

BAKALÁŘSKÁ PRÁCE

Vývoj docházkového systému pro společnost Školička Krásná z.s.

Development of Attendance System for company Školička Krásná z.s.

STUDIJNÍ PROGRAM

Ekonomika a management

STUDIJNÍ OBOR

Řízení a ekonomika průmyslového podniku

VEDOUCÍ PRÁCE

Ing. Bc. Pavel Andres, Ph.D., ING.PAED. IGIP, Oddělení pedagogických a psychologických studií

KAUFMANN

MAREK

2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Kaufmann Jméno: Marek Osobní číslo: 439557
Fakulta/ústav: Masarykův ústav vyšších studií (MÚVS)
Zadávací katedra/ústav: Oddělení pedagogických a psychologických studií / Masarykův ústav vyšších studií
Studijní program: Ekonomika a management
Studijní obor: Řízení a ekonomika průmyslového podniku

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:
Vývoj docházkového systému pro společnost Školička Krásná z.s.

Název bakalářské práce anglicky:
Development of Attendance System for Company Školička Krásná z.s.

Pokyny pro vypracování:

CÍL: Cílem bakalářské práce je nalezení a popsání procesů, které jsou obsaženy v projektu vývoje aplikace. Rovněž dojde k vytvoření aplikace pro společnost Školička Krásná z.s., která bude primárně sloužit jako nástroj pro tvorbu docházkového systému s ohledem na legislativní náležitosti a klientovy požadavky.

PŘÍNOS: Vytvořením této aplikace dojde k redukci administrativního zatížení zaměstnanců dané společnosti a splnění projektových podmínek.

OSNOVA: 1. Úvod; 2. Teoretická část - požadavky pro tvorbu aplikace, předvývojová analýza, popis procesů vývoje; 3. Praktická část - analýza současného stavu, předvývojová fáze, vývojová fáze, odevzdání projektu; 4. Závěr.

Seznam doporučené literatury:

KANISOVÁ, Hana a Miroslav MÜLLER. UML srozumitelně. Computer Press, 2006.

ARLOW, Jim a Ila NEUSTADT. UML 2 a unifikovaný proces vývoje aplikací: objektivě orientovaná analýza a návrh prakticky. Computer Press, 2007.

LACKO Luboslav. Vývoj aplikací pro Android. Computer Press, 2015.

NEZMAR Luděk. GDPR: Praktický průvodce implementací. Grada, 2017.

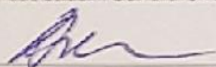
Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Bc. Pavel Andres, Ph.D., ING.PAED.IGIP, Oddělení pedagogických a psychologických studií

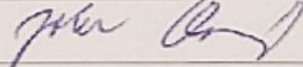
Jméno a pracoviště konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: 5.12.2017 Termín odevzdání bakalářské práce: 5.5.2018

Platnost zadání bakalářské práce: 31.8.2019



Podpis vedoucí(ho) práce



Podpis vedoucí(ho) ústavu/katedry

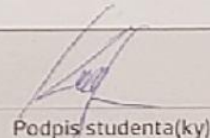


Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

21-03-2018

Datum převzetí zadání



Podpis studenta(ky)

KAUFMANN, Marek. Vývoj docházkového systému pro společnost Školička Krásná z.s..
Praha: ČVUT 2018. Bakalářská práce. České vysoké učení technické v Praze, Masarykův
ústav vyšších studií.



**MASARYKŮV ÚSTAV
VYŠŠÍCH STUDIÍ
ČVUT V PRAZE**

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně. Dále prohlašuji, že jsem všechny použité zdroje správně a úplně citoval a uvádím je v přiloženém seznamu použité literatury.

Nemám závažný důvod proti zpřístupňování této závěrečné práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Praze dne: 03. 05. 2018

Podpis:

Poděkování

Jako autor bych rád poděkoval veškerým osobám, které se podílely na vzniku této práce. Zejména poté vedoucímu práce panu doktoru Andresovi, oponentovi práce, společnosti Školička Krásná z.s. a své rodině, kteří mne podporovali při tvorbě této práce. Bez jejich pomoci by tato práce nemohla vzniknout.

Abstrakt

Cílem bakalářské práce je popsání průběhu takového projektu, kdy dojde k implementaci nového softwaru. V rámci volby řešení tohoto projektu využijeme příslušné moderní nástroje, pomocí kterých můžeme výhodnost jednotlivých řešení ohodnotit. Mezi tyto řešení rovněž zahrneme možnost provést vývoj vlastního softwaru.

Současně v této práci bude řešen projekt na zavedení docházkového systému ve společnosti Školička Krásná z.s., která bude primárně sloužit jako nástroj pro evidenci docházky s ohledem na legislativní a klientovy požadavky.

Závěrem této práce dojde k předání toho systému k užívání do výše uvedené společnosti.

Klíčová slova

Projekt, Doba návratnosti, náklady na projekt, UML, Jednotný modelovací jazyk, Vývoj softwaru, Docházkový systém, Požadavky, Správa požadavků

Abstract

The aim of the bachelor thesis is to describe the course of such a project, when the new software will be implemented. In choosing solutions to this project, we will use the appropriate modern tools to evaluate the value of individual solutions. Among these solutions, we will include possibility to develop software ourselves.

Also, the software for company Školička Krásná z.s. will be developed and its purpose will be mainly attendance system with handling law and client requirements.

In the end of this thesis, the working system will be handed to mentioned company for usage.

Key words

Project, Payback Period, Project Costs, UML, Unified Modeling Language, Software Development, Attendance System, Requirements, Requirement management

Obsah

Úvod.....	5
1 Projekt	7
2 Finanční analýza.....	10
2.1 Doba návratnosti investice	10
2.2 Posouzení investice	10
2.3 Náklady projektu	13
3 Vývoj aplikace	14
3.1 Vysvětlení UML.....	14
3.2 Identifikace Stakeholderů	14
3.3 Správa požadavků.....	15
3.4 Sběr požadavků	17
3.5 Analýza sběru dat.....	23
4 Popis společnosti.....	33
5 Představení projektu.....	33
6 Specifikace hledaného řešení	34
7 Možnosti řešení	36
8 Výběr řešení	36
9 Sběr požadavků	41
9.1.1 Způsob evidence	42
9.1.2 Role	42
9.1.3 Správa dětí	43
9.1.4 Záznam docházky dětí	47
9.1.5 Správa učitelů	50
9.1.6 Záznam docházky učitelé.....	54
9.1.7 Stravné.....	56
9.1.8 Fakturace	59
Závěr	61
Seznam použité literatury	64
Seznam obrázků	65
Seznam tabulek	67

Úvod

Proč se zabývat vývojem aplikace? Motivací pro vytvoření vlastního fungujícího systému může být nespočet. Níže si jich pro ukázkou pár uvedeme.

Například otázka finanční. Vyplatí se nám investovat čas, finance a další zdroje do vývoje vlastní aplikace? V tomto ohledu budeme později řešit, jaká bude muset být životnost vyvinutého softwaru, aby veškerá částka peněz investovaná do vývoje, a to včetně nezbytných prostředků na provoz a údržbu, byla menší než při volbě již existujícího řešení s totožnými možnostmi na trhu, pokud však takové existuje.

Dalším pohledem tedy může být vývoj softwaru jednoduše proto, že trh požadované řešení nenabízí. Je tedy kladena vysoká specifická potřeba na software, a to sice ze strany klienta. Klient může být na rozdíl od legislativních požadavků náročnější a požadovat implementaci takových funkcí, které nejsou běžné a stávající možnosti trhu s nimi nepočítají.

Jiným pohledem může být jednoduchý požadavek klienta. V tomto případě jde o vytvoření vlastního řešení, které je nezávislé na jiných dodavatelích. Tento postoj přináší výhody, mezi které lze zařadit možnost ovlivnit nebo jednoduše upravit chod celého systému. Další výhodou je velmi dobrá znalost, chcete-li pochopení, procesů a jejich průběhu v rámci daného systému.

Poslední motivací může být zdokonalení se v dané oblasti. Ať už se jedná o stránku vývoje softwaru samotného, či získání přehledu o procesech, organizaci, legislativních požadavcích či komunikaci s klientem a příslušnými orgány, kompetentními v dané problematice.

Motivací pro tvorbu této bakalářské práce byly všechny výše zmíněné možnosti a rovněž poukázání na to, že tato disciplína vyžaduje velké úsilí pro úspěšnou realizaci.

Bakalářská práce bude rozdělena na dvě části, a to sice na teoretickou a praktickou část. V teoretické části bude využita literatura k nastínění situace, která je spojena s výběrem vhodného softwarového řešení. Tento proces bude tvořen identifikací klíčových osob, finanční analýzou či sběrem klientových požadavků na funkce softwaru.

V rámci praktické části dojde k výběru řešení, které splňuje všechny klientovy a legislativní požadavky a jehož cílem je implementace docházkového systému do společnosti Školička Krásná z.s. K tomuto určení budou využity pasáže, které budou řešeny v rámci části teoretické. Vzhledem k velikosti a náročnosti tohoto projektu budou témata rozebírána tak, aby dokázala pokrýt provedené práce v rámci praktické části.

TEORETICKÁ ČÁST

V rámci teoretické části této práce bych se rád zaobíral tématy, která jsou spojena s výběrem a implementací nové softwarové aplikace, a to včetně možnosti, že tento požadavek budeme řešit vlastním vývojem. Možnost, kdy volíme vývoj nové aplikace by se na první pohled mohl zdát, jako ryze programátorská záležitost. Samozřejmě to tak uchopit lze, ale pravděpodobnost, že výsledek vrátí investované prostředky (peněžní, hmotné, časové), nemusí být tak velká, jako když využijeme moderních nástrojů pro organizaci, komunikaci a následně i pro samotnou práci programování. Jednoznačnou výhodou použití nástrojů a principů, které jsou zmiňovány v této práci, je že nepracujeme naslepo, protože již dopředu známe požadovaný výsledek.

1 Projekt

Jak již bylo řečeno, tato bakalářská práce se zabývá případem, kdy potřebujeme zavést do stávající společnosti nový software. Hovoříme tudíž o projektu. To, co a proč můžeme považovat za projekt si vysvětlíme v následujících řádcích. Ve velkých společnostech můžeme nalézt celé útvary, které se zabezpečením řízení projektu zabývají a pro provoz své agendy mají vymezené nemalé prostředky. Pokud budeme hledat řešení pro malou společnost o několika zaměstnancích s nízkým rozpočtem, tak se koncepce projektu nemusí jevit jako nezbytná. Nicméně použití níže popsanych pravidel není rozhodně kontraproduktivní a může nám pomoci i při rozsahově malém projektu.

Co je a není projektem je občas problém rozlišit. Pro jistotu si zde rozeberme, podle jakých ukazatelů poznáme, zdali se jedná o projekt jedná či nikoliv, a to sice dle ukazatelů, kterými jsou:

- jedinečnost
- rozsah
- časové omezení
- finanční omezení
- zdrojové omezení
- složitost
- riziko

(Doležal, a další, 2017)

V dostupné literatuře není definice projektu jednotná čili jednoznačné vymezení neexistuje. Nicméně dobrým příkladem může být to, že se jedná o běh činností, které vedou k naplnění určitého cíle. Názorné to může být na aktivitách nespecifikovaného automobilového závodu. Zatímco vývoj nového automobilu jedinečný je, tak jeho následná výroba již není. V prvním případě se jedná o specifické záležitosti, jejíž výsledek je nejistý. V případě výroby se jedná o každodenní neměnní se činnosti. Rozsah je položka, která říká, co vše bude v rámci projektu vyprodukováno. Do tohoto rozsahu řadíme samotné výstupy projektu, ale rovněž i dokumentaci, která je s ním spojená. Zdroje jsou pro nás veškeré hmotné, nehmotné a lidské zdroje, které jsou pro

daný projekt v jeho časovém a finančním ohraničení k dispozici pro dosažení cíle. Cílem poté chápeme stav, v němž chceme, aby se projekt na jeho konci nacházel. Lépe řečeno se o soubor činností, jejichž výsledkem je určitý druh výstupu. Důvodem proč projekt realizujeme, jsou nejčastěji jeho přínosy. Projekt jako takový má určitou úroveň složitosti, což ho opět spolu s rizikem neúspěchu dělá jedinečným. Nemůžeme říct, že kontinuální výroba již otestovaného automobilu je riziková ve smyslu projektu. Pokud budeme hovořit o vývoji specifického softwaru pro určitého klienta, můžeme tedy konstatovat, že se o projekt jedná. Uvedme si příklady rizik, kterým můžeme v rámci projektu čelit. V průběhu této práce s nimi budeme pracovat velmi často a za hlavní tedy dle literatury považujeme:

- nedostatek zdrojů
- nejednoznačné zadání
- špatně řízené vztahy se stakeholdery
- změny zadání

(Doležal, a další, 2017)

Všimněte si v průběhu této práce, jak se tato rizika budou postupně objevovat. Sice ne vždy budou pojmenována stejně, ale jejich význam, respektive důsledek ano.

Dalším důležitým pojmem spojeným s projektem je jeho řízení. Projektové řízení můžeme chápat jako kontinuální proces organizování, jehož výsledkem je dosažení vytyčených výstupů projektu. Těch dosáhneme použitím postupů, pravidel a nástrojů, díky kterým povedeme náš projektový tým, při dodržení stanoveného rozpočtu zdrojů (hmotné, nehmotné, finanční a časové). Toto řízení je realizováno projektovým manažerem. (Doležal, a další, 2017)

Nyní se zaměříme na projekt z hlediska participace osob. Na projektu se dle jeho velikosti podílí různé množství lidí, nicméně každý projekt má následující účastníky:

- sponzor
- zákazník
- manažer

(Doležal, a další, 2017)

Ve skutečnosti osob spojených se projektem je mnohem více. Tyto osoby budeme později nazývat stakeholdery, a jejich výčet a určení si rozebereme v jiné kapitole.

Sponzor je taková osoba, která ovlivňuje směřování projektu jako takového. Tato osoba poskytuje zdroje, které jsou následně užívány pro chod projektu. (Doležal, a další, 2017)

Zákazník jako osoba představuje toho, kvůli komu se projekt realizuje. Definuje požadavky na výstup projektu, v našem případě tedy na software. Je zde možnost, že zákazník je tou samou osobou, jako sponzor. (Doležal, a další, 2017)

Projektový manažer je klíčová osoba. Je pověřena řízením celého projektu. Pojem řízení projektu jsme si definovali v minulé podkapitole, proto jej zde nebudeme více rozebírat. (Doležal, a další, 2017)

Projekt v průběhu svého života prochází nejrůznějšími etapami. Každá je specifická svými potřebami, zdroji a dalšími náležitostmi. Obecně rozeznáváme tyto fáze:

- předprojektová
- zahájení
- plánování
- realizace
- ukončení

(Doležal, a další, 2017)

Předprojektová fáze je prvotní fází, kdy samotný projekt ještě není v běhu. V této fázi dochází především k teoretické činnosti, respektive k vytvoření základní obecné koncepce projektu (určení popisu, cíle, přínosů, omezení zdrojů a dalších parametrů). Tyto údaje tvoří dokument zvaný Business Case a zdůvodňuje nám, proč by bylo vhodné tento projekt realizovat. Na základě tohoto dokumentu se rozhoduje, zda se projekt zahájí či nikoliv. Pokud tuto fázi vztáhneme na projekt softwarového vývoje, můžeme hovořit o tom, zdali je vývoj softwaru výhodný či nezbytný. (Doležal, a další, 2017)

Fáze zahájení následuje, pokud byla myšlenka projektu přijata. V rámci toho kroku je přiřazena role projektového manažera, sponzora, zákazníka a založen dokument zvaný „zakládací listina projektu“. V této listině jsou uvedeny zmíněné osoby projektu. Po jejím schválení je projekt oficiálně spuštěn. Opět pokud se budeme bavit o naprogramování systému, jedná se o okamžik, kdy se zástupce vývojáře aplikace dohodne se zákazníkem. (Doležal, a další, 2017)

Po fázi projektové následuje plánování. V tomto kroku dochází k podrobnému určení časových, finančních a zdrojových omezení. Rovněž upřesníme způsob realizace a kontrolních mechanismů. V případě softwarové aplikace můžeme hovořit správě požadavků. Tento proces bude podrobně rozebrán v rámci této práce. (Doležal, a další, 2017)

Předposlední fáze čili realizace, je samotný jádrem projektu. Projektový manažer koordinuje projektový tým, komunikuje se stakeholdery, a to vše tak, aby došlo k naplnění cílů projektu. Na konci této fáze předáváme výsledek projektu. V našem případě je to tedy samotná práce na požadovaném výstupu. (Doležal, a další, 2017)

Ukončení je zastřešení naší výsledné práce na daném projektu. Následuje už pouze vyhodnocení projektu, které by nám mělo pomoci při realizaci projektů následných. (Doležal, a další, 2017)

2 Finanční analýza

Jedna z velmi důležitých otázek každého projektu je, jaké a jak velké prostředky budeme muset vynaložit, abychom dosáhli požadovaného cíle. Dále nastává otázka, za jakou dobu od ukončení projektu se nám investované prostředky vrátí. Možností, jak se k tomuto závěru dopracovat, jsou ukazatele, kterých máme na výběr širokou paletu. Pro potřeby našeho malého projektu nebudeme potřebovat tolik komplexní ukazatele, jelikož našich vstupů bude minimum.

2.1 Doba návratnosti investice

Ukazatel doby návratnosti, označovaný jako „ DN “, nám říká, za jak velké časové období bude suma příjmů z investice „ P_n “ větší, než suma všech výdajů pro realizaci projektu „ I “. Rovněž musíme uvažovat vývoj hodnoty peněz v čase, proto přidáváme diskontní úrokovou míru „ i “, která reprezentuje ponížení hodnoty peněz v čase. Vzorec je poté následující:

$$I = \sum_{n=1}^{DN} \frac{P_n}{(1+i)^n}$$

- I suma výdajů
- DN doba návratnosti
- n doba životnosti projektu
- P_n Peněžní příjem za období n
- i diskontní úroková sazba

(Máče, 2006)

2.2 Posouzení investice

Určení, zdali se vyplatí investovat do daného projektu finanční prostředky, 200je možné opět více způsoby. Jedna z možností může být zakomponování doby návratnosti investice, a to sice v případě, kdy investice generuje zisk. To spočívá v porovnání s ostatními možnostmi, z hlediska časové návratnosti investovaných prostředků. (Máče, 2006)

Rok	0	1	2	3	4	5	6	7
Investiční výdaj	600	400	–	–	–	–	–	–
Peněžní příjem A	0	200	320	440	440	380	80	0
Peněžní příjem B	0	660	320	180	100	50	30	0

Obrázek 1 – tabulka investic - (Máče, 2006)

V tabulce máme jako příklad dvě investice, které mají shodnou dobu životnosti a množství investovaného kapitálu. Co však nemají stejné, je peněžní příjem za období n . Nyní pojďme získat informace z dat, které nám tabulka poskytuje. Z tabulky vyplývají následující informace (kde p.j. jsou peněžní jednotky): (Máče, 2006)

- Celkový peněžní příjem z A 1 860 p.j.
- Celková výše investice A 1 000 p.j.
- Celkový zisk z investice A 860 p.j.
- Doba návratnosti investice A zhruba 3 roky a 1 měsíc

- Celkový peněžní příjem z B 1 340 p.j.
- Celková výše investice B 1 000 p.j.
- Celkový zisk z investice B 340 p.j.
- Doba návratnosti investice B zhruba 2 roky a 1 měsíc

Z výše uvedeného vyplývají dva závěry. Byť je z hlediska času investice „B“ výhodnější, tak už nikoliv z hlediska celkového zisku z investice, kde investice „A“ přinese větší prostředky. Nicméně jsme neuvažovali diskontní úrokovou sazbu, která by výsledná čísla ponížila. Se získanými informacemi souvisí tři pojmy, a to sice:

- likvidita
- výnosnost
- riziko

(Máče, 2006)



Obrázek 2 – investiční trojúhelník - (Máče, 2006)

Obrázek výše popisuje vztah mezi těmito 3 parametry. Pokud budeme chtít minimalizovat (levý dolní vrchol), tak sice to na úkor vysoké časové likvidity a nízké výnosnosti. Stejně tak bychom mohli popsat zbylé případy. Je tedy evidentní, že pokud chceme minimalizovat, nebo maximalizovat jeden z těchto ukazatelů, tak se to projeví na parametrech zbylých. (Máče, 2006) Nyní si popíšme tyto ukazatele.

Likvidita je takový ukazatel, který nám říká, za jak dlouho jsme schopni dostat svým závazkům, který v rámci investice bereme jako investované prostředky. (Máče, 2006)

Výnosnost nám hovoří o sumě získaných prostředků z realizace daného projektu. Může nabývat kladných, tak i záporných hodnot. (Máče, 2006)

Poslední faktor je riziko. Riziko nám představuje tu možnost, že projekt, tak jak je plánovaný, nebude uskutečněn. To znamená, že může dojít k navýšení investovaných prostředků, snížení peněžních příjmů či snížení doby životnosti investice. Všechny výše popsané situace mají negativní dopad na investici, respektive její výnosnost. Riziko si můžeme stanovit jako poměr mezi likviditou a výnosností. Riziko jsme mohli vidět ve vzorci pro výpočet doby návratnosti, kde je reprezentováno jako diskontní míra. Pokud tedy budeme uvažovat investici s vysokou mírou rizika, budeme požadovat vyšší výnosnost za podstoupení tohoto rizika. (Máče, 2006) Faktory, které nám rovněž mohou navýšit míru rizika jsou například:

- doba návratnosti
- odvětví
- trh
- legislativa
- další

Naším cílem pro co největší potlačení rizika spojené s investicí do našeho projektu, je minimalizace doby návratnosti, jejíž výpočet jsme nastínili výše. To tedy znamená, že čím dříve se nám navrátí investované peníze, tím je investice bezpečnější.

Odvětví společně s trhem, kde bude investice provedena, může významně ovlivnit rizikovost naší investice. V rámci odvětví se může jednat o hledání příležitosti na trhu. Rizikovým trhem může být například nestabilní země či oblasti světa, kde hrozí například přírodní katastrofy.

Legislativou je myšlena změna právního stavu, kdy se můžeme dostat do situace, že naše počínání bude protizákonné. Tudíž nestabilní politická situace či jiné účelové jednání může znamenat riziko zastavení našich záměrů v rámci našeho projektu.

Co když ale naše investice přímo negeneruje reálný zisk a je vyložena investicí, která má přinést pouze větší užitek. V takové případě není vhodné použít dobu návratnosti investice, jelikož by vždy vyšla záporně, ale použijeme raději „NVP“ (net present value), kterou můžeme vyjádřit následovně: (Máče, 2006)

$$NPV = \sum_{t=0}^n \frac{CF_t}{(1+r)^t}$$

- NPV čistá současná hodnota
- CF_t peněžní tok v roce t
- n doba trvání projektu
- r diskontní úroková míra
- t vybraný rok

Můžeme vidět, že oproti době návratnosti investice došlo ve vzorci k drobným úpravám. Tento ukazatel vyjadřuje množství peněz, které projekt za svou živostnost vygeneruje, nebo naopak zkonzumuje. Metoda NPV se používá pro krátkodobé investiční záměry do 10 let životnosti a je vhodná na porovnání více investičních možností. Využít tuto metodu můžeme při porovnání peněžních toků více variant projektu. (Máče, 2006).

2.3 Náklady projektu

Náklady neboli počáteční investice do našeho projektu představuje velmi významný parametr, zdali projekt vůbec započneme, nebo zvolíme jiný způsob řešení. Musíme však stanovit, co mezi tyto náklady řadíme, abychom je byli schopni při přípravě projektu kvantifikovat. Prvotní rozdělení nákladů v rámci našeho projektu můžeme pojmut následovně:

- náklady realizace projektu
- provozní náklady
- náklady na údržbu

(Máče, 2006)

Náklady na realizaci projektu jsou spojené přímo s prvotními investicemi, jejichž výsledkem je zamýšlený cíl, či výstupy. Výše jsme tuto položku zmiňovali jako „I“ - suma výdajů. Do těchto nákladů můžeme v našem případě zahrnout tedy veškeré náklady spojené s vývojem softwaru. Jmenovitě poté osobní náklady, náklady na pořízení potřebných prostředků, poradenství, licencí pro práci s jiným softwarem a další. (Máče, 2006)

Provozní náklady jsou takové náklady, které jsou generované provozem dokončeného projektu. Tyto náklady jsme výše neviděli sice přímo, nicméně jsou zahrnuté v peněžním příjmu za vybrané období, kde jsou tyto náklady odečteny od tržeb realizovaných v daném období. Mohou to být například měsíční poplatky za správu databází či dohodnutá fixní částka za určité období za užívání softwaru. (Máče, 2006)

Náklady na údržbu můžeme považovat za takové náklady, které jsou spojené s aktualizacemi softwaru. V případě, že bychom se bavili o upgradu, tak zde ve většině případů nelze hovořit o nákladech na údržbu, nýbrž o migraci čili novém projektu. (Máče, 2006)

3 Vývoj aplikace

Vývoj softwaru se může pro některé jevit jako nepochopitelná záležitost, nicméně se tato oblast tolik neliší například oproti zakázkové výrobě bot. Zákazník si objedná produkt, na kterém specifikuje svoje požadavky. Například typ bot, jejich barvu, materiál a mnoho dalších. Na konci tohoto procesu si odnáší své nové boty, pokud jsou takové, jaké předem specifikovat. Těžko si lze představit, že při objednání společenských polobotek se zákazník spokojí s holinami. Ne jinak tomu je při objednání a vývoji softwaru. Abychom jsme se vyhnuli podobným nehodám, využíváme moderních prostředků, které nám s celým procesem vývoje pomáhají. Jedním z nich je například UML.

3.1 Vysvětlení UML

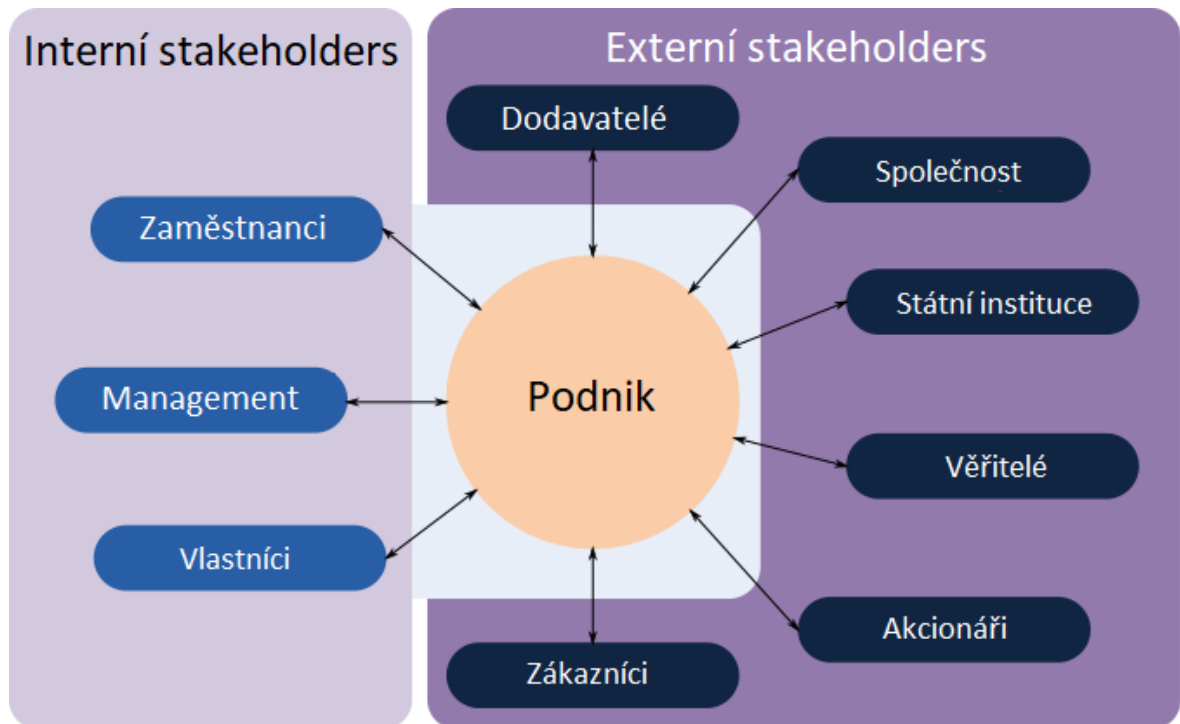
Značná část této práce se bude opírat o poznatky z modelovacího jazyka UML, a proto je vhodné uvést, o jakou problematiku se jedná. UML (Unified modeling language) je jednotný modelovací jazyk, jehož cílem je objektově orientovaná analýza a jeho využití je globální. Jednodušeji řečeno se jedná o grafické a textové vyjádření stavby projektu v případě návrhu softwarového systému. Pro tuto vizualizaci je využíván nejrůznější grafický software, který může být zdarma, či si za jeho použití zaplatíme. (Kanisová, a další, 2012) UML je tedy pro nás klíčem, pro takový návrh aplikace, který bude srozumitelný a bude splňovat požadavky našich stakeholderů. O stakeholderech se dočtete v další kapitole.

3.2 Identifikace Stakeholderů

Stakeholderi neboli zainteresované strany jsou osoby, které mají nějakým způsobem zájem na realizaci projektu a jsou to právě osoby, které případně mohou vznášet požadavky na funkce softwaru. (Doležal, a další, 2017) Stakeholdery můžeme dělit podle jejich příslušnosti a dle situace, ve které se nacházíme. Například pokud budeme hovořit o projektu na vytvoření e-shopu v určité společnosti, máme zde interní a externí stakeholdery. Řízení vztahů a výběr těchto zainteresovaných osob je kompetencí projektového manažera, který by měl disponovat znalostmi pro výběr těchto osob dle náplně a cílů projektu. Nedostatečné zapojení či špatné určení stakeholderů může vést k vytvoření takového softwaru, který uživatelům nebude vyhovovat. (Kanisová, a další, 2012) Mezi zdroje požadavků (stakeholdery) můžeme tedy zařadit:

- zadavatel projektu
- management organizace
- finální uživatelé
- hmotné zdroje, na nichž bude systém provozován
- legislativa
- zákazníci

- konkurenční společnosti
- (Arlow, a další, 2007)



Obrázek 3 – schéma stakeholderů - (What is a Stakeholder?, 2017)

Po nalezení všech našich zdrojů požadavků se přesouváme na získávání požadavků stakeholderů a činnosti s tím spojené.

3.3 Správa požadavků

Stanovení stakeholderů v přechozí podkapitole sloužilo především k tomu, abychom získali zdroje požadavků na vývoj našeho softwaru. Nyní se dostáváme k požadavkům jako takovým. Celý proces získávání požadavků se nazývá správa požadavků. Tento proces můžeme rozdělit do těchto dílčích částí:

- sběr dat/požadavků
- analýza dat/požadavků
- řízení požadavků

(Kanisová, a další, 2012)

Tato předvývojová část je z pohledu funkčnosti softwaru jednou z nejdůležitějších částí a je potřeba jí věnovat značné úsilí. Dobře fungující správou požadavků získáme veškeré požadavky před vytvářením samotného softwaru, což nám umožňuje:

- porovnání stávajícího řešení na trhu
- implementaci softwaru dle představ stakeholderů
- stanovení doby implementace
- zpřesnění kalkulace nákladů

(Kanisová, a další, 2012)

Špatná správa požadavků či špatné stanovení stakeholderů přinese pouze negativní důsledky mezi něž můžeme zařadit tyto:

- výběr nevhodného řešení projektu
- implementaci odlišného softwaru, než stakeholdeři požadují
- neplnění časového plánu
- navyšování nákladů projektu

(Kanisová, a další, 2012)

Nyní k definici požadavku. Požadavek je specifikace nebo popis určité funkce či možnosti softwaru, na nějž vzniknul požadavek na implementaci ve výsledném produktu. Požadavky dělíme na dva základní typy:

- funkční
- nefunkční

(Arlow, a další, 2007)

Funkční požadavek se vyznačuje tím, že určuje požadavky na funkčnost systému jako takového. Příklad může být funkce na tvorbu zákazníků v databázi.

(Arlow, a další, 2007)

Nefunkční požadavek poté upřesňuje požadavky funkční. Vytvoření záznamu zákazníka v databázi nesmí trvat déle, než 5 vteřin. To mohl být požadavek nefunkční. (Arlow, a další, 2007)

Zdroje těchto požadavků jsme specifikovali výše. Nyní tento seznam doplníme ještě o dvě položky. Zdroj stakeholderi byl již specifikován výše, proto se jim nyní už nebudeme zabírat, co se týká specifikace. Zdroje požadavků jsou tedy:

- stakeholderi
- naše znalosti problematiky
- prostředí zákazníka

(Kanisová, a další, 2012)

Pod pojmem znalosti problematiky chápeme dosavadní získané zkušenosti v daném oboru, které mohou odhalit případné nedostatky ostatních zdrojů požadavků.

Prostředí zákazníka je poté výchozí situace osoby, pro kterou je vývoj softwaru realizován. Přesněji poté současné hardwarové a softwarové podmínky.

(Kanisová, a další, 2012)

Nyní je potřebné, abychom určili, jak správně provádět sběr požadavků. Tento proces se liší projekt od projektu, nicméně získaný požadavek by měl ve všech případech splňovat určité parametry. Dle metodiky UML má požadavek jasně říkat, co má systém nabízet, nikoliv jak toho dosáhnout. (Kanisová, a další, 2012) Vznesený

požadavek by měl být tedy co nejvíce kvantifikovatelný. Pro formulaci požadavku můžeme využít potenciál metody SMART, která spočívá v určení takového požadavku/cíle, který je:

- specific
- measureable
- achievable
- realistic
- timely

(Keller, a další, 2007)

Prvním bodem přeloženo z angličtiny je specifičnost daného požadavku. To znamená nejlépe kvantifikovat dané cíle. Příkladem může být vytvoření e-shopu, který bude zaručeně kompatibilní s určitým webovým prohlížečem. (Keller, a další, 2007)

Dalším bodem je měřitelnost, která je úzce spjatá s předchozím bodem. Přesněji řečeno jde o stanovení metodiky, pomocí níž budeme ověřovat plnění cíle. Jako příklad může posloužit zvolení metody pro sledování počtu zařízení, která využívají určitý druh softwaru. (Keller, a další, 2007)

Třetím bodem potom můžeme rozumět dosažitelnost nebo rovněž akceptovatelnost. Je potřeba určit takový požadavek, který je v rámci projektu a zdrojích přijatelný. Jako příklad uvedu dobu vývoje aplikace, která musí být v souladu s možnostmi vývojáře. (Keller, a další, 2007)

Následujícím bodem rozumějme realističnost požadavku. Každý projekt má svá omezení a požadavek by neměl toto omezení přesahovat. Opět příkladem můžeme uvést funkce aplikace, které musí být vývojář schopen korektně vyvinout. (Keller, a další, 2007)

Poslední bodem je poté stanovení délky realizace určitého požadavku. Tento bod zajišťuje časové vymezení pro realizaci daného požadavku. Opět volba tohoto bodu by měla korespondovat s body předcházejícími. Jako vždy uvedu příklad, který by mohl být například zprovoznění databáze do určitého data. (Keller, a další, 2007)

Dodržování všech výše popsanych principů a metod snižuje riziko, že náš projekt bude neúspěšným z důvodu špatného sběru požadavků. Požadavky takto získané jsou reálné, měřitelné, časově vymezené a dosažitelné. Soubor všech získaných požadavků nám poté tvoří cíle projektu. (Arlow, a další, 2007) Nyní zbývá pouze určit, jakým způsobem, respektive metodami budeme tyto požadavky zdrojů určovat.

3.4 Sběr požadavků

Pro úspěšné navržení softwarové aplikace je klíčové zvolit správnou metodu sběru požadavků. Volba této metody může ovlivnit správnost a úplnost sběru požadavků, a to sice v závislosti na výchozí situaci, v níž se projekt nachází. Než začneme popisovat jednotlivé metody získávání dat, objasněme si nejprve základní pojmy pro zjednodušení následné práce. Prvně si rozvedme, jaký rozdíl mezi daty a informacemi. (Kanisová, a další, 2012)

Data jsou jakékoliv záznamy, které byly vytvořeny specifickým způsobem. Jejich forma může být papírová či elektronická. Data jsou tedy například záznamy z dotazníků, verbální vyjádření, obrázky, zvukové stopy či video. Data si ještě rozdělíme dle jejich původ:

- primární
- sekundární

(Tahal, a další, 2017)

Primární data jsou veškerá data, která neexistovala před zahájením projektu a jejich pořízení je za účelem zkoumané problematiky. Jakékoliv použití takto pořízených dat, mění tato data na data sekundární. Výhodou takto pořízených dat je jejich aktuálnost. Opačnou stranou mince je časová a eventuálně finanční náročnost, která může být vyšší než využití dat sekundárních. (Tahal, a další, 2017)

Sekundární data jsou taková data, která existovala již před zahájením našeho výzkumu. Můžeme je získávat z těchto zdrojů:

- interní
- externí

(Tahal, a další, 2017)

Interní zdroje poskytují taková data, která nejsou veřejně přístupná pro osoby mimo daný zdroj dat. Příkladem může být informace o odběru zboží jednotlivých partnerů. (Tahal, a další, 2017)

Externí zdroje jsou veřejně dostupné, což znamená že je nalezneme i mimo daný interní zdroj. Například tedy výroční zprávy. (Tahal, a další, 2017)

Informací rozumějme výstup, který jsme získali analýzou pořízených dat. Příkladem může být získaný průměrný věk respondenta dotazníku, který jsme vypočítali na základě získaných dat. (Tahal, a další, 2017)

Zabývejme se nyní, jakými metodami můžeme od našich stakeholderů sběr dat provést. Mezi dostupné metody pro sběr požadavků dle použité literatury můžeme zařadit dotazníky a konzultace. (Arlow, a další, 2007) My si však tento seznam rozšíříme ještě o následující metody, které si následně popíšeme:

- dotazník/anketa
- rozhovor/konzultace
- pozorování

- kombinace

(Tahal, a další, 2017)

První metodou v našem seznamu je tvorba dotazníku. Tento způsob získávání dat se řadí mezi nejčastěji používané metody. Tato metoda sběru dat je vhodná tehdy, pokud náš výběrový soubor je rozsáhlý, ne dobře dosažitelný a nemáme s respondenty osobní vztah. Struktura, velikost a koncepce dotazníku je rovněž stanovena dle plánovaných respondentů. Každý dotazník by měl splňovat určitá pravidla. Tato pravidla například říkají, že se máme ptát pouze na ty otázky, které jsou v rámci našeho sběru. Dále otázky musí být naším respondentům srozumitelné a délka dotazníku by se měla pohybovat do 25 minut na vyplnění. Výše na výčtu metod jsme na stejné pozici mohli vidět i anketu. Anketa je rovněž tvořena otázkami, na které respondent poskytuje odpovědi, nicméně se zpravidla skládá z méně otázek (obvykle jedna otázka) a je volbou respondenta, zdali anketu vyplní či nikoliv. S anketou se můžeme nejčastěji setkat na internetových stránkách. Dnes můžeme vidět využití ankety často v obchodních řetězcích, které pomocí hodnotících pultů monitorují spokojenost svých zákazníků. Tyto řetězce umísťují hodnotící pulty k východům svých provozoven a zákazníci mohou v případě zájmu využít možnost hodnotit, jak byli dnes s nabízenými službami/zbožím spokojeni. (Tahal, a další, 2017) (Arlow, a další, 2007)

Vraťme se zpět k dotazníku. V něm můžeme narazit na dva typy otázek:

- uzavřené
- otevřené

(Tahal, a další, 2017)

Otázky uzavřené jsou takové otázky, kde respondent vybírá již z předem vytvořených odpovědí. Výhodou těchto otázek je jednoznačnost odpovědí a následné vyhodnocení dat z nich plynoucí. Nevýhodou může být neznalost problematiky respondenta, což může vést nepřesnému vyplnění. Rovněž je obtížnější správně naformulovat otázky a odpovědi tohoto druhu otázek. Mezi otázky uzavřené můžeme zařadit:

- dichotomické/trichotomické
- více variant
- likertova škála
- sémantický diferenciál
- stupnice důležitosti
- stupnice známek
- stupnice úmyslu něco koupit

(Keller, a další, 2007)

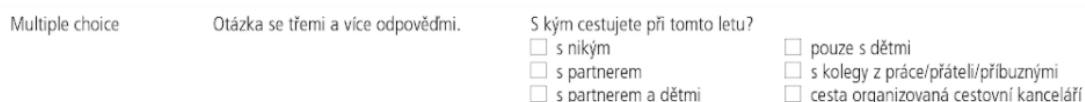
Pro naše potřeby zajištění sběru požadavků stakeholderů pro vývoj funkčního softwaru využijeme pouze určité výše zmíněné typy uzavřených otázek, kde můžeme lehce ověřit naplnění metody SMART. Z této kategorie si tedy vybereme dichotomické/trichotomické, více variant.

Otázky dichotomické jsou strukturovány tak, že respondent může na položenou otázku odpovědět „ano“ či „ne“. Může nastat situace, kdy dotazovaný nemůže zvolit ani jednu zmíněnou variantu, protože nepochopil kontext otázky či jednoduše nezná odpověď, což by vedlo ke zvýšení chybovosti. Tento problém lze například řešit použitím trichotomické struktury odpovědí, kdy mezi odpovědi „ano“ a „ne“, přidáme navíc možnost „nevím“. Jednoduchou otázkou dichotomického typu může být jako na obrázku níže. Na tuto otázku nelze odpovědět jinak, než výše popsaným způsobem a pro zadávacího z toho plyne jednoznačná informace o daném respondentovi. Nicméně je zde nutný předpoklad, že tato otázka bude položena jenom relevantním respondentům. (Keller, a další, 2007)



Obrázek 4 – dichotomické otázky - (Keller, a další, 2007)

Druhým typem strukturalizace uzavřené otázky pro potřeby vývoje funkčního softwaru, je možnost výběru z více variant. Tato struktura je vhodná tehdy, pokud víme, že odpovědi našich respondentů se budou výrazně lišit a zároveň známe položky, mezi kterými budou jejich odpovědi oscilovat. Je však důležité zvolit, jestli použijeme možnost výběru pouze jedné odpovědi, či připustíme odpovědi více. V případě volby pouze jedné možnosti odpovědi bývá obvyklé užití koleček, v druhém případě čtverečků. Ukázka níže připouští tedy více odpovědi. (Keller, a další, 2007)



Obrázek 5 – výběr z více možností - (Keller, a další, 2007)

Otázky otevřené jsou oproti uzavřeným otázkám jednodušší na tvorbu, a můžeme pomocí nich získat takové informace, které by nás běžně nemuseli nenapadnout. Nevýhodou je možnost respondenta odpovědět libovolně, a to sice ve smyslu následného zpracování dat, které je obtížnější. Další nepříjemností může být fakt, že respondent neodpoví na danou otázku, či případ, kdy to z jeho odpovědi není možné kvalifikovat. U otázek otevřených máme možnosti struktury otázek dotazníku následující:

- zcela nestrukturované
- slovní asociace
- dokončení věty
- dokončení příběhu
- obrázek
- test tematické apercepce

(Keller, a další, 2007)

Znovu pro naše potřeby nejsou veškeré výše uvedené možnosti validní, a proto budeme uvažovat pouze první uvedenou možnost stavby otázky. Otázka zcela nestrukturovaná bývá nejčastějším reprezentantem otázek otevřených, přičemž je potřeba korektně otázku formulovat, aby splňovala veškeré vlastnosti, které jsme si popsali v minulém odstavci. Problémem stále zůstává názorová diverzita, která zde může nastat, jako například u otázky níže. (Keller, a další, 2007)

Po dotazníku přecházíme k rozhovoru. Druhy rozhorů jsou následující:

- strukturovaný

Zcela nestrukturované	Otázka, na niž může respondent odpovědět mnoha různými způsoby.	Jaký máte názor na společnost American Airlines?
-----------------------	---	--

Obrázek 6 – zcela nestrukturovaná otázka - (Keller, a další, 2007)

- polostrukturovaný
- nestrukturovaný

(Švaříček, a další, 2007)

Strukturovaný rozhovor je takový rozhovor, kde se máme předem na definovaná témata a otázky a striktně drží jejich pořadí a strukturu. (Švaříček, a další, 2007)

Polostrukturovaný rozhovor je podobný jako strukturovaný, nicméně oproti němu je připuštěno mírné odklonění od tématu či nedodržení pořadí otázek. (Švaříček, a další, 2007)

Nestrukturovaným rozhovorem je poté takový rozhovor, kdy je respondentovi položena otázka a na základě jeho vyjádření mu jsou tazatelem pokládány otázky doplňující, či rozšiřující. (Švaříček, a další, 2007)

Strukturu daného rozhovoru volíme s ohledem na cíl získat požadavky na náš software. Proto je tedy nezbytné se na daný dialog připravit. První část přípravy spočívá ve studiu prostředí daného respondenta. Lépe řečeno zajistit si informace a spojitosti diskutovaného tématu. Dalším bodem je příprava otázek a způsob, jakým je dotazovanému položí. Zanedbaní této přípravy může vést k tomu, že tazatel ztratí zájem rozhovor vést, či bude odpovídat na jinou otázku, což můžeme vidět na příkladu níže. (Švaříček, a další, 2007) Příprava otázek a způsob jejich položení pro účel sběr požadavků je tedy stěžejní. Otázky dle použité literatury dělíme na níže uvedené:

- úvodní
- hlavní
- navazující
- ukončovací

(Švaříček, a další, 2007)

Úvodní otázky, nebo úvod jako takový, slouží k uvedení dotazovaného do prováděného výzkumu. Těmito otázkami bychom měli navodit v respondentovi myšlenkový tok spojený s výzkumem a chuť, aby na naše otázky korektně odpovídal. (Švaříček, a další, 2007)

Hlavní otázky tvoří nejpodstatnější část rozhovoru a jsou to tedy otázky přímo spojené s prováděným výzkumem, v našem případě sběru požadavků. Úkolem tazatele je formulovat otázky tak, aby docházelo k relevantnímu sběru dat, ze kterých vytvoříme následně požadavky. Opět v tomto případě není nic proti ničemu využít metody SMART, kdy formulace otázek a jejich následných odpovědí naplňují body této metody. V tomto bodě získáváme většinou požadavky funkční. (Švaříček, a další, 2007)

Navazující otázky jsou otázkami rozšiřujícími pro otázky hlavní. Podstatné je zde získat potřebné detaily pro konkretizaci daného požadavku, či jeho lepší vysvětlení. Vlastně tedy můžeme hovořit o nefunkčních požadavcích. (Švaříček, a další, 2007)

Otázky ukončovací slouží k rekapitulaci získaných dat a následnému ukončení rozhovoru což je ukončení sběru dat. (Švaříček, a další, 2007)

Tazatel: „A teď něco jiného. Můžeš o nějakém žákovi ve třídě říci, že má malou, nebo velkou představivost?“

Učitelka Eva: „Prostorovou představivost, nebo fantazii? Ve vztahu k příběhům, nebo k matematickým veličinám? Geometrická nebo mapy?“

T: „Co se týče například řešení problémů...“

Eva: „Takže sociální jednání...“

T: „Co může být třeba důležité při učení, samostatném učení...“

Eva: „...ano, jsou děti, které se samostatně učí, které se učí strukturovaněji, ty, co se dobře odhadnou.“

Obrázek 7 – ukázka rozhovoru bez přípravy - (Švaříček, a další, 2007)

Pozorování je metoda, která dle literatury je metodou nejnáročnější. Zároveň nelze použít tuto metodu ve všech případech výzkumu. Metoda spočívá v kontinuálním sledování aktivit, procesů, které se v cílovém prostředí odehrávají. V průběhu tohoto sledování dochází k souvislému pořizování dat, která jsou následně transformována v požadavky. Pozorování jako každou metodu můžeme provádět více způsoby. Prvním dělením může být následující:

- přímé
- nepřímé

(Švaříček, a další, 2007)

Přímé pozorování spočívá v přímé účasti pozorovatele analyzovaného prostředí či procesu. Naopak nepřímé pozorování probíhá pomocí vyhodnocení proběhlých činností z dostupných záznamů. (Švaříček, a další, 2007)

Dalším dělením pro potřeby našeho výzkumu je takové pozorování:

- strukturované

- nestrukturované
(Švaříček, a další, 2007)

Strukturované pozorování se vyznačuje tím, že hledáme odpovědi na již předem určené otázky, zatímco pomocí nestrukturované pozorování postupně upřesňujeme určitou hypotézu. (Švaříček, a další, 2007)

Pro úspěšné pozorování za účelem sběru dat je opět nezbytné provést přípravu pozorovatele. Před zahájením procesu samotného, pozorovatel musí chápat kontext vykonávaných činností, které jsou předmětem pozorování a rovněž stanovit jeho cíle, opět s použitím metody SMART. Od osoby která výzkum provádí, se očekává, že bude pořizovat mimo hlavního sběru dat rovněž terénní poznámky, které můžou následně pomoci upřesnit požadavky jako takové. Rovněž tyto poznámky slouží k lepšímu vybavení provedeného výzkumu. (Švaříček, a další, 2007)

Kombinací zmíněných metod můžeme dosáhnout lepších výsledků ve sběru požadavků. Znovu je klíčové volit kombinaci těchto metod dle okolností. Mezi ně jsme výše zařadili množství respondentů, osobní vztah. Například lze použít dotazník s navazujícím rozhovorem, kde můžeme dialogem s respondentem upřesnit jeho odpovědi. Opět si připomeňme že volba jakékoliv metody by měla vycházet z našich zkušeností, počet a typ stakeholderů, vzájemné osobní vazby. (Švaříček, a další, 2007)

3.5 Analýza sběru dat

Po provedeném sběru dat zbývá provést analýzu těchto dat. Pro tuto analýzu využijeme našich znalostí, ale především nástrojů CASE, BPMN, případy užití a diagram případu užití. Analýza dat je především v naší kompetenci jako vývojářů softwaru. To znamená, že bychom měli disponovat know-how pro provedení tohoto kroku. Úkolem je tedy zhodnotit, jestli a jak lze převést požadavky stakeholderů ve skutečnost. K dosažení vytyčeného cíle nám slouží nástroje, které si níže rozebereme. Dle použité literatury je postup po sběru požadavků následující:

- stanovení funkčních požadavků
- stanovení nefunkčních požadavků
- stanovení případů užití a jejich propojení s funkčními požadavky
- implementace nefunkčních požadavků do systému

(Kanisová, a další, 2012)

První je tedy stanovení funkčních požadavků. Po jejich stanovení můžeme provést analýzu případu užití. Nejprve si však požadavky zaneseme do elektronické podoby, a to sice nástroje CASE. CASE je takový nástroj, kdy do softwaru pro práci s UML zaneseme jednotlivé části našeho projektu, jako Business model (o něm budeme hovořit později), požadavky, případy užití a další. V tomto případě hovoříme tedy o požadavcích, které tedy zaneseme včetně jejich názvu a popis daného požadavku. Je doporučeno každému požadavku přiřadit pořadové číslo, pro následné zjednodušení prací. Ukázku tohoto zanesení pomocí nástroje case můžeme vidět na obrázcích níže (Kanisová, a další, 2012).

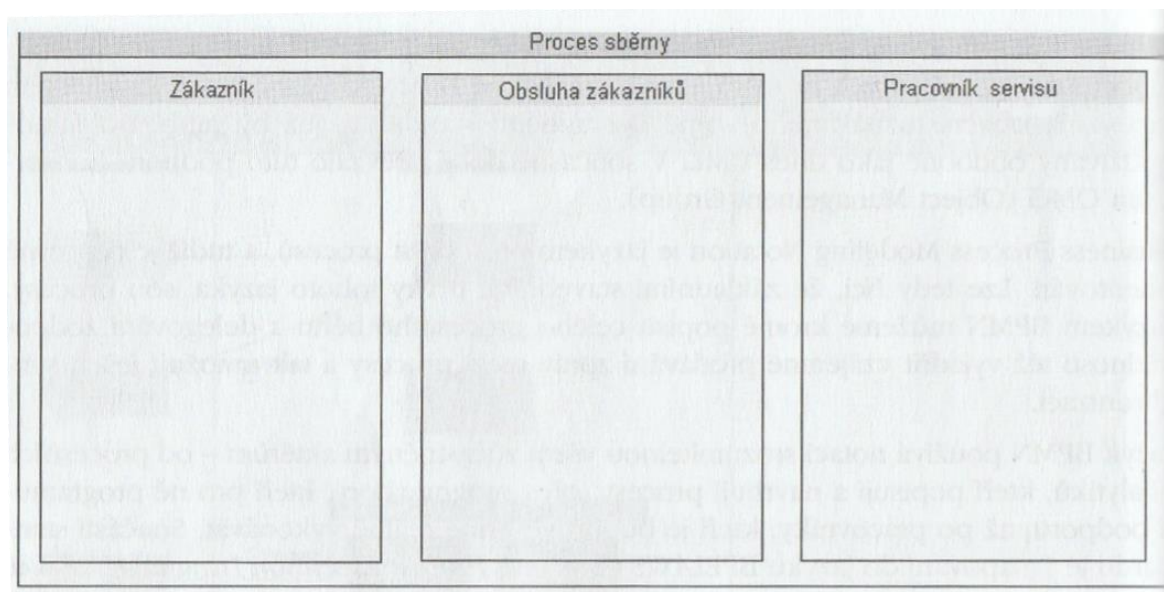


Obrázek 8 – ukázka požadavků zanesených do softwaru - (Kanisová, a další, 2012)

Požadavky si rovněž zaneseme dle notace BPMN. Jedná se o grafické znázornění procesů, které je globálně uznávaným a univerzálním postupem při návrhu podnikových procesů, respektive našich požadavků na aplikaci. Výhodou tohoto jazyka je jeho rozšířenost, srozumitelnost a možnost lehce definovat pravidla, podmínky, vstupů či výstupů. Toto grafické řešení obsahuje níže uvedené prvky:

- bazén
- události
- činnosti
- toky
- brány
- artefakty

Bazén je vymezení části grafického řešení pro procesy, lépe řečeno aktéry. Je vhodné pro každého aktéra volit jeden bazén. Příkladem bazénu je zákazník, obsluha zákazníků a pracovník sběry, jako na obrázku níže.



Obrázek 9 – bazény - (Kanisová, a další, 2012)

Událost je událostí v námi vymezeném systému, která ovlivňuje chod procesů, přičemž je dělíme na:

- počáteční
- průběžná
- koncová

(Kanisová, a další, 2012)

Počáteční událost zahajuje proces, přičemž musí být alespoň jedna. Průběžná událost slouží k propojení procesů mezi sebou a událost koncová ukončuje řetězec procesů. Opět musí být alespoň jedna. (Kanisová, a další, 2012)



Obrázek 10 – typy událostí - (Kanisová, a další, 2012)

Činnosti jsou zástupné zobrazení úkolů, který mají být provedeny. Máme činnosti dílčí a složené. Dílčí je úkol jako takový. Složená činnost obsahuje činnost navíc, která může být buď spojena s daným úkolem, či kompletně nezávislá.

(Kanisová, a další, 2012)

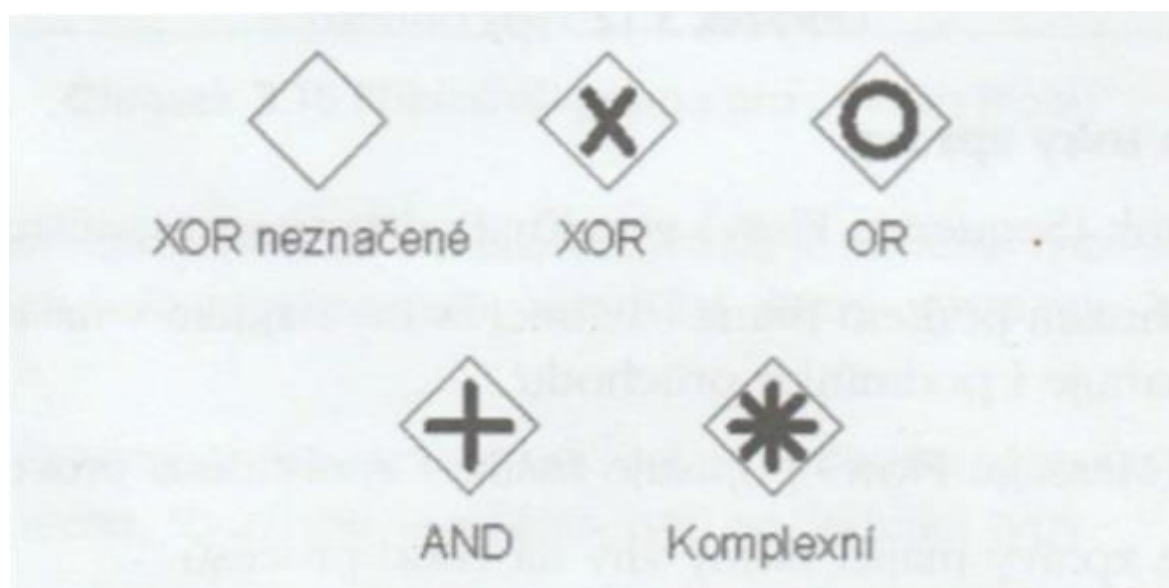
Toků rozeznáváme 2 druhy a sice buď toky zpráv či sekvenční toky. Sekvenční toky vyjadřují vazby mezi prvky grafického znázornění. Toky zpráv poté předávání informací mezi jednotlivými procesy. (Kanisová, a další, 2012)

Brány nám zastupují rozhodovací prvky v závislosti na okolnostech dané situace. Známe 4 druhy těchto bran a to tak, abychom mohli pokrýt všechny potřeby rozhodování:

- XOR
- OR
- AND
- Komplexní

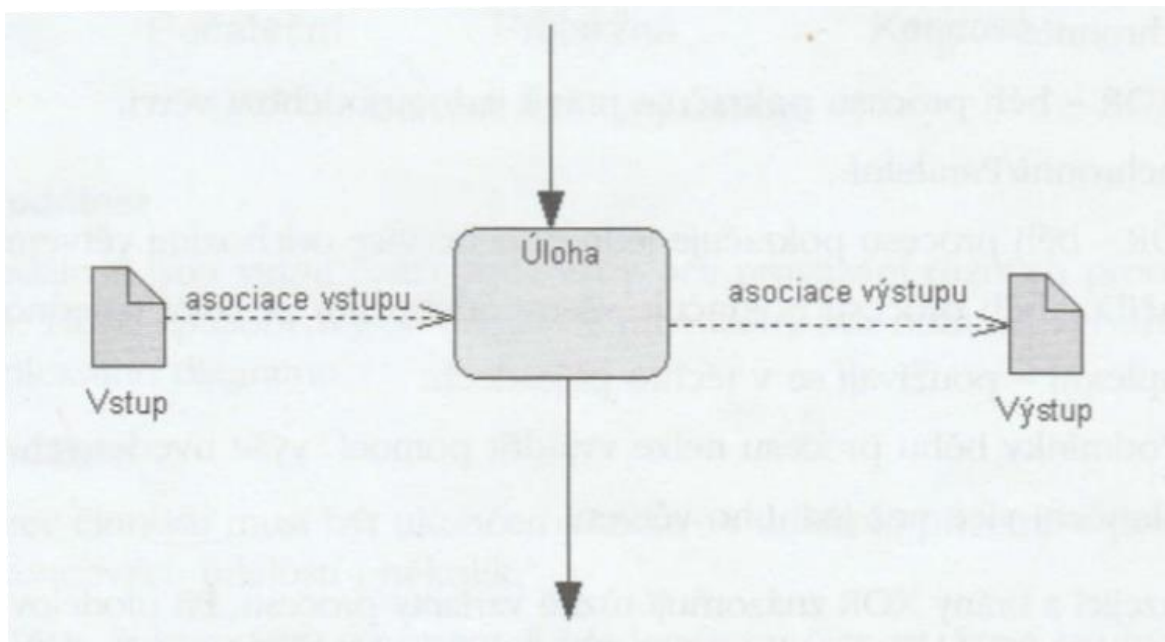
(Kanisová, a další, 2012)

XOR se typ brány, kdy běh činností pokračuje pouze jednou cestou. Brána OR disponuje možností běhu více větvemi najednou. AND brána pokračuje všemi větvemi, které jsou touto vazbou připojeny. Komplexní brána je poté zástupcem takové množnosti, kdy všechny ostatní nestačí pro popsání běhu procesů. (Kanisová, a další, 2012)



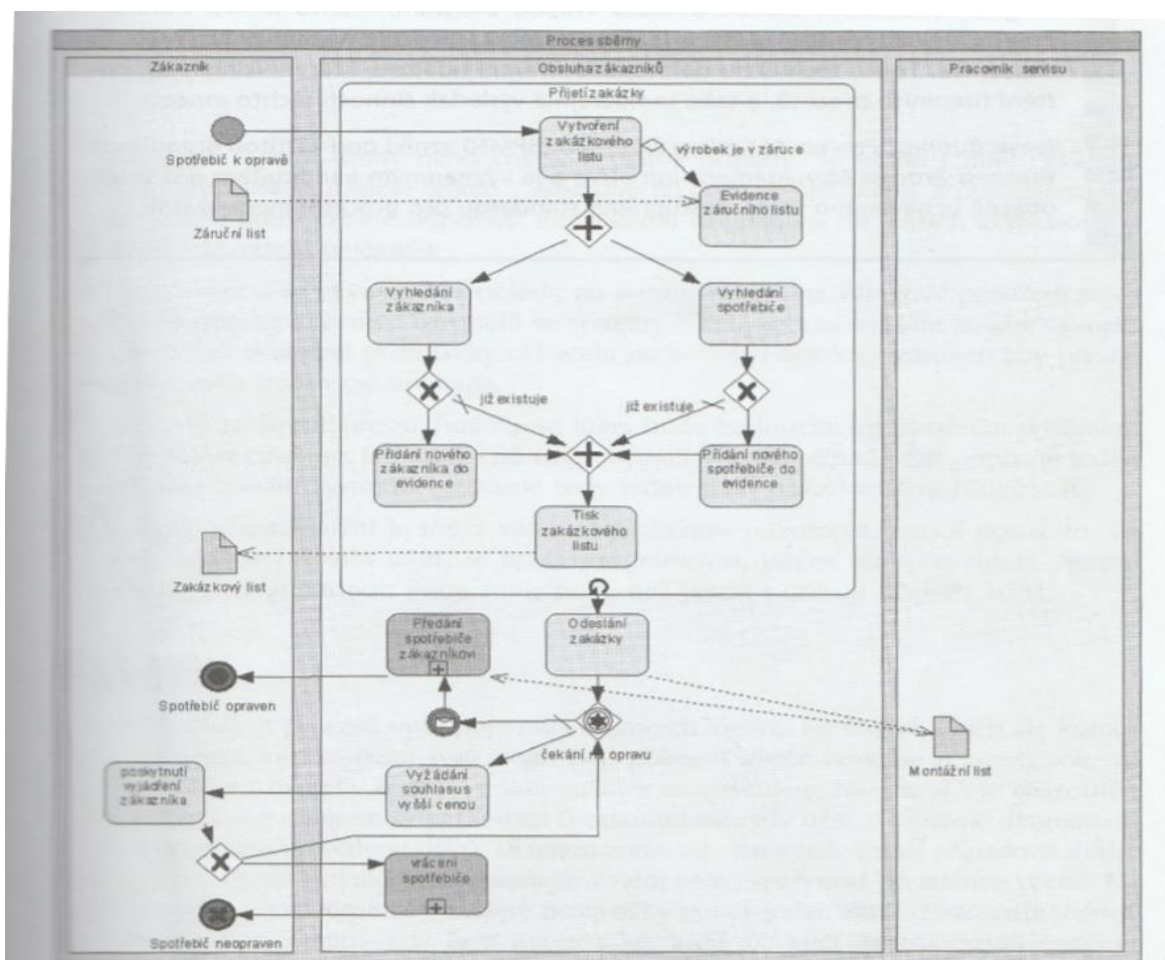
Obrázek 11 – typy bran - (Kanisová, a další, 2012)

Artefaktem je jakýkoliv prvek, který má funkci vstupu či výstupu pro určitou činnost. Vstup v grafu umísťujeme na levou stranu od vybrané činnosti a naopak výstup. (Kanisová, a další, 2012)



Obrázek 12 – artefakty - (Kanisová, a další, 2012)

Zapojení všech prvků a jejich znázornění na praktickém příkladu můžeme vidět na obrázku pod tímto textem.



Obrázek 13 – kompletní BPMN diagram - (Kanisová, a další, 2012)

Nyní s těmito podklady můžeme přejít k případům užití. Tím můžeme odhalit nedostatky získaných požadavků na funkce či naopak jejich duplicitnost. Případy užití totiž rozpadají požadavky na jednotlivé činnosti, kdy na konci každé činnosti je jeden výstup. Jednodušeji poté sada možností, která v rámci daného požadavku může nastat. Případ užití v zásadě popisuje vybraný proces, a to z hlediska účastníků a jejich akcí, jak můžeme vidět na obrázku. (Kanisová, a další, 2012)

Případ užití: Příjem spotřebiče do opravy		
Případ užití začíná, když zákazník přináší spotřebič do opravy		
Krok	Role	Akce
1	Obsluha	dá pokyn k založení zakázkového listu
2	Systém	zobrazí formulář zakázkového listu se zpřístupněnými údaji
3	Obsluha	aktivuje výběr zákazníka ze seznamu existujících
4	Systém	zobrazí seznam existujících zákazníků sběrný
5	Uživatel	POKUD zákazník dosud neexistuje v evidenci sběrný, pořídí údaje zákazníka ... JINAK vybere zákazníka z evidence KONEC-POKUD
6	Systém	předá údaje zákazníka do zakázkového listu
7

Obrázek 14 – případ užití - (Kanisová, a další, 2012)

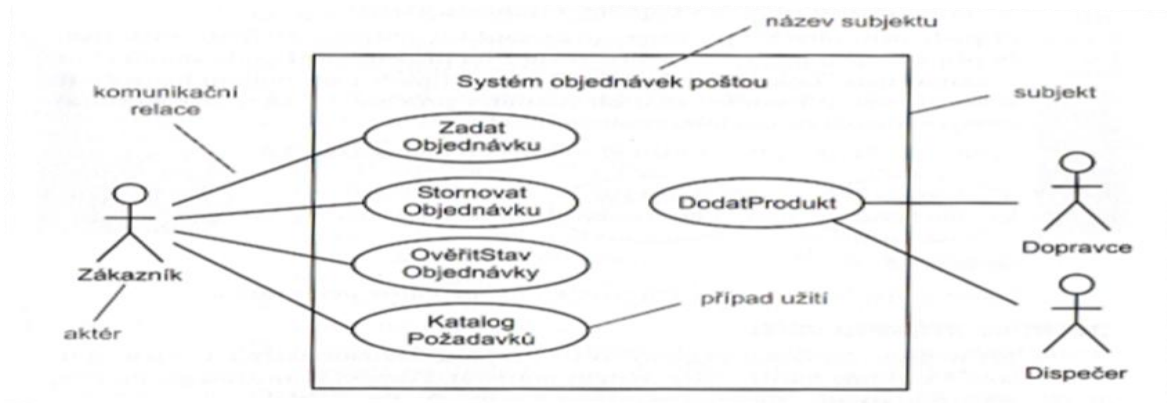
I tento nástroj má pro své správné použití určitá pravidla, která bychom měli respektovat. Mezi ně můžeme zařadit co nejpřesnější formulaci názvů případu užití, chronologické číslování kroků, včetně možnosti rozdílných scénářů či použití jednoduchého, jednotného a srozumitelného jazyka. Výstupem poté je vyjádření, co systém uživateli nabídne, a naopak co systém bude žádat. (Kanisová, a další, 2012)

Doporučeným, nicméně ne nutným je poté diagram případů užití. Použití tohoto nástroje má za úkol zpřehlednit případy užití. To znamená, že se jedná o výstupově stejný zápis, jen v grafickém provedení a propojuje více případů užití dohromady. Aby bylo možné znázornit graficky i obtížně zapsané případy použití, musíme se seznámit se metodickou formou toho zápisu. Prvně jsou to grafické prvky:

- postava
- elipsa
- spojnice
- rámeček

(Kanisová, a další, 2012)

Postavy reprezentují aktéry. Elipsy jsou následně jednotlivé případy užití. Spojnice můžou být dvojího druhu. První spojnice je spojnice plná, která reprezentuje tok informací mezi případem užití a aktérem. Přerušovaná do případu užití zahrnuje další případy užití, které jsou nezbytné pro jejich dokončení. Poslední rámeček udává hranice vybraného systému. (Kanisová, a další, 2012)



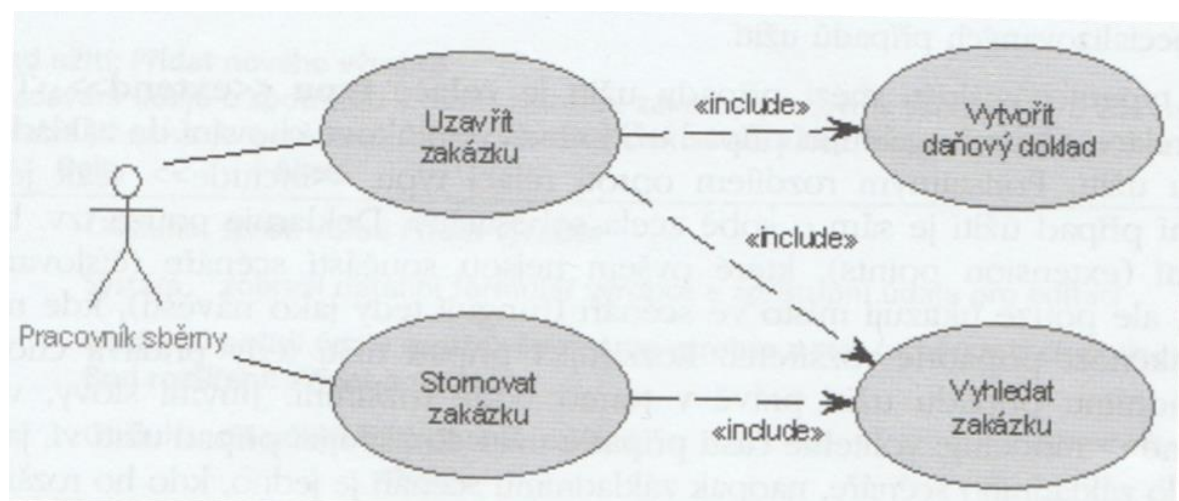
Obrázek 15 – diagram případu užití - (Arlow, a další, 2007)

Dále mezi jednotlivými případy užití mohou být vzájemné vztahy:

- include
- extend

(Kanisová, a další, 2012)

Relace include nám říká, že výchozí případ užití je závislý na jiném případě užití. Na obrázcích níže můžeme vidět kombinaci případu užití a jeho převodu do diagramu případu užití s vazbou include pro lepší pochopení. (Kanisová, a další, 2012)



Obrázek 16 – diagram případu užití s vazbou include - (Kanisová, a další, 2012)

Případ užití: Uzavřít zakázku		
Případ užití začíná, když si zákazník přichází vyzvednout opravený spotřebič		
Krok	Role	Akce
1	Obsluha	vyhledá zakázku – viz případ užití Vyhledat zakázku
2	Systém	zobrazí detailní okno zadané zakázky
3	Obsluha	seznámí zákazníka s výsledkem opravy a dá pokyn k sestavení daňového dokladu – viz případ užití Vytvořit daňový doklad
4

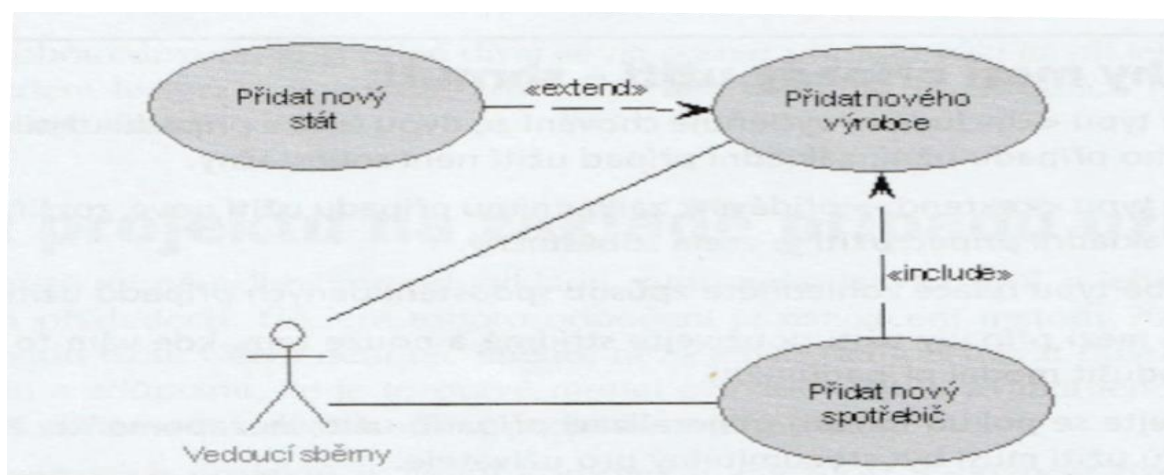
Obrázek 17 – případ užití s vazbou include - (Kanisová, a další, 2012)

Naproti tomu relace extend je situace, kdy je výchozí případ užití zcela nezávislý na rozšiřující činnosti. Na obrázcích případ užití, kde se nám vyskytuje vazba extend. (Kanisová, a další, 2012)

Zápis pomocí případu užití je následující:

Případ užití: Přidat nového výrobce		
Při zadávání údajů o spotřebiči v rámci založení zakázky pracovník sběrný ověří, zda pro daný spotřebič již existuje v evidenci sběrný výrobce; pokud ne, založí nový záznam výrobce.		
Krok	Role	Akce
1	Obsluha	spustí volbu <i>Přidat výrobce</i>
2	Systém	zobrazí detailní formulář výrobce a zpřístupní údaje pro editaci
3	Obsluha	pořídí údaje spotřebiče - název výrobce a stát (výběrem z číselníku států) Bod rozšíření: Přidat nový stát
4	Obsluha	dá pokyn k uložení informací
5

Obrázek 18 – případ užití s vazbou extend - (Kanisová, a další, 2012)



Obrázek 19 – diagram případu užití s vazbou extend - (Kanisová, a další, 2012)

Zaměříme se na obrázek výše. Na první pohled se může jevit zvláštní použití vazby extend v tomto systému, ale tento případ je ještě o trochu složitější. Aktér „Vedoucí sběrný“ může inicializovat dva případ užití a to sice „Přidat nový spotřebič“ a „Přidat nového výrobce“. Případ užití „Přidat nového výrobce“ můžeme inicializovat 2

způsoby, a to sice z „Přidat nový spotřebič“ či rovnou jako samostatná možnost. Příklad „Přidat nový stát“ se pak nebude nikdy spouštět samostatně, ale pouze doplňkově k „Přidat nového výrobce“. Můžeme tedy vidět, že to, jakým způsobem k případu přistupujeme a jak si stakeholderi stanoví, ovlivní vazby mezi jednotlivými případy. (Kanisová, a další, 2012)

Případy užití nám jsou výchozími body pro otestování, jestli všechny popsané případy užití, které vycházejí z požadavků, disponují nezbytnými informacemi a prostředky. Další využití případu užití dle použité literatury spočívá v řízení projektu podle těchto ukazatelů. Projekt vývoje softwarové aplikace je totiž nestabilní záležitostí ve smyslu neustálých změn. Pomocí toho nástroje můžeme společně s našimi stakeholdery korigovat pořadí, v jaké budou jednotlivé požadavky implementovány. Použití nástrojů CASE a případů užití nám umožňuje adaptivně reagovat přírůstky požadavků, které budeme projektovému týmu dávat k vypracování. (Arlow, a další, 2007)

Znovu si připomeňme výhody těchto nástrojů, které spočívají v jejich jednoduchosti, což můžeme využívat při komunikaci s našimi partnery, kteří nutně nemusí rozumět programování či vývoji softwaru jako takového. Naším grafickým znázorněním ovšem porozumí daleko snadněji a můžeme, tak odhalit nedostatky, kterých jsme si doposud nebyli vědomi.

Modelovací jazyk UML poskytuje další velmi přesné instrukce, jakým způsobem přímo již programovat samotnou aplikaci, respektive umožňuje nám stanovit vazby mezi jednotlivými objekty, třídami a dalšími prvky našeho softwaru. Pro rozsah a jednoduchost této práce však tyto nástroje nebudeme využívat a bude nám stačit použití těch nástrojů, které jsme doposud řešili.

PRAKTICKÁ ČÁST

4 Popis společnosti

Společnost, ve které byla vykonána praktická část této bakalářské práce je Školka Krásná z.s. Tato společnost byla založena 1.6.2017 jako spolek v soukromých prostorech žadatele. V rámci dotací byla žadateli částečně zaplacená rekonstrukce prostorů pro provoz dětské skupiny a jednou za 6 měsíců je zaslána dotace, která částečně kryje provozní náklady tohoto zařízení. Společnost se tedy zaměřuje na provoz dětské skupiny, jejímž hlavním úkolem je možnost umístění dítěte ve věku (1 až 3 let stáří), které z nějakého důvodu nemohou být umístěny do běžné státní školky. Provoz toho typu zařízení umožňuje zvýšit konkurenci schopnost a zaměstnanost rodičů umístěných dětí. Umožňuje tedy například matkám vrátit se do práce alespoň na částečný úvazek, či se připravovat na výkon nového povolání při dostání jejích povinností.

Kapacita zařízení je v současné době 10 dětí a následujícím obdobím je plánováno rozšíření této kapacity o dalších 10 míst. Školka v současné chvíli má 2 zaměstnance na HPP, jednu učitelku MŠ a jednu chůvu. Tyto zaměstnanci musí disponovat patřičným vzděláním, a to sice dle zákona o dětské skupině.

Podmínka pro získání dotace na provoz je vykazování veškeré administrativy spojené s chodem školky. Mezi ten hlavní můžeme uvést záznam o docházce, na jehož základě dochází k určení výše této dotace. Docházka dětí je závislá na volbě programu, který rodiče volí při registraci dítěte do dětské skupiny. Tento program je možné kdykoliv zaměnit za jiný.

Příčinou výběru této společnosti pro potřeby této práce je volba řešení evidence docházky dětí a učitelů pro plnění výše zmíněné povinnosti, které bude splňovat klientovi a právní požadavky. Za neplnění tohoto požadavku hrozí krácení dotací, které zařízení na jeho provoz obdrží.

5 Představení projektu

Výše popsaná organizace musí v rámci svých legislativních povinností vést záznamy o docházce jednotlivých osob, které toto zařízení navštěvují, a to sice v přesně stanovené elektronické podobě. Dále organizace vyžaduje, jak vyplynulo z rozhovorů, nad rámec výše zmíněného požadavku, například vedení agendy stravného a obecnou správu uživatelů. Společnost tedy potřebuje zajistit plnění těchto požadavků a musí se pro své fungování rozhodnout, jakým způsobem tuto nastalou situaci vyřeší. Může buď zvolit nějaké ze stávajících řešení na trhu, či poptat vývoj vlastního softwaru. Shrňme si tedy, co je naplní projektu.

Cílem projektu je dosažení stavu, kdy v popsané organizaci bude funkční docházkový systém, a to včetně veškerých náležitostí a způsobu provedení, které zadavatel a současně legislativa vyžaduje.

Instalaci systému je nezbytné realizovat do 1.6.2018. Tehdy začíná monitorovací období a započíná tedy povinnost docházku sledovat. Datum zahájení projektu je přitom 4.1.2018.

Pro projekt je vymezena částka 40 000 Kč bez DPH v rámci jeho 5leté doby životnosti, což je minimální doba udržitelnosti, po kterou bude nutné vést docházku. Budeme tedy hledat řešení v této cenové hladině, kde budou splněny všechny požadavky na funkčnost systému při dodržení termínu implementace.

Výsledné řešení by mělo být pro personál uživatelsky přívětivé a generovat nízké provozní náklady.

6 Specifikace hledaného řešení

Řešení, které hledáme pro úspěšnou realizaci projektu, musí splňovat jak funkce specifikované klientem, ale rovněž i funkce, které jsou dány platnou legislativou. Začneme legislativními požadavky.

Společnost je úspěšným účastníkem výzvy 76, která byla vyhlášena evropským sociálním fondem a stanovuje tedy na její účastníky požadavky. Mezi ně patří právě povinnost vést docházku dětí a to v „elektronickém docházkovém systému, který využívá docházkové čtečky“. Je tedy patrné, že náš systém bude muset pracovat s nějakým druhem čipových karet. Pro záznam docházky dětí je před vytvořena šablona, do které mají být záznamy o docházce ukládány a při kontrole je tento dokument výchozím podkladem pro ověření činnosti dětské skupiny. Systém tedy musí pracovat s touto šablonou a musí umět exportovat tento dokument ve formátu „xlsx“.

Dalším požadavkem ve výzvě 76 je omezení počtu dětí a minimálního množství pečujících osob. Ten říká, že pokud je počet přítomných dětí od 0 do 6, musí být přítomná alespoň jedna pečující osoba. Dále pro 7 až 12 dětí jsou potřeba nejméně 2 pečující osoby a pro 13 až 24 dětí nejméně 3 pečující osoby. Kapacita zařízení může být v rozmezí 5 až 24 dětí, přičemž je tato kapacita stanovena dle hygienických faktorů. V případě naší dětské skupiny je 10 dětí. Oba zmíněné požadavky vychází z dokumentu „Specifická část pravidel pro žadatele a příjemce v rámci OPZ pro projekty s jednotkovými náklady zaměřené na podporu zařízení péče o děti předškolního věku“, který upravuje chod dětské skupiny.

Nezbytným požadavkem je implementace „GDPR“. GDPR je zkratka pro nařízení o ochraně osobních údajů, která vstupuje v platnost od 25.5.2018. Vybraná aplikace musí splňovat body tohoto nařízení.

Při studiu dokumentů spojených s projektem na vznik dětské skupiny jsme našli žádné další body, které by specifikovali provoz docházkového systému a přesunuli jsme se k získání požadavků od klienta, které by měly zastřešit specifikaci řešení. Tyto požadavky byly pořízeny pomocí metody rozhovoru. Rozhovor byl strukturovaný a v jeho úvodu došlo obeznamení klienta s legislativními požadavky na aplikaci. V další části rozhovoru došlo ke sběru požadavků klienta. Klient vždy představil svou vizi příslušného požadavku a doplňujícími a rozvíjejícími otázkami byl tento požadavek přesně specifikován. Získali jsme tak přehled o požadavcích funkčních i nefunkčních. Na konci rozhovoru došlo k rekapitulaci získaných podkladů a popisu následující

[illegible]

| 35

7 Možnosti řešení

V této části se zaměříme na analýzu obecných řešení, které mohou být využity pro splnění podmínky evidence docházky neboli našeho projektu. Tuto analýzu provedeme hledáním dostupných řešení, které splňují veškeré náležitosti projektu. Z těchto řešení budeme v následující kapitole vybírat to, které bude implementováno. Vyhledání možných řešení proběhlo pomocí internetového vyhledávače. Společnosti zahrnuté v analýze nabízejí dvě varianty řešení:

- mobilní aplikace
- pevný terminál

Varianta mobilní aplikace je založena na použití chytrého telefonu s operačním systémem android, který podporuje technologii NFC. Tato technologie podporuje komunikaci telefonu s čipovými kartami a zastupuje tedy roli pevné čtečky. Výhodou této varianty je možnost využít telefon s popsáním operačním systémem, což se velmi kladně promítne v počátečních nákladech. Rovněž je čtečka jednoduše přenositelná a lze na ní provádět práci kdekoli. Ovšem pro jakoukoliv práci na ní je zapotřebí připojení k internetu, a to ať už pomocí mobilních dat či Wi-Fi. Telefon, který splňuje tyto požadavky lze pořídit od zhruba 3 000 Kč bez DPH (Pro naše potřeby budeme kalkulovat s telefonem typu Nexus 5X, který zadané požadavky splňuje a jeho v kalkulacích zanesena částkou 5 500 Kč bez DPH). Nevýhodou může být případ, kdy se baterie telefonu vybije či dojde ke krádeži telefonu.

Druhou variantou je pevný terminál, který slouží jako čtečka čipových karet. Výhodou tohoto řešení může být například nezávislost na výdrži baterie, protože terminál má vlastní přívod napájení a rovněž bude obtížné provést krádež takového zařízení. Nevýhodou je následně vyšší pořizovací cena.

Ať už je zvolena libovolná varianta, je vždy nutné zajistit rovněž čipové karty, které jsou využívány k zaznamenávání docházky při přiložení ke čtečce.

8 Výběr řešení

. Udržitelnost zařízení je plánováno na minimálně dalších 5 let, proto naše kalkulace nákladů budou počítat s tímto časovým rozpětím. Analýza nákladů byla provedena na základě veřejně dostupných informací vybraných společností a v případě nejistoty byla telefonicky ověřena či doplněna. Zároveň veškerá uvedená data jsou platná k měsíci zahájení projektu, a to sice leden 2018. Dále je kalkulace provedena pro 2 dětské skupiny. Do kalkulací nebudou zahrnuty žádné jiné náklady kromě těch přímo zmíněných. Například není tedy uvažována spotřeba elektrické energie, připojení k internetu nebo například časová hodnota peněz.

Nyní přejdeme k jednotlivým společnostem a jejich řešením. První společností v našem rozboru je společnost Aperus, s.r.o. a její systém „Webooker“. Cena řešení této společnosti je závislá na volbě typu řešení a programu pořízení. Společnost nabízí obě

zmíněné možnosti řešení a náklady na zřízení a provoz po dobu 5 let jsou následující (všechny částky jsou bez DPH a v Kč):

Tabulka 1 – Webooker řešení pomocí mobilní aplikace bez závazku

Webooker - mobilní aplikace bez závazku					
Rok / Typ nákladu	1	2	3	4	5
Investiční náklady	15 690	-			
Provozní náklady	8 400	8 400	8 400	8 400	8 400
				Celkem:	57 690

Při této variantě financování docházkového systému, dojde na počátku k nákupu mobilního telefonu (5 500), licence softwaru (6 990), 30 čipových karet (1 200), rozšíření o další dětskou skupinu (2 000). Poté jsou hrazeny jen základní měsíční poplatek (500) a poplatek za rozšíření dětské skupiny (200). Celková částka za 5 let poté činí zaokrouhleně 58 000 Kč.

Tabulka 2 – Webooker řešení pomocí mobilní aplikace se závazkem

Webooker - mobilní aplikace se závazkem					
Rok / Typ nákladu	1	2	3	4	5
Investiční náklady	6 700	-			
Provozní náklady	10 200	10 200	10 200	10 200	10 200
				Celkem:	57 700

Při volbě této varianty financování docházkového systému, dojde na počátku k nákupu mobilního telefonu (5 500), 30 čipových karet (1 200). Poté je hrazen jen základní měsíční poplatek (850). U tohoto kontraktu je 2letá fixační doba. Celková částka za 5 let poté činí zaokrouhleně 58 000 Kč.

Tabulka 3 – Webooker řešení pomocí čtečky bez závazku

Webooker - čtečka bez závazku					
Rok / Typ nákladu	1	2	3	4	5
Investiční náklady	17 200	-			
Provozní náklady	8 400	8 400	8 400	8 400	8 400

Celkem:	59 200
----------------	---------------

Při zvolení této varianty financování docházkového systému, dojde na počátku k nákupu čtečky (9 000), licence softwaru (9 000), 30 čipových karet (1 200), rozšíření o další dětskou skupinu (2 000). Poté jsou hrazeny jen základní měsíční poplatek (500) a poplatek za rozšíření dětské skupiny (200). Celková částka za 5 let poté činí zaokrouhleně 60 000 Kč.

Tabulka 4 – Webooker řešení pomocí čtečky se závazkem

Webooker - čtečka se závazkem					
Rok / Typ nákladu	1	2	3	4	5
Investiční náklady	1 200	-			
Provozní náklady	13 800	13 800	13 800	13 800	13 800
				Celkem:	70 200

Poslední variantou financování docházkového systému u této společnosti, dojde na počátku k nákupu 30 čipových karet (1 200). Poté je hrazen jen základní měsíční poplatek (1 150). U tohoto kontraktu je 2letá fixační doba. Celková částka za 5 let poté činí zaokrouhleně 71 000 Kč.

Aplikace nabízí všechny požadované funkce a může být za peněžitou útratu upravena na míru. Co se týče časové omezení, dodržení termínu pro realizaci by byl dodržen. Ovšem finanční omezení, které máme stanoveno při volbě jakékoliv varianty nebylo dodrženo. Riziko je zde však minimální, a to vzhledem k množství organizací, kde systém již funguje

Další analyzovaným řešením je nabídka společnosti Dignus services, s.r.o. Její systém „Dignus“ umožňuje řešení pomocí mobilní aplikace. Přehled nákladů je následující:

Tabulka 5 – Dignus řešení pomocí mobilní aplikace

Dignus - mobilní aplikace					
Rok / Typ nákladu	1	2	3	4	5
Investiční náklady	-				
Provozní náklady	11 528	12 576	12 576	12 576	12 576
				Celkem:	61 832

Vidíme, že při využití služeb této společnosti máme nulové investiční náklady, jelikož mobilní telefon pro použití jejich softwaru dodá samotná společnost. Platíme

tedy pouze měsíční poplatky za základní užívání softwaru (699) a rozšíření o další dětskou skupinu (349), přičemž první měsíc je na zkoušku zdarma. Celková částka činí zaokrouhleně 62 000 Kč.

Aplikace této společnosti opět vyhovuje požadavkům. Její úprava může být realizována po dohodě a dle rozsahu zpoplatněna. Doba implementace je dodavatelem stanovena na 3 dny. Rovněž aplikace funguje již v několika organizacích, tedy riziko opět nízké. Nicméně finanční omezení opět není splněno

Poslední analyzovanou společností je META IT s.r.o. s jejím systémem „Skolkis“. Nabízené řešení je realizováno opět pomocí chytrého telefonu s následujícími náklady:

Tabulka 6 – Skolkis řešení pomocí mobilní aplikace

Skolkis - mobilní aplikace					
Rok / Typ nákladu	1	2	3	4	5
Investiční náklady	5 000	-			
Provozní náklady	7 500	7 500	7 500	7 500	7 500
				Celkem:	42 500

Náklady jsou tvořeny počáteční investicí (5 000), která zahrnuje čipové karty a mobilní telefon. Provozní náklady se skládají ze základní poplatku za užívání (10 000) a za rozšíření (5 000). Suma nákladů za 5 let je zhruba 43 000 Kč.

Funkce, které aplikace nabízí jsou v souladu s funkčními požadavky klienta. Realizace je poté v řádech dnů. Aplikace je v současné době využívána několika organizacemi. Riziko znovu nízké. Ani ve třetím případě nedošlo ke splnění finančního kritéria.

Po představení dostupných řešení na trhu se podíváme na řešení pomocí vývoje vlastního softwaru. Pravděpodobné náklady na realizaci vývoje mého softwaru popisovaného rozsahu jsou následující:

Tabulka 7 – Marek Kaufmann řešení pomocí mobilní aplikace

Marek Kaufmann - mobilní aplikace					
Rok / Typ nákladu	1	2	3	4	5
Investiční náklady	17 800	-			
Provozní náklady	3 600	3 600	3 600	3 600	3 600
				Celkem:	35 800

Mezi investiční náklady je zahrnut nákup mobilního telefonu, práce na vývoji softwaru a čipové karty. Aplikace jako taková nebude generovat téměř žádné provozní náklady. Jediným provozním nákladem budou služby pověřence. Celková suma nákladů na dobu 5 let by činili 36 000 Kč.

Funkce, které by vyvíjená aplikace nabízela, budou přesně na míru konstruovány dle požadavků legislativy a zákazníka. Prostředky pro vytvoření takového systému disponuji. Ohledně délky realizace vývoje aplikace, můžeme hovořit v řádu měsíců. V tomto případě můžeme hovořit o riziku, že by se předání produktu nemuselo stihnout v termínu, který byl v projektu stanoven. Oproti všem ostatním řešení zde došlo k naplnění finančního požadavku.

Tabulka 8 – porovnání jednotlivých řešení pomocí NPV

Porovnání jednotlivých řešení pomocí kumulovaných CF					
Rok / Společnost	1	2	3	4	5
Marek Kaufmann	21 400	3 600	3 600	3 600	3 600
Webooker - mobilní aplikace bez závazků	24 090	8 400	8 400	8 400	8 400
Rozdíl kumulovaných CF	- 2 690	- 7 490	- 12 290	- 17 090	- 21 890
Marek Kaufmann	21 400	3 600	3 600	3 600	3 600
Webooker - mobilní aplikace se závazkem	16 900	10 200	10 200	10 200	10 200
Rozdíl kumulovaných CF	4 500	- 2 100	- 8 700	- 15 300	- 21 900
Marek Kaufmann	21 400	3 600	3 600	3 600	3 600
Webooker - čtečka bez závazku	25 600	8 400	8 400	8 400	8 400
Rozdíl kumulovaných CF	- 4 200	- 9 000	- 13 800	- 18 600	- 23 400
Marek Kaufmann	21 400	3 600	3 600	3 600	3 600
Webooker - čtečka se závazkem	15 000	13 800	13 800	13 800	13 800
Rozdíl kumulovaných CF	6 400	- 3 800	- 14 000	- 24 200	- 34 400
Marek Kaufmann	21 400	3 600	3 600	3 600	3 600
Dignus	11 528	12 576	12 576	12 576	12 576
Rozdíl kumulovaných CF	9 872	896	- 8 080	- 17 056	- 26 032
Marek Kaufmann	21 400	3 600	3 600	3 600	3 600
Skolkis	12 500	7 500	7 500	7 500	7 500
Rozdíl kumulovaných CF	8 900	5 000	1 100	- 2 800	- 6 700

V tabulce výše můžeme vidět porovnání jednotlivých nabídek s nabídkou vývoje vlastní aplikace pomocí metody NPV. Z tabulky vyplývá, že při investici jako je tato se vyplatí zvolit variantu vlastního vývoje, pokud hovoříme o finanční stránce projektu. Po prezentaci možných řešení, jejich přínosů a zvážení možných rizik se ředitelka školky rozhodla zvolit možnost poslední, sice vývoj vlastního systému. Zároveň bylo

stanoveno, že je nezbytné spustit do časového omezení takový systém, který bude naplňovat minimální legislativní požadavky. Projekt byl tedy tímto aktem odstartován.

9 Sběr požadavků

Jak jsme si rozvedli v teoretické části, před započítím jakýchkoliv prací na vývoji softwaru máme náležitosti, které samotnému vývoji předchází. Těžko si lze představit vývoj specifického docházkového systému právě bez specifikací na funkce a parametry vyvíjeného softwaru. Vývoj provádíme pro konkrétní organizaci jejímž vedení máme osobní vztah. Tomu tedy přizpůsobíme procesy, které samotnému vývoji předchází.

V rámci rozsahu této práce je seznam zainteresovaných osob oproti běžné situaci při vývoji aplikace minimální. Pojďme si tedy tyto osoby uvést:

- ředitelka organizace
- legislativa

Před vývojem aplikace proběhnul výše popisovaný rozhovor s ředitelkou spolku. Rovněž před tímto rozhovorem došlo k samostudiu dokumentu, který ukládá povinnost vést elektronickou docházku dětí a učitelů v daném spolku společně se studiem možností konkurenčních softwarových společností. Níže nalezneme tabulku, kde můžeme vidět název požadavku, typ požadavku, a zdroj tohoto požadavku:

Tabulka 9 – přehled požadavků

Název požadavku	Typ požadavku	Zdroj získání požadavku
Způsob evidence	funkční	legislativa
Role	funkční	ředitelka
Správa dětí	funkční	ředitelka
Stáří dítěte	nefunkční	legislativa
Detaily rodiče	nefunkční	ředitelka
Úpravy dítěte	nefunkční	ředitelka
Oprávnění	nefunkční	ředitelka
Záznam docházky dětí	funkční	legislativa
Čas příchodu	nefunkční	ředitelka
Pravidlo 6x/1x	nefunkční	legislativa
Počet přítomností	nefunkční	ředitelka
Kapacita dětské skupiny	nefunkční	legislativa
Správa učitelů	funkční	ředitelka
Úpravy učitele	nefunkční	ředitelka
Oprávnění	nefunkční	ředitelka
Záznam docházky učitelů	funkční	legislativa
Pravidlo 6x/1x	nefunkční	Legislativa
Počet přítomností	nefunkční	ředitelka

Stravné	funkční	ředitelka
Minimální výše objednávky	nefunkční	ředitelka
Omezení ohlášky a přihlášky	nefunkční	ředitelka
Fakturace	funkční	ředitelka
Oprávnění	nefunkční	ředitelka

9.1.1 Způsob evidence

Při studiu dokumentů bylo zjištěno, že docházkový systém musí mít elektronickou podobu a záznam docházky musí být prováděn pomocí čipových karet. Aplikace tedy musí umět pracovat čipovými kartami a záznamy o docházce vkládat do příslušného dokumentu. Nesmíme zapomenout na možnost exportovat tento dokument ve formátu „xlsx“.

9.1.2 Role

V naší vyvíjené aplikaci se budou vyskytovat 3 druhy osob:

- dítě
- učitel
- administrátor

Dítě má v tomto systému roli klienta, tudíž tato role nebude disponovat přístupem do aplikace a bude s ní pouze operováno. U této osoby budeme evidovat docházku, tvořit na jejím základě reporty a zaznamenávat stravné.

Učitel je personál školky, u něhož je plánována základní součinnost s aplikací. Role učitel tedy bude disponovat vlastním přístupem do aplikace, nicméně nebude moci provádět všechny operace. U této osoby budeme pouze evidovat docházku.

Administrátor je stejná role, jako ta předchozí, s tím rozdílem, že v rámci aplikace může provádět všechny operace. Současně však nebudeme u tohoto uživatele evidovat docházku ani stravné.

Případ užití: Přihlášení uživatele

Systém při zapnutí či odhlášení uživatele vyzve operátora k aktéru

Krok	Role	Akce
1	Systém	vyzve k zadání uživatelského hesla
2	Aktér	zadá své uživatelské heslo
3	Systém	POKUD zadané heslo odpovídá vybrané osobě, tak systém provede přihlášení a zobrazí ty možnosti, které se vážou k jeho roli JINAK informuje o nesprávnosti hesla pro vybranou a vyzve k opakování akce

Obrázek 21 – případ užití přihlášení operátora

9.1.3 Správa dětí

V rámci rozhovoru s ředitelkou školky se došlo k závěru, že je potřeba evidovat následující údaje o dětech, které školku navštěvují:

- jméno
- příjmení
- rodné číslo
- návštěvní dny
- typ návštěvních dnů
- jméno rodiče 1
- příjmení rodiče 1
- telefon rodiče 1
- jméno rodiče 2
- příjmení rodiče 2
- telefon rodiče 2
- email

Jméno a příjmení dítěte je nezbytné z důvodu korektní identifikace daného dítěte. Rodné číslo slouží k ověření, zdali je věk dítěte větší, než 1 rok. Návštěvní dny slouží pro objednávání obědů a svačin, což bude rozebráno v samostatném požadavku na zřízení stravného. Pro upřesnění objednávky stravy slouží ještě parametr typ návštěvních dnů. Jméno, příjmení a telefon slouží v rámci aplikace pro možnost kontaktovat zákonné zástupce v nezbytně nutných případech. Email jako poslední je využíván pro zasílání vyúčtování stravného.

Z výše uvedeného nám plynou tyto případy užití:

- Založení nového dítěte
- Zobrazení údajů o dítěti
- Editace stávajícího dítěte
- Vymazání dítěte

Založení nového dítěte je prvotní zadání dítěte do systému. To umožňuje následně další operace, které jsou pro chod systému klíčové.

Případ užití: Založení nového dítěte

Po obdržení všech nezbytných údajů pro registraci dítěte, dojde jeho zařazení do systému pro tvorbu docházky

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	spustí možnost "Přidat nové dítě"
2	Systém	vytvoří formulář pro založení nového dítěte
3	Aktér	vyplní systémem vytvořený formulář pro založení dítěte
4	Systém	POKUD jsou zadány všechny nezbytné údaje a současně správně, systém provede vytvoření dítěte do cloudové a následně interní databáze telefonu. Dále vyzve systém k přiložení čipu, na nějž bude nahráno specifické číslo dítěte JINAK není splněn určitý požadavek na minimální věk dítěte, nebo nejsou vyplněny všechny nezbytné údaje pro jeho založení, systém informuje o této skutečnosti administrátora a proces se vrací do bodu 4
5	Aktér	přiloží čip
6	Systém	zapiše specifické číslo dítěte na čip
7	Aktér	předá čip zákonnému zástupci k užívání

Obrázek 22 – případ užití založení nového dítěte

Zobrazení údajů o dítěti bude nejvyužívanější každodenní operace systému, která zabezpečí tvorbu záznamu o docházce, ale rovněž pro editaci či vymazání dítěte.

Případ užití: Zobrazení údajů o dítěti

V případě potřeby zjištění například telefonního čísla na zákonného zástupce je tyto údaje možné zobrazit

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen libovolný uživatel

Krok	Role	Akce
1	Aktér	spustí možnost "Seznam dětí", v následně zobrazeném seznamu klikne na ikonku informace a vybere dítě, jehož údaje hledá
2	Systém	zobrazí veškeré údaje o dítěti

Obrázek 23 - případ užití zobrazení údajů o dítěti

Editace stávajícího dítěte má sloužit k těm případům, kdy se například změní telefonní číslo některých z rodičů či návštěvní dny vybraného dítěte.

Případ užití: Editace stávajícího dítěte

Při obdržení nových údajů dítěte, dojde k přepsání údajů, a to jak cloudově, tak i interně v telefonu

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	spustí možnost "Seznam dětí", v následně zobrazeném seznamu klikne na ikonku tužky a vybere dítě, které chce editovat
2	Systém	vytvoří formulář, kde budou předvyplněny stávající údaje spojené s dítětem
3	Aktér	změní data v těch polích, kde došlo ke změně údajů
4	Systém	POKUD jsou zadány všechny nezbytné údaje a současně správně, systém provede vytvoření dítěte do cloudové a následně interní databáze telefonu JINAK nejsou vyplněny všechny nezbytné údaje pro jeho založení, systém informuje o této skutečnosti administrátora a proces se vrací do bodu 4

Obrázek 24 - případ užití editace stávajícího dítěte

Vymazání dítěte je poslední případem užití požadavku správy dětí. Sloužit bude k odstranění dítěte z aplikace, což znemožní zaznamenávat další docházku zvoleného dítěte, přičemž veškeré záznamy provedené do doby před odstraněním nebudou odstraněny.

Případ užití: Vymazání dítěte

Při ukončení docházení dítěte do spolku, je údaje nutné vymazat

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	spustí možnost "Seznam dětí", v následně zobrazeném seznamu klikne na ikonku odpadkového koše a vybere dítě, které chce vymazat
2	Systém	zobrazí informační dialog s detaily dítěte, které chce aktér vymazat
3	Aktér	potvrdí správnost informací stisknutím tlačítka "Ano"
4	Systém	vymaže dítě z cloudové a rovněž z interní databáze telefonu. Následně vyzve operátora k vymazání specifického čísla na čipu
5	Aktér	přiloží čip
6	Systém	vymaže údaje na čipu

Obrázek 25 - případ užití vymazání dítěte

Nyní doplníme nefunkční požadavky, které jsou přímo spojené s požadavkem na správu dětí a jejich případy užití:

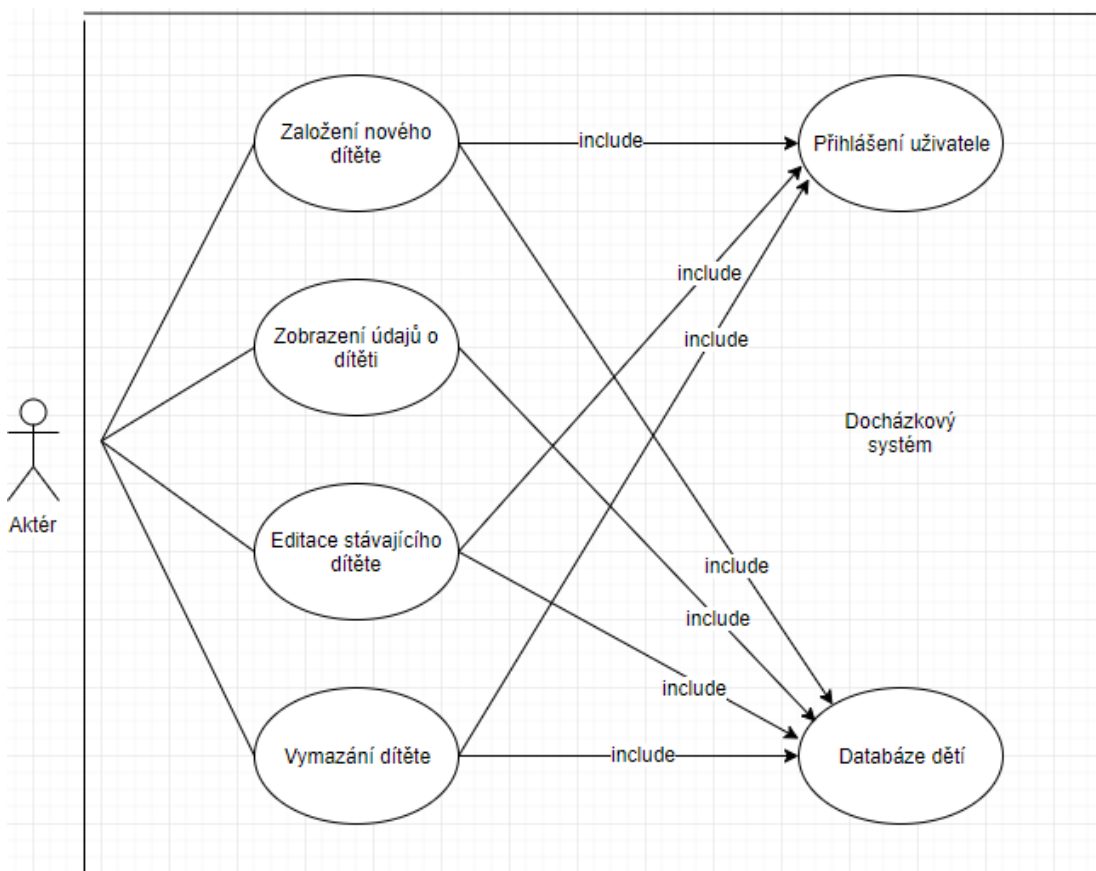
- Stáří dítěte
- Detaily rodiče
- Úpravy dítěte
- Oprávnění

Stáří dítěte, které spolek musí být větší než jeden rok. Proto pokud systém zjistí, že dítě této hranice nedosáhlo, systém odmítne požadavek na založení tohoto dítěte

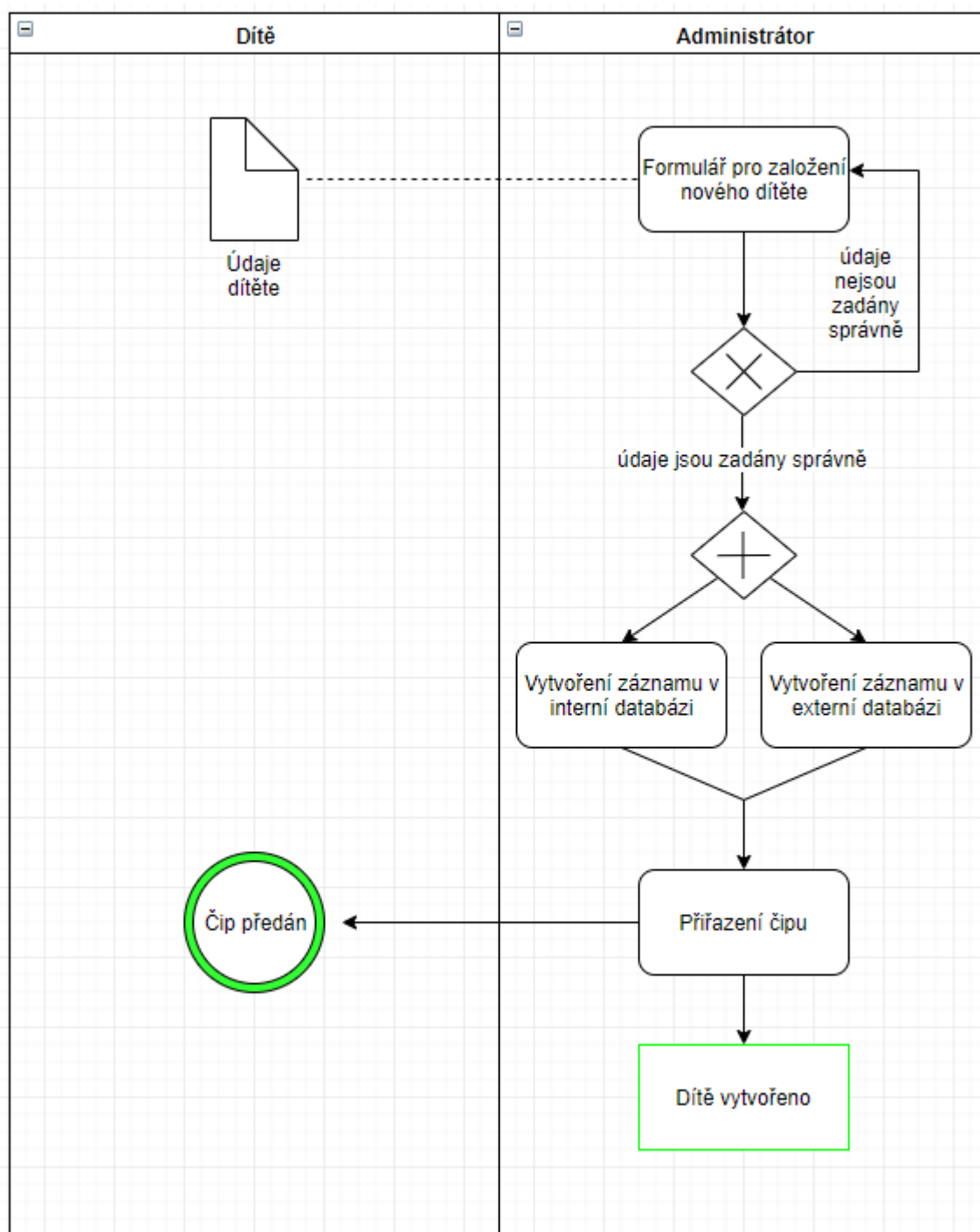
Detailem na rodiče je zde myšleno, že některé dítě nemusí mít nutně dva zákonné zástupce, proto systém přijme požadavek na založení dítěte pouze s jedním vyplněným rodičem. Nicméně systém na tuto skutečnost uživatele upozorní.

K úpravám dítěte může dojít pouze tehdy, pokud má uzavřenou docházku. To znamená, že z aplikace nebude možné odstranit či editovat dítě, které systém eviduje jako přítomné.

Oprávnění zde představuje omezení, kdy přidat, editovat a vymazat dítě může pouze osoba s oprávněním administrátor.



Obrázek 26 - diagram případ užití požadavku správa dětí



Obrázek 27 – BPMN založení dítěte

Způsob, jakým byl rozpracován tento požadavek, slouží jako východisku pro požadavky zbylé. To znamená že bude obsahovat všechny případy užití, diagram případů užití a případně jeden BPMN diagram.

9.1.4 Záznam docházky dětí

Hlavní funkce docházkového systému je pochopitelně evidence docházky. Přichází však otázka, jak způsobem chceme záznam realizovat, jaké údaje má záznam obsahovat a poslední způsobem kde budeme záznamy skladovat. V našem projektu

na použití elektronické čtečky čipů, evidence záznamu dat pak v excelové tabulce, kde zároveň dochází výpočtu naplněnosti zařízení, na jejímž základě poté projekt obdrží určitou výši prostředků. Po prostudování tohoto formuláře vyplynulo, že jeden kompletní záznam docházky dítěte musí obsahovat tyto údaje:

- datum
- jméno
- příjmení
- čas příchodu
- čas odchodu

Případy užití jsou:

- záznam příchodu dítěte
- záznam odchodu dítěte

Záznam příchodu dítěte probíhá při fyzickém příchodu dítěte do zařízení spolku. Zákonný zástupce, či jiný doprovod dítěte použije čip na čtečce, čímž dojde k zaznamenání příchodu. Při tomto kroku je odesláno do databáze datum, jméno, příjmení a čas příchodu dítěte.

Případ užití: Záznam příchodu dítěte

Při fyzickém příchodu dítěte potvrdí jeho doprovod tuto skutečnost přiložením osobního čipu ke čtečce

- Podmínky:**
- 1) Záznamem příchodu dítěte nebude porušeno pravidlo 6x/1x
 - 2) U vybraného dítěte nebyl dosud proveden záznam příchodu v daný den
 - 3) Čas v aplikaci je menší nebo roven 10:00
 - 4) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	přiloží přidělený čip ke čtečce čipů
2	Systém	zaznamená příchod dítěte do školky v databázi

Obrázek 28 – případ užití záznam příchodu dítěte

Záznam odchodu dítěte naopak probíhá při fyzickém odchodu dítěte ze školky. Osoba, která dítě vyzvedává opětovně použije čip se čtečkou, čímž dojde k zaznamenání odchodu. Do databáze je odesláno datum, jméno, příjmení a čas odchodu. Pomocí prvních třech parametrů dojde k vyhledání záznamu v databázi a následnému doplnění času odchodu, čímž je docházka za daný den uzavřena.

Případ užití: Záznam odchodu dítěte

Při fyzickém odchodu dítěte potvrdí jeho doprovod tuto skutečnost přiložením

Podmínky: 1) vybrané dítě je evidováno jako přítomné

2) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	přiloží přidělený čip ke čtečce čipů
2	Systém	zaznamená odchod dítěte ze školky v databázi

Obrázek 29 – případ užití záznam odchodu dítěte

Opět si doplníme nefunkční požadavky, které jsou pro tento požadavek následující:

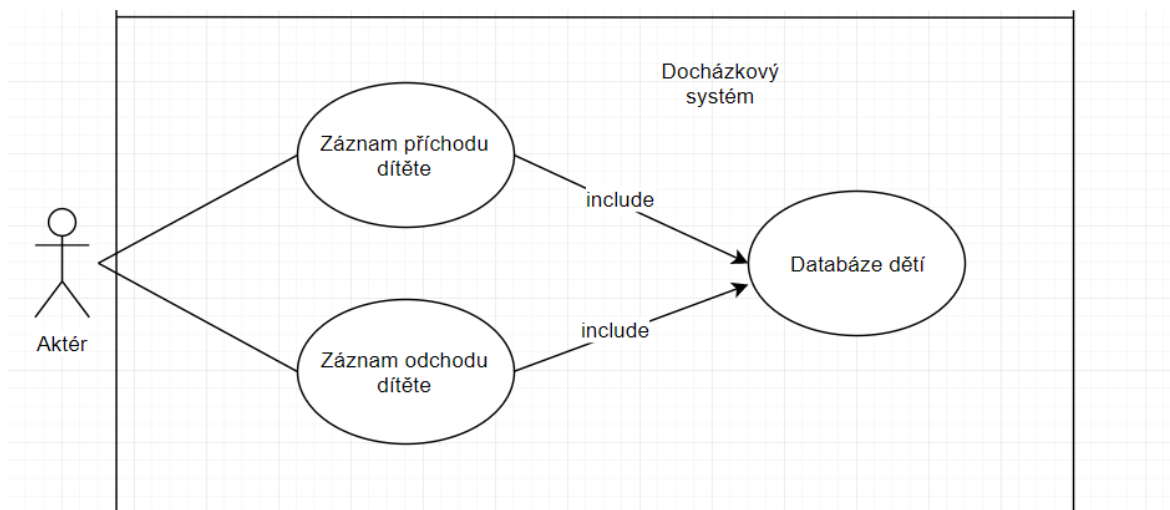
- čas příchodu
- pravidlo 6x/1x
- počet přítomností
- kapacita dětské skupiny

Čas příchodu je interním požadavkem, kdy příchod dítěte má být realizován do 10 hodiny dopolední. Po této době systém odmítne zaznamenat docházku daného dítěte.

Pravidlo 6x/1x reprezentuje nařízení, které stanoví, že jedna opatrovatelka může najednou poskytnout péči pouze pěti dětem. To znamená, že nesmí nastat případ, kdy ve školce bude 7 dětí a jedna opatrovatelka, či 1 dítě a žádný učitel. Systém tedy monitoruje, kolik je přítomných učitelů, podle čehož následně vyhodnotí, jestli může být zaznamenán příchod zvoleného dítěte.

Počet přítomností limituje každé dítě z hlediska evidence docházky pouze na jeden příchod a jeden odchod.

Kapacita dětské skupiny je legislativní omezení, které limituje v jeden okamžik počet přítomných dětí ve školce. V tomto případě hovoříme o čísle 10. Systém tedy odmítne zaznamenat příchod dítěte, pokud ve školce je registrováno již 10 přítomných dětí.



Obrázek 30 – diagram případu užití požadavku záznam docházky dětí

9.1.5 Správa učitelů

Velmi podobně jako je požadavek na vytvoření správy dětí, existuje požadavek na správu učitelů. Seznam použitých údajů je následující:

- jméno
- příjmení
- telefon číslo
- emailová adresa
- heslo

Jméno a příjmení slouží pro identifikaci učitele. Telefon slouží pro možnost kontaktovat vybraného učitele zbylým personálem. Emailová adresa slouží pro potřeby obnovení hesla. Heslo poté k možnosti se do systému přihlásit a pracovat s ním.

Z výše uvedeného nám plynou tyto případy užití:

- založení nového učitele
- zobrazení údajů o učiteli
- editace stávajícího učitele
- vymazání učitele

Založení nového učitele slouží k počátečnímu přidání učitele do systému. To znamená, že bude moci přistupovat do systému.

Případ užití: Založení nového učitele

Po obdržení všech nezbytných údajů pro registraci učitele, dojde jeho zařazení do systému pro tvorbu docházky

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	spustí možnost "Založit nového učitele"
2	Systém	vytvoří formulář pro založení nového učitele
3	Aktér	vyplní systémem vytvořený formulář pro založení učitele
4	Systém	POKUD jsou zadány všechny nezbytné údaje a současně správně, systém provede vytvoření učitele do cloudové a následně interní databáze telefonu. Dále odešle heslo pro přihlášení na email nového učitele JINAK nejsou vyplněny všechny nezbytné údaje pro založení učitele, systém informuje o této skutečnosti uživatele a proces se vrací do bodu 4

Obrázek 32 – případ užití založení nového učitele

Zobrazení údajů o učiteli můžeme využít k zobrazení údajů jako například telefonního čísla pro případ nutnosti daného učitele kontaktovat.

Případ užití: Zobrazení údajů o učiteli

V případě potřeby zjištění například telefonního čísla na učitele je tyto údaje možné zobrazit

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s libovolnou rolí

Krok	Role	Akce
1	Aktér	spustí možnost "Seznam učitelů", v následně zobrazeném seznamu klikne na ikonku informace a vybere učitele, jehož údaje hledá
2	Systém	zobrazí veškeré údaje o učiteli

Obrázek 31 – případ užití zobrazení údajů o učiteli

Editace stávajícího učitele slouží ke změně na příklad emailové adresy, telefonního čísla či hesla.

Případ užití: Editace stávajícího učitele

Při obdržení nových údajů učitele, dojde k přepsání údajů, a to jak cloudově, tak i interně v telefonu

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	spustí možnost "Seznam učitelů", v následně zobrazeném seznamu klikne na ikonku tužky a vybere učitele, kterého chce editovat
2	Systém	vytvoří formulář, kde budou předvyplněny stávající údaje spojené s učitelem
3	Aktér	změní data v těch polích, kde došlo ke změně údajů
4	Systém	POKUD jsou zadány všechny nezbytné údaje a současně správně, systém provede vytvoření dítěte do cloudové a následně interní databáze telefonu JINAK nejsou vyplněny všechny nezbytné údaje pro jeho založení, systém informuje o této skutečnosti administrátora a proces se vrací do bodu 4

Obrázek 33 – případ užití editace stávajícího učitele

Vymazání učitele využijeme v případě, kdy učitel nebude již do zařízení docházet.

Případ užití: Vymazání učitele

Při ukončení docházení učitele do spolku, je údaje nutné vymazat

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	spustí možnost "Seznam učitelů", v následně zobrazeném seznamu klikne na ikonku odpadkového koše a vybere učitele, které chce vymazat
2	Systém	zobrazí informační dialog s detaily učitele, které chce aktér vymazat
3	Aktér	potvrdí správnost informací stisknutím tlačítka "Ano"
4	Systém	vymaže učitele z cloudové a rovněž z interní databáze telefonu

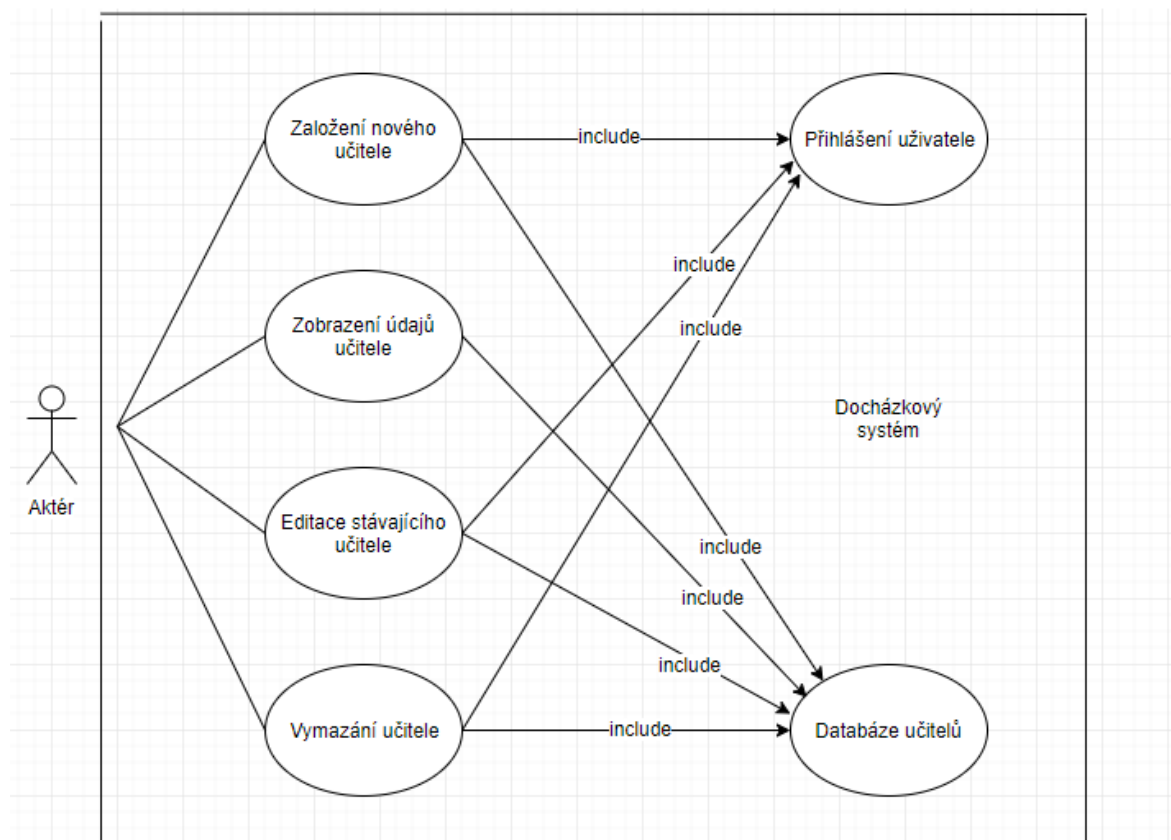
Obrázek 34 – případ užití vymazání učitele

Jako nefunkční požadavky evidujeme:

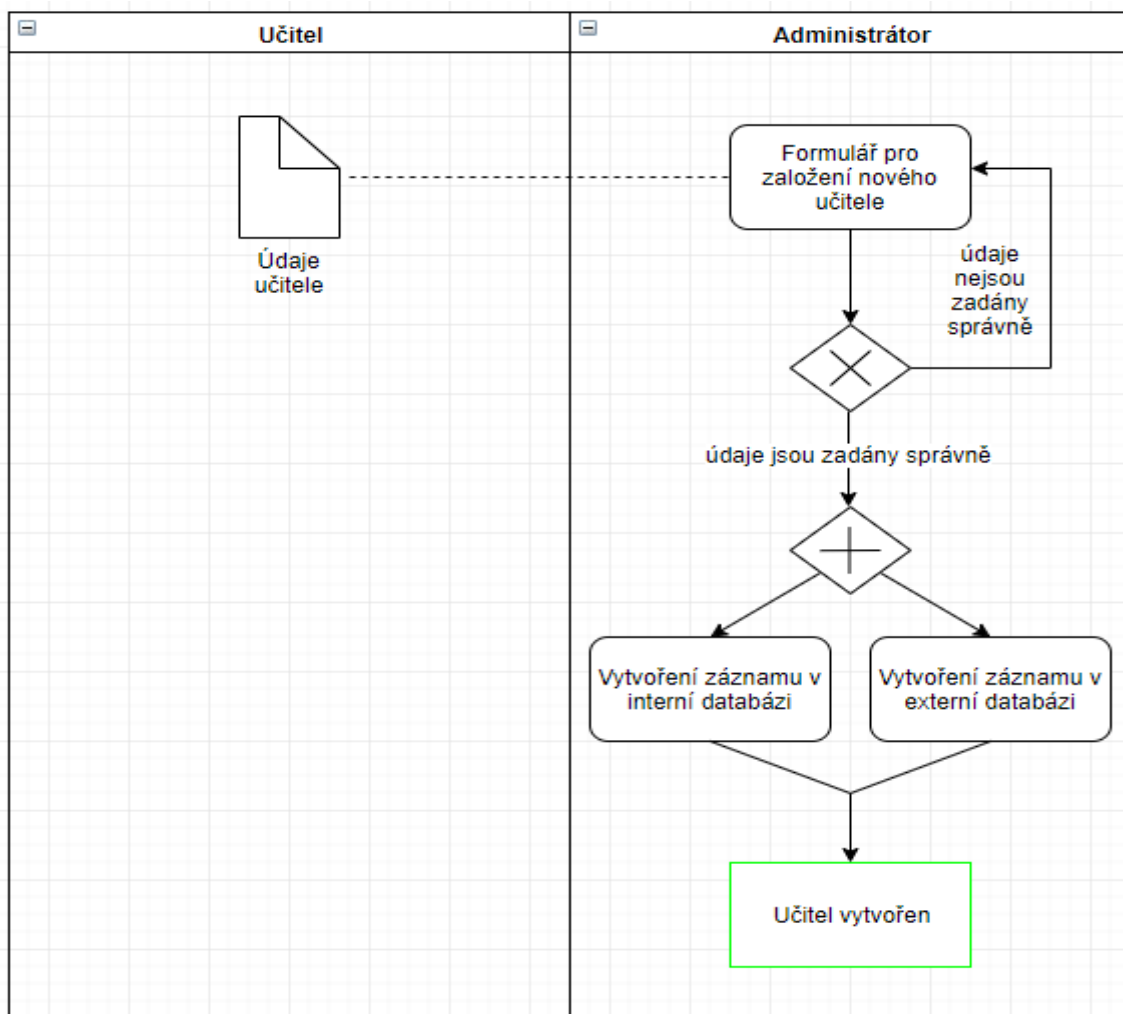
- úpravy učitele
- oprávnění

K úpravám učitele může dojít pouze tehdy, pokud má uzavřenou docházku. To znamená, že z aplikace nebude možné odstranit či editovat dítě, které systém eviduje jako přítomné.

Oprávnění zde představuje omezení, kdy přidat, editovat a vymazat dítě může pouze osoba s oprávněním administrátor.



Obrázek 35 – diagram případů užití požadavku správu učitelů



Obrázek 36 – BPMN založení nového učitele

9.1.6 Záznam docházky učitelé

Náš docházkový systém vyžaduje jak evidenci docházky dětí, které spolek navštěvují, tak rovněž evidenci docházky učitelů. Tato náležitost vychází z legislativních požadavků, kdy musí být dodržena pravidla, která jsme zmínili v požadavku záznamu docházky dětí a rovněž tyto údaje se reportují na konci monitorovacího období v tabulce, do které údaje zanášíme. Na obrázku můžeme vidět, že mezi požadované údaje řadíme tyto:

- datum
- jméno a příjmení
- čas příchodu
- čas odchodu

Hlavní rozdílem oproti evidenci docházky dětí je, že na jednom řádku evidujeme všechny přítomné učitele za monitorovaný den, čemuž musíme aplikaci přizpůsobit.

Případy užití jsou tyto:

- záznam příchodu
- záznam odchodu

Příchod je zaznamenán při prvním přihlášení učitele do aplikace určitý den. Při dalším přihlášení vybraný den již k této akci nedochází.

Případ užití: Záznam příchodu učitele

Při fyzickém příchodu učitele potvrdí tuto skutečnost akcí v aplikaci

Podmínky: 1) Aplikace je zapnutá

2) V aplikaci je přihlášen aktér s libovolnou rolí

3) Čas v aplikaci je menší nebo roven 10:00

4) U vybraného učitele nebyl dosud proveden záznam příchodu v daný den

5) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	vybere možnost "Seznam učitelů", zde vybere možnost s ikonkou "kalendáře"
2	Systém	vytvoří přihlašovací formulář
3	Aktér	zadá přihlašovací údaje
4	Systém	POKUD případ jsou zadane údaje správně, zaznamená příchod JINAK vyzve k opakování zadání a vrací se do kroku 2

Obrázek 37 – případ užití záznam příchodu učitele

Odchod je zaznamenán při odhlášení uživatele, kdy při této akci vybere možnost, že požaduje zaznamenání odchodu.

Případ užití: Záznam odchodu učitele

Při fyzickém odchodu učitele potvrdí tuto skutečnost akcí v aplikaci

Podmínky: 1) Aplikace je zapnutá

2) V aplikaci je přihlášen aktér s libovolnou rolí

3) Záznamem odchodu učitele nebude porušeno pravidlo 6x/1x

4) Vybraný učitel je evidován jako přítomný

5) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	vybere možnost "Seznam učitelů", zde vybere možnost s ikonkou "kalendáře"
2	Systém	vytvoří přihlašovací formulář
3	Aktér	zadá přihlašovací údaje
4	Systém	POKUD případ jsou zadane údaje správně, zaznamená odchod JINAK vyzve k opakování zadání a vrací se do kroku 2

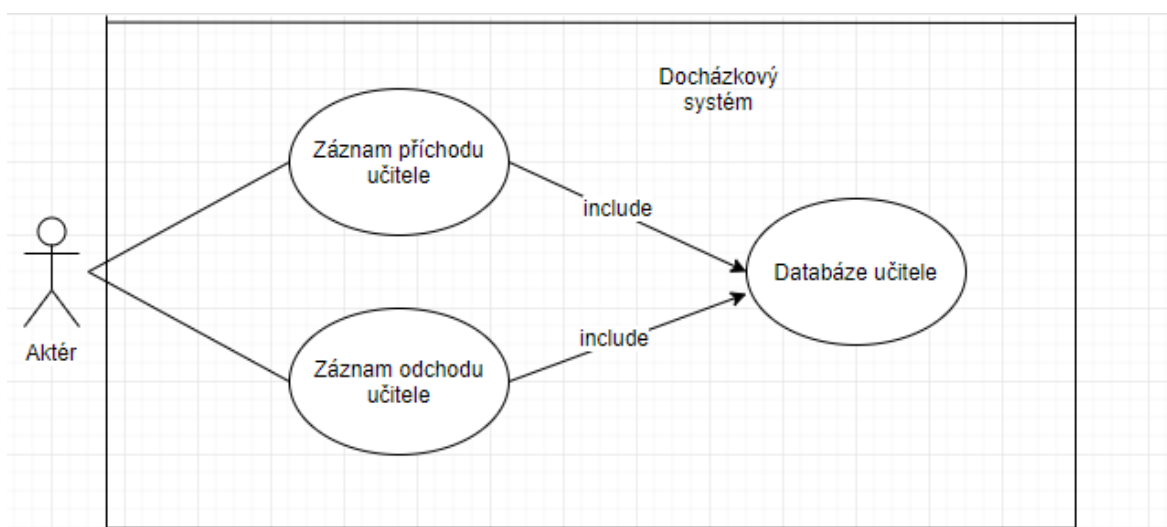
Obrázek 38 – případ užití záznam odchodu učitele

Jako nefunkční požadavky jsme identifikovali tyto:

- pravidlo 6x/1x
- počet přítomností

Pravidlo 6x/1x jsme si definovali již v požadavku záznamu docházky dětí. To znamená, že nemůžeme zaznamenat odchod učitele, pokud by nastal případ, že na jednoho učitele připadne více jak 6 dětí, nebo by nastala situace, že by školce byly nějaké děti a zároveň žádný učitel.

Počet přítomností stejný požadavek jako u záznamu docházky dětí.



Obrázek 39 – diagram případů užití požadavku záznamu docházky učitele

9.1.7 Stravné

Aplikace by současně při zaznamenávání docházky jednotlivých dětí, měla také umět evidovat počet odebíraných obědů a svačin potřebných na den. Tento požadavek se týká samotného objednávání stravného. Proto jsme v požadavku na správu dětí zavedli údaj návštěvní dny, díky kterému víme, kolik obědů budeme potřebovat na jednotlivé dny. Každé dítě, které má označenou docházku jako půldenní, má objednáno dopolední svačinu s obědem. Děti s docházkou celodenní mají ještě navíc odpolední svačinu. Objednání stravného je realizováno v průběhu týdne a na týden následující.

Případy užití u požadavku stravného evidujeme tyto:

- objednání stravného
- odhlášení stravného
- přihlášení stravného
- zobrazení stravného

Objednáním stravného je myšlena realizace předběžné objednávky stravy na následující týden u dodavatelské společnosti. Proces objednávky stravného je zahájen administrátorem jakýkoliv pracovní den v týdnu na týden následující.

Případ užití: Objednání stravného

Aktér chce provést objednání stravného

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	vybere možnost "Stravné" a následně "Objednávka stravného"
2	Systém	zobrazí plánované množství jednotlivých položek stravného na následující týden a vyzve uživatele k potvrzení tohoto množství
3	Aktér	POKUD souhlasí s uvedeným množstvím a povrdí jej JINAK upraví množství odebíraných produktů a vybere ty děti, u nichž došlo k změně a následně potvrdí objednávku
4	Systém	odešle email dodavatelské společnosti s přehledem množství objednaných produktů

Obrázek 40 – případ užití objednání stravného

Odhlášení stravného je případ, kdy dojde k odhlášení objednané stravy u dítěte, které se z libovolného důvodu nebude ve školce přítomné.

Případ užití: Odhlášení stravného

Aktér chce provést odhlášení určitého dítěte ze stravného vybraný den

Podmínky: 1) Aplikace je zapnutá
2) Telefon má přístup k internetu
3) Dítě u něhož chceme provést odhlášení stravného má stravné objednané

Krok	Role	Akce
1	Aktér	vybere možnost "Seznam dětí", zde označí ikonku "panáčka s vlaječkou" a vybere dítě u něhož chce ohlášení provést
2	Systém	vyzve uživatele k potvrzení odhlášení
3	Aktér	POKUD souhlasí a odhlášení je provedeno JINAK odhlášení není provedeno
4	Systém	upraví objednané množství, odešle zprávu o změně

Obrázek 41 – případ užití odhlášení stravného

Přihlášení stravného je taková situace, kdy naopak dojde k objednání stravy pro dítě, které jinak běžně stravu neodebírá.

Případ užití: Přihlášení stravného

Aktér chce provést odhlášení určitého dítěte ze stravného vybraný den

Podmínky: 1) Aplikace je zapnutá
2) Telefon má přístup k internetu
3) Dítě u něhož chceme provést přihlášení stravného nemá stravné objednané

Krok	Role	Akce
1	Aktér	vybere možnost "Seznam dětí", zde označí ikonku "panáčka s vlaječkou" a vybere dítě u něhož chce ohlášení provést
2	Systém	vyzve uživatele k potvrzení přihlášení
3	Aktér	POKUD souhlasí a přihlášení je provedeno JINAK přihlášení není provedeno
4	Systém	upraví objednané množství, odešle zprávu o změně

Obrázek 42 – případ užití přihlášení stravného

Zobrazení stravného slouží k náhledu celkové množství odebrané stravy na vybraný den, či u vybraného dítěte za měsíc.

Případ užití: Zobrazení stravného

Aktér chce zobrazit objednané stravné

Podmínky: 1) Aplikace je zapnutá
2) V aplikaci je přihlášen aktér s rolí administrátor
3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	vybere možnost "Stravné" a následně "Zobrazení stravného". Zde vybere datum, na které chce zobrazit množství objednané stravy
2	Systém	zobrazí plánované množství jednotlivých položek stravného

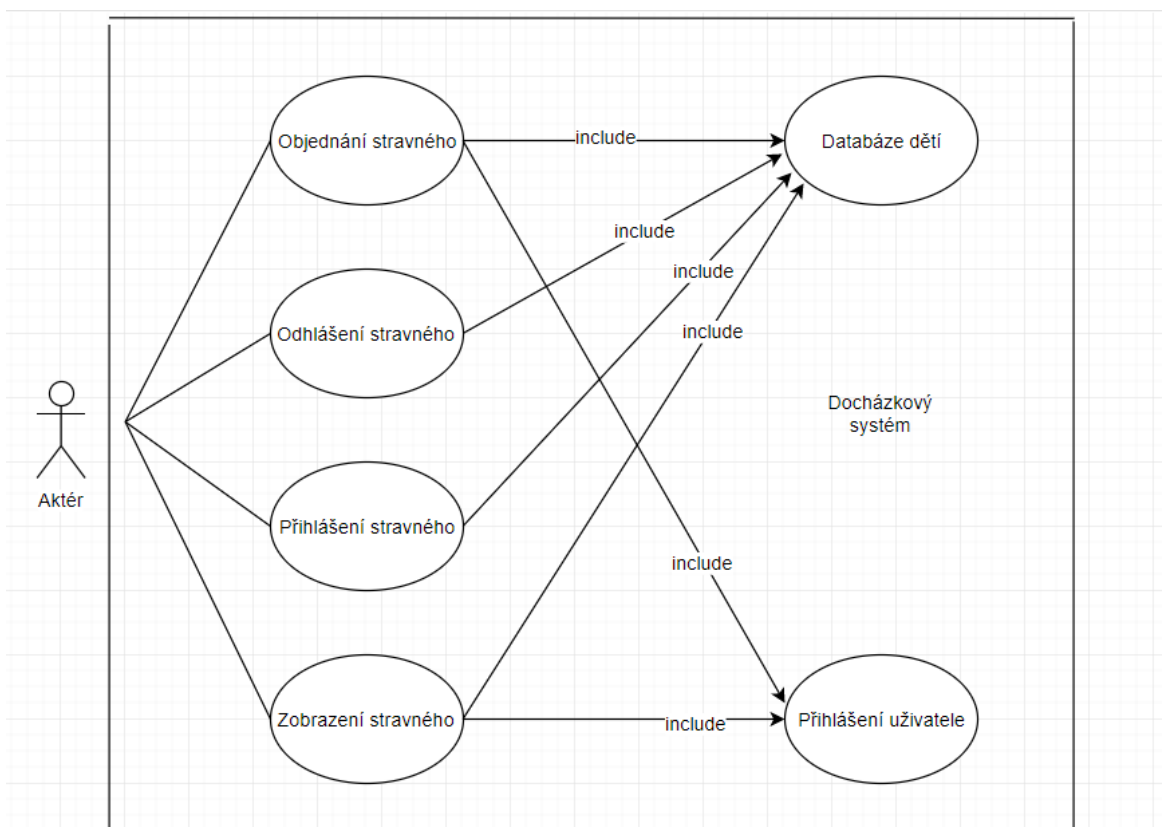
Obrázek 43 – případ užití zobrazení stravného

Rovněž u požadavku stravného evidujeme požadavky nefunkční a to následující:

- minimální množství stravného
- omezení odhlášky a přihlášky stravného
- oprávnění

Minimální výše objednané stravy je smluvně stanovena na 5 kusů obědů. Proto i v případě, že daný den budou přítomny pouze 4 děti, tak stejně dojde k objednávce 5 obědů.

Omezení ohlášení či přihlášení stravného je limitace, kdy danou akci lze provést do 16. hodiny předcházející dni, na kdy je stravné objednané.



Obrázek 44 – diagram případů užití požadavků stravného

9.1.8 Fakturace

Posledním funkčním požadavkem aplikace je tvorba faktur, za docházku do školky společně s fakturací stravného. Případ užití u stravného je pouze tento:

- tvorba faktur

Tvorba faktur je proces, kdy na základě záznamu docházky je vytvořena faktura za pobyt dítěte ve školce. Zároveň je vytvořena faktura za odebranou stravu. V rámci tohoto procesu je aktérem zkontrolována jejich částka a po schválení odeslána na email vedený u daného dítěte.

Případ užití: Odeslání faktur

Aktér chce odeslat faktury za poskytnuté služby

Podmínky: 1) Aplikace je zapnutá

2) V aplikaci je přihlášen aktér s rolí administrátor

3) Telefon má přístup k internetu

Krok	Role	Akce
1	Aktér	vybere možnost "Fakturace". Zde se zobrazí částka faktur pro každé dítě. Po kontrole stiskne tlačítko odeslat
2	Systém	odešle faktury na email rodičů

Obrázek 45 – případ užití odeslání faktur

Jediným nefunkčním požadavkem je oprávnění, kdo může tento proces iniciovat. V tomto případě to může být pouze administrátor

Tímto požadavkem a jeho analýzou jsme uzavřeli sběr požadavků, což nám dovolí přesun k samotnému vývoji softwaru.

Závěr

Náplní této práce byl projekt na zavedení docházkového systému do společnosti Školička Krásná z.s. Při výběru řešení došla společnost k závěru, že při daných nabídkách zvolí služby Marka Kaufmanna neboli řešení pomocí vývoje nové aplikace. Po nashromáždění všech dat, které byly získány studiem dokumentů, monitoringem konkurence a rozhovorem s ředitelkou zmíněné společnosti, došlo k jejich transformaci do podoby požadavků, což vedlo k zahájení samotného vývoje aplikace. Samotný vývoj byl proveden v prostředí softwaru „Android Studio“, což je bezplatné vývojové prostředí určené k tvorbě softwaru pro operační systém Android. Souběžně s vývojem v tomto softwaru proběhla i tvorba cloudové databáze, jejíž provoz je zajištěn pomocí nástroje „Google Scripts“, který tento provoz umožňuje.

V průběhu vývoje aplikace a databáze byly mezivýsledky pravidelně prezentovány a konzultovány se zadavatelem projektu. V současné chvíli je aplikace již v mateřské škole zavedena, a to v režimu testování. Aktuálně obsahuje funkce záznam docházky, správa uživatelů a nejeví vážné nedostatky, které by mohly omezit finální nasazení systému do provozu. Testováním dochází k odhalování nedostatků, které by se mohly projevit v ostrém provozu. Funkce stravného a fakturace budou implementovány v nadcházející verzích. Projekt od počátku nejeví žádné známky hrozeb, které by mohly ovlivnit jeho zdárné dokončení.

Přínosem dokončení toho projektu bude stabilní, spolehlivý software, který může společnost Školička Krásná z.s. bez omezení využívat pro svůj bezproblémový chod. Dalším přínosem této práce je možnost toto řešení distribuovat i do dalších společností s podobným zaměřením.

Závěrem tedy můžeme konstatovat, že došlo k naplnění cílů projektu. To znamenalo naleznout optimální řešení pro projekt implementace docházkového systému do společnosti Školička Krásná z.s., došlo tedy k vývoji aplikace, která plní legislativní požadavky i požadavky zadavatele. Veškeré případy užití byly zpracovány pomocí nástroje „Microsoft Excel“, diagramy využitím online nástroje „Draw.io“.

Nové dítě

Údaje o dítěti

Jméno
Například: Marek

Příjmení
Například: Hofman

Rodné číslo
Například: 9610980873

Dny a jejich typ, kdy bude dítě ve školce


<input type="checkbox"/> PO	<input type="checkbox"/> ÚT	<input type="checkbox"/> ST	<input type="checkbox"/> ČT	<input type="checkbox"/> PÁ
<input type="radio"/> 1	<input type="radio"/> 1	<input type="radio"/> 1	<input type="radio"/> 1	<input type="radio"/> 1
<input type="radio"/> 0.5	<input type="radio"/> 0.5	<input type="radio"/> 0.5	<input type="radio"/> 0.5	<input type="radio"/> 0.5

Detaily rodiče 1

Jméno
Například: Stanislav

Příjmení
Například: Hofman

Telefonní číslo
Například: 774876973



Obrázek 46 – ukázka aplikace „formulář pro nové dítě“


Přihlášení

Vítejte v aplikaci Školička Krásná

Vyberte osobu ze seznamu pro přihlášení

1 - Marek Kaufmann

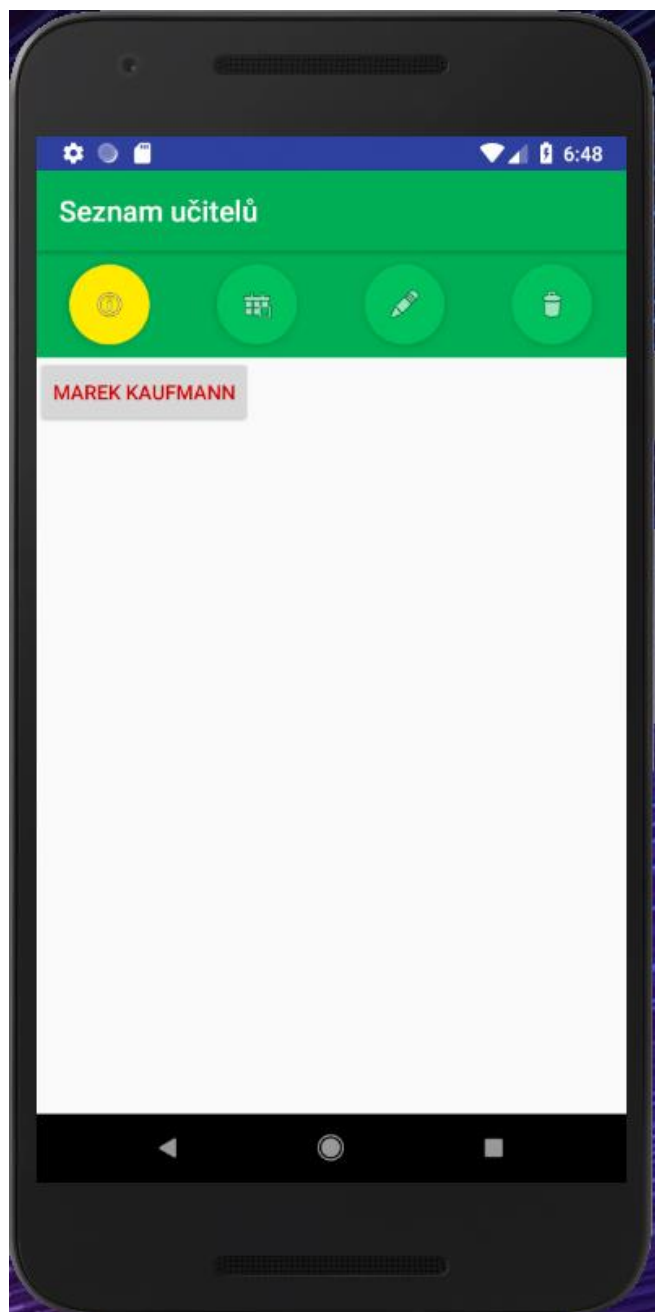
Zadejte vaše heslo



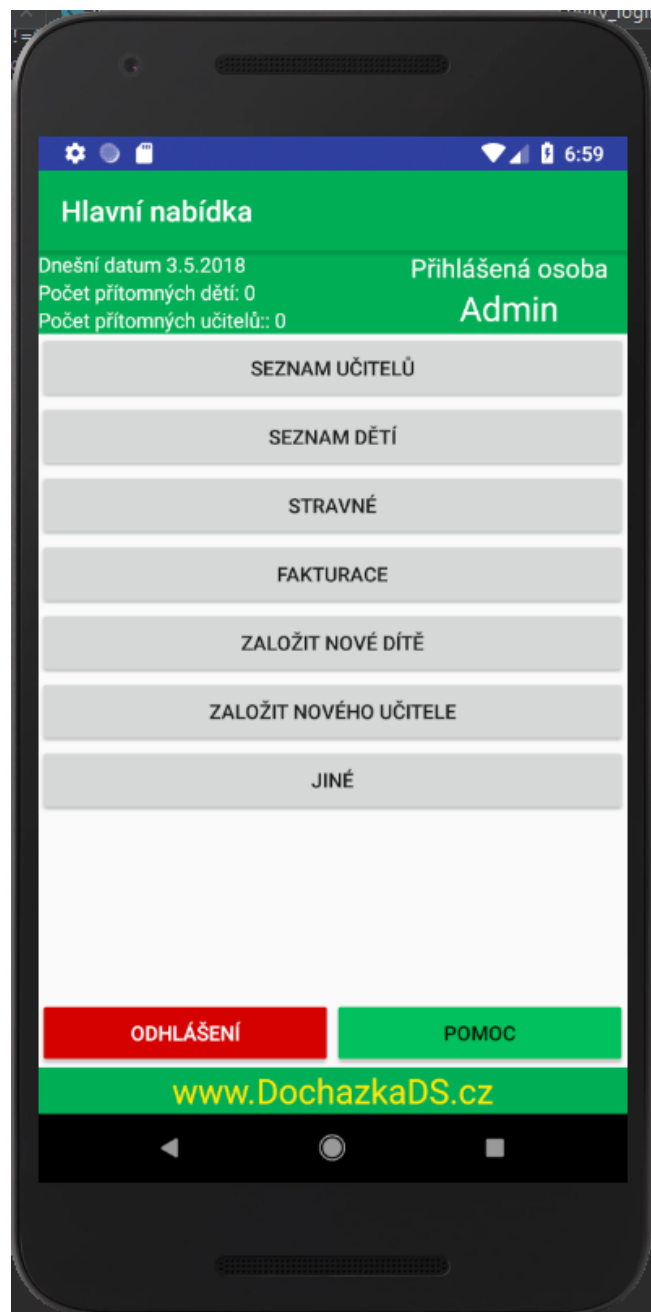
Verze databáze: 4

ZAPOMENUTÉ HESLO?

Obrázek 47 – ukázka aplikace „přihlašovací okno“



Obrázek 49 – ukázka aplikace „zobrazení seznamu dětí“



Obrázek 48 – ukázka aplikace „hlavní menu“

Seznam použité literatury

- BusinessVize. 2010. SMART aneb jak definovat cíle. *BusinessVize*. [Online] 3. 1 2010. [Citace: 3. 4 2018.] <http://www.businessvize.cz/planovani/smart-aneb-jak-definovat-cile>.
- Doležal, Jan a Krátký, Jiří. 2017. *Projektový management v praxi*. Praha : Grada Publishing, a.s., 2017.
- Kanisová, Hana a Müller, Miroslav. 2012. *UML srozumitelně*. Brno : Computer Press, 2012. 978-80-251-1083-6.
- Keller, Kevin Lane a Philip, Kotler. 2007. *Marketing management*. Praha : Grada Publishing, a.s., 2007. 978-80-247-1359-5.
- Listverse Comparision. 2016. INTERNAL VS EXTERNAL STAKEHOLDERS. *Differ Two*. [Online] 3. 5 2016. [Citace: 3. 4 2018.] <http://www.listverse.co/internal-vs-external-stakeholders/>.
- Máče, Miroslav. 2006. *Finanční analýza investičních projektů*. Praha : Grada Publishnig, a.s., 2006.
- Švaříček, Roman a Šedřová, Klára. 2007. *Kvalitativní výzkum v pedagogických vědách*. Praha : Portál, s.r.o., 2007.
- Tahal, Radek a Kolektiv. 2017. *Marketingový výzkum*. Praha : Grada Publishing, a.s., 2017.
- What is a Stakeholder?. 2017. Project Manager. [Online] 13. 6 2017. [Citace: 3. 4 2018.] <https://www.projectmanager.com/blog/what-is-a-stakeholder>.

Seznam obrázků

Obrázek 1 – tabulka investic.....	11
Obrázek 2 – investiční trojúhelník.....	11
Obrázek 3 – schéma stakeholderů.....	15
Obrázek 4 – dichotomické otázky	20
Obrázek 5 – výběr z více možností.....	20
Obrázek 6 – zcela nestrukturovaná otázka.....	21
Obrázek 7 – ukázka rozhovoru bez přípravy	22
Obrázek 8 – ukázka požadavků zanesených do softwaru	24
Obrázek 9 – bazény.....	25
Obrázek 10 – typy událostí.....	25
Obrázek 11 – typy bran.....	26
Obrázek 12 – artefakty	27
Obrázek 13 – kompletní BPMN diagram	27
Obrázek 14 – případ užití.....	28
Obrázek 15 – diagram případu užití	29
Obrázek 16 – diagram případu užití s vazbou include	29
Obrázek 17 – případ užití s vazbou include.....	30
Obrázek 18 – případ užití s vazbou extend	30
Obrázek 19 – diagram případu užití s vazbou extend.....	30
Obrázek 20 – šablona pro záznam docházky	35
Obrázek 21 – případ užití přihlášení operátora.....	42
Obrázek 22 – případ užití založení nového dítěte	44
Obrázek 23 - případ užití zobrazení údajů o dítěti.....	44
Obrázek 24 - případ užití editace stávajícího dítěte.....	45
Obrázek 25 - případ užití vymazání dítěte	45
Obrázek 26 - diagram případ užití požadavku správa dětí.....	46
Obrázek 27 – BPMN založení dítěte	47
Obrázek 28 – případ užití záznam příchodu dítěte	48
Obrázek 29 – případ užití záznam odchodu dítěte	49
Obrázek 30 – diagram případu užití požadavku záznam docházky dětí.....	50
Obrázek 31 – případ užití zobrazení údajů o učiteli.....	51
Obrázek 32 – případ užití založení nového učitele.....	51
Obrázek 33 – případ užití editace stávajícího učitele.....	52
Obrázek 34 – případ užití vymazání učitele	52
Obrázek 35 – diagram případů užití požadavku správu učitelů.....	53
Obrázek 36 – BPMN založení nového učitele	54
Obrázek 37 – případ užití záznam příchodu učitele.....	55
Obrázek 38 – případ užití záznam odchodu učitele.....	55
Obrázek 39 – diagram případů užití požadavku záznamu docházky učitele.....	56
Obrázek 40 – případ užití objednání stravného.....	57
Obrázek 41 – případ užití odhlášení stravného	57

Obrázek 42 – případ užití přihlášení stravného	58
Obrázek 43 – případ užití zobrazení stravného	58
Obrázek 44 – diagram případů užití požadavků stravného	59
Obrázek 45 – případ užití odeslání faktur.....	59
Obrázek 46 – ukázka aplikace „formulář pro nové dítě“	62
Obrázek 47 – ukázka aplikace „přihlašovací okno“	62
Obrázek 49 – ukázka aplikace „hlavní menu“	63
Obrázek 48 – ukázka aplikace „zobrazení seznamu dětí“	63

Seznam tabulek

Tabulka 1 – Webooker řešení pomocí mobilní aplikace bez závazku	37
Tabulka 2 – Webooker řešení pomocí mobilní aplikace se závazkem	37
Tabulka 3 – Webooker řešení pomocí čtečky bez závazku	37
Tabulka 4 – Webooker řešení pomocí čtečky se závazkem	38
Tabulka 5 – Dignus řešení pomocí mobilní aplikace	38
Tabulka 6 – Skolkis řešení pomocí mobilní aplikace	39
Tabulka 7 – Marek Kaufmann řešení pomocí mobilní aplikace	39
Tabulka 8 – porovnání jednotlivých řešení pomocí NPV	40
Tabulka 9 – přehled požadavků	41

Evidence výpůjček

Prohlášení:

Dávám svolení k půjčování této bakalářské práce. Uživatel potvrzuje svým podpisem, že bude tuto práci řádně citovat v seznamu použité literatury.

Jméno a příjmení: Marek Kaufmann

V Praze dne: 03. 05. 2018

Podpis:

Jméno	Oddělení/ Pracoviště	Datum	Podpis