



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH A TVORBA SOFTWAREOVÉ APLIKACE

DESIGN AND IMPLEMENTATION OF A SOFTWARE APPLICATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Eva Chlpíková

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Dydowicz, Ph.D.

BRNO 2023

Zadání bakalářské práce

Ústav: Ústav informatiky
Studentka: **Eva Chlpíková**
Vedoucí práce: **Ing. Petr Dydowicz, Ph.D.**
Akademický rok: 2022/23
Studijní program: Manažerská informatika

Garant studijního programu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh a tvorba softwarové aplikace

Charakteristika problematiky úkolu:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrh řešení, přínos práce
Závěr
Seznam použité literatury

Cíle, kterých má být dosaženo:

Cílem této bakalářské práce je navrhnout a následně vytvořit softwarovou aplikaci v prostředí Xcode a jazyce Swift, který se používá na platformách Apple zařízení. Pro splnění zmíněného cíle nejdříve provedu analýzu firmy a trhu, na základě které sestavím návrh. Po návrhové fázi, aplikaci vytvořím a implementuji ve firmě.

Základní literární prameny:

BASL, J. a R. BLAŽÍČEK. Podnikové informační systémy. Podnik v informační společnosti. Praha: Grada, 2008. 283 s. ISBN 978-80-247-2279-5.

MOLNÁR, Z. Automatizované informační systémy. Praha: Strojní fakulta ČVUT, 2000. 126 s. ISBN 80-01-02269-2.

MOLNÁR, Z. Efektivnost informačních systémů. Praha: Grada Publishing, 2000. 142 s. ISBN 80-7169-410-X.

PECINOVSKÝ, R. Myslíme objektově v jazyku Java: kompletní učebnice pro začátečníky. Praha: Grada, 2009. 570 s. ISBN 978-80-247-2653-3.

SODOMKA, P. a H. KLČOVÁ. Informační systémy v podnikové praxi. Brno: Computer Press, 2010. 501 s. ISBN 978-80-251-2878-7.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně dne 5.2.2023

L. S.

Ing. Jiří Kříž, Ph.D.
garant

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Bakalářská práce se zaměřuje na návrh a tvorbu analytické aplikace pro konkrétní společnost. Aplikace je navržena pro operační systém tvOS běžící na zařízeních Apple TV. Kromě návrhu samotné aplikace se práce zabývá také analýzou současného řešení společnosti a možným alternativám. Součástí návrhové části je představení uživatelského rozhraní, tvorba vlastní databáze a návrh funkcí aplikace.

Klíčová slova

tvOS, vývojový diagram, ER diagram, datové modelování, funkční modelování, Swift, Xcode, EPC diagram

Abstract

The bachelor thesis focuses on the design and development of an analytical application for a specific company. The application is designed for the tvOS operating system running on Apple TV devices. In addition to the design of the application itself, the thesis also deals with the analysis of the company's current solution and possible alternatives. The design part includes introducing the user interface, creating a custom database, and designing application features.

Keywords

tvOS, flow chart, ERD, data modelling, process modelling, Swift, Xcode, EPC diagram

Bibliografická citace

CHLPÍKOVÁ, Eva. *Návrh a tvorba softwarové aplikace* [online]. Brno, 2023 [cit. 2023-05-15]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/152093>.
Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Ing. Petr Dydowicz, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracovala jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušila autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 15. 5. 2023

Eva Chlpíková

autor

Poděkování

Ráda bych velmi poděkovala vedoucímu bakalářské práce panu Ing. Petrovi Dydowiczovi, Ph.D. za jeho rady a trpělivost při psaní práce. Dále bych chtěla poděkovat firmě Eventee za poskytnuté informace, rady a tipy. A na závěr také poděkovat všem, kdo mě při psaní práce podporovali a povzbuzovali.

OBSAH

ÚVOD	12
VYMEZENÍ PROBLÉMU A CÍLE PRÁCE.....	13
1 TEORETICKÁ VÝCHODISKA PRÁCE.....	14
1.1 Společnost Apple	14
1.1.1 Historie.....	14
1.1.2 Výrobky	15
1.2 Programovací jazyk Swift.....	15
1.2.1 Vlastnosti	16
1.2.2 Proměnné a datové typy	16
1.3 Apple TV	17
1.3.1 tvOS	17
1.4 Integrované vývojové prostředí	18
1.4.1 Xcode	18
1.4.2 Architektura MVVM	19
1.4.3 Storyboard.....	20
1.5 UI design.....	21
1.6 Wondershare Mockitt	21
1.6.1 Wireframe	21
1.7 API.....	22
1.7.1 REST API.....	22
1.8 Analýzy	23
1.8.1 SWOT	23
1.9 Datové modelování.....	23

1.9.1	ER diagram	24
1.10	Funkční modelování	24
1.10.1	Proces.....	24
1.10.2	Diagram případů užití	24
1.10.3	Vývojový diagram	25
1.10.4	EPC diagram	26
1.11	Strategie zavádění aplikace.....	26
1.11.1	Souběžná strategie	26
1.11.2	Pilotní strategie	26
1.11.3	Postupná strategie	26
1.11.4	Nárazová strategie.....	27
2	ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE.....	28
2.1	Základní údaje o společnosti.....	28
2.2	Představení společnosti	28
2.3	Aplikace Eventee	29
2.3.1	Webová administrace.....	29
2.3.2	Aplikace pro účastníky	29
2.4	Požadavky společnosti na nové řešení	30
2.5	Analýza konkurenčních aplikací.....	31
2.5.1	Numerics.....	31
2.5.2	Databox.....	32
2.5.3	Klipfolio.....	33
2.6	Analýza současného stavu	34
2.7	SWOT analýza současného řešení	35

2.7.1	Silné stránky	35
2.7.2	Slabé stránky.....	36
2.7.3	Příležitosti	36
2.7.4	Hrozby	36
2.8	Shrnutí.....	36
3	VLASTNÍ NÁVRH ŘEŠENÍ, PŘÍNOS PRÁCE.....	38
3.1	Návrh aplikace	38
3.1.1	Uživatelské rozhraní a funkcionalita	38
3.2	Datový model aplikace	47
3.2.1	Entita View	48
3.2.2	Entita Event.....	48
3.2.3	Entita User	48
3.2.4	Entita Dashboard.....	49
3.2.5	Entita Metrics.....	49
3.2.6	Entita Company	50
3.3	Funkční model aplikace	51
3.3.1	Diagram případů užití	51
3.3.2	Přihlášení uživatele.....	53
3.3.3	Obnova dat po deseti minutách.....	56
3.3.4	Ověřování tokenu při získávání dat ze serveru	59
3.4	Zavedení systému	61
3.5	Přínos práce.....	61
3.6	Ekonomické zhodnocení.....	61
	ZÁVĚR.....	63

SEZNAM POUŽITÝCH ZDROJŮ	65
SEZNAM POUŽITÝCH OBRÁZKŮ.....	69
SEZNAM POUŽITÝCH TABULEK.....	70

ÚVOD

Tato bakalářská práce se bude věnovat návrhu, sestavení a implementaci aplikace pro společnost Eventee s.r.o. Hlavní myšlenkou je vytvořit aplikaci s přehledným uživatelským rozhraním, která by společnosti usnadnila získávání analytik z jejich databáze, umožnila jejich zobrazení v jednoduchých grafech, a tak zefektivnila pracovní procesy.

První kapitola této bakalářské práce bude věnovaná teoretickým východiskům, tedy pojmům, které budu využívat při analýze, návrhu nebo samotné tvorbě aplikace. Zabývat se tak budu pojmy jako SWOT analýza, kterou využiji pro analýzu současného řešení ve společnosti. Dále také vysvětlím datové a funkční modelování, kde popíšu zejména diagramy – vývojové, datové a entitně-relační, prezentované v praktické části návrhu aplikace. A zaměřím se i na samotné vývojové prostředí Xcode, platformu tvOS a programovací jazyk Swift – vlastnosti, proměnné a datové typy.

V druhé části představím společnost Eventee s.r.o., pro kterou danou aplikaci navrhuji a vytvářím. Nejdříve se budu věnovat obecnému popisu společnosti, kde uvedu datum vzniku, právní formu, základní kapitál, sídlo a logo. Poté popíšu historii společnosti a služby, které nabízí. Z hlediska zmíněných služeb, se budu bavit o stejnojmenné aplikaci Eventee, která je platformou pro organizaci a pořádání konferencí. Kromě již zmíněného, dále zanalyzuji současný stav, požadavky na aplikaci a možné alternativní aplikace.

V poslední, třetí, kapitole se budu zabírat již konkrétním návrhem a sestavením. Aplikaci tak nejdříve představím z hlediska návrhu samotných obrazovek a později databáze a jejích entit. Poté využiji diagramy, představené v teoretických východiscích, pro znázornění vybraných činností aplikace. Na závěr celé této kapitoly vývoj aplikace zhodnotím a zmíním přínos pro společnost.

VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

Cílem této bakalářské práce je navrhnout softwarovou aplikaci pro firmu Eventee s.r.o., která se zabývá vývojem stejnojmenného softwaru „Eventee“. Tato aplikace řeší problém organizace a správy konferencí, ale také zvýšení interakce účastníků těchto událostí.

Vzhledem k tomu, že Eventee s.r.o. je „startupová“ společnost, tak mnoho jejích procesů není pevně zavedeno a řeší se tzv. „až za chodu“. V současné době je jedním z těchto chybějících procesů neexistující rozhraní pro sledování interních analytik o využívání aplikace Eventee. Potřebná data se sice dají získat ze serveru, ale bez jakéhokoli uživatelského rozhraní, je lze získat pouze skrze back-end vývojáře. Zde nám tak nastávají dva největší problémy s tímto řešením. Prvním je, že proces je velmi neefektivní a marné časové prostředky vývojáře, který nad tím musí strávit čas, i manažera, který musí čekat. Druhým je, že výsledkem tohoto exportu je tabulkový soubor, spustitelný např. v aplikaci Microsoft Excel, který musí manažer dále zpracovat, aby z něj získal potřebné grafy nebo jiný přehled informací.

Z těchto důvodů společnost Eventee vyžaduje návrh aplikace, která analytická data získá ze serveru a sama je převede do požadovaného formátu – zejména grafů a seznamů. Aplikace by tak měla usnadnit získání a zpracování dat, a také umožnit sledování těchto dat v reálném čase. V konečném důsledku by se díky aplikaci měla zvýšit celková efektivita práce.

K dosažení cíle budou v práci nejprve vymezena teoretická východiska, ze kterých budu později vycházet při analýze problému a současné situace, ale také při vlastním návrhu řešení. Ve zmíněné analýze bude nejdříve představena společnost, její základní údaje, historie a nabízená služba Eventee. Dále budou rozebrány požadavky společnosti na novou aplikaci a bude provedena analýza konkurenčních aplikací. V té budou představeny aplikace, které částečně splňují požadavky společnosti. Následovně bude provedena analýza současného stavu, kde popíšu, jak společnost momentálně získává jednotlivá data. Na závěr celé kapitoly o analýze problému bude provedena SWOT analýza současného řešení a po ní bude následovat shrnutí všech provedených analýz. Na základě těchto informací bude navrhována, vytvořena a implementována aplikace za využití jazyka Swift a prostředí Xcode pro platformu Apple TV.

1 TEORETICKÁ VÝCHODISKA PRÁCE

V této části budou vysvětleny a popsány pojmy potřebné k pochopení této bakalářské práce. Vzhledem k tomu, že práce pojednává o vývoji aplikace pro tvOS (operační systém vyvinutý pro zařízení Apple TV), tak se budu nejdříve věnovat společnosti Apple, jejímu vývojovému prostředí, programovacímu jazyku a architekturám. Následovně se podívám také na prostředky použité pro návrh a samotnou tvorbu jako např. datové a funkční modelování, prototypy a diagramy. Na závěr celé kapitoly ještě krátce představím různé metody implementace softwaru.

1.1 Společnost Apple

1.1.1 Historie

Společnost Apple Computer byla založena dne 1. dubna 1976 Stevem Jobsem, Stephanem Wozniakem a Ronaldem Waynem. Kapitál společnosti byl pouhých 1 300 dolarů, které Jobs utržil za prodej svého auta a Wozniak za programovatelnou kalkulačku. Jobs a Wozniak v té době také předčasně opustili vysokou školu. Ronald Wayne, který byl třetím zakladatel, se měl starat především o administrativu. Nicméně, svého 10% podílu se Ronald vzdal pouhých 12 dnů po založení. Z velké části hlavně protože nevěřil Jobsově finančnímu plánu a celý projekt se mu zdál šílený, bez šance na úspěch. [1]

První počítač společnosti Apple Computer byl Apple I. Ten Jobs s Wozniakem vytvořili v Jobsově garáži, a prodával se bez monitoru, klávesnice i skříně. Celkově se prodalo 200 kusů s cenou 666,66 dolarů za jeden. V únoru roku 1977 se v obchodech s elektronikou objevil nástupce pojmenovaný Apple II. Ten disponoval barevným monitorem a za krátko se z něj stal prodejní hit. [1]

Od roku 1977 až do druhé poloviny 80. let proběhlo ve společnosti několik změn. V roce 1980 vstoupila na burzu, 1983 Wozniak opustil společnost a byl nahrazen Johnem Scullym, a později roku 1985, dokonce i Jobs opustil společnost. Až do roku 1990 společnost dále prosperovala, ale potom přišel zlom a v roce 1996 se věřilo, že společnost zanikne. [2]

Roku 1996 společnost Apple Computers, nyní již bez Wozniaka a Jobse, koupila firmu NeXT, kterou Jobs založil po odchodu z Apple. Z Jobse se tak opět stal ředitel Apple a

operační systém NeXTStep, který vyvinul v NeXTu, se stal základem pro pozdější Mac OS X. [1]

V srpnu 2011 Jobs rezignoval ze své pozice výkonného ředitele a jeho místo zaujal Tim Cook. O dva měsíce později, 5. října 2011 Steve Jobs zemřel. [1]

Společnost se již znovu nedostala blízko k bankrotu. Ve fiskálním roce 2020 dosáhla obrátu 274,5 miliard USD a dostala se tak mezi nejhodnotnější společnosti na světě. [1]



Obrázek 1: Logo společnosti Apple
(Zdroj: [3])

1.1.2 Výrobky

V současné době Apple nabízí celou řadu výrobků:

- Mobilní telefon iPhone – poprvé představený v roce 2007, v současnosti je nejnovější modelem iPhone 14.
- Tablet iPad – poprvé představen v roce 2010.
- Televize Apple TV – představená společně s iPhonem v roce 2007.
- Hodinky Apple Watch – přestaveny v roce 2015.
- Sluchátka AirPods – nejnovější přídavek, představeny v roce 2016.
- Notebooky a počítače Mac

Kromě fyzických zařízení, Apple nabízí, provozuje nebo vyvíjí také spoustu softwarových aplikací jako je Apple TV+, App Store, vývojové prostředí Xcode a další. [1]

1.2 Programovací jazyk Swift

Swift je programovací jazyk vytvořený společností Apple pro platformy iOS, iPadOS, macOS, tvOS a watchOS. Jedná se o objektový, kompilovaný jazyk s typovou kontrolou.

Jeho první verze byla vydána v roce 2014 a nahradila tak, dnes již zastaralý, Objective-C. [4]

Jak již bylo zmíněno, tak Swift je nástupcem jazyků C a Objective-C. Obsahuje tak nízko úrovně primitivy, jako jsou typy, řízení toku a operátory. Ale zároveň poskytuje i objektově orientované funkce, jako jsou třídy, protokoly a generika. [5]

Swift je otevřeně (tzv. „open source“) vyvíjen na webu Swift.org, kde jsou k dispozici zdrojové kódy, nástroje pro sledování chyb, fóra a pravidelné vývojové builds. Na vývoji Swiftu pracuje široká komunita vývojářů, jak uvnitř společnosti Apple, tak z celého světa. [5]

Přestože je Swift určen primárně pro vývoj na platformách Apple zařízení, tak se jedná o multiplatformní jazyk, který lze volně využít i mimo Apple. [5]

1.2.1 Vlastnosti

Mezi nejdůležitější vlastnosti Swiftu řadíme:

- je objektově orientovaný,
- je case-sensitive – rozeznává velká a malá písmena,
- je open source – každý se může podílet na jeho tvorbě,
- řádky kódu není potřeba oddělovat středníkem,
- je multiplatformní – lze ho použít i na vývoj pro jiná zařízení než Apple,
- dokáže koexistovat s jazykem Objective-C (v jednom souboru je tak možné mít zároveň Swift i Objective-C). [5]

1.2.2 Proměnné a datové typy

Proměnná nám poskytuje pojmenované úložiště, se kterým může náš program manipulovat. Každá proměnná ve Swiftu má specifický typ, který určuje velikost a uspořádání paměti proměnné, rozsah hodnot, které lze v této paměti uložit, a sadu operací, které lze na proměnnou aplikovat. [6]

Každá proměnná ve Swiftu musí mít zpravidla jeden datový typ. Mezi základní datové typy podporované ve Swiftu 4 a vyšších patří následující: [6]

Tabulka 1: Základní datové typy ve Swiftu
(Zdroj: Vlastní zpracování podle [6])

Název	Vysvětlení
Int nebo UInt	Používá se pro celá čísla.
Float a Double	Používá se pro desetinná čísla.
Bool	Reprezentuje logickou hodnotu, která je buď true, nebo false.
String	Jedná se o uspořádanou kolekci znaků.
Character	Jedná se o jednoznakový řetězcový literál.

1.3 Apple TV

Apple TV můžeme jednoduše popsat jako multimediální chytrý box, který dodá chytré funkce téměř každé televizi nebo monitoru. Jedinou podmínkou, kterou musí připojené zařízení splnit, je mít HDMI port pro připojení Apple TV. [7]

Apple TV je v mnoha ohledech podobná iPhoneům či iPadům. Stejně jako mobilní zařízení totiž disponuje procesory řady A, které Apple používá v iPhonech a iPadech. Stejně je to i s operačním systémem Apple TV pojmenovaným tvOS, který také vychází z mobilního iOS. Tuto podobnost lze vidět v samotném designu tvOS, ale také třeba v jeho členění nastavení. Na rozdíl od mobilního iOS je ale tvOS přizpůsoben pro ovládání pomocí ovladače Apple Siri Remote. [7]

1.3.1 tvOS

TvOS je operační systém specificky vyvinutý pro zařízení Apple TV od společnosti Apple Computers. Z velké části vychází z operačního systému iOS, nicméně je upravený pro optimalizaci aplikací na televizním rozhraní. [11] To umožňuje uživatelům instalaci aplikací třetích stran jako je YouTube, HBO GO nebo Netflix, ale také na míru optimalizovaný zážitek u aplikací od samotné společnosti Apple. [8]

První tvOS byl představen v září roku 2015 a byl dostupný na zařízeních Apple TV čtvrté generace o měsíc později. Předchozí generace (třetí a nižší) využívali upravenou verzi iOS, která se jmenovala „Apple TV Software“. S představením oficiálního tvOS byla

přidána také podpora vývoje televizních aplikací, a zároveň byl zpřístupněn softwarový obchod přímo pro Apple TV. [7]

Od prvotního přestavení v roce 2015, každý rok vychází nová verze tvOS. V současnosti se tak jedná o verzi tvOS 16. [7]

1.4 Integrované vývojové prostředí

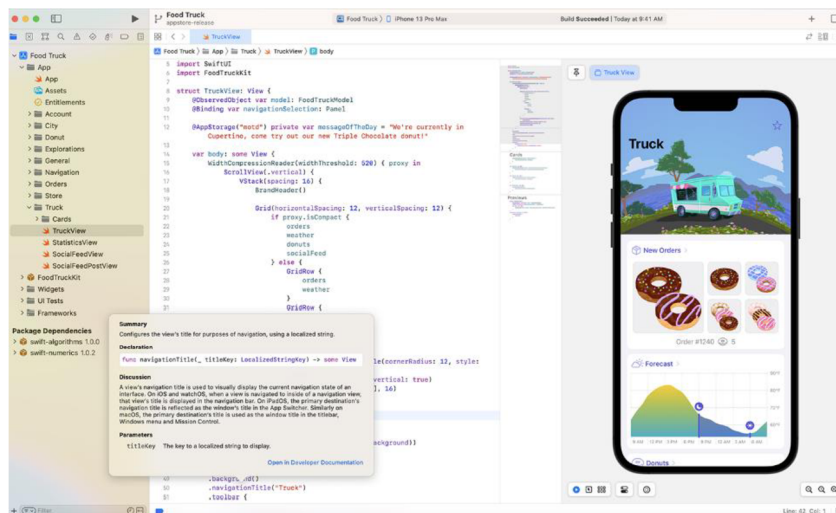
Integrované vývojové prostředí (IDE) je softwarová sada, která sdružuje základní nástroje potřebné k psaní a testování softwaru. Vývojové nástroje často zahrnují textové editory, knihovny kódu, překladače a testovací platformy. Bez IDE by vývojář musel všechny tyto nástroje vybírat, nasazovat, integrovat a spravovat sám. Nástroje obsažené ve vývojovém prostředí jsou navrženy tak, aby zjednodušily vývoj softwaru a zároveň dokázaly identifikovat chyby a překlepy. [9]

1.4.1 Xcode

Xcode je oficiální vývojové prostředí pro tvorbu a testování softwaru na všechny platformy společnosti Apple: iOS, iPadOS, tvOS, watchOS a macOS. [10]

Xcode obsahuje všechny nástroje, které jsou potřeba pro vytvoření aplikace – od designu před tvorbu až po zveřejnění. Kromě hlavního programovacího jazyka Swift, podporuje také jeho předchůdce Objective-C, a další populární jazyky jako C, C++, Java nebo Python. [10]

Xcode je možné získat a používat pouze na platformě macOS. Nepodporuje tedy konkurenční operační systémy Windows, Linux ani jiné unixové platformy. [10]



Obrázek 2: Xcode prostředí
(Zdroj: [10])

1.4.1.1 Simulátor

Součástí nástrojů vývojového prostředí Xcode je také simulátor všech zařízení vyráběných společností Apple. Simulátor napodobuje aplikace, které běží v zařízení iOS, tvOS, macOS a dalších. Umožňuje tak spustit a provozovat tyto aplikace bez nutnosti použití fyzického zařízení. Simulátor můžeme najít v prostředí Xcode, kde si přímo zvolíme, jaké zařízení chceme napodobit. [11]

Simulátory se využívají zejména ke dvěma hlavním případům – testování aplikací bez fyzického zařízení nebo přístup k těmto aplikacím. [11]

1.4.2 Architektura MVVM

Architektura MVVM (Model-View-ViewModel) je v návrhovém vzoru používaným při vývoji mobilních aplikací. Oproti dříve používanému modelu MVC (Model-View-Controller) řeší problém oddělení logiky uživatelského rozhraní a obchodní logiky. [12]

Logika uživatelského rozhraní (UI) je část programu, která zpracovává, co se děje na uživatelském rozhraní. Tedy to, co uživatel vidí a s čím může interagovat (tlačítka, tabulky, popisky). [12]

Obchodní logikou máme na mysli proces, který zpracovává komunikaci mezi databází a rozhraním koncového uživatele. To znamená, že zaručuje zpracování a aktualizaci dat, ale také jejich mazání nebo formátování datových typů. [12]

1.4.2.1 Komponenty MVVM

- **Model** – Chová se stejně jako v modelu MVC, obsahuje objekty pro přístup k datům. Je využíván View Modelem a aktualizován ve chvíli, když View Model pošle nová data.
- **View** – Je zodpovědný za zobrazení uživatelského rozhraní. Mezi View a Modelem není žádná přímá interakce a nemusí se již starat o obchodní logiku. Vyžaduje data od View Modelu ve formátu připravenému k zobrazení.
- **View Model** – Zodpovědný za všechnu obchodní logiku a zcela nezávislý na View. Jakmile obdrží informaci od View, tak ji zpracuje a pošle zpátky. [12]

Konečným cílem MVVM je izolovat odpovědnost za obchodní logiku z View do View Modelu. Ve View Modelu navíc provádíme přípravu dat pro vizuální prezentaci a také nasloucháme aktualizacím modelu. [12]

Kromě rozdělení odpovědností mezi komponenty přináší MVVM také další výhody:

- Lepší testování – Je snadnější vytvořit testy, když komponenty nesdílí data a funkcionality.
- Transparentní komunikace – View Model nabízí View transparentní rozhraní pro interakci s jednotlivými vrstvami. Výsledkem je pak transparentní komunikace mezi jednotlivými vrstvami aplikace. [12]

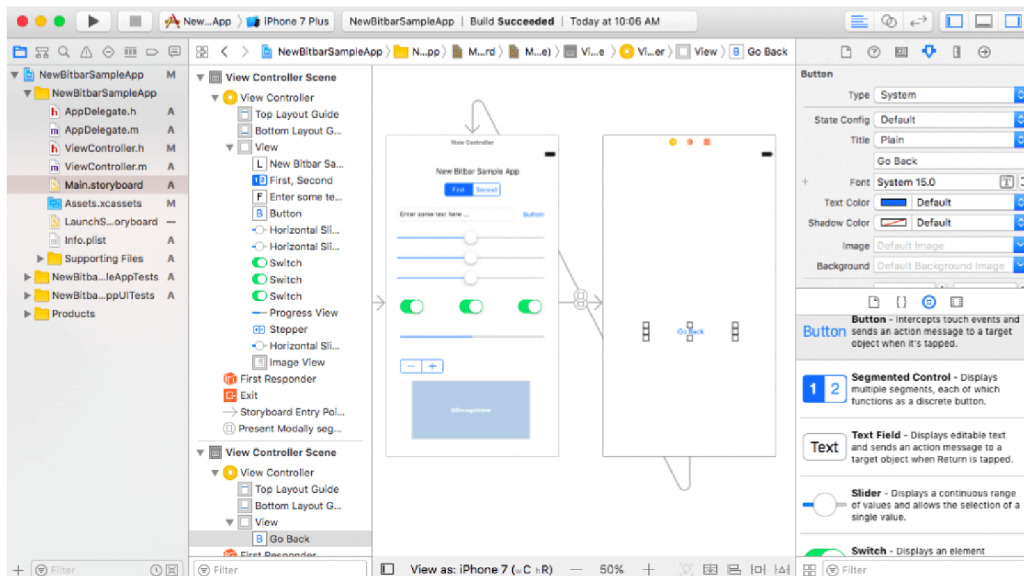


Obrázek 3: MVVM aplikace
(Zdroj: [13])

1.4.3 Storyboard

Storyboard je jedním z nástrojů v prostředí Xcode, který umožňuje navrhovat, vytvářet a upravovat uživatelské rozhraní vyvíjené aplikace. [14]

Všechny projekty vytvořené v Xcode jsou dodávány s alespoň jedním storyboardem připraveným k použití, ten se jmenuje: Main.storyboard. V něm je možné vytvořit libovolný počet rozhraní, z nichž každé představuje jeden view controller v aplikaci. V Main.storyboard lze také navrhovat přechody mezi view controllers a to vše bez jediného řádku kódu. [14]



Obrázek 4: Ukázka vytváření Storyboardu
(Zdroj: [15])

1.5 UI design

UI design, také design uživatelského rozhraní, se zabývá vizuální podobou navrhovaného systému. Při jeho tvorbě je klíčové, jak vypadá produkt, prostředí či konkrétní element (tlačítko, textové pole, buňka), se kterými zákazník přichází do kontaktu, nikoliv to, jak fungují. [16]

1.6 Wondershare Mockitt

Wondershare Mockitt je univerzální platforma pro návrh UX/UI designu, která umožňuje navrhovat a vytvářet prototypy aplikace. Kromě prototypů (wireframes) je v této platformě možné také vytvářet finální designové návrhy. [17]

1.6.1 Wireframe

Wireframe slouží jako návod pro vytváření webu, systému nebo aplikace. Jedná se o “polografický” návrh, ve kterém jsou vyobrazeny jednotlivé elementy. Ty se většinou

volí jako šedé čtverečky. Je nutné upozornit, že wireframe není finální návrh aplikace. [18]

Mezi hlavní funkce wireframu patří ukázat základní rozložení informací nebo prvků na obrazovce, vytvořit a přiblížit strukturu celé aplikace a vytvořit podklad pro další práci grafika. [18]

1.7 API

API (Application Programming Interface; aplikační programové rozhraní) slouží pro komunikaci dvou nebo více systémů. V kontextu aplikací se používá pro předávání dat mezi dvěma softwarovými systémy. V jednom ze systémů je API implementována a nabízí potřebné funkce, knihovny, protokoly nebo i třídy. Všechny tyto prvky API jsou zdokumentovány a popsány v technické dokumentaci. Na základě této dokumentace se poté v druhé aplikaci programově implementuje volání těchto funkcí, využívání knihoven, protokolů a tříd. [19]

1.7.1 REST API

REST (Representational State Transfer) je datově orientovaná architektura rozhraní navržená pro distribuované prostředí. Jejím zdrojem mohou být jak data, tak stavy aplikace, pokud je lze popsat konkrétními daty. Všechny zdroje mají vlastní identifikátor URI a REST definuje čtyři základní metody pro přístup k nim. [20]

Zmíněné metody jsou známé pod označením CRUD:

- Create – stará se o vytvoření dat, volanou metodou pro vytvoření je POST.
- Retrieve – stará se o získání dat, volanou metodou pro získání dat je GET.
- Update – stará se o změnu dat, konkrétní volanou metodou pro změnu dat je PUT.
- Delete – stará se o smazání dat, konkrétní volaná metoda je stejnojmenná. [20]

1.8 Analýzy

1.8.1 SWOT

SWOT analýza je univerzální analytická technika používaná pro zhodnocení vnitřních a vnějších faktorů ovlivňujících úspěšnost organizace nebo nějakého konkrétního záměru (například nového produktu či služby). Nejčastěji je SWOT analýza používána jako situační analýza v rámci strategického řízení a marketingu. Autorem SWOT analýzy je Albert Humphrey, který ji navrhl v šedesátých letech 20. století. [21]

Její podstatou je identifikovat klíčové silné a slabé stránky uvnitř, tedy v čem je organizace (nebo její část) dobrá a v čem špatná. Stejně tak je důležité znát klíčové příležitosti a hrozby, které se nacházejí v okolí organizace. Cílem SWOT analýzy je identifikovat a následně omezit slabé stránky, podporovat silné stránky, hledat nové příležitosti a znát hrozby. Organizace by měla využívat příležitostí, které se nabízejí a umět předcházet hrozbám. [21]



Obrázek 5: SWOT analýza
(Zdroj: [21])

1.9 Datové modelování

Datové modelování se zabývá problematikou dat, která potřebujeme pro informační systém. [22, str. 76]

Cílem datového modelování je zaznamenat vlastní data, bez zbytečných duplicit a nepřesností, a dále také zachytit jejich vzájemné vztahy, vazby a jiné souvislosti. [22, str. 3]

1.9.1 ER diagram

Diagram vztahů entit (ER diagram), známý také jako model vztahů entit, je grafické znázornění, které zobrazuje vztahy mezi osobami, objekty, místy, koncepty nebo událostmi v rámci systému. ER diagram používá techniky datového modelování, které mohou pomoci definovat obchodní procesy a sloužit jako základ relační databáze. [23]

1.10 Funkční modelování

Funkční modelování se zabývá zkoumáním a algoritmizací činností a procesů, které v informačním systému (nebo aplikaci) probíhají. [22, str. 76]

1.10.1 Proces

Při popisu činností ve funkčním modelování provádíme hierarchický rozklad funkcí od nejobecnějších až k elementárním. Počet úrovní takovéto dekompozice není omezen a liší se podle konkrétního řešeného případu. Nejnižší úrovní tohoto členění je tzv. elementární funkce. Vzhledem k tomu, že k popisu činnosti si nevystačíme s elementárními funkcemi, tak je spojíme dohromady a tak vzniká tzv. proces. [22, str. 77]

1.10.2 Diagram případů užití

Diagram případů užití (Use Case Diagram) zobrazuje chování systému z pohledu uživatele. Účelem diagramu je popsat funkcionalitu systému (v tomto případě aplikace), tedy co od něj klient nebo návrhář očekává. Diagram vypovídá pouze o tom, co má systém umět – jeho funkcionality, ale ne jak se to má udělat – programová implementace. Většinou se tak jedná o první diagram, který při návrhu systému vytváříme. [24]

Diagram případů užití se skládá z případů užití (use case), aktérů (actors) a vztahů mezi aktéry. [24]

1.10.2.1 Případ užití

Případ užití je sada několika akcí, které vedou k dosažení určitého cíle. Takovým případem může být např. registrace uživatele, přihlášení nebo přidání komentáře. Dá se tedy říct, že definuje jednu funkcionalitu, kterou by navrhovaný systém měl umět. [24]

Případy užití vychází buď z představ zákazníka o systému nebo od nás, pokud vyvíjíme vlastní systém. [24]

Každý jeden případ užití se značí jako elipsa s názvem funkce uvnitř. [24]

1.10.2.2 Aktér

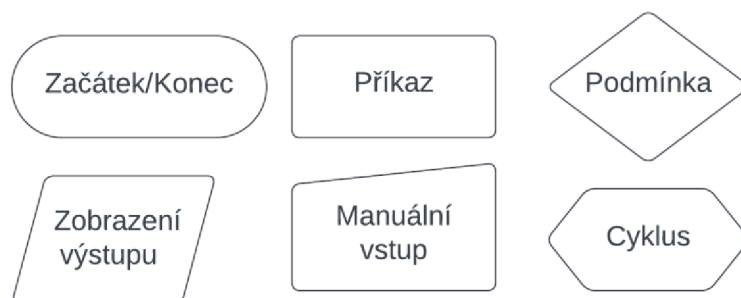
Aktér je role, která komunikuje nebo vykonává jednotlivé případy užití. Tuto roli může zaujímat uživatel, ale i další externí systém. Aktérem tedy může být uživatel, administrátor, SMS server nebo dokonce čas. [24]

Aktéra znázorňujeme jako postavu z čar s názvem napsaným pod ní. [24]

1.10.3 Vývojový diagram

Vývojový diagram je druh diagramu, který slouží ke grafickému znázornění jednotlivých kroků algoritmu nebo popisu procesu. Diagram používá pro znázornění jednotlivých kroků symboly, které jsou navzájem propojeny pomocí orientovaných šipek. Hlavní výhodou použití vývojového diagramu je možnost velmi dobře zachytit větvení při splnění či nesplnění požadovaných podmínek. [25]

Vývojové diagramy se skládají z grafických značek. Značky se kombinují, tím se simulují různé situace a různé příkazy. Do těla těchto značek se vepisují upřesňující údaje jako podmínky, vstupy, výstupy apod. [25]



Obrázek 6: Vybrané značky vývojového diagramu
(Zdroj: Vlastní zpracování podle [25])

1.10.4 EPC diagram

EPC diagram (diagram procesu řízeného událostmi) je grafický modelovací jazyk založený na vývojovém diagramu, který lze použít k popisu podnikových procesů a pracovních postupů. [26]

Na rozdíl od jiných diagramů, je EPC diagram vždy spuštěn a ukončen nějakou událostí. Tato událost se graficky značí šestiúhelníkem. Za každou událostí pak může následovat jedna nebo více funkcí (každá značená obdélníkem) a za každou funkcí zase jedna nebo více událostí. Propojení těchto funkcí a událostí je zprostředkováno skrze výrazy OR, AND a XOR. [26]

První EPC diagram sahá do roku 1992, kdy ho vytvořil prof. Scheer a jeho kolegové ze Sárské univerzity. [26]

1.11 Strategie zavádění aplikace

V případě, že potřebujeme stávající informační systém, aplikaci nebo jiný software zcela nebo i částečně nahradit, tak je potřeba k tomu zvolit vhodnou strategii. Celkově se nabízí čtyři druhy strategie – souběžná, pilotní, postupná a nárazová. [27, str. 123]

1.11.1 Souběžná strategie

Při souběžné strategii jsou oba systémy provozovány současně po určitou dobu. Během této doby dochází k ověření funkčnosti nového systému, zaškolení pracovníků a získání jistoty, že nový systém je připraven k použití a starý systém může být opravdu opuštěn. [27, str. 123]

1.11.2 Pilotní strategie

V případě použití pilotní strategie nejdříve zavedeme nový systém v jedné pobočce (nebo oddělení) firmy a zbylá část firmy nadále používá starý systém. Po odzkoušení systému na něj přechází zbytek firmy. [27, str. 124]

1.11.3 Postupná strategie

Podstatou postupného zavádění je odebírání částí starého systému a jejich nahrazování částmi nového systému. Hlavní výhodou tohoto druhu zavádění je bezpečnost.

Postupné zavádění se používá zejména při inovaci rozsáhlejších systémů. [27, s.tr 124]

1.11.4 Nárazová strategie

Nárazová strategie je vysoce rychlá a účinná, nicméně také velmi riskantní. Podstatou této strategie je, že se starý systém ukončí tzv. „ze dne na den“ a zcela se nahradí novým. [27, str. 124]

2 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE

V této části se nejdříve budu věnovat popisu společnosti a služeb, které nabízí svým zákazníkům. Dále se budu soustředit na požadavky společnosti na novou aplikaci a analýzu již existujících současných řešení. Následovně provedu rozbor současného řešení v rámci společnosti a na závěr udělám upravenou SWOT analýzu tohoto řešení.

2.1 Základní údaje o společnosti

Tabulka 2: Základní údaje o společnosti

(Zdroj: Vlastní zpracování podle [37])

Název společnosti:	Eventee, s.r.o.
Právní forma:	Společnost s ručením omezeným
Sídlo:	Kopečná 940/14, Staré Brno, 602 00 Brno
Identifikační číslo:	29307236
Základní kapitál:	200 000 Kč
Datum vzniku:	31. leden 2012



Obrázek 7: Logo společnosti

(Zdroj: [28])

2.2 Představení společnosti

Společnost Eventee, s.r.o. je start-upová společnost, která působí na trhu již od roku 2012. Ve svých začátcích se společnost věnovala zakázkové výrobě aplikací pro platformy Android, iOS a webové prohlížeče. [28]

V roce 2015 společnost Eventee, v té době ještě pod názvem Touch Art s.r.o., vyvinula stejnojmennou aplikaci Eventee jako reklamu k propagaci jejich agentury. Díky velmi pozitivnímu ohlasu od organizátorů a účastníků se společnost rozhodla si aplikaci ponechat jako vedlejší projekt. [28]

V roce 2018 byla společnost vybrána jako účastník akceleračního programu VentureOut pro start-upové společnosti, aplikaci Eventee tak měla možnost prezentovat v New Yorku. Tento okamžik by se také dal popsat jako první „opravdový“ začátek Eventee aplikace. [28]

Od roku 2019 se společnost plně věnuje vývoji aplikace Eventee a ukončuje práci na všech zakázkových projektech pro jiné firmy. [28]

V současnosti společnost sídlí v Brně na adrese Kopečná 14, zaměstnává do 15 zaměstnanců a nemá formální organizační strukturu. Vzhledem k tomu, že firma je relativně novější a její tým je malý, tak mnoho procesů se řeší až tzv. „za chodu“.

2.3 Aplikace Eventee

Jak již bylo uvedeno při představení, tak společnost v současné době nabízí pouze jedinou službu, a to softwarovou aplikaci Eventee. Ta slouží k organizaci a správě konferencí. Nabízí webovou administraci pro organizátora, ale také nativní aplikace pro iOS, Android a webový prohlížeč pro účastníky. [29]

2.3.1 Webová administrace

Z hlediska webové administrace lze:

- Vytvořit přednášky, workshopy a přestávky
- Vytvořit profily přednášejících a sponzorů
- Propojit s externími aplikacemi umožňujícími prodej vstupenek (vlastní vstupenkový portál Eventee zatím nenabízí)
- Informovat účastníky o novinkách skrze funkci zpráv [29]

2.3.2 Aplikace pro účastníky

Z hlediska aplikace pro účastníky lze:

- Vytvořit si tzv. „Vlastní program“ – tedy zaregistrovat se na workshopy, uložit si přednášky
- Komunikovat s dalšími účastníky skrze chat
- Sledovat přímý přenos přednášek
- Hodnotit přednášky
- Pokládat dotazy a hlasovat v anketách
- Přidávat příspěvky na sociální zeď konference [29]

Eventee je tedy určena pro pořadatele konferencí, eventové agentury, ale i jednotlivé uživatele. Soustředí se zejména na činnosti, které probíhají při konferenci – vysílání přednášek, možnost pokládat otázky, pořádat ankety apod. [29]

2.4 Požadavky společnosti na nové řešení

Společnost požaduje aplikaci, která by sloužila jako jejich interní dashboard pro přehled nejrůznějších metrik a informací v reálném čase. Aplikace by měla zobrazovat jednotlivé údaje zejména v grafech, ale také číselně a v tabulkách.

V prvotní verzi bude aplikace zobrazovat interní data využívání Eventee, později by se měly dodělat i metriky zaměřené více na business (odliv zákazníků, příchozí platby apod.). Z hlediska interních analytik Eventee se jedná o celkový počet konferencí, účastníků, zákazníků, aktivních uživatelů, procentuální zákaznickou konverzi a míru loajality. Tyto údaje by mělo být možné prohlížet jako přehled za posledních 12 měsíců, ale také detailněji za posledních 7 nebo 30 dní. U přehledu není potřeba zobrazovat grafy, stačí pouze přehledný jednoduchý výpis. Obrazovka obsahující detailnější zobrazení by měla prezentovat většinu informací ve vhodných grafech. Nicméně, u analytik jako je konverze nebo míra loajality, postačuje společnosti jedno číslo za celé období.

Další funkcionalitou by měla být možnost zobrazovat základní informace přímo o nadcházejících a již proběhlých konferencích jako je jméno, datum, počet relací (přednášek a workshopů) a počet účastníků. Podobné informace by mělo být možné získat i o posledních zákaznících a uživateli – tedy jméno, kontaktní email, využívaný plán a datum registrace.

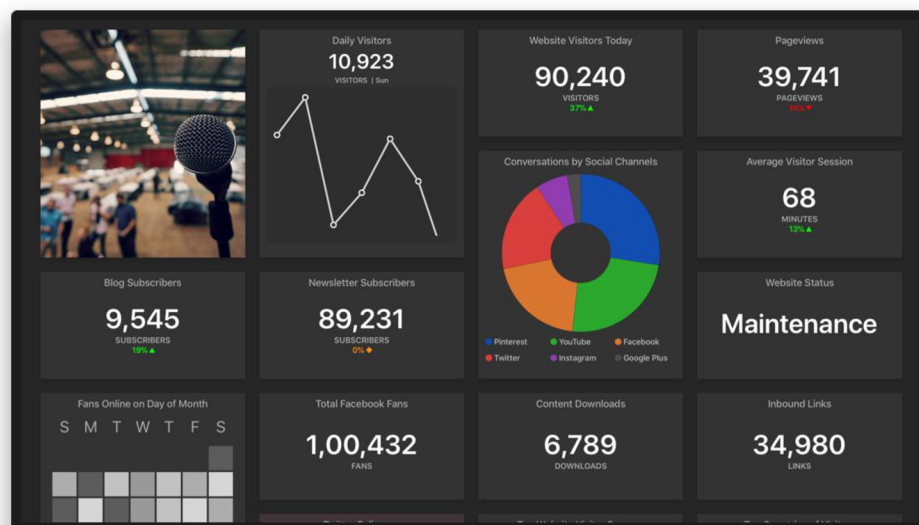
Celou aplikaci společnost požaduje vytvořit pro platformu Apple TV, kde by v reálném čase bylo možné sledovat jednotlivé analytiky a informace.

2.5 Analýza konkurenčních aplikací

Na základě požadavků společnosti uvedených v kapitole 2.4 byly vybrány 3 konkurenční aplikace, které by společnost mohla zvážit jako alternativu navrhované aplikace. Tyto aplikace v následujících podkapitolách detailněji představím a vysvětlím jejich klady a zápory.

2.5.1 Numerics

Numerics je aplikace pro vytváření dashboardů, která umožňuje navrhovat vlastní rozhraní pro ukazatele výkonnosti a jiné metriky. Aplikace Numerics je od základu navržena na míru pro iPad, aby poskytovala to uživatelsky nejpřívětivější zobrazení dashboardů. V jednotlivých dashboardech tak lze sledovat informace o analýze webu, zapojení na sociálních sítích, postupu projektů, zůstatcích na účtu nebo jakékoliv další klíčové firemní ukazatele. [30]



Obrázek 8: Ukázka aplikace Numerics
(Zdroj: [31])

2.5.1.1 Výhody Numerics

Mezi výhody aplikace Numerics můžeme zařadit:

- Podpora všech zařízení od Applu – Mac OS, Apple TV, iPhone a další.
- Nabízí widgety – tedy miniaplikace, které lze připnout např. na plochu počítače.

- Nabízí uživatelský manuál na svých stránkách.
- Zkouška zdarma – aplikaci lze stáhnout, ale většina funkcí je zamčena. [30]

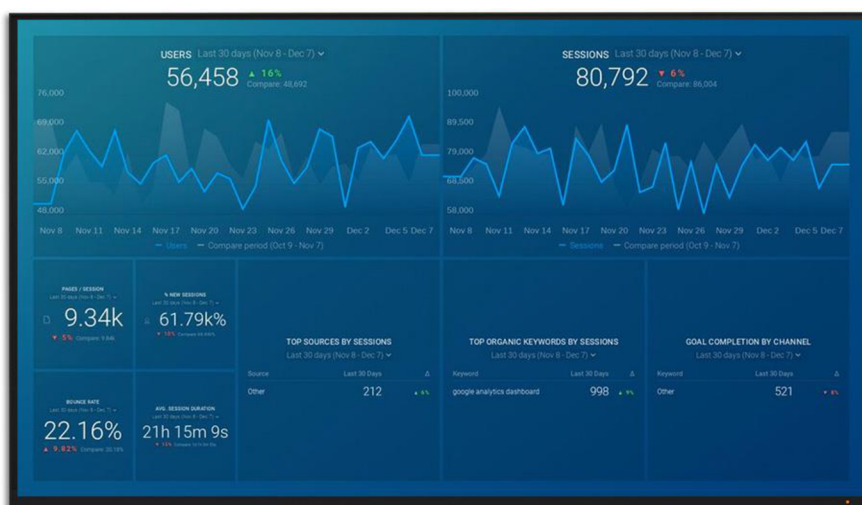
2.5.1.2 Nevýhody Numerics

- Žádná hodnocení v Apple obchodě s aplikacemi a minimum na internetu.
- Cena od 100 dolarů ročně za jednoho uživatele.
- Nelze napojit přímo na serverová data – takže nelze ukázat data z Eventee databáze. [30]

2.5.2 Databox

Databox je obchodní analytická platforma, která shromažďuje všechna vaše data na jednom místě, takže můžete sledovat výkonnost a zjišťovat informace v reálném čase. [32]

Databox nabízí desítky integrací se zdroji jako jsou Google Analytics, Salesforce, HubSpot, Mixpanel, Facebook, Shopify a další. [32]



Obrázek 9: Ukázka aplikace Databox
(Zdroj: [32])

2.5.2.1 Výhody Databoxu

Mezi výhody Databoxu řadíme:

- Je k dispozici přehledná produktová dokumentace.
- Jednoduché uživatelské rozhraní.

- Data lze získat také z Google tabulek a dalších firemních aplikací. [33]

2.5.2.2 Nevýhody Databoxu

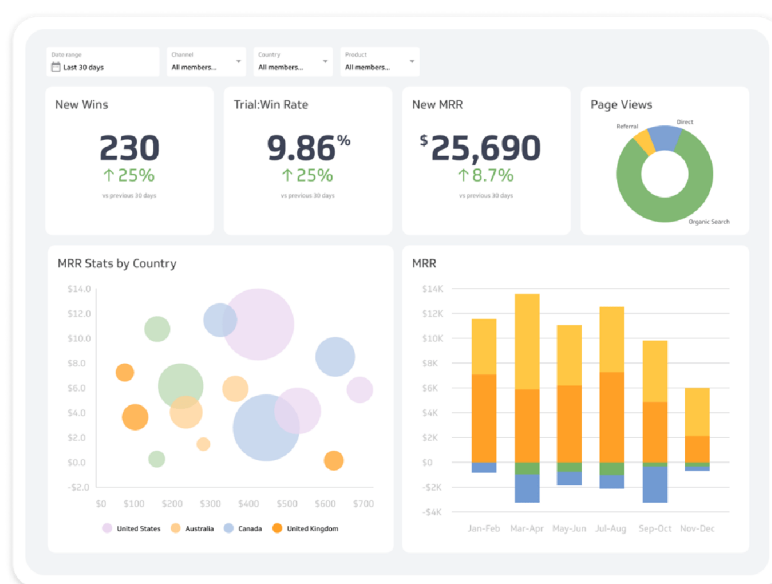
Nevýhody aplikace Databox jsou následovné:

- Aktualizace dat po dnech (společnost vyžaduje hodiny a minuty; liší se podle obrazovky).
- Velmi omezená customizace obrazovek a grafů.
- Uchovávání dat starších dvou let vyžaduje nejdražší plán.
- Ve verzi zdarma jsou k dispozici maximálně 4 různá připojení k datům.
- Nepodporuje Apple TV.
- Pro splnění požadavků by společnost musela pořídit balíček za minimálně 236 dolarů měsíčně. [33]

2.5.3 Klipfolio

Klipfolio je cloudová webová aplikace, ve které je možné vizualizovat a sledovat klíčové ukazatele výkonnosti a metriky, které jsou pro společnosti důležité. [34]

Pomocí aplikace Klipfolio lze shromažďovat, sdílet a zobrazovat data v reálném čase. Ale také je sledovat za určité období, což umožňuje historická srovnání pro sledování pokroku a růstu společnosti. [34]



Obrázek 10: Ukázka aplikace Klipfolio
(Zdroj: [35])

2.5.3.1 Výhody Klipfolia

- Přívětivé uživatelské prostředí.
- Data může importovat i člověk bez programovacích schopností.
- Možnost importu vlastních dat z REST API. [36]

2.5.3.2 Nevýhody Klipfolia

- V základní verzi zobrazení dat maximálně dva měsíce starých.
- Nepodporuje nativně Apple TV.
- Pro splnění požadavků společnosti cena od 229 dolarů za měsíc. [36]

2.6 Analýza současného stavu

Jak jsem již zmínila během představení společnosti, tak se jedná o menší start-upovou společnost, jejíž procesy nejsou pevně zavedené. Na základě rozhovoru s jednatelem společnosti jsem popsala požadavky společnosti na novou aplikaci v kapitole 2.4. V této kapitole bych se ráda věnovala tomu, jak firma získávala jednotlivá data do teď.

V současné době se k údajům z aplikace lze dostat pouze ručním exportem údajů z databáze. To tedy znamená, že každý takový export má na starosti programátor s databázovým přístupem. Ten musí přerušit svoji práci, načíst databázi a manuálně odtud extrahovat potřebná data. V praxi taková činnost zbytečně zabírá čas programátora, který exportuje data, i osoby, která musí na export čekat. Je potřeba také zmínit, že výsledkem exportu údajů je tabulkový soubor, který musíme ještě dále upravovat, aby z něj bylo možné přehledně vyčíst všechny požadované údaje. Pokud se pak jedná o data za delší časový interval může být nutné sloučit více souborů dohromady, a pak je upravovat.

Navrhovaná aplikace by ve výsledku měla odstranit riziko korupce nebo změny dat při přímém zásahu do databáze, a zároveň zvýšit efektivitu práce – nebylo by potřeba žádné slučování souborů, převod dat do přehlednější formy nebo čekání na vyexportování.

2.7 SWOT analýza současného řešení

Tato část se bude zabývat modifikovanou SWOT analýzou, která bude zaměřena konkrétně na současné řešení společnosti, kterým je manuální export dat ze serveru.

SWOT analýzu nejdříve znázorním tabulkou s přehledem hlavním silných, slabých stránek, hrozeb a příležitostí, a poté se budu věnovat každé oblasti ve vlastní podkapitole.

Tabulka 3: SWOT analýza současného řešení
(Zdroj: Vlastní zpracování)

Silné stránky	Slabé stránky
Vysoká variabilita řešení Nízké provozní náklady (databázi musí mít společnost v každém případě) Bezpečnost dat	Ztráta časových prostředků Menší pracovní produktivita Menší efektivita práce
Příležitosti	Hrozby
Možnost vytvořit novou aplikaci s jednodušším rozhraním	Možnost korupce dat Možnost přepsání dat

2.7.1 Silné stránky

Mezi nejdůležitější silné stránky současného řešení patří vysoká variabilita exportu údajů. Jednatel nebo jiný zaměstnanec si může přímo nadiktovat, jaký přesně by potřeboval výstup dat. Jednou se tak může jednat o seznam konferencí za minulý rok, jindy jejich počet nebo druh. Nemusí se tak obávat, že by výstup nebyl podle jeho představ nebo, že by z něj musel ještě dále něco počítat.

Za další silnou stránku můžeme brát i fakt, že společnost nemusí využívat externí řešení, které by opět nemuselo ukazovat údaje v požadovaném formátu. Ale také by externí řešení s velkou pravděpodobností muselo být placené.

Poslední, z hlediska silných stránek, bych také zmínila bezpečnost dat. Vzhledem k tomu, že k datům přistupuje přímo oprávněný zaměstnanec, tak není potřeba bát se úniku nebo

zneužití dat. U jiné, externí aplikace zvláště v dnešní době, by mohla být data více ohrožena.

2.7.2 Slabé stránky

Hlavní slabou stránkou současného řešení jsou zejména promarněné časové prostředky, jak zaměstnance, tak jednatele nebo jiného zaměstnance vyžadující příslušná data.

Další podstatnou slabou stránkou je fakt, že ve firmě je v současné době pouze jeden zaměstnanec schopný provádět export dat z databáze. Pokud má tento zaměstnanec dovolenou, tak se na něj musí čekat, aby bylo možné získat data. Tato slabá stránka sice není zcela způsobena současným řešením, ale může být eliminována řešením prezentovaným v této práci.

2.7.3 Příležitosti

Hlavní příležitostí je zcela určitě implementace nové aplikace. Ta by poskytla chybějící funkcionalitu a zároveň nabídla přívětivější uživatelské rozhraní. Když vezmeme v potaz i fakt, že data v aplikaci by byla živá a okamžitě zobrazovaná v grafech a přehledných tabulkách, tak by se samozřejmě zvýšila i efektivita a produktivita osob, které s daty budou pracovat.

2.7.4 Hrozby

Hrozbou současného řešení je především práce se samotnou databází. Přestože s ní pracuje zkušený zaměstnanec, tak nelze zcela vyloučit, že při exportu dat se něco nepokazí a data se nepoškodí. Vzhledem k tomu, že se zde bavíme o datech přímo souvisejících se zákazníky společnosti, tak jejich poškození by mohlo mít i fatální následky.

2.8 Shrnutí

Na základě provedených analýz a zjištěných nedostatků existujících řešení, se společnost rozhodla k vytvoření vlastní aplikace, která bude splňovat jejich specifické požadavky. Vlastní vývoj aplikace je pro společnost nejvhodnějším řešením vzhledem k zaměření společnosti, tedy že se jedná o softwarovou společnost, a tak mohou aplikaci později spravovat sami. Dále také vzhledem k jejich start-upovému charakteru, kdy si nemohou

dovolit dlouhodobě financovat externí řešení. Společnost očekává vyšší náklady na realizaci. Nicméně, z dlouhodobého hlediska budou postupně tyto náklady nižší než využívání konkurenčních aplikací. Také se při řešení použijí nejnovější technologie, což sníží riziko korupce databázových dat v porovnání se současným řešením manuálního zásahu do databáze.

3 VLASTNÍ NÁVRH ŘEŠENÍ, PŘÍNOS PRÁCE

Tato část práce se bude věnovat návrhu vlastního řešení včetně jeho dílčích částí. Nejdříve budu popisovat jednotlivé obrazovky aplikace, a to z hlediska designu, funkčnosti a programové implementace. Nakonec zpracuji ekonomické zhodnocení celého řešení a zhodnotím přínosy práce.

3.1 Návrh aplikace

Při návrhu této aplikace jsem se snažila vytvořit co nejjednodušší uživatelské rozhraní, které by bylo přehledné a zároveň nic nepostrádalo. Celý návrh a jednotlivé prototypy obrazovek jsou v angličtině, protože firma se soustředí a má zákazníky zejména na zahraničních trzích – USA, Spojené království apod.

S ohledem na účel aplikace a seznam platforem, na které je společnost schopná vyvíjet, bylo rozhodnuto, že aplikaci je vhodné udělat na tvOS, tedy operační systém zařízení Apple TV.

3.1.1 Uživatelské rozhraní a funkcionality

V této části budu probírat návrh a požadovanou funkcionality jednotlivých obrazovek aplikace. Návrhy, které budou představeny, byly vytvořeny skrze nástroj Mockitt od společnosti Wondershare. Později byly upraveny na základě zpětné vazby od společnosti a celý tento proces trval přibližně 1 měsíc.

Data, která jsou uvedena v jednotlivých návrzích neodpovídají skutečnosti, ale slouží pouze pro ukázkou umístění a zobrazení jednotlivých prvků v aplikaci.

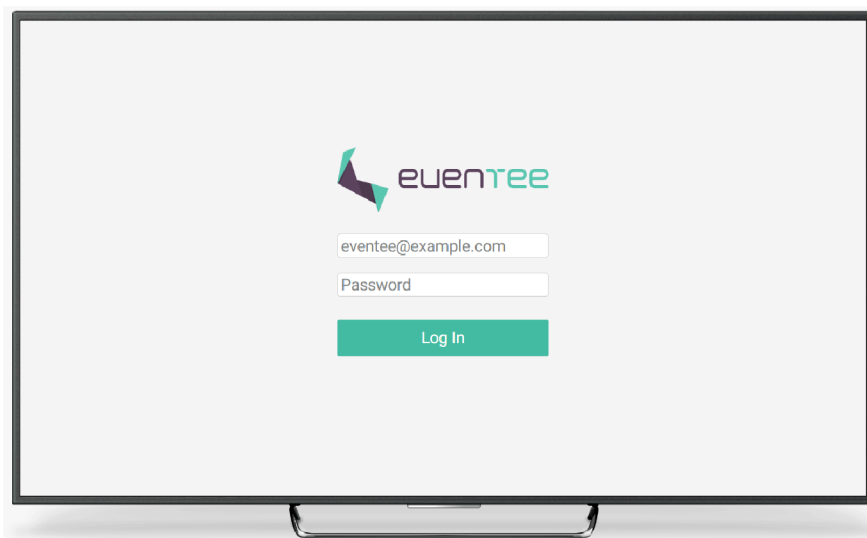
3.1.1.1 Přihlašovací obrazovka

Začneme popisem přihlašovací obrazovky, která se zobrazí okamžitě po prvotní spuštění aplikace.

Na obrazovce se nachází logo Eventee, ve kterém je kromě ikony zapracováno i jméno. Pod logem se nachází dvě textová pole – jedno pro e-mail a druhé pro heslo. Každé pole má nastavený příslušný vstup, a tak se automaticky při zadávání hesla zobrazí hvězdičky pro zachování bezpečnosti. Pod textovými poli se nachází tlačítko pro přihlášení.

V případě, že uživatel nevyplní všechna pole nebo zadá invalidní údaje, tak se také objeví dialogové okno hlásící chybu.

Pokud je uživatel již přihlášen, tak je přihlašovací obrazovka přeskočena.



Obrázek 11: Přihlašovací obrazovka
(Zdroj: Vlastní zpracování)

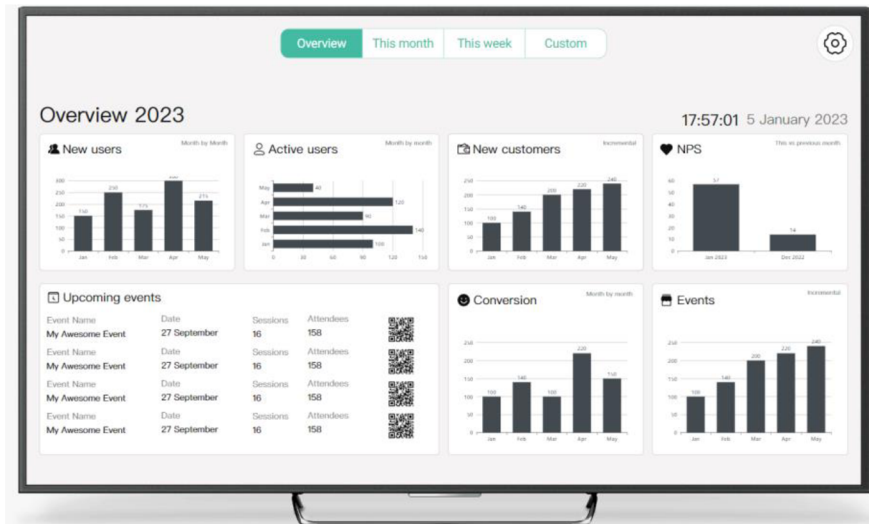
3.1.1.2 Overview obrazovka

Overview neboli přehled, je obrazovka, která se zobrazí hned po přihlášení do aplikace, popř. jako první obrazovka, když je uživatel již přihlášen.

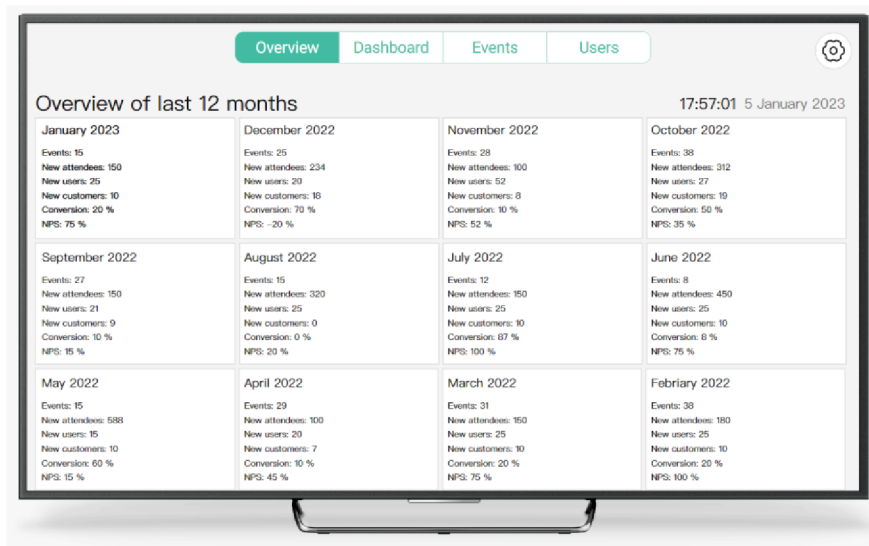
Vize za touto obrazovkou, jak již název vypovídá, byl přehled nejdůležitějších statistik z Eventee. Vzhledem k tomu, že jsem od společnosti neměla příliš specifických požadavků, tak můj prvotní návrh obsahoval různé grafy o jednotlivých metrikách za současný kalendářní rok. A dále také seznam nejbližších budoucích konferencí s jejich jménem, datem, počtem pozvaných účastníků, počtem relací a QR kódem vedoucím do administrace v aplikaci Eventee. Tento návrh se nasetkal s příznivou zpětnou vazbou, a postupně byl přepracován na souhrn posledních dvanácti měsíců. V každém kontejneru, obsahujícím jeden měsíc, jsou data o počtu konferencí konaných ten měsíc, o počtu nových účastníků, nových uživatelů (tedy všech, co se registrovali do Eventee administračního rozhraní) a nových zákazníků. Poslední dva řádky zobrazují procentuální konverzi a NPS. Za konverzi se v tomto případě považuje kolik uživatelů se stalo

platícími zákazníky. Procentuální NPS značí spokojenost zákazníků s aplikací, ptá se totiž na otázku, zda by produkt osobně doporučili na stupnici od 1 do 10.

Na obrazovce se také nachází tlačítko pro přesměrování do nastavení v pravém horním rohu. Pod tímto tlačítkem se nachází text zobrazující živě dnešní čas a datum. Nahoře uprostřed se nachází jednotlivé lišty/záložky pro přepínání obrazovek.



Obrázek 12: Prvotní návrh obrazovky Overview
(Zdroj: Vlastní zpracování)



Obrázek 13: Výsledný návrh obrazovky Overview
(Zdroj: Vlastní zpracování)

3.1.1.3 Dashboard obrazovka

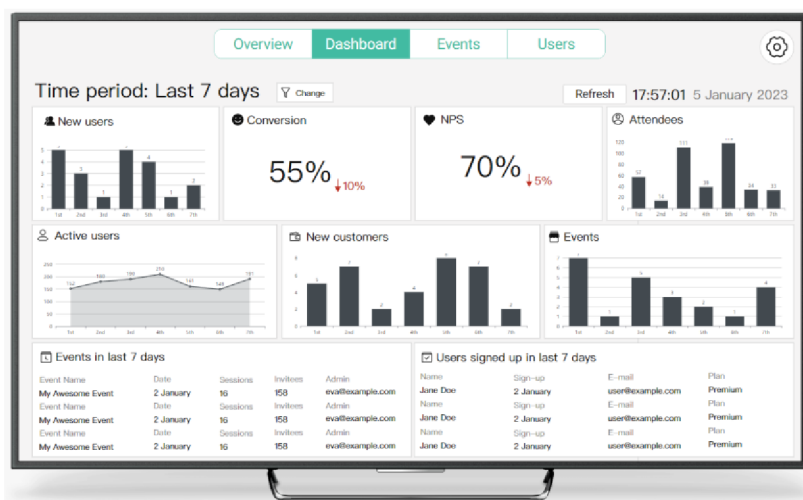
Tato obrazovka ukazuje detailní analytiku ve zvoleném časovém rozmezí. Výchozím intervalem se po dohodě zvolilo posledních 7 dní.

Na obrazovce jsou tři řádky informací. V první řádce se nachází graf pro nové uživatele, konverzi, NPS a nové účastníky. Uživatelský a účastnický graf mají vzhled sloupcového grafu, kdy osa X ukazuje jednotlivé dny a osa Y počet. NPS a konverze jsou zobrazeny jako konečné číslo za daný interval. Vedle hlavního čísla je také napravo menším fontem spočten rozdíl oproti předchozímu intervalu. V tomto případě oproti předchozím sedmi dnům. V druhém řádku jsou tři grafy – aktivní uživatelé, noví zákazníci a konference. Pouze aktivní uživatelé mají graf jiného druhu a to spojnicový graf. Osa X u všech grafů zobrazuje jednotlivé dny, osa Y počty.

V posledním řádku jsou dva kontejnery – uživatelé a konference. Oba tyto kontejnery zobrazují detailnější informace o konferencích, které proběhly v daném časovém intervalu a uživateli, kteří se v něm registrovali.

Při kliknutí na jakýkoliv graf je uživatel přesměrován na novou obrazovku, kde vidí pouze konkrétní graf. Tato funkce se hodí zejména v případech, že časový interval je příliš rozsáhlý, aby informace v grafech byly dobře čitelné v menším rozlišení.

Data na této obrazovce se automaticky obnovují každých 10 minut. Nicméně je zde i tlačítko pro manuální obnovení. Umístěné je vedle času pod názvem „Refresh“.

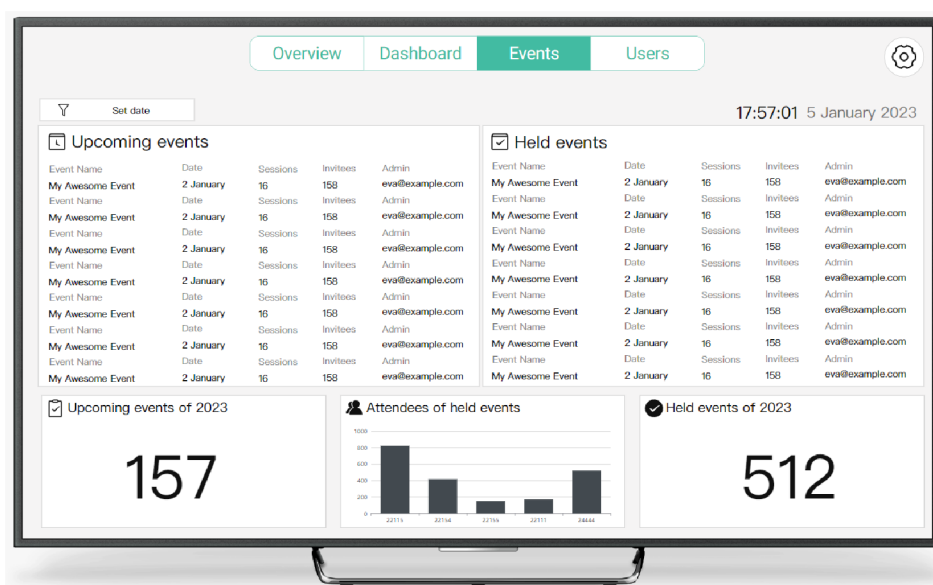


Obrázek 14: Obrazovka Dashboard
(Zdroj: Vlastní zpracování)

3.1.1.4 Events obrazovka

Na obrazovce Events se zobrazují jednotlivé konference v Eventee. Skrze tlačítko „Set date“ v levém horním rohu lze zvolit datum a podle něj se zobrazí konference před a po. Výchozím je opět dnešní den. Při kliknutí na jakoukoliv konferenci ze seznamu se zobrazí její detail, který je popsán v následující podkapitole.

Pod kontejnery s konferencemi jsou tři informace – celkové množství nadcházející a proběhlých konferencí za tento rok a množství účastníků v předešlých konferencích.

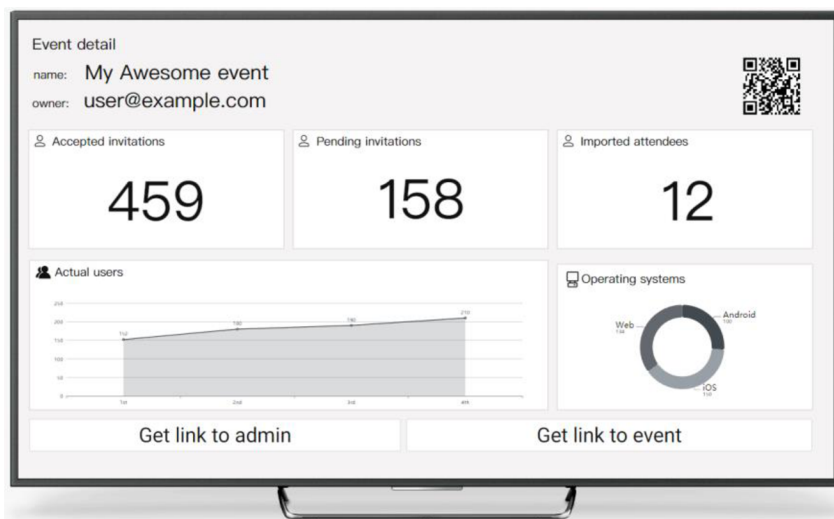


Obrázek 15: Obrazovka Events
(Zdroj: Vlastní zpracování)

Obrazovka detailu konference

Po kliknutí na konkrétní konferenci je uživatel přesměrován do jejího detailu. Nahoře na obrazovce detailu je jméno konference a e-mail na organizátora, napravo od těchto informací je QR kód do konference. Po touto sekci jsou tři řádky informací.

V prvním řádku je vyobrazeno kolik má konference účastníků, kolik ještě nepřijalo pozvánku a kolik pozvánka nebyla poslána (v případě invalidní e-mailové adresy). V druhém je počet uživatelů v konferenci a vedle něj koláčový graf představující jednotlivé operační systémy používané uživateli. Třetí řádek obsahuje tlačítka, která zašlou link do konference nebo administrace v Eventee do připojeného zařízení – telefonu nebo notebooku značky Apple.



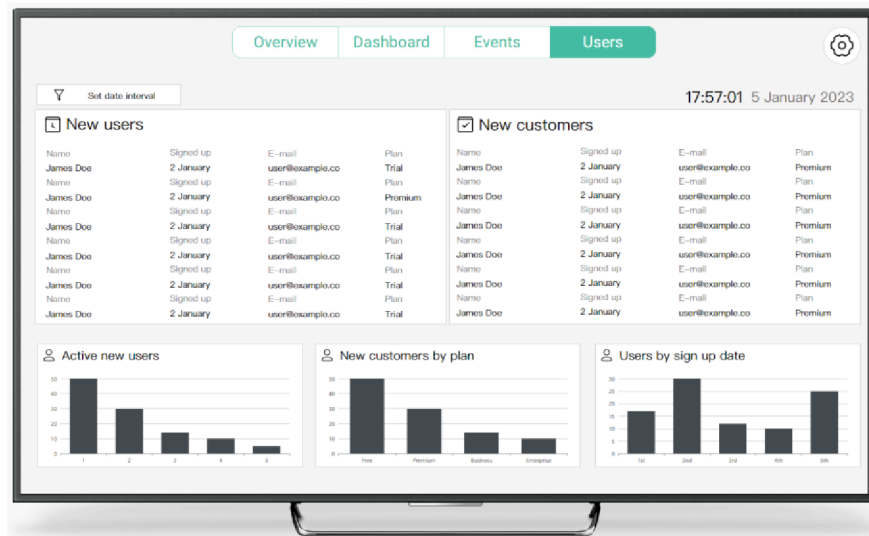
Obrázek 16: Obrazovka detailu konference
(Zdroj: Vlastní zpracování)

3.1.1.5 Users obrazovka

Obrazovka Users zobrazuje uživatele i zákazníky. Výchozím časovým interval je zde doba od dnešního dne do začátku roku.

Vlevo nahoře je tlačítko pro změnu časového intervalu, které otevře obrazovku pro výběr počátečního a koncového data. Pod ním se nachází kontejner pro nové uživatele, tedy všechny, co se registrovali v daném intervalu. Hned vedle se nachází obdobný kontejner se zákazníky. Oba listy jsou řazené sestupně podle data. Každý záznam v obou listech lze rozkliknout a podívat se na bližší informace o dané osobě.

Dole na obrazovce jsou zobrazeny tři grafy – aktivní noví uživatelé, nový zákazníci podle plánu a uživatelé podle data registrace. Grafy jsou opět sloupcové a lze na ně kliknout pro zobrazení na celou obrazovku.



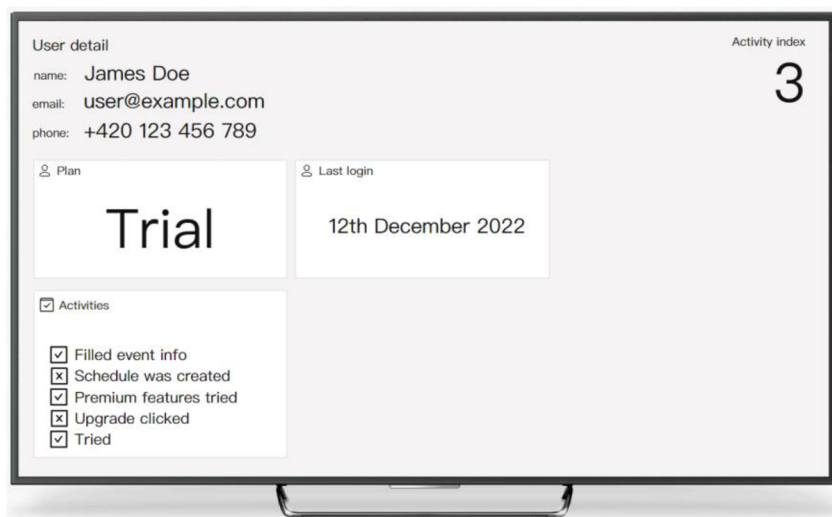
Obrázek 17: Obrazovka Users
(Zdroj: Vlastní zpracování)

Obrazovka detailu uživatele

Jak bylo zmíněno v předchozí části, tak uživatele je možné rozkliknout. Po rozkliknutí se zobrazí tato obrazovka – detail uživatele. Na ní se v levém horním rohu nachází jméno uživatele, email a telefonní číslo. Naproti těmto údajů na pravé straně je tzv. „Activity index“, který zobrazuje kolik aktivit uživatel vykonal z celkových pěti měřených. Těmi jsou:

- zda uživatel vyplnil základní informace o konferenci,
- vytvořil program konference,
- zapnul některé z prémiových funkcí,
- kliknul na tlačítko „Upgrade“ (pro zobrazení ceníku),
- vyzkoušel Eventee – tedy kliknul na tlačítko zobrazení účastnického rozhraní.

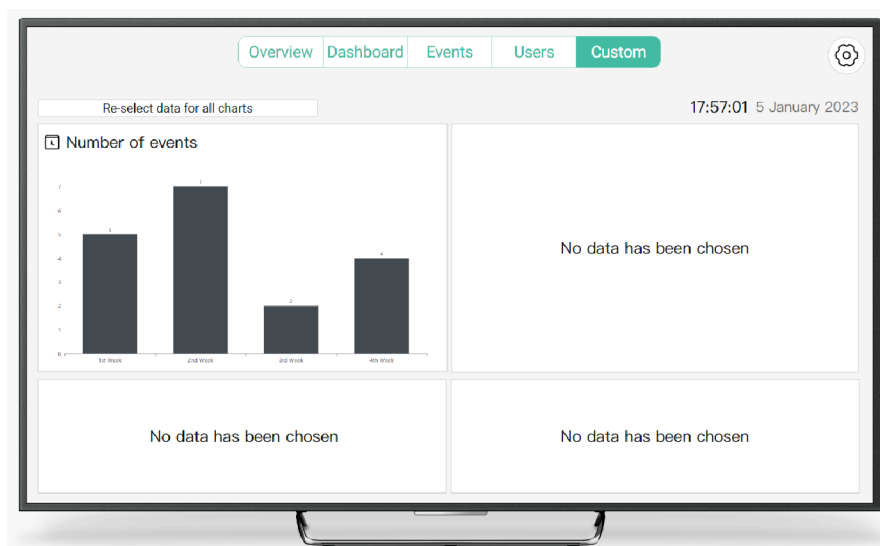
Pod těmito textovými údaji jsou tři kontejnery s plánem uživatele, dnem posledního přihlášení a výpisem provedených aktivit.



Obrázek 18: Detail uživatele
 (Zdroj: Vlastní zpracování)

3.1.1.6 Custom obrazovka

Jedná se o obrazovku, na které si uživatel může sám zvolit grafy a pojmenovat je. Nicméně, tato obrazovka je součástí horního menu pouze v případě, že je zapnutá skrze nastavení. Ve výchozím stavu není zapnutá – toho si jde povšimnout i na předchozích obrázcích aplikace. V levém horním rohu se nachází tlačítko, kterým si uživatel může zvolit data a pojmenování pro všechny grafy najednou. Jinak se dá každé grafové pole vybrat i samostatně. Po zvolení příslušného datového setu jsou data buď získána z lokální databáze nebo serveru. V případě serveru může získání trvat několik sekund.

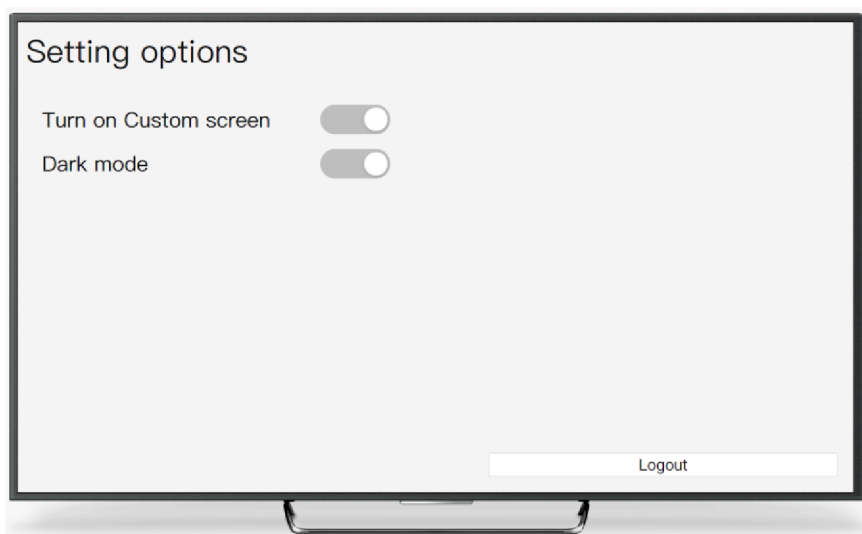


Obrázek 19: Custom obrazovka s jedním grafem
 (Zdroj: Vlastní zpracování)

3.1.1.7 **Obrazovka s nastavením**

Settings nebo také obrazovka s nastavením obsahuje dva přepínače a jedno tlačítko. První přepínač je pro zobrazení Custom obrazovky představené v předchozí kapitole 3.1.1.6. Druhý pro zapnutí a vypnutí tmavého režimu. V pravém dolním rohu se pak nachází tlačítko odhlášení, které přestože nepředpokládám, že by se hojně užívalo, tak raději jsem dodělala.

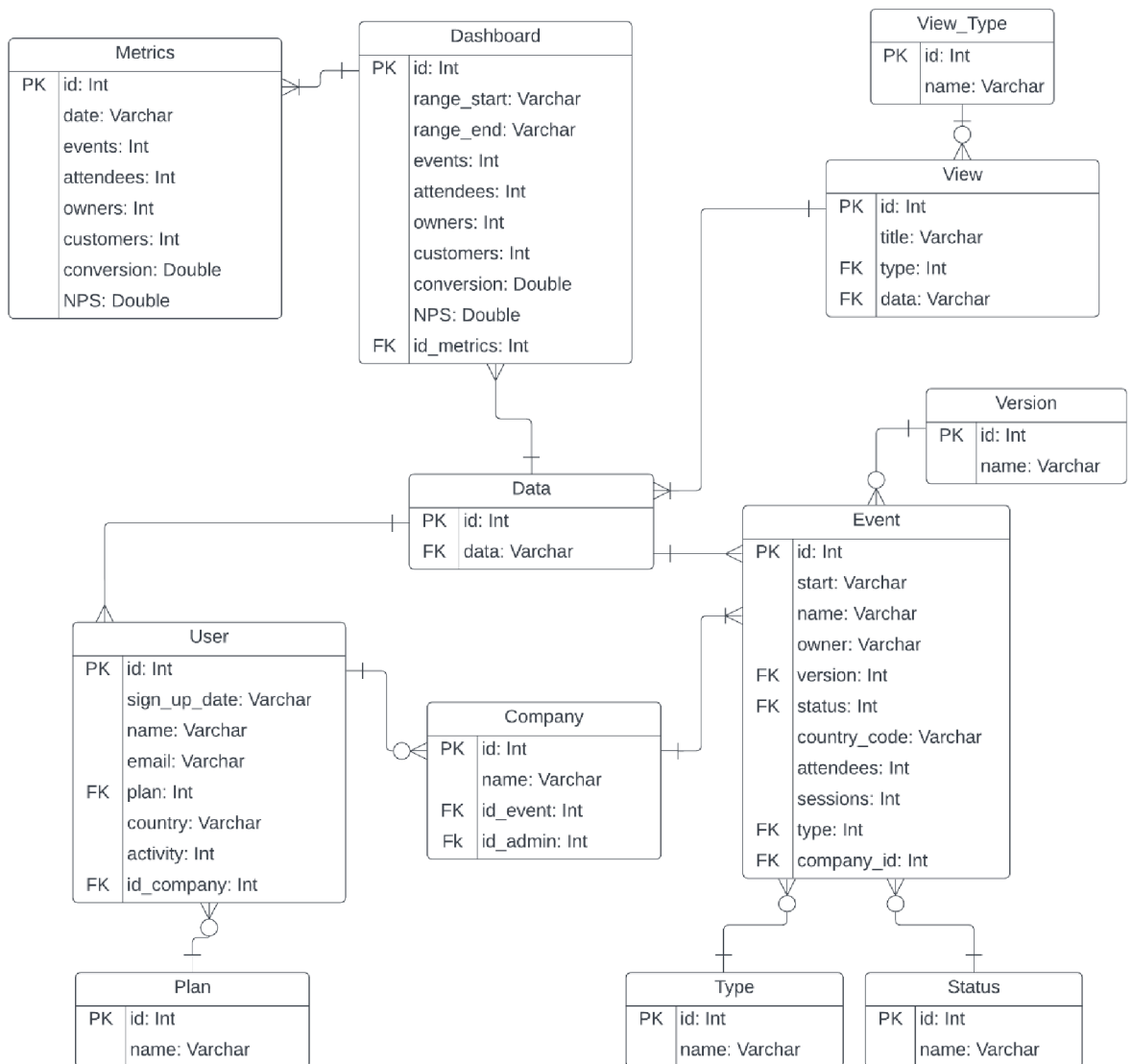
Po kliknutí na odhlášení bude uživatel přesměrován zpět na přihlašovací obrazovku. Bez opětovného přihlášení není možné prohlížet jednotlivé analytiky ani jiné údaje.



Obrázek 20: Obrazovka s nastavením
(Zdroj: Vlastní zpracování)

3.2 Datový model aplikace

Pro ukládání dat v aplikaci bylo potřeba vytvořit lokální databázi pomocí MongoDB. Nicméně, pro samotnou komunikaci se serverem používá aplikace REST API, která zabezpečuje všechny potřebné funkce jako jsou GET, POST, PUT, DELETE a PATCH. Níže představím vytvořenou aplikační databázi, její entity a některé z důležitějších atributů.



Obrázek 21: ER diagram aplikační databáze
(Zdroj: Vlastní zpracování)

3.2.1 Entita View

Entita View uchovává údaje o grafech a tabulkách zobrazovaných přímo v aplikaci. Mezi její atributy patří:

- id – primární klíč, automaticky se zvětšuje,
- title – název tabulky nebo grafu,
- type – cizí klíč do tabulky View_Type (typ zobrazení),
- data – cizí klíč do tabulky Data.

3.2.2 Entita Event

Entita Event neboli konference, slouží k ukládání dat o konferenci. Mezi její atributy patří:

- id – primární klíč, automaticky se zvětšuje,
- start – datum začátku konference, uložené ve formátu YYYY-MM-DD HH:mm:ss Z,
- name – jméno konference,
- owner – email organizátora konference,
- version – cizí klíč do tabulky Version (verze konference),
- status – cizí klíč do tabulky Status (status konference),
- country_code – kód země, např. CZ,
- attendees – počet účastníků,
- sessions – počet relací,
- type – cizí klíč do tabulky Type (typ konference),
- company_id – cizí klíč do tabulky Company.

3.2.3 Entita User

Tato entita uchovává údaje o uživateli a má následující atributy:

- id – primární klíč, automaticky se zvětšuje,
- sign_up_date – datum registrace, uložené ve formátu YYYY-MM-DD,
- name – celé jméno uživatele,
- email – emailová adresa uživatele,

- plan – cizí klíč pro entitu plan,
- country – země uživatele zadaná kódem, např. CZ,
- activity – počet aktivit uživatele,
- id_company – cizí klíč pro entitu company (společnost).

3.2.4 Entita Dashboard

Entita Dashboard ukládá údaje o celkovém počtu účastníků, konferencí, aktivních uživatelů apod. za určité časové období. Její atributy jsou:

- id – primární klíč, automaticky se zvětšuje,
- range_start – začátek časového intervalu ve formátu YYYY-MM-DD,
- range_end – konec časového intervalu ve formátu YYYY-MM-DD,
- events – celkový počet konferencí,
- attendees – celkový počet účastníků,
- owners – celkový počet organizátorů konferencí,
- customers – celkový počet platících zákazníků,
- conversion – procentuální hodnota konverze,
- NPS – procentuální hodnota NPS,
- id_metrics – cizí klíč do tabulky Metrics.

3.2.5 Entita Metrics

Entita Metrics stejně jako Dashboard ukládá údaje o celkovém počtu účastníků, konferencí nebo aktivních uživatelů. Nicméně, jejím časovým intervalem je jeden den.

- id – primární klíč, automaticky se zvětšuje,
- date – datum ve formátu YYYY-MM-DD,
- events – celkový počet konferencí,
- attendees – celkový počet účastníků,
- owners – celkový počet organizátorů konferencí,
- customers – celkový počet platících zákazníků,
- conversion – procentuální hodnota konverze,
- NPS – procentuální hodnota NPS.

3.2.6 Entita Company

Tato entita uchovává údaje o společnostech, ve kterých jsou jednotlivé konference.

Atributy této entity jsou:

- id – primární klíč, automaticky se zvětšuje,
- name – název společnosti,
- id_event – cizí klíč do tabulky Event (konference),
- id_owner – cizí klíč do tabulky User (uživatel).

3.3 Funkční model aplikace

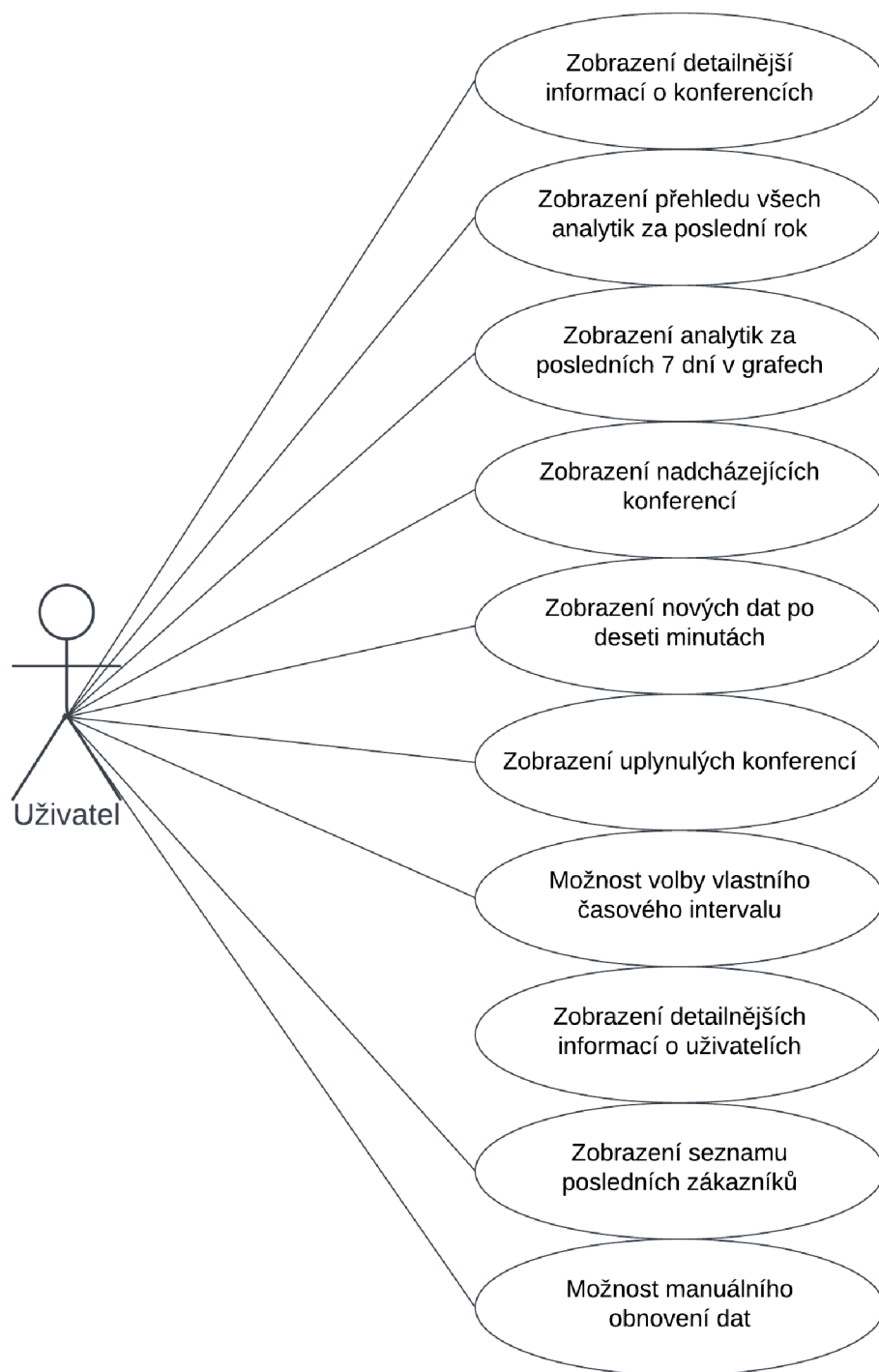
V následující kapitole se budu věnovat praktické implementaci vybraných činností aplikace. Nejdříve představím diagram případů užití, kde budou vypsány nejdůležitější funkce, které aplikace obsahuje. Potom začnu popisovat tři konkrétní činnosti v aplikaci – přihlášení uživatele, obnovení (refresh) dat v aplikaci a postup ověřování tokenu. První dvě činnosti budou popsány pomocí vývojového a EPC diagramu. Poslední prezentovaná činnost bude popsána pouze vývojovým diagramem.

3.3.1 Diagram případů užití

V níže uvedeném diagramu jsem znázornila 10 hlavních případů užití, které může provádět osoba Uživatelé ve vytvořené aplikaci.

Osoba uživatele může:

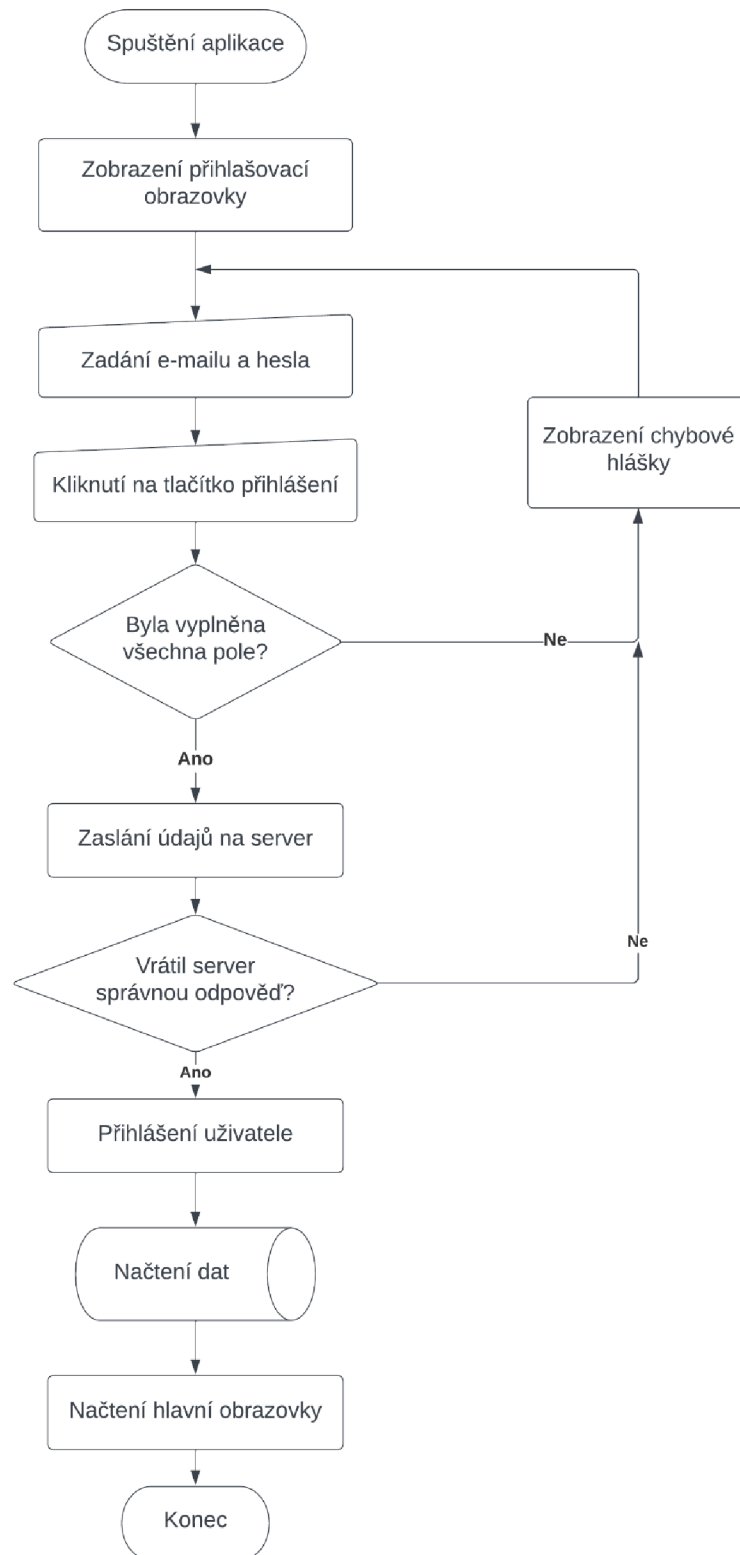
- Zobrazit detailnější informace o konferencích – mezi ty patří třeba název konference, datum, počet účastníků a QR kód ke konferenci.
- Zobrazit přehled všech analytik za poslední rok – zobrazení všech informací za posledních 12 měsíců.
- Zobrazit analytiky za posledních 7 dní – konverze, celkový počet účastníků, noví uživatelé apod.
- Zobrazit nadcházející konference – list nadcházejících konferencí obsahující jméno, datum, organizátora.
- Obnovit data (manuálně nebo automaticky) – obnovit nebo zobrazit obnovená data.
- Zobrazit uplynulé konference – list uplynulých konferencí.
- Vybrat vlastní časový interval – 7 dní, 30 dní nebo poslední rok.
- Zobrazit detailní informace o uživateli – datum registrace, jméno, email, telefonní číslo.
- Zobrazit seznam posledních zákazníků – licence, datum registrace, jméno apod.



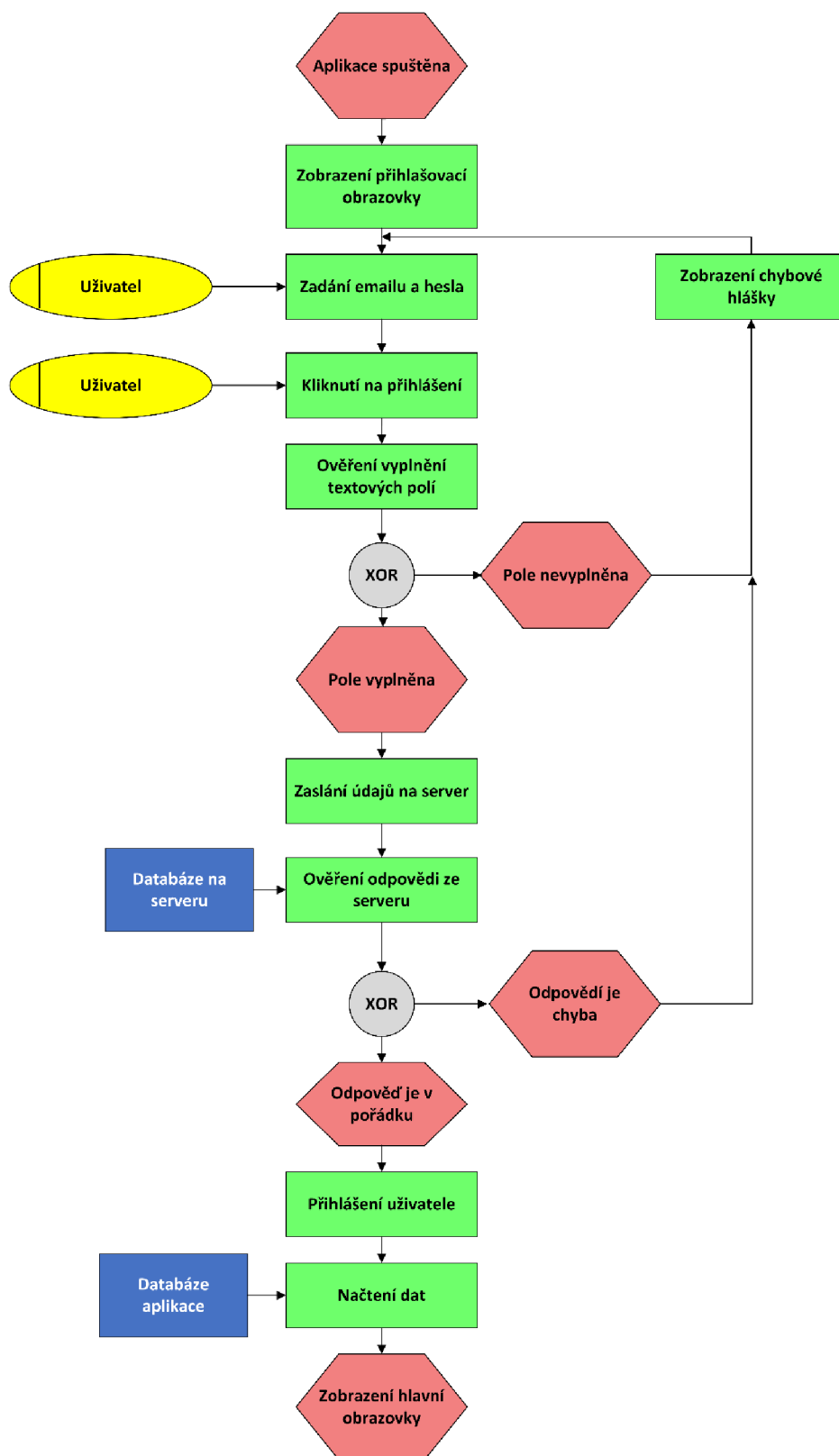
Obrázek 22: Diagram případů užití
(Zdroj: Vlastní zpracování)

3.3.2 Přihlášení uživatele

První popisovanou činností je přihlášení uživatele. Tato činnost začíná spuštěním aplikace, po kterém následuje zobrazení přihlašovací obrazovky. Na této obrazovce může uživatel vidět dvě textová pole – email a heslo – a tlačítko přihlášení. Po kliknutí na toto tlačítko se zavolá podmínka. Ta ověří, zda uživatel vyplnil obě textová pole a pokud ne, tak zobrazí chybovou hlášku a uživatel zůstane na přihlašovací obrazovce. Pokud je podmínka splněna a pole jsou vyplněna, tak jsou zadané údaje poslané na server. Poté se ověřuje, zda server vrátil odpověď bez chyby a pokud ne, tak se opět objeví chybová hláška a uživatel opět zůstává na přihlašovací obrazovce. Pokud server vrátil bezchybnou odpověď, tak je vše v pořádku, podmínka je splněna a uživatel je přihlášen. Po jeho přihlášení jsou načtena data z lokální databáze aplikace a otevřena hlavní obrazovka.



Obrázek 23: Vývojový diagram přihlášení uživatele
(Zdroj: Vlastní zpracování)



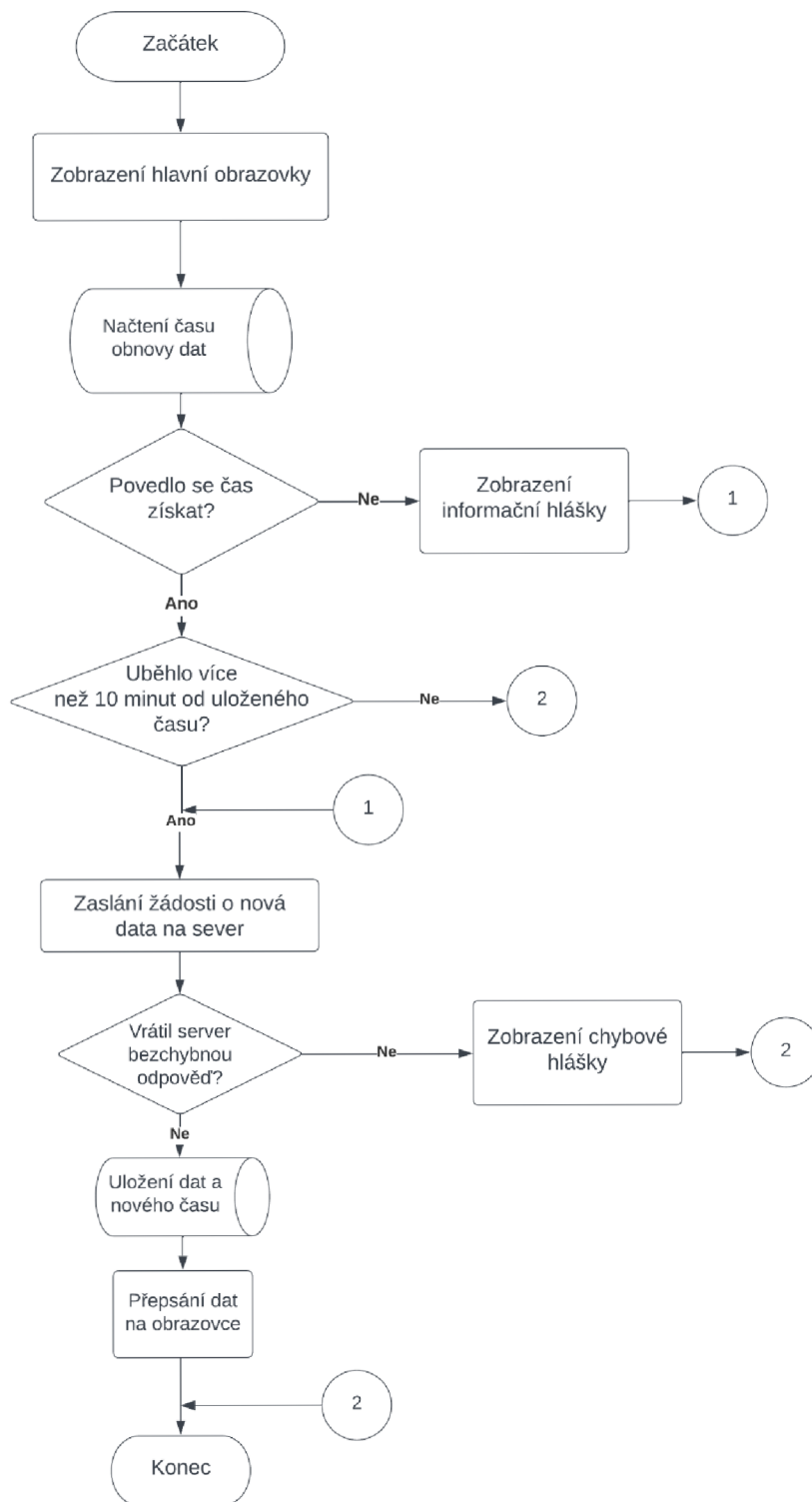
Obrázek 24: EPC diagram přihlášení uživatele
(Zdroj: Vlastní zpracování)

3.3.3 Obnova dat po deseti minutách

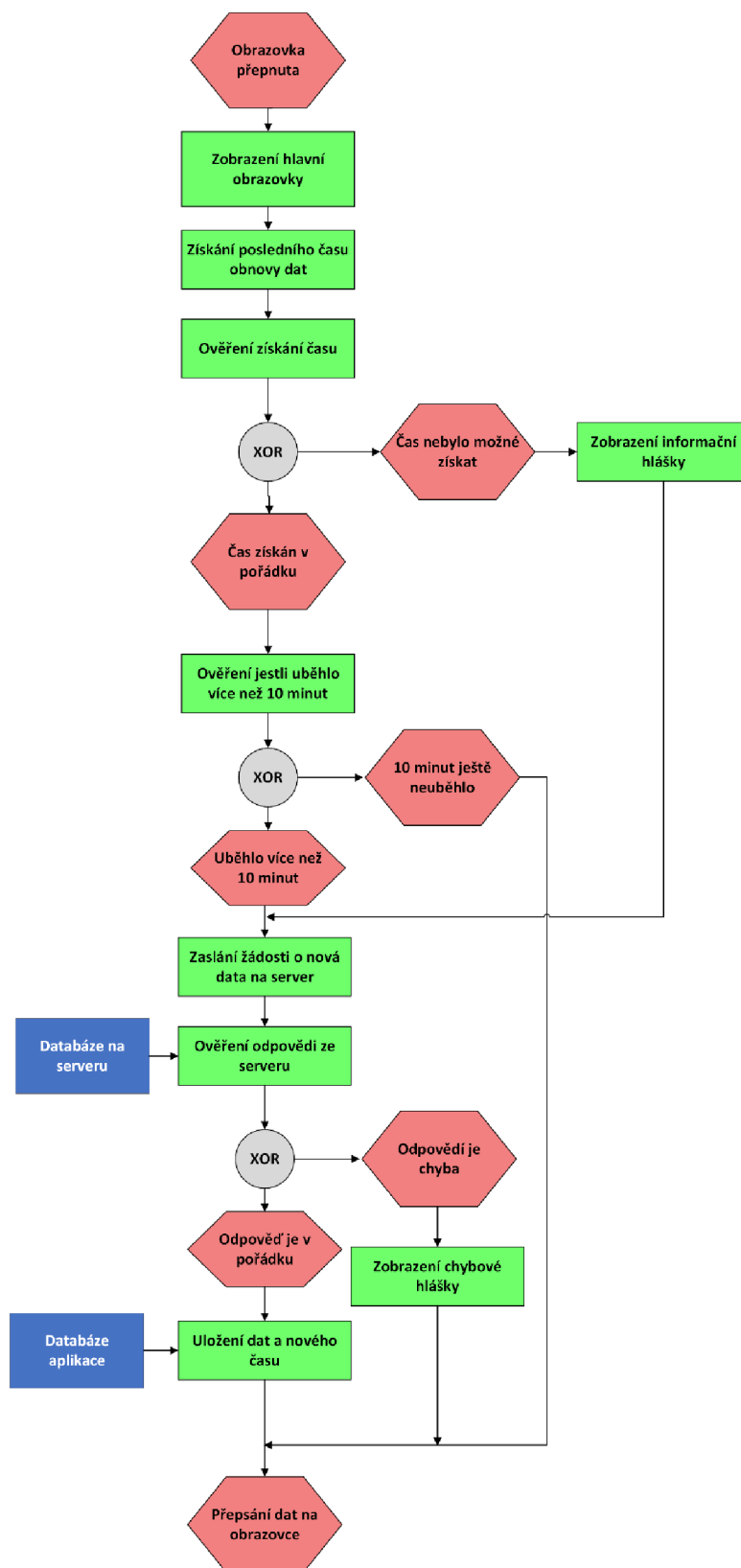
Další popisovanou činností je obnova dat po deseti minutách, může být spuštěna jak automaticky (při načtení obrazovky), tak manuálně skrze tlačítko. V níže zakresleném grafu se jedná o automatické obnovení. K popsání této činnosti je potřeba také zmínit, že čas obnovení dat se ukládá do databáze aplikace ve formě tzv. „časová značka“.

Tato činnost tedy začíná zobrazením hlavní obrazovky, kdy se zavolá funkce, která se snaží získat původní čas obnovení z lokální databáze. Zde se nám činnost začíná lehce větvit. Pokud se povede čas získat, tak se spustí další podmínka, která ověřuje, zda již uběhlo více než 10 minut od uloženého času, pokud ne, není se o co starat, data jsou v našem případě stále aktuální. Pokud ano – uběhlo více než 10 minut, tak je potřeba získat nová data. Nicméně, pokud hned první podmínka selže a čas se nepovede získat, tak to znamená, že buď se data ještě nikdy neobnovovala nebo nastala jiná neočekávaná chyba. V takovém případě již neověřujeme jestli uběhlo deset minut – nemáme ani jak, když neznáme čas – a rovnou získáváme nová data.

V procesu získávání nových dat nejdříve pošleme žádost o nová data na server. Jakmile získáme zpět odpověď, tak ověříme zda v ní byla nějaká chyba a pokud ano, tak chybu vypíšeme a tím končí celá činnost. Pokud vše proběhlo v pořádku a získali jsme zpět odpověď bez chyby, tak uložíme nová data do lokální databáze společně s časovou značkou. Nakonec ještě přepíšeme data na obrazovce a činnost je u konce.



Obrázek 25: Vývojový diagram obnovy dat po deseti minutách
(Zdroj: Vlastní zpracování)



Obrázek 26: EPC diagramu obnovení dat po deseti minutách
(Zdroj: Vlastní zpracování)

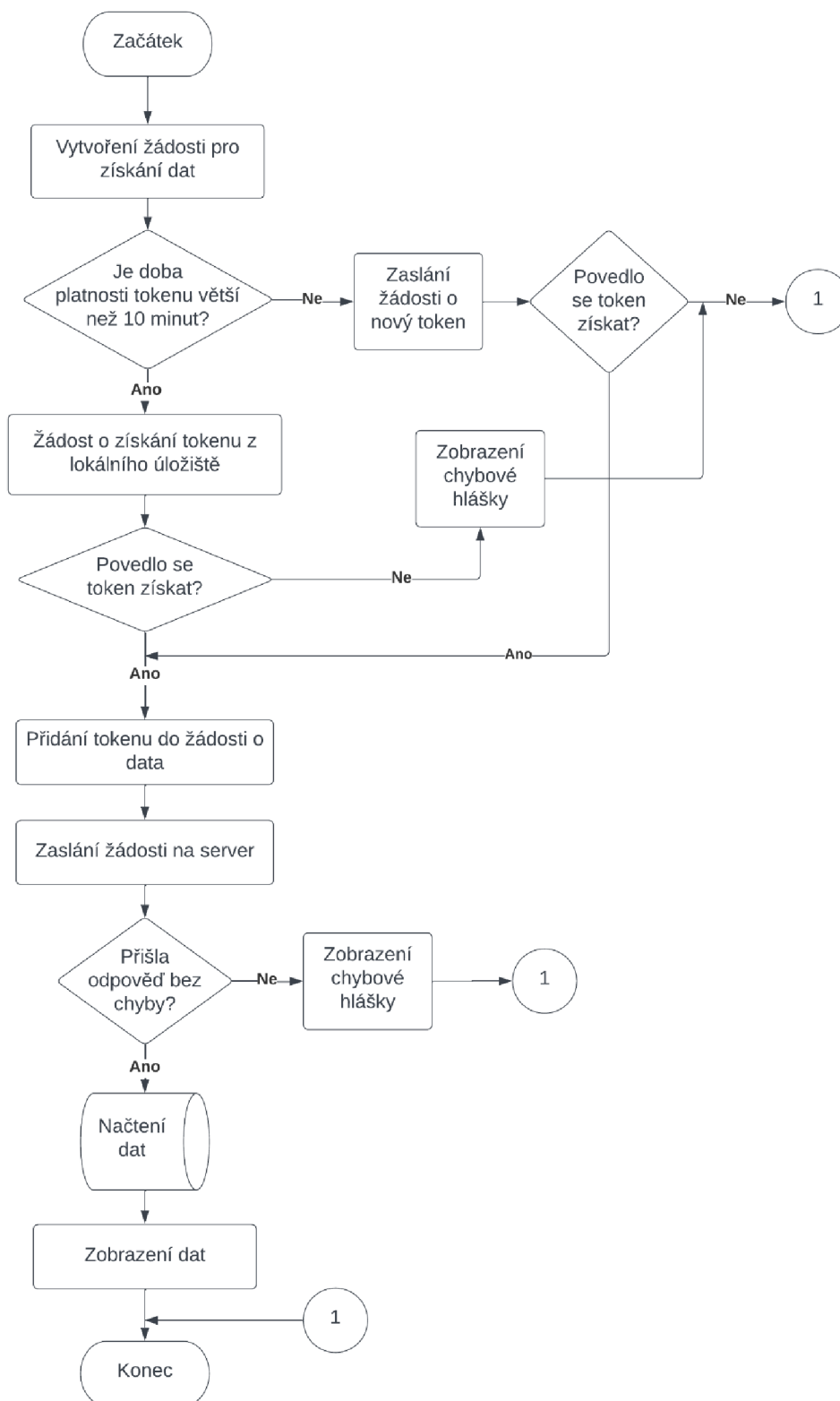
3.3.4 Ověřování tokenu při získávání dat ze serveru

Poslední popisovanou činností je ověřování tokenu při získávání dat. Tato činnost už je více technicky zaměřená, ale vzhledem k tomu, že jsem nic podobného předtím nedělala, tak jsem se rozhodla ji zjednodušeně popsat i v této práci. V této části se nebudu zaměřovat na to, co je to vlastně token nebo proč byla tato funkce v aplikaci potřebná, ale budu pouze popisovat programovou funkcionalitu.

Na začátku této činnosti vytvoříme žádost pro získání potřebných dat (třeba při jejich obnovení, které bylo popsáno v předchozí podkapitole). Po vytvoření žádosti ověříme, zda je doba platnosti (expirace) tokenu větší než 10 minut. Pokud ano, token získáme z lokální databáze, přidáme do žádosti a tu pošleme na server. Pokud ne, tedy token buď propadl nebo propadá někdy v následujících 10 minutách, tak je potřeba zaslat žádost o nový token. Nicméně, pokud by se token nepodařilo získat ani po této žádosti, tak je celá činnost u konce a data nelze získat. V takovém případě by byla zobrazena chybová hláška.

Po úspěšném (získání a) přidání tokenu do žádosti, ji pošleme na server a čekáme na odpověď. Jestliže odpověď je bez chyby, tak můžeme načíst data, tentokrát z databáze serveru a zobrazit je. Pokud by přišla chybná odpověď, tak se opět zobrazí příslušná chybová hláška a činnost je u konce.

Proces ověřování tokenu probíhá při každé žádosti posílané na server, což znamená, že může proběhnout i 10krát za minutu (pokud jsou na jedné obrazovce různá data).



Obrázek 27: Ověření tokenu při získávání dat
(Zdroj: Vlastní zpracování)

3.4 Zavedení systému

Při zavádění aplikace jsem použila nárazovou strategii – tedy nahrazení původního systému novým tzv. „ze dne na den“. Vzhledem k tomu, že společnost do této doby vlastně nepoužívala alternativní aplikaci, kdy všechna data se získávala manuálně z databáze, tak jinou strategii nebylo možné použít.

3.5 Přínos práce

Aplikace vytvořená v této bakalářské práci pro firmu Eventee s.r.o. bude mít přínos především pro její zaměstnance. V analýzách provedených v kapitole 2, bylo zjištěno, že současným řešením je manuální export údajů z databáze. Nová aplikace by tak měla zaměstnancům ušetřit čas, kdy už není potřeba manuálně provádět export údajů z databáze, čekat na exportovaná data, převádět do vhodných grafů, tabulek nebo jiné požadované formy. Všechny funkce a zobrazení aplikace poskytuje sama na základě dříve požadavků společnosti získaných při analýze. Další velkou výhodou vidím v možném zefektivnění pracovních procesů, kdy s využitím aplikace mají zaměstnanci k neustále k dispozici aktuální data.

3.6 Ekonomické zhodnocení

Tato kapitola se věnuje ekonomickému zhodnocení vypracovaného řešení. Ve zhodnocení nejsou brány v úvahu náklady na provoz, které se nedají konkrétně určit.

Hodinová mzda společná pro všechny dílčí části, návrh, vývoj a testování, byla stanovena na 320 Kč. Prototypování neboli návrh funkcionalit a designu zabral jeden měsíc, když počítáme i nepracovní dny. V převodu na pracovní dny se tak jednalo o přibližně 20 dní. Samotný vývoj aplikace byl původně stanoven na přibližně 50 až 55 pracovních dní, ale nakonec trval delší dobu kvůli nepředpokládaným komplikacím. Mezi ty patřily třeba chyby ve vývojovém prostředí, neexistující dokumentace k použitým knihovnám nebo zrušená podpora některých komponent. Celková doba vývoje aplikace tedy trvala 65 dní. Závěrečné testování aplikace proběhlo již jen manuálně, a tak trvalo pouze 10 dní.

V tabulce níže počítám jeden pracovní den jako 7,5 hodin.

Tabulka 4: Ekonomické zhodnocení

(Zdroj: Vlastní zpracování)

Položka	Trvání ve dnech	Trvání v hodinách	Částka v Kč
Prototypování	20	150	48 000
Vývoj aplikace	65	487,5	156 000
Testování	10	75	24 000
Celkem	-	-	228 000

Konečná částka, spočtená na 228 000 korun českých, byla společnosti vystavená ve formě faktury. Částka je tedy pro společnost konečná, o odvody sociálního a zdravotního pojištění se již společnost nestará.

ZÁVĚR

Cílem této práce bylo navrhnout, vytvořit a implementovat analytickou aplikaci pro společnost Eventee. Aplikace měla za úkol zejména zefektivnit pracovní procesy, kdy již nikdo nemusí čekat na vygenerování reportů z databáze, ale může je okamžitě vidět v aplikaci.

První část práce se zabývala analýzou teoretických východisek. Představila tak důležité pojmy, jak z oblasti vývoje, tak návrhu, analýzy a implementace. Mezi tyto pojmy patřila platforma tvOS, vývojové prostředí Xcode, jazyk Swift, datové a funkční modelování včetně příslušných diagramů.

Druhá část práce se věnovala představení společnosti Eventee, její historii a nabízeným službám. Potom bylo představeno současného řešení společnosti, jímž byl manuální export údajů z databáze. Následovně se kapitola věnovala požadavkům na aplikaci a možným alternativám aplikace. Na závěr kapitoly byla také provedena upravená SWOT analýza současného řešení, a to jak slovně, tak i v tabulce.

Poslední částí práci byl samotný návrh aplikace. Ten zahrnoval, jak návrh uživatelského rozhraní, tak databáze a jednotlivých funkcí. Po těchto kapitolách byly představeny i některé konkrétní činnosti aplikace a to přihlášení, obnovení dat a ověření tokenu před získáním informací ze serveru. Závěr vlastního řešení byl věnován ekonomickému zhodnocení a přínosům práce.

Kromě přínosu pro zaměstnance firmy Eventee s.r.o. měla tato práce určitě i přínos osobní. Díky této práci jsem získala celou spoustu nových vědomostí z oblasti návrhu a vývoje aplikací, ale i analýzy. Aplikace se zaměřovala pouze na platformu tvOS, a tak jsem měla příležitost prozkoumat, jak se s touto platformou pracuje, jaké jsou její specifické komponenty a co vše je možné i nemožné. Vzhledem k tomu, že aplikaci jsem i navrhovala, tak jsem se také seznámila s prototypováním v aplikaci Wondershare Mockitt. Často aplikace vytvářím bez jasných návrhů a pak je několikrát předělávám. Takže tato zkušenost, kde jsem navrhovala obrazovku po obrazovce, byla zcela novým zážitkem, který mi ukázal jak je důležité navrhovat systém dopředu. Především, jak moc času ušetřím při následné programové implementaci. Nicméně ani zkušenosti z oblasti analýzy, navrhování databáze nebo sestavování diagramů nemohu opomenout. Díky nim

jsem měla možnost vyzkoušet si znalosti, které jsem získala v průběhu studia. Ale také mít dopředu jasnou představu, jak chci navrhnout svoji aplikační databázi a jak budu později s daty pracovat.

SEZNAM POUŽITÝCH ZDROJŮ

- 1) KAPOUN, Jan. Historie společnosti Apple. *CIO Business World* [online]. 2021 [cit. 2023-02-28]. Dostupné z: <https://www.cio.cz/clanky/historie-firmy-apple-15438/>
- 2) The Founding of Apple Computer, Inc. *Library of Congress* [online]. [cit. 2023-05-12]. Dostupné z: <https://guides.loc.gov/this-month-in-business-history/april/apple-computer-founded>
- 3) Apple vylepšuje svůj rekord, pomohl soud i iPhone 5. *Aktuálně.cz* [online]. [cit. 2023-2-2]. Dostupné z: <https://zpravy.aktualne.cz/applelogo/r~i:photo:477758/r~i:article:755684/>
- 4) HIMMER, Matyáš. Swift akademie pro začátečníky: začínáme psát první řádky kódu #2. Český *Mac* [online]. 2020 [cit. 2023-02-28]. Dostupné z: https://www.cesky_mac.cz/swift-akademie-pro-zacatecniky-zaciname-psat-prvni-radky-kodu-2/
- 5) Swift. Apple *Developer* [online]. 2023 [cit. 2023-02-28]. Dostupné z: <https://developer.apple.com/swift/>
- 6) DWIVEDI, Harshit. Data Types in Swift — Where it All Starts. *Medium* [online]. 2019 [cit. 2023-02-28]. Dostupné z: <https://medium.com/coding-blocks/data-types-in-swift-where-it-all-starts-c1311fa93368>
- 7) FILIP, Jiří. Apple TV: Co to je a proč ji chtít. *Zboží.cz* [online]. 2021 [cit. 2023-02-28]. Dostupné z: https://www.zbozi.cz/magazin/c/apple-tv-informace/?_st=MTY3NzYxMDc5Njg4OQ%3D%3D
- 8) LUTKEVICH, Ben. TvOS definition. *TechTarget* [online]. 2022 [cit. 2023-02-28]. Dostupné z: <https://www.techtarget.com/searchmobilecomputing/definition/tvOS>
- 9) Definition integrated development environment (IDE). *TechTarget* [online]. [cit. 2023-04-30]. Dostupné z: <https://www.techtarget.com/searchsoftwarequality/definition/integrated-development-environment>
- 10) EKREN, Esat Kemal. What Is Xcode and How to Use It?. *Netguru* [online]. 2022 [cit. 2023-02-28]. Dostupné z: <https://www.netguru.com/blog/what-is-xcode-and-how-to-use-it>

- 11) HINDI, Daniel. What is an iOS App Simulator and How Does it Work?. *Buildfire* [online]. 2023 [cit. 2023-02-28]. Dostupné z: <https://buildfire.com/ios-app-simulator-and-how-it-works/>
- 12) SHARMA, Krupanshu. MVVM in iOS – A Quick Walkthrough. *Clarion Technologies* [online]. 2023 [cit. 2023-02-28]. Dostupné z: <https://www.clariontech.com/blog/mvvm-in-ios-a-quick-walkthrough>
- 13) JAFFEE, Andrew. Introduction to MVVM: Refactoring a MVC App Using the MVVM Design Pattern. *Appcoda* [online]. 2018 [cit. 2023-02-28]. Dostupné z: <https://www.appcoda.com/mvvm-vs-mvc/>
- 14) HUDSON, Paul. What is a storyboard?. *Hacking with Swift* [online]. 2019 [cit. 2023-02-28]. Dostupné z: <https://www.hackingwithswift.com/example-code/language/what-is-a-storyboard>
- 15) HELPPI, Ville-Veikko. Using Xcode Storyboard for UI Building and Testing. *Smartbear* [online]. 2017 [cit. 2023-02-28]. Dostupné z: <https://smartbear.com/blog/using-xcode-storyboard-for-ui-building-and-testing/>
- 16) LEŠKA, Šimon. Co je to UX a UI design?. *BlueGhost* [online]. 2020 [cit. 2023-04-30]. Dostupné z: <https://www.blueghost.cz/clanek/co-je-to-ux-a-ui-design/>
- 17) If you're a designer, here's why you need to try out Wondershare Mockitt. *Pocket-lint* [online]. 2021 [cit. 2023-04-30]. Dostupné z: <https://www.pocket-lint.com/apps/news/157510-if-you-are-a-designer-here-s-why-you-need-to-try-out-wondershare-mockitt/>
- 18) Co je to wireframe, prototyp a mockup?. *Simonjun* [online]. [cit. 2023-04-30]. Dostupné z: <https://www.simonjun.cz/blog/co-je-to-wireframe-prototyp-mockup>
- 19) STŘELEČ, Michal. Co je to API?. *Michal Střelec* [online]. 2023 [cit. 2023-02-28]. Dostupné z: <https://www.strelec.pro/napsal/co-je-to-api>
- 20) MALÝ, Martin. REST: architektura pro webové API. *Zdroják* [online]. 2009 [cit. 2023-04-30]. Dostupné z: <https://zdrojak.cz/clanky/rest-architektura-pro-webove-api/>
- 21) SWOT analýza. *Management Mania* [online]. 2016 [cit. 2023-02-28]. Dostupné z: <https://managementmania.com/cs/swot-analyza>
- 22) KOCH, Miloš a Beunard NEUWIRTH. *Datové a funkční modelování*. Brno: CERM, 2006. ISBN 978-80-214-4125-5.

- 23) Definition Entity Relationship Diagram (ERD). *TechTarget* [online]. [cit. 2023-04-30]. Dostupné z: <https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD>
- 24) ČÁPKA, David. Lekce 2 - UML - Use Case Diagram. *IT Network* [online]. [cit. 2023-04-30]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>
- 25) Vývojové diagramy. *Matematická biologie* [online]. [cit. 2023-05-01]. Dostupné z: <https://portal.matematickabiologie.cz/index.php?pg=zaklady-informatiky-pro-biology--algoritmizace-a-programovani--navrh-algoritmu-i--vyvojove-diagramy>
- 26) Event-driven process chain (EPC). *ARIS Community* [online]. 2023 [cit. 2023-05-08]. Dostupné z: <https://ariscommunity.com/event-driven-process-chain>
- 27) KOCH, Miloš, Jan DOVRTĚL, Tomáš HRŮZA a Hana NENIČKOVÁ. *Management informačních systémů*. Brno: CERM, 2010. ISBN 978-80-214-4157-6.
- 28) Seznamte se s lidmi, kteří stojí za Eventee. *Eventee* [online]. [cit. 2023-01-31]. Dostupné z: <https://eventee.co/cs/about>
- 29) Event App Features for Enhancing the Attendee Experience. *Eventee* [online]. 2023 [cit. 2023-05-12]. Dostupné z: <https://eventee.co/en/features/>
- 30) Numerics. *Sourceforge* [online]. [cit. 2023-05-01]. Dostupné z: <https://sourceforge.net/software/product/Numerics/>
- 31) Numerics Home. *Numerics* [online]. [cit. 2023-05-01]. Dostupné z: <https://cynapse.com/numerics/>
- 32) Business dashboards for tracking data from all the different tools you're using. *Databox* [online]. [cit. 2023-05-01]. Dostupné z: <https://databox.com/product/databoards>
- 33) Track & improve your performance at an affordable price. *Databox* [online]. [cit. 2023-05-01]. Dostupné z: <https://databox.com/pricing>
- 34) What is Klipfolio?. *Klipfolio Help Center* [online]. [cit. 2023-05-01]. Dostupné z: <https://support.klipfolio.com/hc/en-us/articles/225162928-What-is-Klipfolio->
- 35) Dashboards and reports. *Klipfolio* [online]. [cit. 2023-04-30]. Dostupné z: <https://www.klipfolio.com/features/dashboards-reports>
- 36) Rock-solid metrics without the price tag of traditional BI. *Klipfolio* [online]. [cit. 2023-05-01]. Dostupné z: <https://www.klipfolio.com/pricing>

37) Eventee s.r.o., Brno IČO 29307236 - Obchodní rejstřík firem. *Rejstřík firem* [online].
[cit. 2023-05-14]. Dostupné z: <https://rejstrik-firem.kurzy.cz/29307236/eventee-sro/>

SEZNAM POUŽITÝCH OBRÁZKŮ

Obrázek 1: Logo společnosti Apple (Zdroj: [3])	15
Obrázek 2: Xcode prostředí (Zdroj: [10]).....	19
Obrázek 3: MVVM aplikace (Zdroj: [13])	20
Obrázek 4: Ukázka vytváření Storyboardu (Zdroj: [15])	21
Obrázek 5: SWOT analýza (Zdroj: [21])	23
Obrázek 6: Vybrané značky vývojového diagramu (Zdroj: Vlastní zpracování podle [25])	25
Obrázek 7: Logo společnosti (Zdroj: [28])	28
Obrázek 8: Ukázka aplikace Numerics (Zdroj: [31])	31
Obrázek 9: Ukázka aplikace Databox (Zdroj: [32])	32
Obrázek 10: Ukázka aplikace Klipfolio (Zdroj: [35])	33
Obrázek 11: Přihlašovací obrazovka (Zdroj: Vlastní zpracování).....	39
Obrázek 12: Prvotní návrh obrazovky Overview (Zdroj: Vlastní zpracování)	40
Obrázek 13: Výsledný návrh obrazovky Overview (Zdroj: Vlastní zpracování).....	40
Obrázek 14: Obrazovka Dashboard (Zdroj: Vlastní zpracování)	41
Obrázek 15: Obrazovka Events (Zdroj: Vlastní zpracování).....	42
Obrázek 16: Obrazovka detailu konference (Zdroj: Vlastní zpracování).....	43
Obrázek 17: Obrazovka Users (Zdroj: Vlastní zpracování)	44
Obrázek 18: Detail uživatele (Zdroj: Vlastní zpracování).....	45
Obrázek 19: Custom obrazovka s jedním grafem (Zdroj: Vlastní zpracování)	45
Obrázek 20: Obrazovka s nastavením (Zdroj: Vlastní zpracování).....	46
Obrázek 21: ER diagram aplikační databáze (Zdroj: Vlastní zpracování)	47
Obrázek 22: Diagram případů užití (Zdroj: Vlastní zpracování).....	52
Obrázek 23: Vývojový diagram přihlášení uživatele (Zdroj: Vlastní zpracování).....	54
Obrázek 24: EPC diagram přihlášení uživatele (Zdroj: Vlastní zpracování)	55
Obrázek 25: Vývojový diagram obnovy dat po deseti minutách (Zdroj: Vlastní zpracování).....	57
Obrázek 26: EPC diagramu obnovení dat po deseti minutách (Zdroj: Vlastní zpracování)	58
Obrázek 27: Ověřování tokenu při získávání dat (Zdroj: Vlastní zpracování).....	60

SEZNAM POUŽITÝCH TABULEK

Tabulka 1: Základní datové typy ve Swiftu (Zdroj: Vlastní zpracování podle [6])	17
Tabulka 2: Základní údaje o společnosti (Zdroj: Vlastní zpracování podle [37])	28
Tabulka 3: SWOT analýza současného řešení (Zdroj: Vlastní zpracování)	35
Tabulka 4: Ekonomické zhodnocení (Zdroj: Vlastní zpracování)	62