID – jednoznačný identifikátor
- RFID – „Radio Frequency Identification“ neboli identifikace na rádiové frekvenci
- NFC – „Near field communication“ – technologie bezdrátové komunikace mezi elektronickými zařízeními na velmi krátkou vzdálenost

# 1 Úvod

Asi většina z nás se účastnila nějaké konference, semináře, výstavy nebo jiné podobné události, na které bylo potřeba se orientovat. Způsob, jakým je účastník takovéto akce navigován, záleží výhradně na organizátorech. V některých případech dostane každý návštěvník vytištěný plánek jednotlivých sálů s programem, v jiných si informace musí zjistit z informačních tabulí či plakátů. Jelikož ale žijeme v době, kdy většina lidí vlastní mobilní telefon, tablet nebo jiné chytré zařízení, tak i pořadatelé využívají tyto technologie a informace nám zobrazují přímo v našem zařízení prostřednictvím webových stránek či aplikací.

Nicméně tato možnost je závislá na prostředcích, které mají být využity právě pro účely informovanosti účastníků, a ne každý si může dovolit vytvoření vlastního informačního webu nebo aplikace. Z tohoto důvodu vznikají stránky a programy, ve kterých lze zobrazovat vlastní data, obrázky a jiné informace potřebné k zřehlednění organizace dané akce.

Cílem mé diplomové práce bylo tento koncept rozvinout do podoby, kde by bylo možné nejen zobrazit nabízené informace, ale i možnost celou aplikaci upravit do podoby reprezentující společnost (komunitu), která akci pořádá.

Jeden z hlavních problémů je využívání aplikace na různých zařízeních s různými platformami. V tomto případě by mělo být zajištěno, že se aplikace bude chovat na všech zařízeních identicky a bude poskytovat stejné informace. Další důležitý aspekt tvoří celkový vzhled aplikace, který by se po jejím spuštění na zařízeních s rozdílným rozlišením či formátem měl přizpůsobit tak, aby byl zachován pořadatelem navržený design.

Tato práce obsahuje návrh a popis takovéto multiplatformní mobilní aplikace, která může posloužit jako jednoduchý informační systém pro výstavy a konference, a do které je zakomponována možnost jednoduché změny jejího vzhledu a chování dle požadavků pořadatele bez nutnosti rozsáhlé aktualizace či instalace.

## 2 Podrobnější představení problematiky

Jak bylo naznačeno v úvodu, na každé akci je nutné účastníkům poskytnout informace o její organizaci nebo uspořádání, a to co nejsnazším a nejpřehlednějším způsobem. Je samozřejmé, že ne vždy je vhodné nebo potřebné využít právě mobilní aplikaci, ale zaměříme se například na větší vývojářskou konferenci, na které je pořádáno velké množství přednášek v několika sálech. Na takovéto akci se předpokládá, že má drtivá většina účastníků k dispozici chytrý mobilní telefon a veškeré informace hledá výhradně prostřednictvím webu nebo informační aplikace.

### 2.1 Co by měla aplikace poskytovat

Pokud se budeme držet příkladu s rozsáhlou vývojářskou konferencí organizovanou ve více prostorách, ve kterých se odehrávají série přednášek, tak si potřebujeme uvědomit, co je zapotřebí poskytnout návštěvníkovi. Každý návštěvník by měl mít k dispozici jednoduchý přístup k mapě jednotlivých hal či sálů s přehledem plánovaných prezentací a přednášek, včetně dalších klíčových informací.

### 2.2 Program přednášek

Pokud se nejedná o událost, kde se odehrává pouze jeden bod programu, ale o konferenci skládající se z dílčích prezentací, přednášek či vystoupení, je nezbytné, aby každý, kdo se chce zúčastnit, byl informován o přesném času a místě události. A proto by aplikace měla zahrnovat i přesný časový rozvrh jednotlivých aktivit. Jistě by bylo přínosné, aby se každý mohl rozhodnout, jaké přednášky se bude účastnit. To mu může být usnadněno také zobrazením podrobnějších informací o vybrané prezentaci.

#### 2.2.1 Plán místa a sálů

Je dobré si uvědomit, že akci navštíví i lidé, kteří jsou v dané lokalitě poprvé a tudíž se v ní dobře neorientují. Proto by jim měl být poskytnut alespoň minimální „návod“, jak se na dané místo dostat. K tomu může posloužit odkaz s GPS pozicí či mapa s vyznačenou trasou. Důležitější ale je zajistit, aby se účastník bez problémů orientoval v celém areálu, neboť to mu většina navigací nedokáže poskytnout. Z toho důvodu by měl mít prostřednictvím aplikace přístupnou mapu celého komplexu s vyznačenými klíčovými místy.



## 2.2.2 Doplnující funkce

Pro doplnění informací o akci se nabízí například i možnost zobrazit jednotlivé sponzory a partnery, kteří se na konferenci podíleli. Můžeme třeba využít sponzoringu od místní taxi služby a umožnit uživateli její jednoduché zavolání.

Známa je také služba *Sli.do* [5] určená nejen pro přímou interakci s přenášejícím. Zde je po jednoduchém přihlášení na webovou stránku možné psát připomínky či dotazy k právě probíhající prezentaci a umožnit tak mluvčímu zahrnout odpověď přímo do projevu.

## 2.3 Způsoby zpracování

### 2.3.1 Webová aplikace

Výhoda webové aplikace spočívá v její multiplatformnosti a dostupnosti. Každé zařízení s internetovým připojením a podporovaným webovým prohlížečem má možnost stránku zobrazit a prohlížet její obsah. Vzhled a chování takového webu je závislý zejména na jeho návrhu, podpoře využívaného prohlížeče a rychlosti internetového připojení. Pro pořadatele je pak velice snadné dodatečně upravovat vzhled, dynamicky měnit zobrazované údaje nebo například přesměrovat návštěvníka na jiné stránky s dalšími informacemi.

Hlavní nevýhodou se ale stává stabilita a dostupnost webu, závislá na rychlosti internetového připojení uživatele. Pokud má návštěvník příliš pomalé připojení nebo je omezen datovým limitem, může se stát, že než se dostane k požadovaným informacím, tak přestanou být aktuální. Toto omezení se dá minimalizovat poskytnutím volného Wi-Fi připojení v areálu konference, ale poté musí být zajištěna její bezpečnost a stabilita.

### 2.3.2 Mobilní aplikace

Mobilní aplikace je, co se týče návrhu, komplikovanější než web, jelikož kompatibilitu s různými systémy neřeší prohlížeč nainstalovaný v zařízení, ale musí ji vyřešit sám programátor. Je nezbytné vytvořit kód pro každou podporovanou platformu a zajistit stejnou funkcionalitu. Může také dojít ke špatnému návrhu jádra aplikace, který může mít za následek například velkou spotřebu baterie či dat.

Výhodou je možnost plného využití hardwaru mobilního zařízení a všech funkcí, které poskytuje. Mezi to patří využití interní paměti telefonu pro uložení nastavení

aplikace či využití fotoaparátu. Díky tomu, že nativní aplikace má svá data uložena v zařízení, není nezbytné, aby k jejich čtení využívala internetové připojení, čímž dochází i k šetření přenášených dat a program není závislý na rychlosti nebo dostupnosti připojení.

## 2.4 Volba aplikace

### 2.4.1 Existující projekt

Jestliže organizátor využije již vytvořený projekt, musí se spolehnout na jeho stabilitu a funkcionalitu. Na výběr má ze širokého spektra různých webů a aplikací, kde každý nabízí diferenciální možnosti. Hlavním parametrem výběru by měly být požadavky návštěvníka, který bude aplikaci využívat. Z tohoto důvodu je vhodné zvolit přehledný systém disponující intuitivním ovládáním, nejlépe i s možností změny jazyka.

Většina již hotových aplikací poskytuje možnost zobrazení vlastních dat či obrázků, ale není povoleno si ji upravit dle svého grafického návrhu. Z těchto důvodů byla tato práce zaměřena právě na návrh softwaru umožňujícího rozsáhlé úpravy vzhledu a chování již nainstalované aplikace, a to pouze prostřednictvím administrátorského rozhraní.

### 2.4.2 Vlastní aplikace

Jestliže chce pořadatel využít mobilní telefon jako jeden z informačních kanálů pro návštěvníka, měl by si pečlivě rozmyslet, co od aplikace očekává a co je do ní ochoten investovat. V současné době existuje velké množství již hotových projektů určených právě pro výstavy nebo konference, které jsou buď volně přístupné, nebo je potřeba zakoupení licence.

Dále je samozřejmě možnost vytvořit svou vlastní aplikaci nebo web, který je navržen přesně dle specifických potřeb a představ. Tato varianta ale vyžaduje čas nutný na její vytvoření a vývojářský tým, který projekt realizuje.

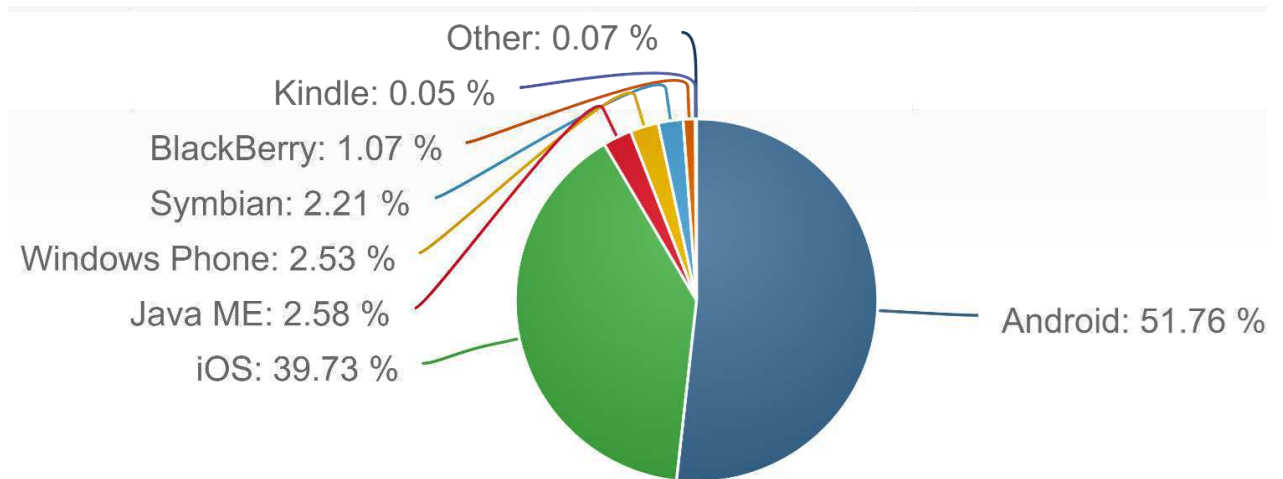
## 2.5 Tvorba multiplatformní aplikace

Před samotným psaním kódu je nezbytné upřesnit, pro které platformy bude aplikace dostupná a tudíž, jaký programovací jazyk bude pro její návrh použit.

### 2.5.1 Průzkum využití

Aktuálně se na světovém trhu vyskytuje několik hlavních operačních systémů využívaných na chytrých mobilních telefonech a tabletech. Většinu trhu ovládají dva operační systémy, a to Android a iOS.

Na obrázku (Obr. 1) je vidět jejich zastoupení ve světě, dle dat získaných z portálu *NetMarketShare* [6].



Obr. 1: Zastoupení mobilních platforem na trhu 2015

Zdroj: *NetMarketShare* [6], 12. 2. 2016

### 2.5.2 Omezení vývoje

Každý vývoj pro specifický operační systém s sebou přináší určitá omezení. Ve většině případů je nezbytné vlastnit alespoň jedno zařízení s cílovým systémem, abychom mohli náš kód otestovat. Existují sice emulátory, které se chovají jako reálné zařízení, často však nezaručují plnou spolehlivost. Jejich velká výhoda spočívá v tom, že s jejich pomocí můžeme simulovat chod aplikace v jakémkoli zařízení a upravovat si velké množství parametrů. Poskytují možnost testovat chování aplikace na různých velikostech displejů s různým rozlišením bez nutnosti použití reálných přístrojů. Některé emulátory disponují i funkcí, která umožňuje dynamicky měnit rychlost internetového připojení, a tak je například možné otestovat, jak by se aplikace mohla chovat v místech s nedostatečným signálem. Dále je možné ručně upravovat data o aktuální GPS poloze telefonu nebo jeho natočení v prostoru.

Velmi důležitějším faktem ovšem je, že každý systém má jinou strukturu. Vývoj pro požadovanou platformu tedy vyžaduje, aby byla aplikace napsána v programovacím

jazyce, který při použití správného kompilátoru, bude přeložen do podoby podporované cílovou platformou.

Například pokud vyvíjíme aplikace pro zařízení s operačním systémem iOS nebo MacOS, je nezbytné využít překladač tohoto systému. Pro tyto platformy se potřebný program vyskytuje jako součást vývojového prostředí Xcode, které je zatím k dispozici pouze na zařízeních s MacOS, a tudíž potřebujeme počítač od společnosti Apple, abychom mohli kód přeložit do spustitelné podoby.

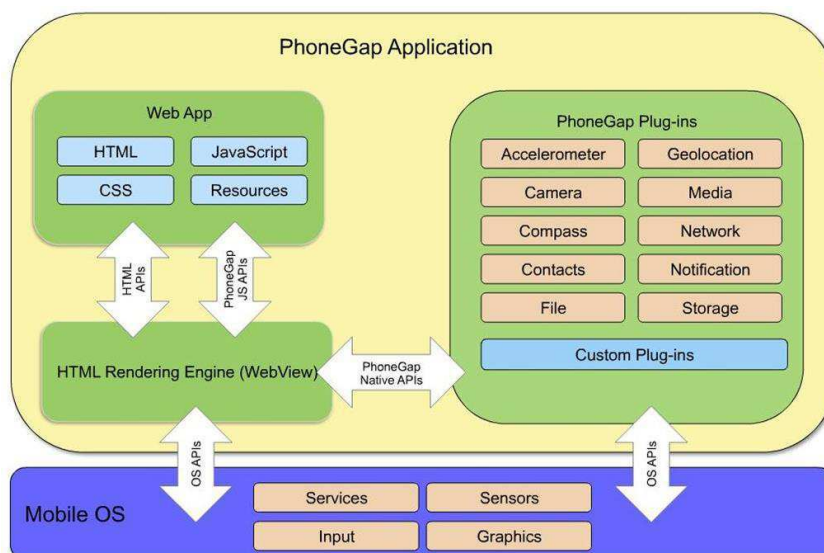
## 2.6 Využívané nástroje a přístupy

Pokud neplánujeme psaní nativní aplikace pro každou cílovou platformu zvlášť, je nezbytné zvolit nástroje, které podporují vývoj multiplatformních projektů. Existuje široké spektrum přístupů, kterými lze vytvořit software spustitelný na různých typech zařízení a výběr spočívá zejména na schopnostech programátora a typu cílové aplikace.

Jestliže je naším cílem vytvořit mobilní hru nebo interaktivní aplikaci využívající velké množství grafických efektů, můžeme k tvorbě využít například vývojové prostředí **Unity**. Jedná se zejména o herní engine vyvinutý společností Unity Technologies [7] využívající programovací jazyky C# a JavaScript. S jeho pomocí lze vytvářet nejen mobilní aplikace, ale také programy pro PC, konzole nebo web.

Další možností je využít například nástroje **PhoneGap** [8] nebo **Sencha** [9]. Jedná se o frameworky, které umožňují napsání kódu ve formě webové stránky a jeho následné přeložení do podoby mobilní aplikace. Výhodou je využití standardních jazyků a přístupů používaných při tvorbě webů jako je HTML, CSS a JavaScript. Vytvořená aplikace se na zařízení spouští v kontejneru „WebView“ a s komponenty telefonu komunikuje prostřednictvím systémových nebo PhoneGap nativních API.

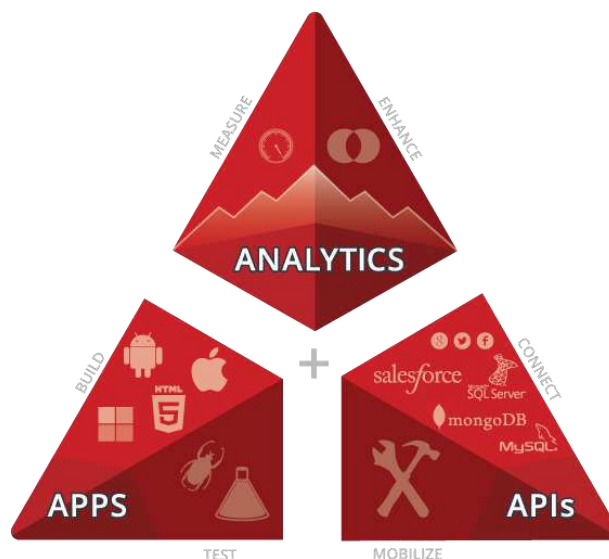
Na obrázku (Obr. 2) je znázorněna zjednodušená struktura architektury z přednášky o programování multiplatformních aplikací z portálu *magin.me* [10].



Obr. 2: Architektura PhoneGap aplikace

Zdroj: [magin.me](http://magin.me) [10]

Tvorba nativních aplikací pro platformy iOS, Android a Windows může být prováděna například s využitím nástroje **Appcelerator Titanium**. Jedná se o otevřený software společnosti Appcelerator [11], ve kterém lze programovat multiplatformní aplikace v jazyce JavaScript. GUI je zde kódováno pomocí jazyka XML. Jsou zde obsaženy nástroje pro jednoduchou komunikaci s nativními komponenty cílového zařízení, jako například akcelerometr, navigační panel, dialogová okna atp.



Obr. 3: Rozdělení částí v Appcelerator Titanium

Zdroj: [appcelerator.com](http://appcelerator.com) [11]

Na obrázku (Obr. 3) je znázorněno rozdělení celé struktury projektu na část pro aplikaci, rozhraní pro komunikaci s různými komponentami a analytické nástroje.

Velice využívaným programovacím jazykem pro tvorbu multiplatformních nativních aplikací je C++. S jeho pomocí lze napsat efektivní kód, který využívá pouze přenositelné knihovny a funkce využitelné v řadě operačních systémů. Tento projekt lze poté přeložit pomocí různých překladačů a využít ho na různých zařízeních přímo jako nativní aplikaci. K tomuto účelu poslouží například vývojové prostředí **Qt Creator**, které je součástí projektu Qt Project [4] původně založeného společností Nokia.

## 2.7 Vývoj v Qt

V prostředí Qt Creator lze vytvářet multiplatformní aplikace při využití programovacího jazyka C++ s různými rozšířeními. Pro překlad na Linuxových nebo FreeBSD systémech využívá sadu překladačů GCC a na operačních systémech Windows může být využit MinGW nebo MSVC. Zároveň je zahrnuta i podpora překladače Clang.

Hlavním programovacím jazykem je C++ s rozšířením o doplňující funkce, jako je například zachytávání takzvaných signálů. Signál je metoda bez těla, která se propojí s určitým slotem. Pokud dojde k vyvolání signálu, je jako reakce spuštěna metoda definovaná ve slotu. Signál může slotu předat libovolný počet argumentů různého typu. Například aplikace nemusí čekat, až budou stažena veškerá data, ale pouze zareaguje, když je vyvolán signál „finished“. Poté se provede metoda „fileDownloaded“ (Obr. 4), které se jako parametr předá informace o datovém přenosu. Ten se vyhodnotí a na základě stažených dat upozorní aplikaci pomocí dalšího signálu. V tomto případě jde o signalizaci pro uživatele, zda došlo k úspěšnému stažení či zda nebyla nalezena požadovaná data.

```
FileDownloader::FileDownloader(QUrl imageUrl, QObject *parent) :
    QObject(parent)
{
    connect(&m_WebCtrl, SIGNAL(finished(QNetworkReply*)),
           SLOT(fileDownloaded(QNetworkReply*)));
    QNetworkRequest request(imageUrl);
    m_WebCtrl.get(request);
}

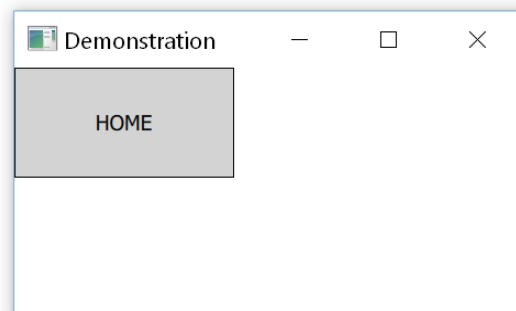
FileDownloader::~FileDownloader() {}

void FileDownloader::fileDownloaded(QNetworkReply* pReply)
{
    m_DownloadedData = pReply->readAll();
    //emit a signal
    pReply->deleteLater();
    if(m_DownloadedData.isNull() || m_DownloadedData.isEmpty()){
        emit noDataDownloaded();
    }
    emit downloaded();
}
```

Obr. 4: Výstřížek kódu z třídy *FileDownloader*

V Qt Creatoru lze kombinovat C++ i s dalšími jazyky, jako je například JavaScript nebo QML. QML je deklarativní vysokoúrovňový skriptovací jazyk určený pro tvorbu uživatelského rozhraní aplikace. Slouží nejen k popisu vzhledu, ale i k definici jeho chování. Jeho zápis ve stromové struktuře s využitím JavaScript syntaxe velice usnadňuje a zpřehledňuje celý návrh. Veškeré stavební elementy jsou brány jako objekty, které mohou být měněny pomocí parametrů či rozšířeny o nové prvky. Všechny komponenty jsou součástí různých knihoven a balíčků.

```
Rectangle {  
    id: homeButton  
    width: 200  
    height: 100  
    color: "lightgray"  
    border.color: "black"  
    anchors.top: parent.top  
    Text {  
        text: qsTr("HOME")  
        anchors.centerIn: parent  
    }  
    MouseArea{  
        anchors.fill: parent  
        onClicked: goHome()  
    }  
}
```



Obr. 5: Ukázka kódu QML

Pokud by vývojář upřednostňoval spíše návrh v grafické podobě místo zdrojového kódu, může využít Qt Designer, který je součástí Qt Creatoru a zobrazuje jednotlivé prvky tak, jak vypadají ve výsledné aplikaci. Na obrázku (Obr. 5) je znázorněné jednoduché vytvoření tlačítka na návrat na domovskou stránku. Zde se využívá vnoření jednotlivých objektů s různými parametry.



### 3 Cíle práce

Cílem této práce bylo vytvořit návrh multiplatformní mobilní aplikace určené pro výstavy a konference, která by poskytovala potřebné informace pro účastníky akce a zároveň disponovala možnostmi rozsáhlé customizace ze strany pořadatele.

#### 3.1 Upřesnění zadání

- Aplikace byla navržena tak, aby uživateli usnadňovala orientaci v místě konference a poskytovala mu důležité informace ve formátu specifikovaném organizátorem. Vše by mělo být navrženo tak, aby bylo možné využít pouze jednu mobilní aplikaci na různých typech výstav či jiných událostech, při zachování funkcionality, ale s možností snadné změny vzhledu či dat.
- Hlavním cílem bylo vytvoření modelu aplikace, jehož chování a vzhled by nebyl závislý na využívané platformě či rozměru zařízení. Z toho důvodu musela být provedena rešerše s účelem zjistit, jaký programovací jazyk pro vytvoření aplikace využít a pro jaké mobilní platformy ji poskytovat.
- Dalším cílem bylo zpřístupnit návštěvníkovi pouze uživatelskou část aplikace sloužící právě pro usnadnění orientace na konferenci a využívající informací a dat poskytnutých administrátorem. Zároveň posoudit, jaké informace jsou pro návštěvníka nejdůležitější a jakým způsobem mu mají být poskytovány, společně s vytvořením návrhu dat využívaných i bez aktivního internetového připojení.
- V sounáležitosti s předchozím bodem bylo nezbytné vytvořit administrátorskou část, ve které by bylo možné pozměnit chování, vzhled uživatelského prostředí a poskytované údaje bez zásahu do zdrojového kódu aplikace. Tímto by byla zajištěna možnost dynamických změn rozhraní bez nutnosti rozsáhlých aktualizací celého jádra.
- Dále navrhnout princip vzájemné komunikace mezi mobilní a serverovou částí aplikace umožňující propagaci změn v datech do již nainstalovaných mobilních aplikací.
- V neposlední řadě umožnit implementaci speciálních prvků požadovaných administrátorem a umožnit rozšiřitelnost aplikace o nové komponenty.
- Jako poslední bod byl zpřístupnit stažení uživatelské aplikace do mobilního zařízení prostřednictvím oficiálních obchodů.

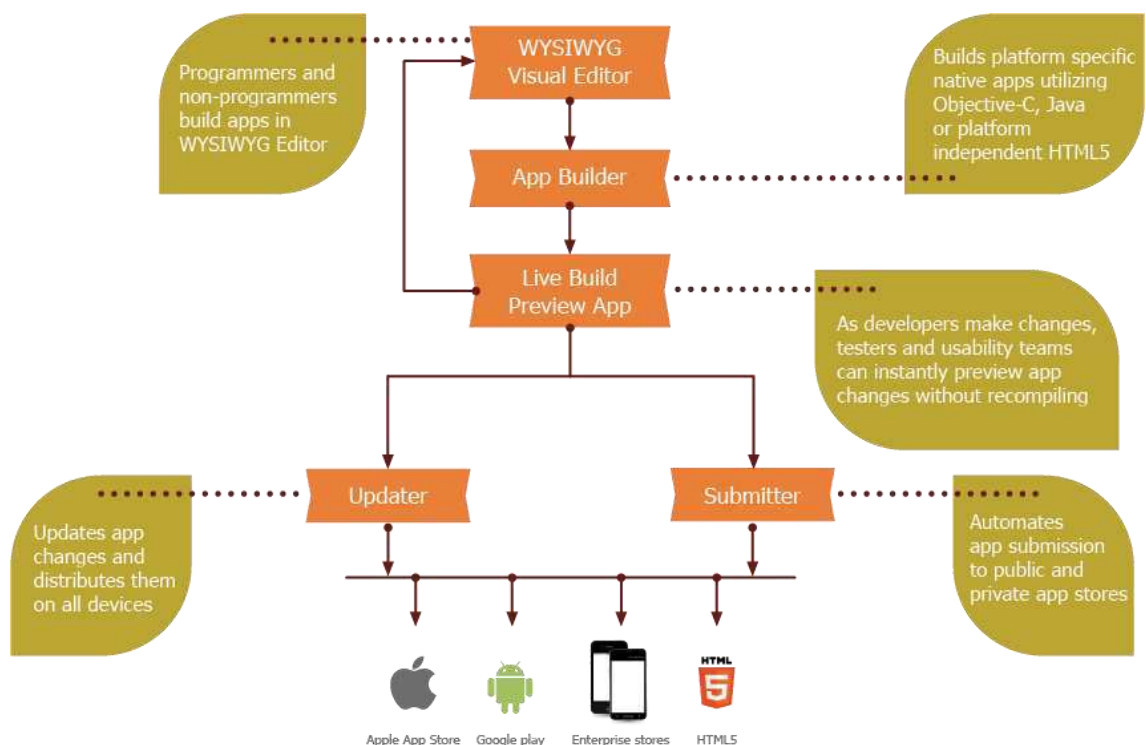
## 4 Návrh řešení

Celý tento projekt byl navržen tak, aby co nejvíce vyhovoval uživateli, který nechce psát svoji vlastní aplikaci, ale požaduje specifický design. S tímto cílem byla u většiny komponent implementována možnost jejich editace s využitím dynamického nastavení. Díky tomu bylo možno vytvořit aplikaci, která pouhou změnou konfiguračního souboru dokázala změnit svůj vzhled bez nutnosti nové instalace či rozsáhlé aktualizace.

### 4.1 Inspirace

#### 4.1.1 Snappii

Snappii [12] je cloudová služba, ve které lze vytvářet multiplatformní aplikace bez nutnosti znalosti programování. Po registraci je uživateli zobrazeno grafické prostředí, kde lze libovolně umisťovat či editovat jednotlivé komponenty. Jedná se o přístup „WYSIWYG“, což je akronym anglického „What you see is what you get“. Tím je dáno, že to, jak si uživatel aplikaci nadefinuje, bude její skutečný vzhled i na cílových zařízeních. Při uložení změn je vše okamžitě zobrazeno v „náhledové aplikaci“, kde lze ověřit správný vzhled a chování. Uživateli je zpřístupněno obrovské množství různých nástrojů a ovládacích prvků a tudíž je schopen rychle vytvořit téměř libovolnou aplikaci.



Obr. 6: Tvorba aplikace ve Snappii

Zdroj: snappii.com [12]

Na obrázku (Obr. 6) je schéma znázorňující průběh tvorby výsledné aplikace, ze kterého je patrné, že se uživatel nemusí starat ani o zdrojový kód, ani o distribuci výsledné aplikace do oficiálních obchodů, ale vše je prováděno automaticky.

#### 4.1.2 Aplikace na konferenci

Existuje velké množství aplikací určených přímo pro potřeby konferencí, jednání, výstav atp. Mezi ně patří například **Conference Companion** [13], **Meeting Application** [14] nebo **QuickMobile** [15]. Tyto projekty poskytují možnost vytvoření vlastních událostí, přidání informací o řečnících a disponují přehledným plánovacím kalendářem. Nicméně postrádají možnost rozsáhlejší editace celého uživatelského rozhraní. Z tohoto důvodu má uživatel (návštěvník) pouze jedno uživatelské rozhraní, ve kterém si ukládá oblíbené události. V těchto aplikacích má ale jen omezené možnosti přizpůsobit vzhled a chování aplikace dle svých představ.

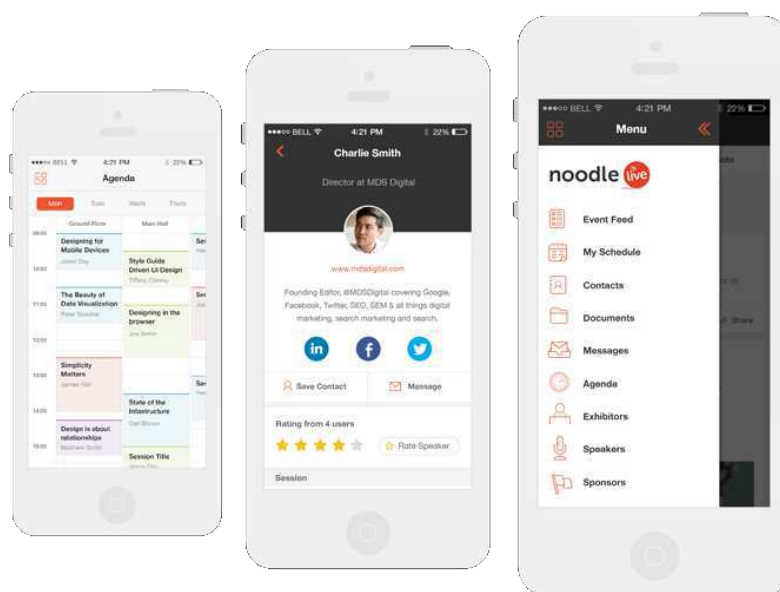


Obr. 7: Ukázka aplikace Conference Companion

Zdroj: [conferencecompanion.com](http://conferencecompanion.com) [13]

Na obrázku (Obr. 7) je zobrazen náhled aplikace Conference Companion, kde je znázorněno využití několika hlavních nástrojů, které byly použity pro účely konference. Aplikace je dělena do jednotlivých stránek, na které uživatel přistupuje z hlavní plochy. Má možnost zobrazit program, zprávy, různé typy map, ukázková videa a další nástroje.

Dalším podobným projektem jsou mobilní aplikace od společnosti **Noodle Live Limited** [16]. Ty mimo jiné obsahují možnost komunikace s různými aktivními prvky, jako jsou RFID a NFC tagy. Tím je uživateli umožněna interakce, například s okolními exponáty, kde se po přiložení mobilního telefonu okamžitě zobrazí například dodatečná informace. Možným využitím je i propojení s mapou sálu uloženou přímo v zařízení, do které mohou být, po načtení aktivního štítku, zaslány údaje o aktuální pozici. Nicméně vytvořená aplikace postrádá dodatečnou možnost editace grafického prostředí, a tudíž je po jakékoliv změně nezbytná její aktualizace. Na obrázku (Obr. 8) je znázorněno

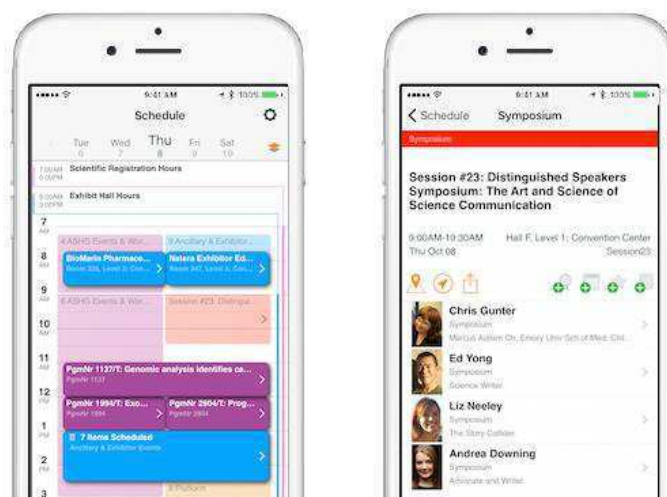


*Obr. 8: Náhled Noodle Live aplikace*

*Zdroj: joedarrington.com [17]*

uživatelské prostředí aplikace, kde je možné si všimnout přehledného menu jednotlivých stránek nebo plánovacího kalendáře.

Mezi aplikace specializované na vědecké, lékařské či technické konference patří **EventPilot** [18], která je určena zejména pro události s velkým množstvím paralelních přednášek. Z tohoto důvodu obsahuje rozsáhlé možnosti filtrování zobrazovaných dat. Veliká výhoda spočívá v možnosti plánování, či zobrazení jednotlivých přednášek bez nutnosti internetového připojení. Díky tomu má uživatel stále přístup k veškerým informacím a není omezen stabilitou nebo dostupností připojení. Ke každé přednášce je možné připojit detailní informace obsahující odborný text, doplňující soubory či fotografie. Na obrázku (Obr. 9) je zobrazen náhled prostředí této aplikace.



Obr. 9: Prostředí aplikace EventPilot

Zdroj: [eventmanagerblog.com](http://eventmanagerblog.com) [18]

Velice využívaná je upravitelná aplikace **EventMobi** [20]. Její velký potenciál je právě v možnosti konfigurace vlastního chování a vzhledu. Uživatel tak může pomocí grafického rozhraní nadefinovat požadovaný design celého projektu a určit, jaké komponenty budou využity. Veškeré změny se projeví přímo v cílovém zařízení. Tato funkcionality je umožněna díky využívání návrhu kódu v HTML a její zobrazení v nativní aplikaci prostřednictvím WebView. Výhodou je kompatibilita s širokým spektrem zařízení s různými operačními systémy a jednoduchá a rychlá možnost úpravy celé aplikace. Příklad vytvořeného projektu pro konferenci je uveden na obrázku (Obr. 10), na kterém jsou znázorněny využití komponenty.



Obr. 10: Příklad využití EventMobi

Zdroj: [EventMobi.com](http://EventMobi.com) [20]

## 4.2 Požadované chování aplikace

Koncept celého projektu vycházel z mé aplikace poskytující základní organizační informace, kterou jsem vytvořil pro účely konference ShopRoku 2014 pořádanou společností Heureka.cz, která návštěvníkovi poskytovala základní organizační informace. Tato událost byla pořádána ve dvou oddělených sálech, kde se odehrávaly jednotlivé přednášky. Návštěvník si mohl pomocí aplikace zobrazit mapy jednotlivých sálů, prohlížet program jednotlivých přednášek nebo přímo zavolat taxi. Základní aplikace byla vytvořena dle zadání pořadatele a mým cílem bylo ji vylepšit do jednoduše upravitelné podoby tak, aby obsahovala optimální kombinaci vlastností výše zmíněných aplikací.

Jednou z možností by bylo vytvoření „hybridní aplikace“, a to přepsáním zdrojového kódu do HTML, který by byl zobrazována přímo v zařízení prostřednictvím WebView. V tomto případě by bylo zapotřebí vytvořit rozhraní, umožňující komunikaci s plánovanými hardwarovými prvky zařízení a přepsat celé jádro do HTML. Výsledný projekt by byl kompatibilní s většinou chytrých zařízení a byla by umožněna jeho snadná upravitelnost.

```
Text {
  id: menuButtonText
  text: "Mapy"
  color: "#CCCCCC"
  anchors.bottom: parent.bottom
  anchors.bottomMargin: 13
  renderType: Text.NativeRendering
  font.pixelSize: 25
  font.family: myHelveticaFont.name
}

Text {
  id: menuButtonText
  text: qsTr(buttonSettings.text)
  color: buttonSettings.textColor
  anchors.bottom: parent.bottom
  anchors.bottomMargin: mainParameter * (-0.002)
  renderType: Text.NativeRendering
  font.pixelSize: headerFontSize
  font.family: myHelveticaFont.name
}
```

Obr. 11: Přidání dynamických parametrů

Další variantou bylo přepsat již hotovou nativní aplikaci tak, aby obsahovala pouze dynamické komponenty (Obr. 11), které se po načtení konfiguračního souboru naformátují přesně dle požadavků. Takto výslednou aplikaci je poté možné měnit přímo v cílovém zařízení pouze pomocí změny jejích parametrů. Veškeré úpravy jsou prováděny v administrační části, která umožňuje dané modifikace uložit a zároveň zamezuje chybnému zadání hodnot. Po uložení nového nastavení je vyslána informace všem aktivním aplikacím s požadavkem na aktualizaci nastavení. Následně se všem komponentám předají aktualizované parametry a jejich vzhled nebo chování se upraví do

nové podoby. Veškerá data jsou uložena přímo v přístroji a jsou připravena pro načtení dané konfigurace i v off-line režimu.

Důraz byl při návrhu kladen zejména na vzhled, rychlost a stabilitu výsledné aplikace. Aby byl zachován návrh zadavatele, bylo zapotřebí, aby se výsledná aplikace nijak nelišila při jejím zobrazení na různých přístrojích a mohla být rozšířena o komponenty efektivně využívající hardwarové komponenty přístroje. Pro splnění těchto vlastností je vhodnější volba „čistá“ nativní aplikace, obsahující objekty s parametrickým nastavením jednotlivých parametrů, které svůj rozměr či pozici mají danou dynamicky vůči ostatním prvkům.

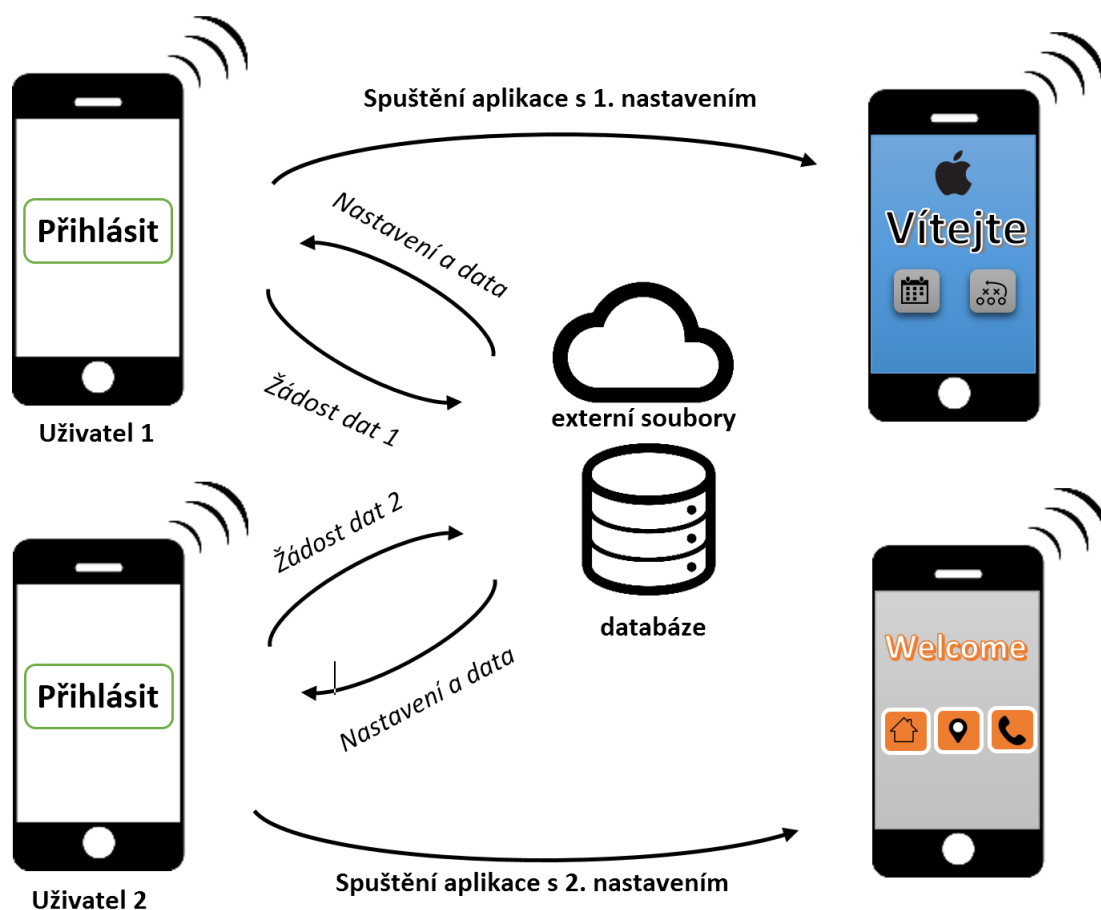
### 4.3 Mobilní aplikace

V prvotním návrhu byla aplikace rozložena na dvě části: uživatelskou a administrátorskou. Každá instalace v sobě měla obsaženy obě tyto části a každý uživatel tedy mohl přistupovat do administrátorské části.

#### 4.3.1 Uživatelská část

V této sekci je uživateli umožněno přihlášení na jakoukoliv konferenci, pokud zná její identifikátor. Po zadání správného názvu konference mu je staženo veškeré nastavení vytvořené administrátorem a dojde ke spuštění aplikace s nadefinovaným vzhledem a chováním (Obr. 12). V takovémto případě se uživatel ocitne na domovské stránce aplikace, kde mu jsou zpřístupněny ovládací prvky celého prostředí. Každý takto přihlášený klient má možnost pouze číst zobrazované informace a přecházet mezi jednotlivými stránkami. Veškeré texty, obrázky či nastavení mu jsou nahrány do zařízení, aby byly k dispozici i při výpadku sítě a také, aby nebylo zapotřebí jejich opakované stažení při příštím spuštění.





Obr. 12: Schéma komunikace uživatelské části aplikace

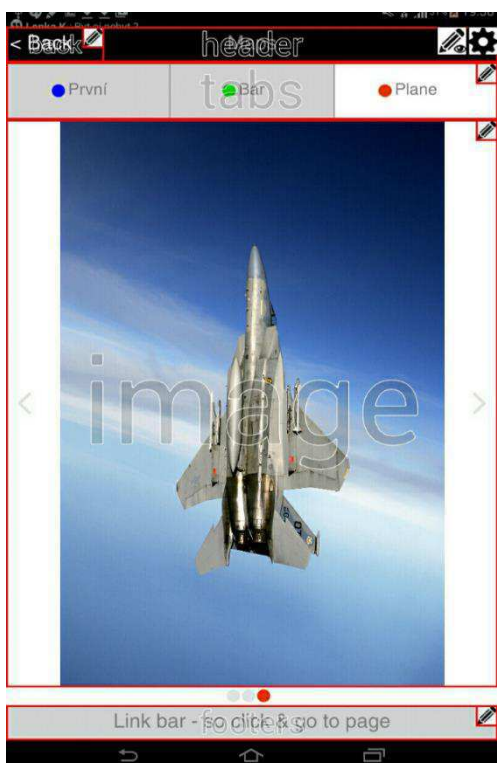
Po odhlášení z aktuální konference je k dispozici možnost přihlášení na jinou událost. Tehdy se provede stažení nového nastavení a celá aplikace se spustí s novými daty.

#### 4.3.2 Mobilní administrace

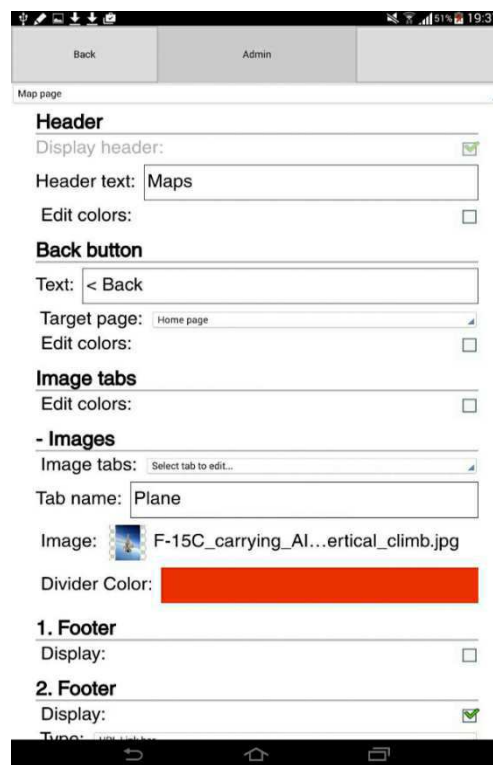
V případě přihlášení do administrátorské části se zobrazí možnost vytvoření nového nastavení nebo úprava již vytvořené konference. Pro úpravu je nezbytné znát název dané konference a přístupové heslo. Po úspěšném ověření správnosti údajů je uživatel přihlášen jako administrátor a zpřístupní se mu možnost úpravy jednotlivých komponent. Spuštění aplikace proběhne identicky jako u normálního klienta, s tím rozdílem, že po načtení hlavní stránky aplikace je odemčena možnost editace jednotlivých prvků.

Každé komponentě je možné změnit základní vlastnosti, jako jsou viditelnost, rozměr, barva pozadí, velikost a barva okrajů atp. Pokud by se jednalo o prvek obsahující text, je umožněna změna celého textu a většina jeho parametrů. Každá změna se okamžitě projeví pouze v upravované aplikaci a tím má administrátor možnost nakonfigurovat vzhled celé aplikace dle svých představ a požadavků. Pro projevení změn ve všech

aplikacích využívajících aktuální nastavení, je nezbytné je uložit na webový server. Poté je každý uživatel informován o novém nastavení, které se následně stáhne a aplikuje.



Obr. 13: Stránka se zvýrazněnými objekty



Obr. 14: Mobilní verze administrace

Na náhledu vlevo (Obr. 13) je znázorněna stránka aplikace určená pro zobrazování prezentačních obrázků či map se spuštěným zvýrazněním editovatelných komponent. Po vybrání libovolného prvku je administrátor přeměrován do sekce s možností upravení jednotlivých parametrů (Obr. 14). Zde jsou jednotlivé elementy popsány pomocí identifikátorů s výpisem upravitelných vlastností. Pro znázornění je zde uvedena například možnost upravit tlačítko pro návrat na domovskou stránku. To je označeno identifikátorem „Back button“ a je možné změnit jeho zobrazovaný text, cílovou stránku na kterou se uživatel po kliknutí dostane, a veškeré barvy (tato funkce je z důvodu přehlednosti skryta).

Pro tvorbu této části bylo nezbytné vytvořit velmi rozsáhlé množství nástrojů a komponent, jelikož v době tvorby aplikace neexistovala jejich multiplatformní podpora. Například bylo zapotřebí vytvořit jednoduchý způsob výběru barev s možností vlastního nastavení všech složek včetně průhlednosti a případným textovým vstupem. Nicméně

celá implementace administrátorského rozhraní do mobilní aplikace byla po nějaké době zrušena a byla nahrazena uživatelsky přehlednějším webovým rozhráním.

#### 4.4 Webová administrace

Jelikož rozsáhlejší úpravy aplikace na malém displeji telefonu působily nepřehledně a bylo by zapotřebí vytvoření velikého množství nových komponent, které by celou aplikaci zpomalovaly, rozhodl jsem se celou administraci přepsat do podoby webové stránky.

Webová administrace byla navržena tak, aby splňovala klíčové vlastnosti původního rozhraní. Stejně jako na mobilu byla vytvořena možnost přihlášení pomocí identifikátoru konference a hesla do sekce s možností úpravy jednotlivých parametrů cílové konference. Mimo přihlášení byla k dispozici možnost vytvoření zcela nového nastavení. Aby bylo zaručeno správné vygenerování nového konfiguračního souboru a aby se mohl uživatel například inspirovat, vše je vygenerováno dle speciálních šablon. Po přihlášení či vytvoření nového projektu je zobrazena nabídka s výběrem jednotlivých stránek aplikace. Při editaci stránek jsou vypsány jednotlivé objekty aplikace společně se všemi nastavitelnými parametry. Na obrázku (Obr. 15) je vidět nastavení stejné komponenty jako v předchozím případě, tedy tlačítka na návrat na domovskou stránku.

Header

Display?

Height [0,00 - 1,00]: 0,05

Text: About

Text color:

Background color:

Back button-text: < Back

Back button-text color:

Back button-background color:

Back button-width [0,00 - 1,00]: 0,2

Target page id 0

Obr. 15: Náhled webové administrace

Hlavními výhodami webového rozhraní je jeho použitelnost na různých interaktivních zařízeních a uživatelsky příjemnější prostředí při editaci většího množství parametrů. Nicméně se jedná o oddělenou část aplikace napsané v odlišném programovacím jazyce.

## 4.5 Systém, jazyk, IDE

Jako cílové podporované systémy byly, na základě jejich největšího zastoupení mezi uživateli zvoleny Android a iOS s plánovaným rozšířením i na Microsoft Windows Phone. Jelikož sestavení aplikace pro systém Android lze provést na většině operačních systémů a tvorba iOS aplikací vyžaduje specifický kompilátor, bylo nezbytné, aby vývoj probíhal pod operačním systémem MacOS.

Celý projekt byl vytvořen pomocí vývojového prostředí Qt Creator, který byl využit i k napsání původní aplikace a nebyla tedy nutnost přizpůsobovat kód novému prostředí. Pro správný překlad jednotlivých buildů bylo nezbytné propojení s IDE Xcode, který obsahuje překladače a veškeré balíčky pro vytvoření kódu kompatibilního se zařízeními využívajícími systém iOS nebo MacOS. Dále byla nastavena podpora vývoje pro Android aplikace. Pro tyto účely byl Qt Creator propojen s již nainstalovaným Android Studiem, které obsahuje SDK pro tvorbu právě android aplikací. Dále bylo potřeba připojit JDK od společnosti Oracle, jež obsahuje sady balíčků určených pro tvorbu aplikací v jazyce Java. V neposlední řadě bylo nezbytné propojení s Android Native Development Kit (NDK), které umožňuje překlad C a C++ knihoven do podoby nativního kódu. Součástí těchto programů jsou i emulátory různých zařízení, na kterých lze výsledné aplikace testovat.

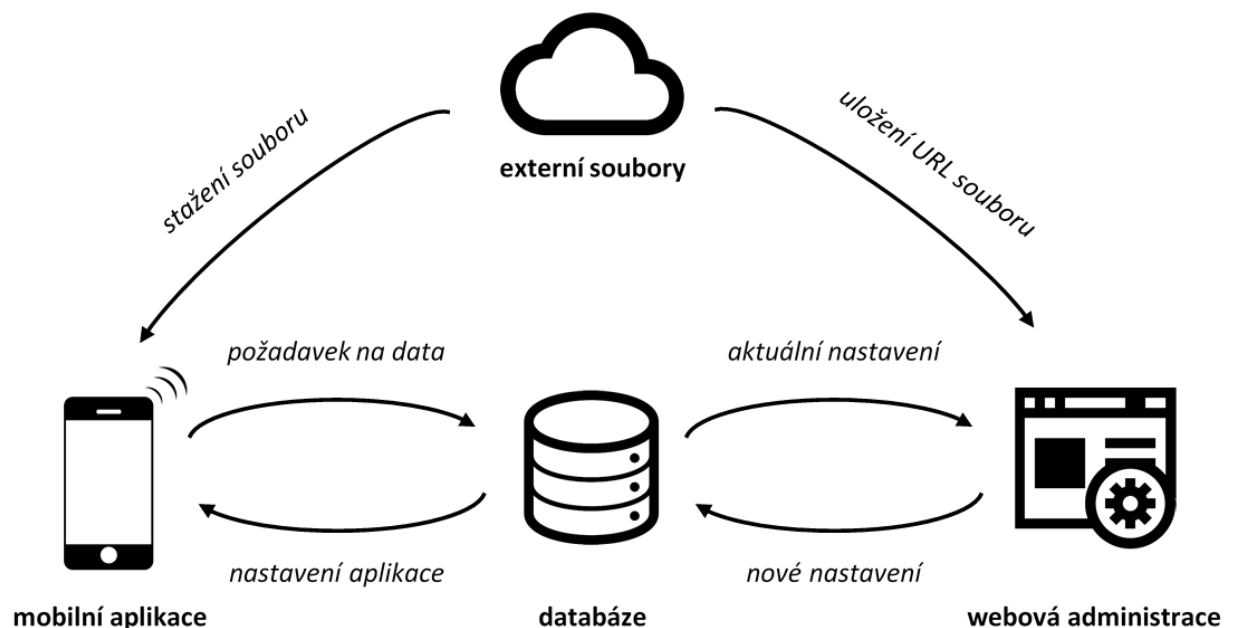
Základním jazykem na tvorbu uživatelského rozhraní byl zvolen jazyk QML, který s podporou JavaScript funkcí slouží pro přehledný zápis jednotlivých prvků. Každá součást GUI byla složena z jednotlivých objektů, jejichž parametry mohly být dynamicky upraveny na základě staženého nastavení. Pro práci s lokálními daty a zpracování stahovaných informací byly vytvořeny nástroje v programovacím jazyce C++. Pro online uložení jednotlivých nastavení byla vybrána databáze na soukromém hostingu se systémem MySQL.

## 5 Realizace řešení

Celý projekt byl rozdělen do tří částí:

- Mobilní aplikace – je určena pouze pro potřeby klienta a jsou v ní zobrazovány veškeré informace definované správcem.
- Webová administrace – v této části jsou prováděny veškeré změny cílové aplikace. Přístupy do jednotlivých nastavení byly zabezpečeny heslem, aby data mohla upravovat pouze ověřená osoba.
- MySQL Databáze – zde jsou ukládány veškeré informace o jednotlivých konferencích. Jelikož nebylo zapotřebí ukládat rozsáhlá data, byla databáze rozvržena pouze do dvou hlavních tabulek. V tabulce „appSettings“, se uchovávají celková nastavení aplikace ve formátu JSON a v tabulce „appEvents“ jsou uloženy údaje o přednáškách či jiných událostech konaných přímo na konferenci.

### 5.1 Komunikace mezi jednotlivými vrstvami



Obr. 16: Schéma komunikace součástí aplikace

Z obrázku (Obr. 16) je patrné, že je výměna dat upravených v administrační části s uživatelskou aplikací prováděna pouze skrze databázi. Tudíž veškeré změny se na klientském či testovacím přístroji projeví až po uložení změn a následné aktualizaci nastavení aplikace. Jelikož je po každé změně zapotřebí překreslení veškerých komponent, je při stažení nového nastavení prováděna kontrola. Ta ověří, zda není nový

konfigurační soubor identický se stávajícím a na základě toho rozhodne, zda se má aplikace obnovit. Při zjištění nového konfiguračního souboru je provedena jeho analýza, která má za účel nalézt odkazy na soubory uložené na webových serverech (loga, ikony atp.). Pokud je určitý soubor využit, je zařazen do fronty a následně stažen a uložen přímo do klientského zařízení. Tímto způsobem se zajišťuje dostupnost všech potřebných souborů v případě pozdějšího spuštění aplikace bez internetového připojení. Veškeré požadavky jsou prováděny s využitím REST API, které má pro každou metodu definovanou vlastní URI.

Při přístupu do webové administrace se provádí kontrola přihlašovacích údajů uložených v databázi. Po jejich ověření jsou zpracovány informace o upravovaném nastavení a je vypsán seznam jednotlivých stránek aplikace s možností jejich editace. Při úpravě jednotlivých stránek se vždy vygenerují komponenty s aktuální konfigurací, rozlišené dle identifikátorů jednotlivých komponent mobilní aplikace. Po změně uložení nových parametrů jsou veškerá data převedena zpět do formátu JSON a následně nahrána do databáze. Obrázky či podobné soubory není zapotřebí nahrávat na stejný server. Do zdrojového souboru jsou předány pouze jejich URL adresy, ze kterých jsou, po jejich načtení, staženy přímo do aplikace.

## 5.2 Mobilní část

Mobilní aplikace byla zaměřena především na přehlednost uživatelského rozhraní a jeho rychlou odezvu. Proto byla celá strukturována do podoby jednotlivých stránek, které byly dle potřeby vykreslovány.

```
if(mainSettings.configData.homePage){
    createPage("HomePage.qml", mainSettings.configData.homePage.appPageId, true);
}

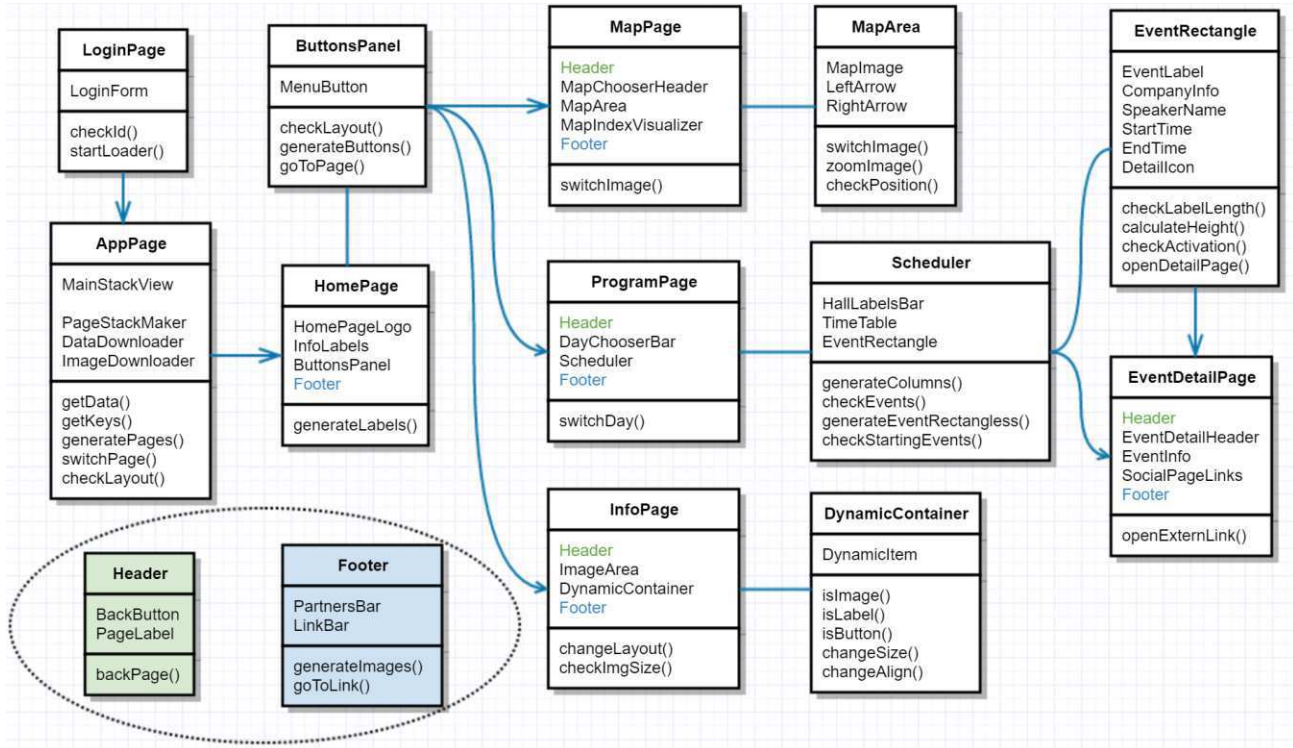
function createPage(targetQML, targetPageIndex, isVisible){
    pageQMLList.append({"source": targetQML,
                       "index": targetPageIndex,
                       "visibility": isVisible
    });
}
```

Obr. 17: Výstřižek kódu pro generování pole stránek

Pro tyto účely byl využit QML objekt „StackView“, který zajišťuje dynamické zobrazování QML komponent v podobě „balíčku karet“ a optimalizované přechody mezi nimi. Na základě staženého nastavení byly vygenerovány jednotlivé stránky. Jejich název zdrojového souboru šablony a identifikátor byly uloženy do pole objektů, které slouží jako seznam možných stránek („karet“) zobrazovaných prostřednictvím „StackView“. Na

obrázku (Obr. 17) je uveden výstřížek funkce ukládající informace do zdrojového objektu.

### 5.2.1 Struktura komponent aplikace



Obr. 18: Základní struktura komponent mobilní aplikace

Na diagramu (Obr. 18) je znázorněna základní struktura jednotlivých komponent aplikace. Je zde naznačeno jejich skládání z menších objektů a využití klíčových funkcí. Z obrázku je patrné, že po spuštění aplikace a přihlášení do požadované konference, se uživateli zobrazí domovská stránka „HomePage“. Zde lze, při kliknutí na některé z tlačítek „MenuButton“, přecházet na ostatní plochy aplikace. Komponenty „Header“ a „Footer“ se mohou vyskytovat na všech stránkách. Jsou navrženy formou šablony, která je vložena do komponenty stránky a je jí předáno požadované nastavení. Tímto způsobem lze jejich vzhled a chování upravovat pouze na vybraných místech při zachování jejich struktury a funkcionality.



## 5.2.2 Výběr konference

Při spuštění aplikace je uživateli zobrazeno pole pro zadání unikátního identifikátoru požadované konference. Prozatím byla databáze vytvořena pouze z testovacích dat a jako klíč posloužil název konference. Po ověření bezchybného zadání hesla je provedena kontrola, zda se již v zařízení nenachází předchozí uložené nastavení. V případě jeho absence je staženo aktuální nastavení z databáze ve formátu JSON. Následně jsou tato data převedena do podoby objektu, na kterém je provedena analýza struktury. Na základě klíčových prvků je objekt rozdělen na jednotlivé stránky, jejichž identifikátory jsou uloženy do pole, které slouží jako seznam „listů“ hlavní aplikace pro komponentu StackView. Poté se pro každou stránku provede algoritmus ke zjištění odkazů na externí soubory.

```
function getKeys(keys, obj, path, settingsImages) {
  for (var key in obj) {
    var currpath = path + '/' + key;
    keys.push([key, currpath]);
    if (key == "links" || key == "link") {
      if (key == "link") {
        var tmp = checkStructure( obj[key].trim() );
        if (tmp != -1 && settingsImages.indexOf(tmp) <=-1) {
          settingsImages.push(tmp);
          WorkerScript.sendMessage({ "messageId":"IMGS found: " + settingsImages.length });
        }
      } else {
        for (var i = 0 ; i < obj[key].length; i++) {
          var tmp = checkStructure( obj[key][i].trim() );
          if (tmp != -1 && settingsImages.indexOf(tmp) <=-1) {
            settingsImages.push(tmp);
            WorkerScript.sendMessage({ "messageId":"IMGS found: " + settingsImages.length });
          }
        }
      }
    }
    if (typeof(obj[key]) == 'object') {
      getKeys(keys, obj[key], currpath, settingsImages);
    }
  }
}
```

Obr. 19: Ukázka kódu na rekurzivní procházení objektem

Vyhledávání probíhá pomocí rekurzivní funkce (Obr. 19), která prochází celou strukturu hlavního objektu a jeho vnořených objektů. Pokud je nalezena adresa odkazu na externí obrázek, okamžitě se zařadí do fronty na stažení a po zařazení posledního nalezeného souboru je automaticky zahájeno jejich stahování přímo do zařízení. Zároveň je o každém nalezeném obrázku uživatel informován pomocí zprávy, která se vyvolává v reakci na signál „sendMessage“ a obsahuje informaci o aktuálním počtu nalezených souborů. Stahování těchto dat je zabezpečeno pomocí vytvořené C++ třídy „FileDownloader“, která pro zpracování požadavků využívá třídu „QNetworkRequest“, jenž je součástí Qt Network Access API. Veškerá sazená data se ukládají do zařízení, pro

opětovné pozdější použití i bez nutnosti internetového připojení. Po úspěšném stažení všech potřebných souborů je spuštěna hlavní část aplikace.

### 5.2.3 Struktura stránek

Každá stránka hlavní části aplikace je uložena v objektu „Rectangle“ a je rozdělena na tři hlavní části: hlavičku, tělo stránky, patičku (Obr. 20). Tyto jednotlivé komponenty jsou nastaveny tak, aby se „HeaderBar“ vždy vykresloval v horní části displeje, „Footer“ vždy ve spodní části displeje a tělo stránky vždy vyplnilo prostor mezi nimi. Tím je zajištěno využití celé plochy aplikace a možnost dynamicky měnit tělo stránky. Každému prvku jsou předávány konfigurační objekty obsahující nastavení určené právě pro daný element. Tato zdrojová struktura je uložena v globálním nastavení celé aplikace, aby byla zajištěna její dostupnost ve všech částech programu. Při aktualizaci nastavení dojde k přepsání uložených dat a veškeré komponenty jsou okamžitě překresleny.

```
Rectangle {
    id: mapPage
    property int appPageIndex: 0
    color:mainSettings.configData.mapPage.bgColor
    HeaderBar {
        id: headerBar1
        myParent: parent
        configObj: mainSettings.configData.mapPage
    }
    MapChooser{
        myParent: parent
        anchors.top: headerBar1.bottom
        anchors.bottom: footer1.top
        configObj: mainSettings.configData.mapPage.mapChooser
    }
    Footer{
        id: footer1
        myParent: parent
        configObj: mainSettings.configData.mapPage
    }
}
```

*Obr. 20: Základní struktura stránky aplikace*

### 5.2.4 Hlavní stránka

Po stažení všech potřebných souborů je uživatel přesměrován přímo do hlavní části aplikace, kde přepínání jednotlivých oken přebírá komponenta StackView. Při načtení je jí předána první položka z pole stránek, která obsahuje šablonu objektu pro hlavní plochu

aplikace. Ta již obsahuje pouze parametrické objekty, kterým je předáno kompletní nastavení přímo ze staženého konfiguračního souboru.

```
Repeater {  
  model: mainSettings.configData.homePage.buttons  
  MenuButton {  
    visible: mainSettings.configData.homePage.buttons[index].display  
    width: buttonPanel.height  
    height: buttonPanel.height  
    buttonSettings: mainSettings.configData.homePage.buttons[index]  
    onSendClicked: {  
      showActualTime = true;  
      switchPage(mainSettings.configData.homePage.buttons[index].targetId);  
    }  
  }  
}
```

*Obr. 21: Výstřížek kódu generující tlačítka*

*Na obrázku (*

Obr. 21) je uvedena malá část kódu určená pro vykreslení panelu tlačítek sloužících k přechodu na jednotlivé stránky. Objekt „Repeater“, na základě velikosti pole „buttons“ předaného z konfiguračního objektu, provede daný počet cyklů, ve kterých postupně vygeneruje nové objekty „MenuButton“. Těm jsou předány parametry udávající veškeré potřebné nastavení.

Na hlavní stránce je také možnost zobrazit logo či obrázek reprezentující společnost organizující celou událost. Tento objekt je vždy upraven dle velikosti zařízení tak, aby byly zachovány jeho proporce a aby si zachoval identický poměr vůči ostatním komponentám. Pod logem jsou k dispozici čtyři elementy, které mohou být pomocí nastavení změněny buď jako textové pole, nebo jako pole obsahující obrázek. Oběma variantám je možné nadefinovat vertikální nebo horizontální zarovnání a lze jim nastavit viditelnost.

### 5.2.5 Mapy a obrázky

Další stránka, která může být v aplikaci využita, je určena pro zobrazení speciální galerie. Zde uvedené obrázky mohou posloužit jako reklamní upoutávky, prezentační materiál nebo je zde možné zobrazit například mapy jednotlivých sálů. Záleží pouze na organizátorovi, zda tuto stránku chce využít a co na ni chce zpřístupnit. Galerie disponuje multidotykovým ovládním, které uživateli umožňuje využít základní ovládací prvky, na které je zvyklý z ostatních aplikací. Je zde možnost přecházení mezi jednotlivými snímky pomocí tažení do strany, přibližování či oddalování snímku dvěma prsty atp.

Pohyb v náhledu obrázku je zpracováván pomocí komponenty „Flickable“, která je, jako většina využitých nástrojů, součástí balíčků QtQuick. V tomto objektu lze zpracovávat několik signálů. „Drag“, který definuje jednoduché tažení obrázku nebo například „Flick“, který při přerušení dotyku nechává prvek plynule zastavit. Další důležitou komponentou je „PinchArea“ reagující na gesta více prsty (přiblížení, natočení atp.). Zde bylo zapotřebí zabezpečit správné chování při změnách přiblížení.

```
if((tempHeight * pinch.scale >= scrollWindow.height ||
    tempWidth * pinch.scale >= scrollWindow.width) && scrollWindow.pinchning){
    scrollWindow.contentX += pinch.previousCenter.x - pinch.center.x;
    scrollWindow.contentY += pinch.previousCenter.y - pinch.center.y;
    if(tempWidth * pinch.scale < scrollWindow.width ||
        tempHeight * pinch.scale < scrollWindow.height ){
        if(tempWidth * pinch.scale < scrollWindow.width){
            scrollWindow.resizeContent(scrollWindow.width,
                tempHeight * pinch.scale, pinch.center);
        }else{
            scrollWindow.resizeContent(tempWidth * pinch.scale,
                scrollWindow.height , pinch.center);
        }
    }else{
        scrollWindow.resizeContent(tempWidth * pinch.scale,
            tempHeight * pinch.scale, pinch.center);
    }
}
```

Obr. 22: Výstřižek kódu pro optimalizaci přiblížování obrázku

Výchozí způsob zvětšování obrázku zachovává střed objektu v původním bodě, ale při zoomu dvěma prsty se očekává přiblížení v oblasti dotyku. Proto bylo nezbytné dynamicky přizpůsobovat pozici obsahu dle středu zachytávaného gesta. Zároveň byla implementována funkce zamezující nekonečné zvětšení či zmenšení obrázku. Náhled použité metody je znázorněn na obrázku (Obr. 22).

Celá galerie je strukturována do podoby záložek, kde každá obsahuje jiný obrázek. Administrátor může, mimo konfigurace veškerých barev, určit i počet jednotlivých listů, obrázky, které se na nich zobrazují a název každé záložky. Pro přehlednost může být každá stránka označena barevným identifikátorem v podobě kolečka. Tento jednoduchý element slouží jak k přehlednosti rozhraní, tak například k názornějšímu rozlišení přednáškových sálů či hal. Přejít mezi jednotlivými kartami je umožněn kliknutím na hlavičku záložky, šipkami pro přechod mezi obrázky v galerii či tažením obrázku do strany.

## 5.2.6 Stránka s programem

Nejsledovanější součástí každé události je její harmonogram, a proto je zapotřebí, aby byl přehledný a poskytoval návštěvníkovi co nejpřesnější údaje. Pro tyto účely slouží v aplikaci šablona pro vygenerování interaktivního výpisu akcí. Administrátor má možnost vytvořit seznam jednotlivých přednášek s podrobnými informacemi a určit, jak mají být v aplikaci zobrazeny.

Do návrhu byla zahrnuta i možnost, že by se událost mohla konat více dní nebo by mohla mít prezentace naplánované na několik různých termínů. Z tohoto důvodu byla do aplikace umístěna možnost výběru libovolného data ze speciálního kalendáře. V jeho náhledu jsou pro přehlednost odlišeny dny, ve kterých se odehrává určitá přednáška. Uživatel může přejít na vybraný termín nebo využít rychlou volbu zobrazení aktuálního dne.

Informace jsou zobrazovány ve sloupcích, kde každý definuje odlišný sál či halu a je rozdělen na časové intervaly. Šířka a hodnota jednotlivých úseků může být upravena v administraci.

```
Flickable {
  property bool slideUpdate: true
  id:scrollWindow
  anchors.fill: parent
  contentWidth: configObj.halls.length * mainParameter * configObj.recWidth
  contentHeight: timeTable1.height
  clip: true
  flickableDirection: Flickable.HorizontalAndVerticalFlick
  Repeater {
    id: repeater1
    model: eventList.length
    EventRectangle{
      //eventLabelText: FScript.getLabelAt(index)
      y: timeTable1.myHook + FScript.calculateTimeToPixelForm(eventList[index].
      height: {
        if(FScript.calculateTimeToPixelForm(eventList[index].endTime) >
          return (FScript.calculateTimeToPixelForm(intervalStop.toString)
        }else{
          return (FScript.calculateTimeToPixelForm(eventList[index].endTime)
        }
      }
    }
    visible: {
      (eventList[index].dayCode === "00000000" && mainSettings.adminF
      ((FScript.checkDayCode(eventList[index].dayCode,mainSettings.sel
      (FScript.calculateTimeToPixelForm(eventList[index].startTime
      (FScript.calculateTimeToPixelForm(eventList[index].startTime
    )
  )
}
```

*Obr. 23: Výstřižek kódu vykreslující objekty reprezentující přednášky*

Aby nemusely být veškeré informace zmenšovány na velikost displeje zařízení, je celý program v objektu „Flickable“ (Obr. 23). Díky nastavení jeho parametru „flickableDirection“ je umožněn jeho horizontální i vertikální posun a není tak omezena

velikost obsahu. Každá přednáška je specifikována barevným obdélníkem objektu „EventRectangle“, jehož výška je určena délkou jeho trvání. Na tomto objektu jsou vypsané základní údaje zahrnující název prezentace, jméno mluvčího, čas začátku a čas konce přednášky.

Pokud je u přednášky povolen náhled detailu, pak má uživatel možnost otevřít stránku obsahující podrobné informace vybrané přednášky. Na této kartě jsou uvedeny stejné údaje jako v programu s několika doplňujícími funkcemi. Například se zde zobrazují loga či fotografie řečníka nebo odkaz na domovskou stránku či sociální sítě. Také jsou zde podrobnější informace o přednášce či jiný text definovaný administrátorem.

Veškerá data o přednáškách jsou uložena ve speciální tabulce v databázi a jsou načítána odděleně od hlavního nastavení aplikace. V této struktuře je zapotřebí dodržet definici identifikátorů jednotlivých sálů, aby byly události vykresleny do správných sloupců. To je ošetřeno na straně webového rozhraní administrace.

### 5.2.7 Stránka s informacemi

Mezi univerzálnější šablonu patří stránka, která nezobrazuje jen informace o aplikaci. Její struktura obsahuje plochu rozdělenou na dvě části, kam je možné umístit libovolný počet elementů definovaných speciálními objekty. Ty jsou poté zobrazeny dle definovaného rozložení s možností posouvání celého panelu. Na ukázce (Obr. 24) je uveden základní prvek „Rectangle“ který pomocí změny parametrů může sloužit buď jako textové pole, objekt obsahující obrázek, tlačítko nebo jednoduchý oddělovač.

```
Rectangle{
    property bool isButton: configObj.labels[index].isButton
    width: isButton ?
        Math.min(mainParameter *configObj.labels[index].width, parent.width) : parent.width
    height: isButton ?
        Math.min(mainParameter *configObj.labels[index].height , parent.height ) : parent.height
    radius: isButton ? bigRadius : 0
    color: isButton ? configObj.labels[index].bgColor : "transparent"
    border.color: isButton ? configObj.labels[index].borderColor : "transparent"
    border.width: isButton ? mainBorderWidth : 0
    anchors.centerIn: parent
    Text {
        text: qsTr(configObj.labels[index].text + configObj.labels[index].fontSizeIndex)
        font.pixelSize:{
            switch(configObj.labels[index].fontSizeIndex){
                case 0 : return smallFontSize;
                case 1 : return mediumFontSize;
            }
        }
    }
}
```

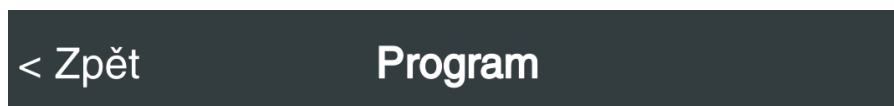
Obr. 24: Ukázka kódu dynamického prvku

V tomto případě je přes proměnnou „isButton“ určeno, zda si má objekt brát parametry ve struktuře pro tlačítko nebo zda bude brán jako textové pole. Například

textové pole (pokud není ohraničeno rámečkem) nepotřebuje mít zaoblené hrany, nebo není zapotřebí kontrola stisknutí. V situaci, kdy je element brán jako tlačítko, pak se vnořený prvek „Text“ bere jako popisec tlačítka. V každém případě je prostřednictvím konfiguračního souboru možné nastavit rozsáhlé množství nastavení.

Tato stránka může mít různý charakter a vše závisí jen na jejím návrhu. Při využití pouze textových polí lze snadno vytvořit stránku obsahující pouze informace o aplikaci či společnosti konající konferenci. Pokud chceme využít vlastnosti dynamického tlačítka, může mu být předána URL pro otevření webové stránky, ID pro přechod na libovolnou stránku aplikace či telefonní číslo pro okamžité vytočení.

### 5.2.8 Hlavička



Obr. 25: Náhled hlavičky aplikace

„HeaderBar“, neboli hlavička aplikace, je jednoduchá, nicméně nezbytná součást každé stránky. Z domovské plochy se lze pomocí tlačítek dostat na jednu z vybraných šablon, dle definovaného indexu. Přechody jsou zpracovávány pomocí komponenty „StackView“, která listy zobrazuje ve stylu balíčku karet. Aby bylo možné přejít na kartu o úroveň níže, je zapotřebí předat stránkovací komponentě signál „o krok zpět“. Na zařízeních disponujících navigačními klávesami je tato funkce propojena přímo s tlačítkem „zpět“. Nicméně například mobilní telefony od společnosti Apple taková tlačítka nemají. Z těchto důvodů je nezbytné, aby aplikace obsahovala tuto navigační klávesu v softwarové podobě.

```
MouseArea{
  anchors.fill: parent
  onClicked: {
    backpage(configObj.header.targetId);
    prevPageIndex = 0;
    eventDetailIndex = 0;
  }
  onPressed: {
    backText.color = configObj.header.backTextColorA;
  }
  onReleased: {
    backText.color = configObj.header.backTextColor;
  }
}

function backpage(targetIndex) {
  appStack.pop();
  actualPageIndex = targetIndex;
  print(appStack.depth);
}
```

Obr. 26: Kód zajišťující návrat na předchozí stránku

Na obrázku (Obr. 26) je zobrazena část kódu objektu tlačítka „Zpět“, která obsahuje komponentu „MouseArea“ umožňující zachytávat vstupy od uživatele. Pokud dojde ke stisknutí tlačítka, je změněna jeho barva na hodnotu definovanou v „backTextColorA“. Po ukončení dotyku je zaznamenán jak signál „onReleased“, ve kterém je nastavena nová barva textu, tak i „onClicked“, kde je vyhodnocena funkce pro přechod na předchozí stránku. Komponenta „StackView“ si informace o jednotlivých přechodech ukládá do struktury zásobníku, kde je na jeho vrchol vloženo metodou „push( idStranky )“ a odebíráno pomocí metody „pop()“.

### 5.2.9 Patička

„Footer“, neboli patička aplikace, je panel, který je určen pro zobrazení doplňujících informací nebo může splňovat funkci tlačítka. Zobrazuje vespod aplikace a je možné ho vypnout skrze administraci. Byly zde vytvořeny dva typy těchto objektů, které mohou být zobrazeny jednotlivě či společně na kterékoliv stránce aplikace.

První typ je označen jako „PartnersBar“, jelikož jeho zamýšlený účel je zobrazovat loga veškerých sponzorů či partnerů konference. Tyto obrázky jsou předány pomocí pole z administrace a jsou zobrazeny v řádku umožňujícím horizontální posouvání. Možné provedení je znázorněno v horní části obrázku (Obr. 27).

Druhý objekt typu „LinkBar“, v dolní části obrázku (Obr. 27), může posloužit buď pro jednoduché zobrazení krátkého textu, nebo je možné aktivovat funkce reagující na vstup od uživatele. V tomto případě ho lze využít jako tlačítko s volitelnou funkcionalitou.

K oběma komponentám je možné zobrazovat nastavitelný titulek, měnit veškeré texty a barvy atp. Nastavení jejich výšky je limitováno maximální hodnotou, která se určuje tak, aby zobrazovaná patička negativně neovlivňovala velikost plochy hlavní části aplikace.



Obr. 27: Náhled patičky aplikace



## 5.2.10 Notifikační panel

V horní části obrazovky aplikace se na základě různých signálů zobrazuje informační panel s oznámením o události. Uživatel je tímto způsobem informován o aktuálním stavu aplikace, jako je například stažení nových dat, ztráta internetového spojení se serverem atp. Hlavní účel této lišty ale spočívá v upozornění návštěvníka na začátek některé z přednášek a rychlé volby jejího zobrazení.

Tyto události jsou vyvolávány jako reakce na vyhodnocení kontrolní funkce, která se provádí v definovaných časových intervalech. Při jejím spuštění jsou prověřeny veškeré přednášky uložené v poli pro aktuální den a je provedena kontrola, zda se čas jejich začátku shoduje se systémovým časem zařízení. V takovémto případě je přednáška označena jako aktivní a je vyvolána notifikační lišta obsahující její název a tlačítko pro rychlý přístup na stránku obsahující podrobnější informace o prezentaci. V situaci, kdy potřebujeme vyvolat více panelů s upozorněním, například pokud začínají dvě různé přednášky ve stejný okamžik, je zobrazena druhá lišta, která se plynule vysune pod první. Veškeré signály vyvolávající tato upozornění jsou zařazovány do fronty, a jakmile se jeden z využitých panelů uvolní, je okamžitě využit pro zobrazení další zprávy.

```
WorkerScript.onMessage = function(message) {
  if(message.showState){
    if(message.firstBar) {
      message.firstEventNotificationReady = false;
      message.topMargin += message.height*message.interval/message.msToLoad;
    }else{
      if(message.firstEventNotificationReady){
        message.topMargin +=message.height*message.interval/message.msToLoad;
      }
    }
    if(message.topMargin>=0){
      message.topMargin = 0;
      message.showState = false;
      message.interval = 5000;
      if(message.firstBar) message.firstEventNotificationReady = true;
    }
  }else{
    message.interval = 25;
    message.firstEventNotificationReady = true
    message.topMargin -= message.height*message.interval/message.msToLoad;
    if(message.topMargin*-1 >= message.height){
      message.topMargin = message.height*-1;
      message.showState = true;
      message.running = false;
      message.visible = false;
      if(message.myNotificationIndex === message.firstEventNotificationIndex){
        message.firstEventNotificationIndex = 0;
      }else{
        message.secondEventNotificationIndex = 0;
      }
    }
  }
}
```

Obr. 28: Část kódu pro řízení notifikačního panelu

Pro zajištění plynulých přechodů a pohybů obou objektů bylo zapotřebí jejich vykreslování v závislosti na jejich vzájemné aktuální poloze. Na obrázku (Obr. 28) je výstrižek JavaScriptové funkce zpracovávající aktuální stavy obou panelů.

Jedná se o funkci prováděnou ve vlastním vlákne, kde za pomoci zpráv jsou hodnoty předávány do vlákna hlavní aplikace. Poloha jednotlivých objektů je řízena pomocí úpravy jejich odsazení vůči horní hraně aplikace, kde se každých 25ms provede požadovaná změna. Po plném vysunutí je zpráva ponechána k nahlédnutí po dobu 5000ms a následně je vykonáno její postupné zasunutí. Jelikož je výška informační lišty závislá na velikosti a rozlišení zařízení, je pro zachování plynulosti animace nezbytné parametrické vypočítávání přírůstku odsazení.

### 5.3 Webová část

Jak bylo zmíněno v druhé kapitole, webová administrace byla zvolena jako přehlednější varianta pro editaci jednotlivých prvků mobilní aplikace.

Její hlavní výhoda spočívá v možnosti zobrazení celého nastavení na zařízeních s mnohem větší pracovní plochou, než mají mobilní telefony. Administrátorovi se také jednodušeji pracuje s externími soubory a případné kopírování jejich URL adres do atributů cílové komponenty je přehlednější.

Celá webová aplikace byla napsána pomocí frameworku Nette a byla rozdělena do jednotlivých stránek. Aby nedošlo k neoprávněným změnám, bylo nastavení zabezpečeno administrátorským heslem. Zároveň je webová stránka prozatím přístupná pouze z vybraných IP adres.

#### 5.3.1 Přidání konference

Po zobrazení webové stránky je administrátorovi umožněna buď editace již vytvořeného nastavení, nebo vytvoření celé nové konference. Každá konfigurace se ukládá do databáze, kde je definována unikátním identifikátorem, ke kterému náleží zabezpečovací heslo a konfigurační soubor ve formátu JSON. Pro usnadnění tvorby celého nastavení je při vytváření nového souboru možnost výběru z již přednastavených a optimalizovaných šablon. Jejich zdrojový kód se po výběru překopíruje přímo k aktuálně vytvořené konferenci a je ihned zpřístupněn k editaci.

```

public function postFormSucceeded($form, $values)
{
    $confTemplateName = $values['tempName'];
    $templateData = $this->database->table('appSettings')->get($confTemplateName);
    if (!$templateData) {
        $this->error('špatné id šablony');
    }
    $toSave['data'] = $templateData['data'];
    $toSave['confName'] = $values['confName'];
    $this->database->table('appSettings')->insert($toSave);
    $this->flashMessage("Konference vytvořena.", 'success');
    $this->redirect('AppSettings:pageSelection', $toSave['confName']);
}

```

Obr. 29: Funkce pro vytvoření nového nastavení konference

### 5.3.2 Úprava konference

Editace jednotlivých položek je prováděna skrze vygenerované komponenty formuláře. Veškerá data potřebná pro jejich vyplnění se získají z konfiguračního souboru staženého z databáze a převedeného z formátu JSON do podoby PHP objektu. Následně je již administrátorovi zobrazen výpis nastavení jednotlivých prvků cílové mobilní aplikace. Veškeré části jsou rozlišeny pomocí identifikátorů definujících upravované elementy.

```

protected function createComponentLabelSettingsContainer($targetForm, $key) {
    $labelContainer = $targetForm->addContainer('labelContainer' . $key);
    $labelContainer->addCheckbox('isImage', 'Is image?');
    $labelContainer->addCheckbox('display', 'Display?');
    $labelContainer->addText('text', 'Text:');
    $labelContainer->addText('textColor', 'Text color:');
        ->setType('color');
    $labelContainer->addText('align', 'Align:');
        ->setType('number')
        ->addRule(Form::INTEGER);
    $labelContainer->addText('link', 'Link:');
}

```

Obr. 30: Funkce pro tvorbu komponenty titulku

Na obrázku (Obr. 30) je náhled funkce vytvářející komponentu určenou pro nastavení parametrů jednoduchých textových popisků či štítků. Ta přidává do cílového formuláře prvky sloužící k upravení nastavení cílového objektu a umožňuje pozměnit jeho text, barvu, zarovnání a viditelnost. Dále je možnost přidat příznak určující, že má být prvek zobrazen pouze jako obrázek a přidat jeho externí odkaz.

Kdyby se například jednalo o nastavení tlačítka, a byl by do položky „link“ zapsán identifikátor stránky mobilní aplikace, pak by po jeho stisknutí byla uživateli zobrazena

právě cílová stránka. Pokud by zde byla vložena URL webové stránky, pak by byl uživatel přesměrován právě na nastavený web prostřednictvím výchozího internetového prohlížeče v zařízení. Pro využití tlačítka jako volby pro přímé zavolání stačí do parametru předat telefonní číslo a označit element příznakem „call“. Následně se do kódu aplikace přidá příznak „tel:%1“ kde, je argument „%1“ nahrazen telefonním číslem a po jeho zavolání je uskutečněn požadovaný hovor.

V případě konfigurace prvků podporujících více vnořených objektů, jako je například panel pro zobrazení sponzorů, stačí do kolonky přidat pole jednotlivých záznamů oddělených čárkami. Poté jsou v aplikaci vykresleny přesně v daném pořadí.

Veškeré nastavení se po jeho uložení převede zpět do podoby souboru JSON a je nahráno zpět na server. Následně každá aplikace při kontrole aktualizací zjistí, že byly provedeny změny a provede obnovení své celkové konfigurace.

### 5.3.3 Editace přednášek

Pole	Typ	Nulový
<b>eventId</b>	int(11)	Ne
<b>confName</b>	varchar(50)	Ne
<b>name</b>	tinytext	Ne
<b>company</b>	tinytext	Ano
<b>pictureURL</b>	text	Ano
<b>label</b>	text	Ne
<b>hall</b>	int(11)	Ne
<b>startTime</b>	varchar(5)	Ne
<b>endTime</b>	varchar(5)	Ne
<b>detail</b>	tinyint(1)	Ne
<b>info</b>	text	Ano
<b>url</b>	text	Ano
<b>facebook</b>	text	Ano
<b>twitter</b>	text	Ano
<b>dayCode</b>	varchar(8)	Ne

Tab. 1: Struktura tabulky databáze pro ukládání prezentací

Obdobným způsobem je řešeno přidávání a editace jednotlivých přednášek. Při zobrazení nastavení konference má administrátor možnost přejít do náhledu prezentací, kde je mu umožněna jejich editace, odstranění nebo přidání nové události. Každý záznam je s konferencí svázán jejím unikátním klíčem a, na rozdíl od nastavení aplikace, veškerá data jsou uložena v jednotlivých sloupcích tabulky databáze.

Při vytváření nového záznamu je povinné vyplnit jméno řečníka nebo řečnicků, „label“ neboli název přednášky, číslo sálu, čas plánovaného začátku a konce přednášky a den jejího konání. Pomocí těchto údajů je komponenta vykreslena v mobilní aplikaci.

## 6 Vyhodnocení aplikace

Výsledná aplikace byla v průběhu vývoje mnohokrát testována na několika přístrojích s různým hardwarem a nastavením. Díky tomu bylo vyzkoušeno její chování v rámci různých operačních systémů a úspěšně ověřeno dynamické přizpůsobování jednotlivých komponent.

Prověřeno bylo chování v různých verzích operačního systému Android. Jako nejstarší byla ozkoušena verze 2.3. Gingerbread na mobilním telefonu HTC Desire S, kde byla aplikace plně funkční, ačkoli její odezva byla pomalejší. Další testy byly prováděny na Samsungu Galaxy S3 mini s operačním systémem Android 4.1. Jelly Bean a nejnovější testovaná verze byla Android 5.1 Lollipop na telefonu Samsung Galaxy S4. Pro prověření správného chodu aplikace na zařízení s větším displejem byl využíván tablet Samsung Galaxy Note 10.1 se systémem Android 4.4.2 KitKat. Na všech novějších testovaných zařízeních byl chod aplikace plynulý a vždy se celé prostředí optimálně přizpůsobilo velikosti displeje.

Stejný způsob testování probíhal i na zařízeních s operačním systémem iOS. K těmto účelům byly využity mobilní telefony Apple iPhone 4s se systémem iOS 5 a Apple iPhone 5s se systémem iOS 9.3. Z větších zařízení byl použit Apple iPad Air. Na všech testovaných zařízeních byl chod aplikace plynulý a stabilní.

### 6.1 Výhody

#### 6.1.1 Upravitelnost

Aplikace, kterou jsem vytvořil, má v porovnání s jinými aplikacemi veliké výhody právě díky její rozsáhlé upravitelnosti. Záleží pouze na návrhu možného nastýlování jednotlivých komponent a tím je možné vytvořit naprosto unikátní vzhled i chování při zachování identického jádra. Díky tomu lze projekt nakonfigurovat tak, aby dodržel design organizace nebo ladil s ostatními aplikacemi společnosti. Také lze použít jen některé z předinstalovaných šablon a využívat například pouze část s programem či galerií. Tímto způsobem lze pro toto rozhraní nalézt velmi rozsáhlé možnosti využití.

#### 6.1.2 Multijazyčnost

V hlavní aplikaci není žádný zobrazovaný text přímo součástí zdrojového kódu, ale bere se jako parametr z konfiguračního souboru. Díky tomu lze rozhraní zobrazit

v libovolném jazyce pouze změnou v administraci. Zároveň lze do textu vkládat libovolné HTML příznaky, což usnadňuje vkládání speciálních symbolů či formátování pouze částí textu.

### 6.1.3 Využití off-line

Jelikož se veškeré nastavení a ostatní soubory stahují přímo do zařízení, je po první aktualizaci možné aplikaci spustit bez nutnosti internetového připojení. Tímto způsobem není uživatel závislý na stabilitě či dostupnosti připojení a má neustále k dispozici původní poskytované údaje.

### 6.1.4 Rychlost

Absence administrace v přímo v mobilní aplikaci a využití „StackView“ jako hlavní řídicí komponenty velice urychlilo chod celého uživatelského prostředí. Veškeré komponenty reagují téměř okamžitě i na málo výkonných zařízeních a tudíž není nijak omezována rychlost interakce s uživatelem. Zároveň je zachována identická funkcionality a rozložení všech prvků.

## 6.2 Nevýhody

Jednotlivé objekty využívají komponenty obsažené v balíčcích QtQuick, které jsou velice rozsáhlé. Z tohoto důvodu má výsledná aplikace větší nároky na prostor v paměti zařízení. Nicméně nejedná se o velikost, která by nějak zásadně ovlivňovala možnost využití na všech podporovaných zařízeních.

Z důvodu uchování veškerých souborů a nastavení přímo v paměti zařízení je urychleno opakované spouštění aplikace bez nutnosti stahování nových dat. Díky tomu je i sníženo využití datového tarifu uživatele. Nicméně zatím nebyla v aplikaci implementována funkce minimalizující velikost stahovaných obrázků, a proto velikost stažených dat záleží i na administrátorovi a jím zvolených souborech.

## 7 Shrnutí práce

### 7.1 Hlavní body jednotlivých kapitol

V úvodní kapitole byla nastíněna problematika výběru informačních prostředků na výstavách či konferencích. Byly zde uvedeny možnosti využití moderních informačních přístupů a jejich výhody a nevýhody. Dále bylo vybráno několik nejdůležitějších informací, které je potřeba návštěvníkovi předat. Také zde bylo vybráno několik možností způsobu zpracování informačního rozhraní a byly uvedeny jejich omezení. Zároveň byla provedena rešerše nejvíce využívaných mobilních platforem dle jejich zastoupení na světovém trhu. Na základě zjištěných informací byla nastíněna možnost využití vývojového prostředí určeného k návrhu multiplatformních aplikací pro návrh vlastního projektu.

V návrhu řešení byla podrobněji popsána problematika návrhu konfigurovatelné mobilní aplikace a byly zde uvedeny přístupy optimalizující její přehlednost a chování. Hlavním cílem celého návrhu bylo upřesnit chování výsledné aplikace, která by obsahovala pouze dynamické komponenty a její identické chování na různých typech a platformách mobilních zařízení. Součástí bylo navržení administrativního rozhraní, jehož prostřednictvím by bylo možné veškeré prvky upravit.

V hlavní části práce byla podrobněji popsána realizace celého projektu zahrnující nejdůležitější funkce a vlastnosti vytvořené mobilní aplikace. Byly zde popsány jednotlivé stránky uživatelského rozhraní, u kterých byla nastíněna jejich informační hodnota pro uživatele. Dále zde byl znázorněn princip komunikace aplikace s ostatními součástmi projektu. V neposlední řadě byl vysvětlen princip úpravy a chování dynamických komponent, zároveň s přiblížením práce v administrátorském rozhraní.

Při vyhodnocení funkcionality výsledné aplikace byly zmíněny její hlavní výhody a nedostatky spojené s návrhem celého projektu. Byly zde vyjádřeny zejména výhody spjaté s dynamičností jednotlivých komponent a možnost rozšířeného pole využití celé aplikace. Bylo upozorněno na nároky datových přenosů a jejich možnou minimalizaci.

### 7.2 Splnění cílů

Hlavním cílem této práce bylo vytvořit univerzální mobilní aplikaci pro výstavy a konference, která může být prostřednictvím administrace upravena do nové podoby. S tím souviselo pochopení problematiky programování multiplatformních mobilních



aplikací a určení optimálních nástrojů sloužících pro zobrazení informací návštěvníkovi konference. Dále bylo nezbytné vytvořit přehledné administrátorské rozhraní sloužící k úpravě dynamických prvků aplikace.

Přestože výsledná aplikace není zatím veřejně dostupná, splňuje veškeré požadavky, a tudíž byl tento cíl splněn.

### 7.3 Plány budoucího rozšíření

Využití tohoto projektu je již nyní široké, ale samozřejmě, jako u všech softwaru, je nezbytný neustálý vývoj a udržování kroku s novými technologiemi. Zároveň je plánováno zoptimalizování databázové vrstvy, aby byla minimalizována redundantnost dat a umožněna editace pouze drobných částí nastavení. Dalším krokem je minimalizace stahovaných dat při využití nástrojů pro zmenšení velikosti obrázků. A v neposlední řadě přidání nových šablon jednotlivých stránek umožňujících práci s dalšími dynamickými komponentami.

## 8 Seznam použité literatury

- [1] FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures* [online]. University of California, 2000 [cit. 2013-10-08]. Dostupné z: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [2] ALLAMARAJU, Subrahmanyam. *RESTful Web services cookbook. 1st ed. Sebastopol, CA.: O'Reilly, 2010, xv, 293 p. ISBN 05-968-0168-8.*
- [3] BOCKLAGE-RYANNEL, Juergen a Johan THELIN. *A Book about Qt5* [online]. 2014 [cit. 2014-10-10]. Dostupné z: <http://qmlbook.org/>
- [4] QT-PROJECT. *Qt 5.3 documentation online* [online]. 2015. vyd. [cit. 2015-10-10]. Dostupné z: <http://qt-project.org/doc/qt-5/index.html>
- [5] Sli.do. *Sli.do - Audience Interaction Made Easy* [online]. 2013-2015 [cit. 2016-05-03]. Dostupné z: <https://www.sli.do/home>
- [6] NetMarketShare. *Market share for mobile, browsers, operating systems and search engines* [online]. 2016 [cit. 2016-02-12]. Dostupné z: <https://www.netmarketshare.com/>
- [7] Unity. *Unity - Game Engine* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <https://unity3d.com/>
- [8] Adobe PhoneGap. *PhoneGap* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <http://phonegap.com/>
- [9] Sencha. *Cross-platform Mobile Web App Development Framework for HTML5 and JS* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <https://www.sencha.com/products/touch/#overview>
- [10] Magin.me. *Building Mobile Applications with HTML/Js/Css and Phonegap* [online]. 2014 [cit. 2016-02-03]. Dostupné z: <http://magin.me/presentation-phonegap-course/>
- [11] Appcelerator. *Mobile App Development Platform & MBaaS* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <http://www.appcelerator.com/>
- [12] Snappii. *Custom apps built for less on ultra-fast mobile platform* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <https://www.snappii.com/>
- [13] ConferenceCompanion. *Mobile app for Paperless Conferences and Meetings* [online]. 2014 [cit. 2016-02-03]. Dostupné z: <http://www.conferencecompanion.com/>
- [14] Meeting Application. *Conference & Event App* [online]. 2012-2014 [cit. 2016-02-03]. Dostupné z: <https://meetingapplication.com/>
- [15] QuickMobile. *Event Apps for Conferences & Meetings* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <https://www.quickmobile.com/>

- [16] Noodle Live. *Event mobile apps for conferences, exhibitions & live events* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <http://www.noodlelive.com/event-mobile-apps/>
- [17] Joe Darrington. *A selection of sites by Joe Darrington* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <http://www.joedarrington.com/img/noodle.png>
- [18] EventPilot. *Conference Apps for Medical and Scientific Meetings* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <http://ativsoftware.com/>
- [19] Event Manager Blog. *EventPilot: Event App for Super Sized Events* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <http://www.eventmanagerblog.com/eventpilot-review>
- [20] EventMobi. *Mobile Apps for Events & Conferences Made Simple* [online]. 2016 [cit. 2016-02-03]. Dostupné z: <http://www.eventmobi.com/>