

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Návrh konfigurátoru informačního systému

Lucie Řípová

Vedoucí: Ing. Pavel Náplava, Ph.D.
Obor: Softwarové inženýrství a technologie
Květen 2021

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Řířpová** Jméno: **Lucie** Osobní číslo: **474412**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadářavající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh konfigurátoru informačního systému

Název bakalářské práce anglicky:

Information System Configurator Design

Pokyny pro vypracování:

Navrhněte a vytvořte aplikaci, která pomůže studentům v předmětu „Informační systémy“ pochopit konfiguraci a cenotvorbu již hotových (předpřipravených) informačních systémů. Postupujte následujícím způsobem:

- 1) Prozkoumejte existující komplexní informační systémy (např. ERP nebo CRM) a zmapujte klíčové oblasti (funkčnosti), které obsahují.
- 2) Identifikujte oblasti, které je vhodné využít v rámci tvorby případové studie a návrhu systému v předmětu „Informační systémy“.
- 3) Na základě identifikovaných oblastí a po dohodě s vedoucí práce navrhněte obecný konfigurátor, který budou moct pro své studie využít studenti.
- 4) Vytvořte návrh aplikace (základní UML diagramy), na základě kterého bude možné navržený konfigurátor implementovat.
- 5) Konfigurátor implementujte alespoň v první verzi, která umožní vytvořit základní návrh konfigurace systému a spočítat náklady na jeho implementaci.
- 6) Funkčnost aplikace ověřte prostřednictvím uživatelského testování.

Seznam doporučené literatury:

[1] T. Bruckner, J. Voříšek, A. Buchalceřová a kolektiv, Tvorba informačních systémů, Principy, metodiky, architektury, Grada Publishing, 2012.
[2] L. Gála, J. Pour, Z. Šedivá, Podniková informatika - Počítačové aplikace v podnikové a mezipodnikové praxi - 3., aktualizované vydání, Grada Publishing, 2015

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D., katedra softwarového inženýrství FIT

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **11.02.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Pavel Náplava, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studentky

Poděkování

Děkuji mému vedoucímu práce Ing. Pavlu Náplavovi, Ph.D. za odborné vedení bakalářské práce, za jeho cenné rady, připomínky a čas, který mi věnoval. Také bych chtěla poděkovat své rodině a svým blízkým za podporu, kterou mi poskytovali po celou dobu mého studia.

Prohlášení

Čestně prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Veškeré použité podklady, ze kterých jsem čerpala informace, jsou uvedeny v seznamu použité literatury a citovány v textu.

Abstrakt

Bakalářská práce se zabývá návrhem a tvorbou aplikace, která pomůže studentům v předmětu Informační systémy pochopit konfiguraci a cenotvorbu již hotových informačních systémů. V první části práce jsou objasněna teoretická východiska, která se vztahují k informačním systémům. Analytická část obsahuje přiblížení výuky informačních systémů a vnímání studentů této problematiky, včetně jejich problémů s vypracováváním jejich prací. Dále obsahuje identifikaci oblastí, které je vhodné využít v rámci tvorby případové studie a návrhu systému. Nabízí porovnání již existujících systémů s konfigurátorem. Praktická část obsahuje implementaci konfigurátoru z identifikovaných oblastí. Obsahuje také základní informace pro vytváření aplikací v Power Apps.

Klíčová slova: návrh konfigurátoru informačního systému, konfigurace, informační systém, Power Apps

Vedoucí: Ing. Pavel Náplava, Ph.D.
Centrum znalostního managementu
(13393)

Abstract

This bachelor's thesis evaluates the design and production of an application that would assist students of the Information Systems subject to understand the configuration and pricing of ready-made information systems. The first part of this paper elucidates the theoretical background relating to the subject. The analytical section investigates the Information Systems teaching method as well as students' understanding of this field, including some of the challenges they may experience whilst completing their assignments. Furthermore, this thesis contains an identification of areas suitable for use during a potential case study and a system design. It offers a comparison of existing systems with the configurator. The practical part covers the implementation of the configurator from the identified areas. It also includes basic information for creating applications in Power Apps.

Keywords: information system configurator design, configuration, information system, Power Apps

Title translation: Information System Configurator Design

Obsah

1 Úvod	1	6 Požadavky a diagramy	23
2 Cíle a metodika práce	3	konfigurátoru	23
2.1 Struktura práce	3	6.1 Funkční požadavky	23
3 Seznámení s pojmy	5	6.2 Příklad užití	24
3.1 Systém	5	6.3 Doménový model	25
3.2 Informační systém	5	6.4 Diagram nasazení	25
3.3 Data	5	6.5 Návrh obrazovek	25
3.4 ICT - Information and Communication Technologies	5	7 Power Apps a implementace	
3.5 ERP – Enterprise Resource Planning	6	konfigurátoru	27
3.6 CRM – Customer Relationship Management	6	7.1 Uživatelské prostředí	28
3.7 DMS – Document management system	6	7.2 Prostor plátna	28
3.8 Maintenance	6	7.3 Sestavení aplikace	29
3.9 Power Apps	6	7.4 Data v Excel souboru	29
4 Výuka informačních systémů	7	7.5 Připojení k datům	30
4.1 Obecná výuka informačních systémů	7	7.6 Návrh aplikace	30
4.2 Výuka informačních systémů v předmětu INS	7	7.7 Funkce	31
4.3 Úvodní studie	8	7.7.1 Nejvíce používané funkce při vývoji konfigurátoru	32
4.4 Nejčastější problémy studentů	8	7.8 Ovládací prvky a jejich vlastnosti	33
4.5 Návrhy na změnu	8	7.8.1 Společné vlastnosti podle kategorie	37
5 Konfigurace	11	7.8.2 Základní vlastnosti ovládacích prvků	37
5.1 Existující systémy	11	7.8.3 Další vlastnosti ovládacích prvků	38
5.1.1 Solitea Varío	11	7.9 Proměnné	39
5.1.2 eWay-CRM	12	7.10 Power Automate a další služby	39
5.1.3 Porovnání systémů Varío a eWay-CRM	12	7.11 Uložení a publikace	41
5.2 Mapování modulů konfigurátoru	13	7.12 Sdílení a spuštění aplikace	41
5.2.1 Licence	14	8 Design a rozvržení konfigurátoru	43
5.3 Funkční bloky modulů	14	9 Testování, výsledky a	
5.3.1 Seznam funkčních modulů	14	optimalizace	47
5.4 Prostor	17	9.1 Testování konfigurátoru	47
5.4.1 Systémový moduly	17	9.1.1 Problém se sdílením a chybná databáze	47
5.5 Klienti	17	9.2 Výsledky testování	48
5.6 Uživatelé	19	9.3 Návrh budoucích optimalizací	49
5.7 Obvyklý průběh implementace	19	10 Závěr	51
5.7.1 Nacnění prací	20	Literatura	53
5.8 Odhady průběhu práce na implementaci	21	A Tabulky modulů	55
5.9 Etapy implementace	22	A.1 Moduly	55
5.10 Součinnosti	22	A.2 Implementace	58

Obrázky

5.1 Průběh implementace	20
6.1 Případy užití	24
6.2 Ukázka domain modelu	26
6.3 Deployment diagram	26
7.1 Prostředí plátna	28
7.2 Tabulky v Power apps	30
7.3 Funkce	32
7.4 Tok pro vytváření PDF souboru v Power Automate	40
8.1 První obrazovka	43
8.2 Obrazovka funkčních modulů ...	44
8.3 Obrazovka odhadů	44
8.4 Obrazovka etap	45
8.5 Obrazovka implementace	45
8.6 Obrazovka celkové ceny za implementaci	46
8.7 Obrazovka finanční analýzy	46
9.1 Analýza chyb	48
9.2 Použití konfigurátoru	49

Tabulky

5.1 Tabulka funkčních modulů	18
5.2 Tabulka systémového modulu ..	18
5.3 Tabulka klientů	19
5.4 Tabulka uživatelů	19
5.5 Ceník prací	21

Kapitola 1

Úvod

Informační systémy (dále jen IS) v této době doprovázejí většinu firem a organizací. Od papíru a tužky se přešlo k systémům, které pomáhají lépe pracovat s daty a jejich zpracováním. Je to dané nástupem průmyslu 4.0, kdy se doba digitalizuje. S tím souvisí automatizace výroby, úspory času a peněz, zvýšení flexibility i změny na pracovním trhu. Zaměstnance nahrazují robotické systémy, které stále více rozvíjejí vnímání svého okolí. Pro dobré fungování je potřeba spolehlivá síť s cloudovými úložišti a kvalitní hardware. IS pak slouží jako ekosystém, který software a hardware spojuje.

Na trhu je mnoho firem, které se specializují na vytváření IS a poskytují je těm, kteří je potřebují. IS pomáhají v řadě odvětví a dokáží propojovat různé okruhy (například výrobu, obchod, personalistiku) tak, aby firma, která je používá prospívala. Většinou mají IS za cíl ušetřit peníze, poskytnout uživatelsky přívětivé prostředí pro práci, komunikaci a také optimalizovat chod firmy. Rozlišujeme proto dva hlavní systémy. Jeden je zaměřený na optimalizaci, organizaci, řízení a plánování chodu firmy. Druhý je určený především pro vztahy se zákazníky, řešení jejich potřeb a komunikaci.

Je důležité také rozlišit IS a aplikaci. IS je složen většinou z několika modulů obsahujících funkčnosti, které mezi sebou mohou mít návaznosti. Pracuje se se společnou databází, takže je důležité postupovat pečlivě, aby chyby neovlivnily i ostatní, kteří v systému pracují. IS pak běží nejčastěji na serveru a komunikace s ním je prostřednictvím režimu klient-server. Výhoda je v tom, že pokud má firma více poboček, tak se jakákoliv informace zadaná do systému objeví ve všech ostatních. Aplikace na rozdíl od IS pro uživatele znamená, že na ní pracuje sám a sám odpovídá za výsledek.

V současné době, kdy se vše digitalizuje, se logicky vyvíjí i výuka informačních systémů. Jelikož je tento obor velmi proměnlivý, tak se musí často aktualizovat a doplňovat o nové informace. Studenti se učí, jak teoreticky, tak prakticky navrhovat vlastní IS. V tomto směru je občas těžce pochopitelné si představit, co takový IS může obsahovat a jaká je jeho cena. Na to navazují další práce, které mohou být opomenuty, studenti s nimi nemusí počítat a pak můžou být překvapeni finální cenou a tím, kolik práce je potřeba na vytvoření IS. Na základě zlepšení výuky vznikla motivace vytvořit konfigurátor IS.

Konfigurátor bude mít k dispozici všechny potřebné funkční moduly, které studenti budou používat při návrhu vlastních IS. Dále je má provést všemi

fázemi vývoje IS a donutí je popřemýšlet nad každou fází, funkcionalitou, činností i cenou. Díky tomu, pak pro studenty nebude problém si uvědomit, jak může být systém komplexní a udělají si o problematice lepší představu, kterou budou moci využít ve svých pracích v rámci výuky, nebo následně také v praxi.

Kapitola 2

Cíle a metodika práce

Hlavním cílem práce je vytvořit konfigurátor IS. Nejdříve byl vytvořen průzkum výuky IS, jak ji vnímají studenti a jestli by konfigurátor využili v tvorbě jejich prací. Dále také prozkoumat existující IS a zmapovat klíčové oblasti, které obsahují a kde je vhodné je využít. Po prozkoumání systémů a oblastí navrhnout a vytvořit konfigurátor, který pomůže studentům v předmětu Informační systémy pochopit konfiguraci a cenotvorbu již hotových IS.

Dalším cílem této práce je vytvořit návod, jak pracovat v prostředí Power Apps, ve kterém jsem konfigurátor vytvořila. Tento nástroj od společnosti Microsoft poskytuje spoustu možností v tvorbě aplikací bez předchozích znalostí programovacích jazyků. Jelikož mezi studenty tato platforma není tolik rozšířená, obsahuje tato práce základy pro tvorbu aplikací právě v Power Apps.

Práce nabídne přehled o IS a informacích, které jsou důležité pro porozumění této problematice. Zaměří se na výuku IS a poskytne návrh na změnu. V rámci výuky seznámí s úvodní studií a nejčastějšími problémy studentů. Dále zmapuje moduly, které budou využity v tvorbě IS studentů, jejich funkcí a prostředí. Poskytne náhled do náročnosti, průběhu implementace, nasazení IS a jaké činnosti jsou k tomu zapotřebí. Praktická část popisuje implementaci konfigurátoru a tvorbu aplikací v Power Apps.

2.1 Struktura práce

Práce je rozdělena do dvou hlavních částí.

V první části jsou popsány základní informace, které provázejí tuto práci. Je zde popsána výuka informačních systémů a jak je studenti vnímají. Přibližuje úvodní studii, kterou studenti vypracovávají. Problémy, se kterými se studenti potýkají a jak s tím může konfigurátor pomoci. Důležitá část je konfigurace, kde se nachází mapování modulů v konfigurátoru a porovnání modulů s již existujícími IS. V jednotlivých modulech jsou provázány funkčnosti a jejich ceny. Představuje fáze vývoje IS a činnosti potřebné k jeho správnému fungování. Jsou zde vyobrazeny požadavky, diagramy a návrhy obrazovek.

Druhá část představuje implementaci konfigurátoru včetně základního návodu, jak vytvářet aplikace v Power Apps. Od prostředí, ve kterém je

konfigurátor implementován, přes sestavení aplikace, až po uložení a publikaci. Nabídne přehled funkcí konfigurátoru a jak je možné je využít k vytváření vlastních aplikací. V této části jsou shrnuté výsledky práce a další možnosti, jak konfigurátor pro další verze optimalizovat.

Kapitola 3

Seznámení s pojmy

Tato kapitola vysvětluje základní pojmy jako je systém, informační systém, druhy systémů, informační a komunikační technologie. Dále pak popisuje data, která jsou důležitým prvkem jakéhokoliv systému a Power Apps, ve kterém je konfigurátor vytvořen.

3.1 Systém

"Systém je množina elementů, které jsou mezi sebou svázány nějakým vztahem, respektive vazbou, a jako celek má tento systém vztah ke svému okolí. Systém není jen souhrnem prvků, ale entit, jejichž vzájemné vztahy vytvářejí celek."[1]

Systém může pomoci zjednodušit práci, nebo zpracovávat informace pro jejich opětovné využití. Nevnímáme ho jen v IT oboru, systém může být i biologický či fyzikální. Hlavní je dívat se na něj jako na celek složený z jednotlivých částí a interakcí mezi nimi.

3.2 Informační systém

Informační systém představuje prostředky pro efektivní řízení a růst firmy či instituce. Usnadňuje přehlednější administraci, řízení vztahů se zákazníky a spolupráci napříč jednotlivými odděleními.[2]

3.3 Data

Data si lze představit jako reprezentaci informací pro komunikaci a zpracování. Mohou být reprezentována libovolnými znaky, nemají samostatně význam, jen pokud jim dáme informace, tak z nich pak můžeme odvozovat znalosti. [1]

3.4 ICT - Information and Communication Technologies

"Informační a komunikační technologie, zkráceně ICT, zahrnují veškeré informační technologie používané pro komunikaci a informatiku (práci s informa-

cemi). Původní koncept byl doplněn o prvek komunikace, kdy mezi sebou začaly komunikovat jednotlivé počítače či uzavřené sítě. ICT jsou jak hardwarové prvky, tak softwarové vybavení." [3]

3.5 ERP – Enterprise Resource Planning

"Systém plánování podnikových zdrojů, kam lze zahrnout skladové hospodářství a řízení zásob nebo zboží, řízení lidských zdrojů nebo finanční řízení podniku. Přestože se takto označuje převážně software, může se jednat také o aplikaci řízení a organizaci podniku." [4]

3.6 CRM – Customer Relationship Management

"Hlavní náplní práce CRM systémů jsou vztahy se zákazníky. Primárními úkoly těchto systémů je shromažďovat a zpracovávat data o zákaznících, pochopit jejich potřeby a komunikovat s nimi." [4]

3.7 DMS – Document management system

"Počítačový systém určený ke správě elektronických dokumentů a/nebo digitalizovaných papírových dokumentů, tj. např. dokumentů převedených do digitální podoby skenováním." [5]

3.8 Maintenance

Maintenance zahrnuje nárok na novou verzi IS. Platí se ročně a standardně je cena vypočítána z ceny všech získaných komponent.

3.9 Power Apps

Power Apps je platforma pro rychlý vývoj aplikací. Je možné zde vytvářet vlastní aplikace pro své potřeby. Aplikace mohou být připojeny k obchodním datům uloženým buď v datovém úložišti (Microsoft Dataverse), nebo v různých online a lokálních zdrojích dat.

Aplikace sestavené pomocí Power Apps nabízejí široké využití a funkce pracovních postupů. Umožňují automatizovat dříve manuální procesy bez nutnosti psaní kódu. Mají navíc přizpůsobivý design a mohou bez problémů běžet v desktopovém prohlížeči nebo na mobilních zařízeních. [6]

Kapitola 4

Výuka informačních systémů

Tato kapitola popisuje výuku IS včetně návrhů na změnu. Dále představuje projekt, který studenti vypracovávají. Je rovněž doplněna o problémy, které byly v rámci počáteční analýzy zjištěny u studentů, kteří již tento předmět absolvovali.

4.1 Obecná výuka informačních systémů

Obecná výuka IS se zaměřuje na ujasnění pojmů a principů v tomto odvětví a následné převedení teorie do praxe. Studenti jsou seznámeni se strategiemi, které se snaží ve svých projektech uplatňovat. Studenti mají za úkol v průběhu semestru navrhnout vhodné řešení podle předem daného tématu a využívat znalosti, které získají na přednáškách a cvičeních. Řešení obsahuje analýzu firmy, ať už jsou to silné či slabé stránky, finanční analýzu, rizika, návrh informačního systému a další informace pro uspokojení potřeb zákazníka.

4.2 Výuka informačních systémů v předmětu INS

Výuka o IS v předmětu Informační systémy, se skládá z přednášek a cvičení. V přednáškách se snaží přiblížit teoretické pojmy a strategie, které by studenti měli znát, aby mohli vytvářet smysluplné projekty. Do předmětu jsou zakomponovány vstupy externistů, kteří na reálných příkladech, ze své zkušenosti popisují, jak je v praxi důležité znát danou problematiku IS.

Cvičení jsou realizována jako konzultace s různými členy firmy. Cílem konzultací je sesbírání co nejvíce informací o fungování dané firmy, aby studenti mohli navrhnout správné řešení.

V semestru je vypracováván projekt, který má za cíl prohloubit znalosti o IS získaných na základě přednášek a cvičení. Studenti se mohou nechat inspirovat z předem připravené úvodní studie. Projekt je realizován jako případová studie, která analyzuje potřeby zákazníka a nabízí řešení, které zákazníkovi přinese návratnost investic. Vypracovávané kapitoly závisí na tom, s kým z firmy zákazníka studenti konzultují své projekty. Studenti si vyzkouší, jak si správně vybrat jednotlivé komponenty a moduly do svého IS, který je obsažen v rámci řešení.

Po prezentaci a schválení úvodní studie zákazníkem je vytvořen prototyp, který pomůže zákazníkovi si lépe představit funkčnost navrženého IS a přínosy nového řešení.

4.3 Úvodní studie

Úvodní studie je dokument, který studenti vypracovávají v předmětu Informační systémy. Studie má za cíl analyzovat potřeby zákazníka a navrhnout řešení. V řešení je zahrnut návrh IS, analýza firmy a návratnost investic.

Studie je rozdělena na několik částí. Je to obecná část, manažerské shrnutí, specifikace požadavků, situační analýza, vize řešení, rizika a roadmapa projektu.

V obecné části je popis verzí, včetně popisu revizí, obsah dokumentu, slovník a přílohy. Manažerské shrnutí, obsahuje stručné a přehledné informace o projektu. Ve specifikacích požadavků je zahrnuto vstupní zadání, záměr, aktuální problémy, ICT požadavky, milníky, požadavky na změnu, technologická omezení. V situační analýze je SWOT, FURPS+ analýza.

Vize řešení nabízí pohled na naplnění očekávání zákazníka, zasazení do strategie zákazníka, potenciál, využitelnost systému, cílový stav a klíčové přínosy.

Další část práce je věnována rizikům, ať už se jedná o rizika ze strany zadavatele, nebo projektu.

Roadmapa rozděluje projekt do etap, definuje součinnosti, harmonogram, jeho milníky a odhad nákladů na realizaci. Úvodní studie je zakončena závěrem, kde je celý projekt vyhodnocen.

4.4 Nejčastější problémy studentů

Studenti, ač jsou obeznámeni se všemi informacemi, tak mají problém správně uchopit dané téma. Ze začátku jsou problémem otázky, na které se mají ptát na cvičeních členů firmy, jelikož netuší, co všechno tvorba IS obnáší. Následně se snaží odhadnout, jak dlouho bude tvorba IS trvat. Problém je v tom, že studenti obvykle nemají přehled, kolik času jednotlivá práce zabere. Dalším problémem je nacenění prací. Studenti si hrají s náhodnými čísly, aby konečné řešení zapadalo do výsledné cash flow. Výsledkem mohou být nereálné ceny, časy jednotlivých fází, nebo IS, který nenaplní očekávání zákazníka.

4.5 Návrhy na změnu

Konfigurátor má pomoci studentům pochopit všechny náležitosti IS.

Konfigurátor obsahuje vybrané funkční moduly a funkčnosti, které si mohou studenti zakomponovat do návrhu vlastního IS v rámci projektu. Studenti se musí zamyslet nad tím, jaké moduly potřebují, protože moduly navíc stojí čas a peníze. Konfigurátor nabízí předpřipravené ceny prací, které reflektují reálné

Kapitola 5

Konfigurace

V první části této kapitoly jsou představené již existující IS, jejich porovnání a promapování modulů, které budou využity v konfigurátoru. K modulům jsou představeny funkčnosti, jejich ceny a návaznosti. Definována jsou prostředí díky kterým je možné navázat IS na externí systémy. Dále na jakých zařízeních je možné používat vytvořené IS.

V druhé části je nastíněn obvyklý průběh implementace a z jakých fází se skládá.

5.1 Existující systémy

Inspirovala jsem se dvěma systémy, které už jsou na trhu zavedené. Jsou jimi Solitea Vario a eWay-CRM.

5.1.1 Solitea Vario

Solitea Vario je variabilní ERP systém, který je možné přizpůsobovat, podle toho, co se od něj očekává. V systému jsou obsaženy finance, účetnictví, CRM, obchodní agendy, personální a mzdový systém, evidence majetku, sledování výrobního procesu, evidence servisů a půjčovny.

Moduly systému Vario.

- Finance – moduly pro kompletní účetní agendu, přehlednou evidenci majetku a controlling.
- Zásobování – Správa produktů, skladování evidence a propracovaný systém nákupu zboží a materiálů.
- Výroba a servis – řešení pro všechny typy výroby. Sledování výrobního procesu a evidence materiálu.
- Řízení – komunikace, sdílení informací, přenosy dat, reporting.
- Lidské zdroje – personální agendy, zpracování mezd.
- Obchod – evidence zákazníků, nákupních zvyklostí, obchodní historie, evidence o plnění objednávek. [7]

■ 5.1.2 eWay-CRM

EWay-CRM je, jak už z názvu vypovídá, CRM systém. Nabízejí tři verze. EWay-CRM pro Outlook, eWay-CRM Web a eWay-CRM Mobile. Verze pro Outlook je plug-in zasazen uvnitř Microsoft Outlook. EWay-CRM Web je webová aplikace, kterou je možné otevřít v jakémkoli prohlížeči. Mobilní verze umožňuje přístup k datům prostřednictvím mobilních telefonů.

EWay-CRM poskytuje řízení vztahů se zákazníky, obchodní příležitosti a projekty. Obsahuje marketingový modul, který umožňuje zasílat hromadné e-maily. Je možné ho napojit na ERP systém a propojit s dalšími nástroji prostřednictvím API.

Moduly systému eWay-CRM.

- Kontakty – evidence, úprava obchodních kontaktů.
- Společnosti – správa firem (odběratelů, dodavatelů, partnerů, konkurentů). Integrace s obchodním rejstříkem a možností založit společnost.
- Příležitosti – evidence poptávek. Obsahuje funkci workflow, která definuje seznam kroků obchodního případu a pomáhá s jeho řešením.
- Projekty – evidence a správa projektů.
- Marketing – umožňuje efektivní a cílenou komunikaci s klienty. Vytváří personifikované e-maily a propojuje s aktivitami.
- Deník – evidence poznámek a úkolů.
- Dokumenty – správa všech souborů.
- Reporty – přednastavené sestavy, které nabízí různé pohledy na uložená data.
- API – navázání na nástroje už fungující ve společnosti. [8]

■ 5.1.3 Porovnání systémů Vario a eWay-CRM

Vario je systém pro plánování podnikových zdrojů. EWay-CRM se zabývá řízením vztahů se zákazníky. Ač jsou tyto systémy odlišné svým zaměřením, některé moduly mají společné. Oba systémy potřebují evidovat kontakty, ať už napříč firmou, nebo klienty. Potřebují mít k dispozici informace a získávat je přehledně například formou reportů. Oba systémy je možné napojovat na již existující systémy, které jsou ve firmě zavedené. Liší se hlavním záměrem.

Vario obsahuje moduly pro výrobu propojenou se systémem pro sběr požadavků na výrobu a sklad, což umožňuje optimalizaci logických procesů a automatizaci výpočtů potřeb na suroviny, nebo výrobky. Nabízí reálný pohled na kapacitní možnosti firmy a plánování strategií. Na optimalizaci a sledování výrobního procesu se váže oblast servisu a údržby.

Pro systém eWay jsou to nepotřebné moduly, jelikož všechny firmy nemusí vyvíjet fyzický produkt a nepotřebují modul pro výrobu a údržbu. EWay

nabízí moduly v rámci zlepšení vztahů se zákazníky. Jejich moduly umožňují efektivní a cílenou komunikaci. V modulu společnosti mají možnost kontrolovat a přizpůsobovat se trhu. Jsou schopni propojovat soubory, kontakty s projekty a plánovat strategie. Sledovat všechny události s vazbou na obchodní případy.

5.2 Mapování modulů konfigurátoru

Z porovnání již existujících modulů systémů vzešel návrh modulů konfigurátoru.

- Ekonomika navazuje na evidenci majetku, faktur a je zde přidána evidence vozového parku.
- Obchod vede záznamy o zákaznících, dodavatelích včetně řízení vztahů jako u obou systémů. Dále pak eviduje informace o skladu.
- Personalistika představuje modul pro řízení a rozvoj lidských zdrojů, což je možné vidět i u systému Vario.
- Modul Výroby je funkčnostmi podobný jako u Varia. Stará se o optimalizaci výroby a zdrojů.
- Modul Projektů se stará o informace ohledně řízení, správy a úloh na jednotlivých projektech, jako v systému eWay.
- Modul Služeb, ve kterém je zahrnut helpdesk a dispečink, což platí pro konfigurátor i pro Vario.
- Reporting využívají oba systémy. Je to efektivní způsob získání informací, které jsou potřeba. Proto byl také zahrnut.
- Vlastní vývoj byl zahrnut pro možnost vytvořit další funkčnosti, které nejsou definované. [8]

K těmto funkčním modulům byly zařazeny systémové moduly, klienti a uživatelé. Díky systémovým modulům je možné napojovat IS na externí systémy, které firma využívá, to je zahrnuté v systémech Vario i eWay-CRM. Klienty se rozumí vývojové prostředí jako Java, .NET, pokud bude zákazník chtít webového klienta, nebo mobilního. Uživatelé představují počet uživatelů, kteří budou mít licenci na konfigurátor.

Vzhledem k problematice a potřebám studentů v projektech byl vyloučen modul, který spravuje finance. Pomocí toho je možné sledovat a zobrazovat stavy toků v penězích. Je možné evidovat peněžní toky a vyhodnocovat, což je pro potřeby ve skutečném světě neodmyslitelné. Modul nebyl zařazen kvůli složitosti a kvůli tomu, že pro studenty není potřeba.

■ 5.2.1 Licence

Licence se používají buď v rámci modulů, nebo podle počtu uživatelů v systému. U licencování na uživatele mohou uživatelé používat všechny nástroje. U modulového modelu, je platba jen za funkčnosti, které jsou v systému zahrnuty.

Do celkové ceny je potřeba zahrnout i maintenance, viz Kapitola 3. Poplatek je 20% z ceny pořízených modulů za každý rok provozu IS.

■ 5.3 Funkční bloky modulů

Každý z modulů má funkční bloky obsahující sadu agend a podpůrných funkcí. Jsou vybírány podle toho, z jakých částí má být vlastní IS postaven. Bloky mohou fungovat samostatně, ale je třeba si uvědomit i provázanost s jinými bloky a jejich agendami. Pro zvýšení uživatelského komfortu, nebo získání lepší funkčnosti je vhodné se zamyslet i nad využitím dalších návazných modulů.

Funkční bloky se liší se podle toho, jestli jsou obsaženy v jádru, nebo ne. Jádro lze chápat jako obsah všech základních funkcí systému. Bloky obsaženy v jádru není potřeba vybírat, ty jsou automaticky zahrnuty. Funkční blok neobsažený v jádru má vlastní peněžní hodnotu, jinak je cena jádra pevně daná a jsou tam započítány všechny bloky v něm obsažené. S těmito bloky je možné rozšiřovat funkčnost systému.

■ 5.3.1 Seznam funkčních modulů

Seznam funkčních bloků a funkčností, které je možné v konfigurátoru využít viz Tabulka 5.1. U některých bloků je uvedeno i vhodné spojení, nebo návaznost. Pokud bude vybrán funkční blok s obsaženou návazností, tak je do IS studentů zahrnut i navazující funkční blok. Je třeba se zamyslet, jestli bude daný blok opravdu využit a jestli má smysl k němu dodávat doporučený blok.

■ Ekonomika

- **Pokladna** – evidence tisku dokladů, evidence přijatých a vydaných faktur. S tím je vhodné spojit v modulu obchod zákazníky a dodavatele.
- **Majetek** – jednoduchá evidence majetku a s tím spojené funkčnosti. Pro tento blok je vhodné popřemýšlet nad návazností na plánování výroby.
- **Vozový park** – seznam aut s parametry, rozšiřuje majetek o specifika evidence vozidel, která jsou spojena například s leasingem, využíváním soukromých vozidel a způsoby používání.
- **Fakturace** – jednoduchý modul umožňující vystavovat faktury.

- **Kniha jízd** – slouží k vedení evidence jízd uživatelů firemními nebo soukromými vozidly. Agendu je možné přes externí rozhraní propojit pro účely automatizace s dalšími systémy.[9]

■ **Obchod**

- **Zákazníci** – evidence zákazníků, ať už jde o adresář, kontakty a seskupování. Pro zákazníky je možné využít blok CRM.
- **Dodavatelé** - evidence dodavatelů.
- **CRM** – funkční blok pro řízení obchodních příležitostí. Od evidence poptávek přes centralizaci komunikace až po uzavření nebo neuzavření zakázky. Součástí je podpora segmentace a klasifikace zákazníků, včetně detailní analýzy jejich historie. Při využití Uživatelského portálu je možné agendu zpřístupnit obchodním zástupcům.
- **Sklady** – pokud se firma zabývá nákupem, nebo prodejem, umožňuje funkční blok jednoduchou evidenci položek, lokality, umístění, možnost hlídání zásob a různé přesuny.
- **E-shop** – umožňuje nakupovat přes web a není nutné komunikovat telefonicky, nebo emailem s obchodními zástupci. Propojení s ekonomikou dává automatickou fakturaci zboží zákazníkům.[9]

■ **Personalistika**

Uživatelé, kteří fungují v systému.

- **Lidské zdroje** – evidence lidských zdrojů, které jsou ve firmě. Mohou to být zaměstnanci organizace nebo externí zdroje. V případě zaměstnanců je vhodné navázat na systémový modul identity management, který zajistí synchronizaci se serverem pro evidenci uživatelů. Je možné provázat i s agendami, kde se lidské zdroje využívají (projekty, výroba,..).
- **Organizační struktura** – zachycování vazby a nadřazenosti mezi zaměstnanci.
- **Docházka** – vedení docházky, plánování dovolených. Je možné navázat na projekty, výrobu.[9]

■ **Výroba**

Podpora jednoduché výroby a montáže.

- **Zdroje** – evidence zdrojů, které jsou ve firmě. Možné provázání s majetkem. Zdroje jsou využity při výrobě, případně v projektech.
- **Řízení výroby** – kontrola a řízení plánu výroby.

- **Plánování výroby** – plány výroby tak, aby se zamezilo plýtvání zdrojů, bylo vždy dost materiálu a zákazníci nemuseli dlouho čekat na produkt. [9]

■ Projekty

Řízení a kooperace týmů.

- **Řízení projektů** – zastřešující funkčnost, která umožňuje definovat, plánovat, řídit projekt.
- **Workflow, úlohy** – lze automatizovat vybrané činnosti, úkolovat členy týmu, plánovaně spouštět vybrané aktivity.
- **Správa dokumentů** – ukládání a řízení dokumentů.
- **Komunikace** – interní komunikace. Přes systémový modul zasílání zpráv je možné napojit systém na poštovní, nebo jinou komunikační platformu. Jejich prostřednictvím pak lze komunikovat s různými subjekty a veškerou komunikaci v systému uchovávat. [9]

■ Služby

- **Helpdesk** – požadavky zákazníků, personálu.
- **Dispečink** – telefonní číslo a adresa, přes které se řeší požadavky a distribuují se dál.
- **Servis** – evidence kontrol, interní informace externistům o pravidelném měření, nebo certifikaci. Společně s Helpdeskem pomáhá evidovat pravidelné, nebo nepravidelné zásahy zákazníků. Propojením s Workflow lze pravidelné činnosti automatizovat.[9]

■ Reporting

Umožnění definování obsahu, nad kterým lze vytvářet analýzy dle potřeb uživatelů. Čerpání dat je možné z interních systémů přes Export/Import, nebo pomocí externích systémů.

- **Business Intelligence** – Manažerský informační systém pokročilé analýzy.
- **Uživatelský portál** – portál, který umožňuje komunikaci. Intranetové stránky, které mohou obsahovat informace z obsažených funkčních bloků v systému. Pro řízení přístupů je vhodné zahrnout modul personalistika.
- **Tvorba reportů** – například seznamu dodavatelů. Obsahuje celou řadu předpřipravených reportů, které je možné upravovat. Je také možné vytvářet vlastní reporty.[9]

■ Vlastní vývoj

System je možné pomocí modulu Vlastní vývoj rozšířit o nové funkčnosti, které nebyly zahrnuty ve stávajících.

- **Uživatelský vývoj** – umožnění vytvořit novou funkčnost, nebo upravit stávající.
- **Webové služby** – napojení na externí systémy.
- **Export/Import** – možnost využít předpřipravené, nebo vytvořit nové, dle potřeby. Při napojení na Workflow je možné modul zautomatizovat.[9]

■ 5.4 Prostředí

Prostředí dává možnost napojovat IS na externí systémy, různě ho upravovat a používat na více zařízeních.

■ 5.4.1 Systémový moduly

Tabulka 5.2.

- **API** – programovatelné rozhraní, umožňující obousměrnou komunikaci s různými systémy.
- **Import/Export** – Import/Export z různých formátů v rámci souborů.
- **Zasílání zpráv** – propojení na poštovní servery. V rámci systému lze zprávy zasílat.
- **Identity Management** – pokud už databáze ve firmě existuje, tak se nemusí zakládat nová, ale napojí se na existující. [9]

■ 5.5 Klienti

Klienti jsou potřeba při používání systému na různých zařízeních. Vybrané agendy je možné používat nejen z běžných zařízení, ale také například na čtečkách čárových kódů. Viz Tabulka 5.3.

- **Webový klient.**
- **Aplikace Java.**
- **Aplikace .NET.**
- **Mobilní klient Android.**
- **Mobilní klient iOS.** [9]

Funkční modul	Funkční blok	Jádro	Cena v Kč	Návaznost
Ekonomika	Pokladna	Ano	0	Ne
Ekonomika	Majetek	Ano	0	Ne
Ekonomika	Vozový park	Ne	5 000	Ne
Ekonomika	Fakturace	Ano	0	Ne
Ekonomika	Knihy jízd	Ne	5 000	Vozový park
Obchod	Zákazníci	Ano	0	Ne
Obchod	Dodavatelé	Ano	0	Ne
Obchod	CRM	Ne	20 000	Ne
Obchod	Sklady	Ano	0	Ne
Obchod	E-shop	Ne	15 000	Uživatelský portál
Personalistika	Lidské zdroje	Ano	0	Ne
Personalistika	Organizační struktura	Ano	0	Ne
Personalistika	Docházka	Ne	5 000	Identity systém
Výroba	Zdroje	Ano	0	Ne
Výroba	Řízení výroby	Ne	15 000	Plánování výroby
Výroba	Plánování výroby	Ne	15 000	Ne
Projekty	Řízení projektů	Ne	15 000	Ne
Projekty	Workflow/ úlohy	Ne	20 000	Ne
Projekty	Správa dokumentů	Ne	20 000	Ne
Projekty	Komunikace	Ne	10 000	Zasílání zpráv
Služby	Helpdesk	Ne	10 000	Ne
Služby	Dispečink	Ne	10 000	Ne
Služby	Servis	Ne	10 000	Ne
Reporting	Business Intelligence	Ne	50 000	Ne
Reporting	Uživatelský portál	Ne	40 000	Webový klient
Reporting	Tvorba reportů	Ano	0	Ne
Vlastní vývoj	Uživatelský vývoj	Ne	40 000	Ne
Vlastní vývoj	Webové služby	Ne	50 000	API
Vlastní vývoj	Export/Import	Ano	0	Export/Import

Tabulka 5.1: Tabulka funkčních modulů

Systémový modul	Systémový blok	Jádro	Cena v Kč	Návaznost
Systémový modul	Identity management	Ne	10 000	Ne
Systémový modul	Zasílání zpráv	Ne	10 000	Ne
Systémový modul	Import/Export	Ano	0	Ne
Systémový modul	API	Ne	0	Webové služby

Tabulka 5.2: Tabulka systémového modulu

Klient	Klient	Jádro	Cena v Kč	Návaznost
Klient	Webový klient	Ne	0	Uživatelský portál
Klient	Aplikace Java	Ne	5 000	Ne
Klient	Aplikace .NET	Ano	0	Ne
Klient	Mobilní klient Android	Ne	5 000	Ne
Klient	Mobilní klient iOS	Ne	5 000	Ne

Tabulka 5.3: Tabulka klientů

5.6 Uživatelé

Počet licencí systému pro uživatele, kteří budou systém využívat. Viz Tabulka 5.4.

- Základních 5 uživatelů.
- 1 uživatel.
- Balíček 3 uživatelů.
- Balíček 5 uživatelů.
- Balíček 10 uživatelů. [9]

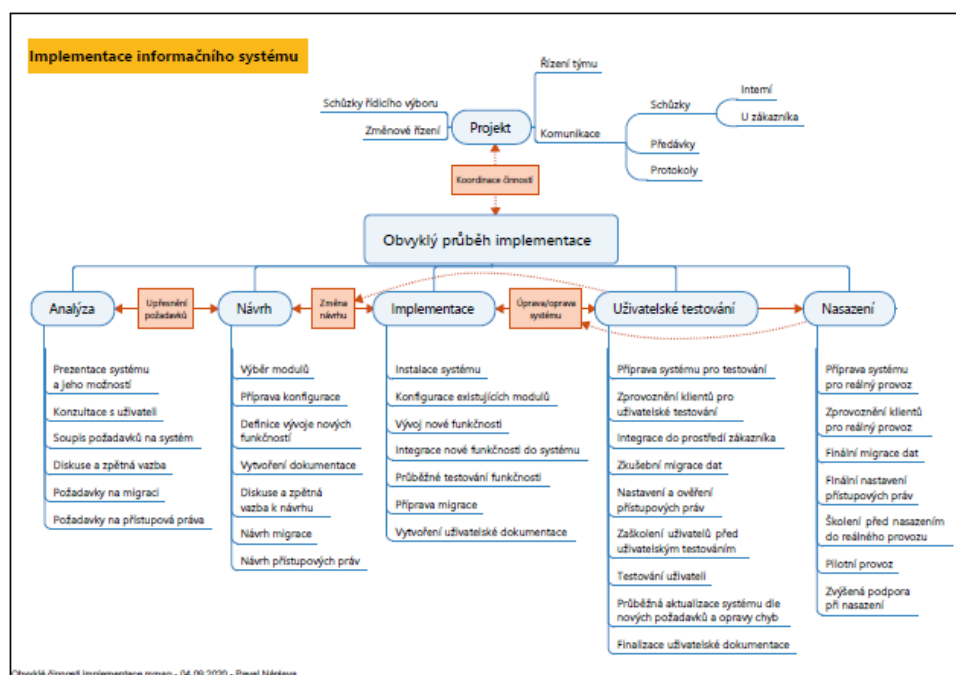
Uživatelé	Počet uživatelů	Jádro	Cena v Kč	Návaznost
Uživatelé	Základních 5 uživatelů	Ano	0	Ne
Uživatelé	1 uživatel	Ne	10 000	Ne
Uživatelé	Balíček 3 uživatelů	Ne	25 000	Ne
Uživatelé	Balíček 5 uživatelů	Ne	40 000	Ne
Uživatelé	Balíček 10 uživatelů	Ne	70 000	Ne

Tabulka 5.4: Tabulka uživatelů

5.7 Obvyklý průběh implementace

Implementace je proces uvedení nového IS do provozu. Práce na systému se skládá obvykle z pěti hlavních částí. Viz Obrázek 5.1.

První část je analýza, která je spojená s prezentací systému a jeho možnostmi, řeší se požadavky na systém. Po upřesnění požadavků se přechází na návrh, kdy jsou řešeny funkčnosti systému. Po návrhu přichází fáze implementace, kdy se konfiguruje funkčnosti, probíhá instalace systému, testování, připravuje se na migraci a vytváří se uživatelská dokumentace. Po úpravě systému přichází fáze uživatelského testování, kdy už probíhá zprovoznění



Obrázek 5.1: Průběh implementace

klienta, nastavování přístupových práv a samotné testování uživateli. Uživatelské testování může odhalit chyby, nebo možnosti systému, které nebyly zprvu zřejmé. Může se ukázat, že nějaká funkčnost nefunguje podle představ, tak je třeba ji opravit ve fázi návrhu a změnit v implementaci. Po aktualizaci systému, kde nebudou žádné chyby a bude dokončená dokumentace, přejde projekt do fáze nasazení a proběhne příprava na reálný provoz. Lze si pod tím představit pilotní provoz systému, kdy probíhá finální migrace, jsou k tomu uzpůsobena školení a je připravena zvýšená podpora, vzhledem k případným problémům. Po doladění všech chyb a zkušebnímu testování je spuštěn ostrý provoz.

Každá z fází je důležitá a žádnou nelze opomenout. Nezapomínejme také, že nejdůležitější na projektu je komunikace, ať už na straně zadavatele, nebo na straně firmy, která tvoří produkt. Při nedostatečné komunikaci je možné ztratit velké množství času, je možné udělat zdánlivě jednoduchý systém složitý a tím mrhat časem i penězi. [9]

5.7.1 Nacenení prací

Všechny ceny prací v konfigurátoru jsou pouze orientační, ale reflektují rozsah cen na trhu. Nacenení je možné vidět v Tabulce 5.5.

- **Analýza** – nacenení úvodní analýzy která se skládá z firemních procesů a informačních systémů. Výstupem může být dokument, který popisuje automatizaci procesů a slouží jako podklad pro implementaci. Je zde definován rozsah prací, jednotlivé kroky a jejich řešení, přesný časový

a finanční harmonogram.

- **Konfigurace** – cena práce za seskupení a přizpůsobení všech funkcností, které mají být obsaženy v IS.
- **Migrace** – pro porovnání všech dat, která se nacházela v aktuálním systému je třeba provést migraci. Ta nám umožňuje převést data pomocí skriptů, nebo různých nástrojů, do nového systému a pak s nimi pracovat. Cena se odvíjí od délky trvání migrace a zkušenostech pověřeného zaměstnance migrací.
- **Zaškolení a pomoc při nasazení** – před zavedením nového systému je třeba naučit uživatele s ním pracovat. Je třeba počítat se zvýšenou podporou v prvních týdnech od zavedení systému. Cena se odvíjí od délky trvání školení a podpory při nasazení.
- **Práva** – cena za identifikaci a nastavení práv.
- **Tvorba uživatelské dokumentace** – cena tvorby dokumentace ke všemu, co IS obsahuje.
- **Vedení projektu** – cena projektového řízení je individuální a vychází z náročnosti projektu. Kvalitní vedení projektu může ušetřit spoustu času a práce. [9] [10]

K těmto pracím je možné přidat i další činnosti jako je instalace a napojení na externí systémy.

Typ práce	Cena za člověkohodinu
Analýza/konzultace	1000
Konfigurace (vývoj)	1000
Migrace	1000
Práva	1000
Tvorba dokumentace	800
Vedení projektu	1500
Ostatní	1000

Tabulka 5.5: Ceník prací

5.8 Odhady průběhu práce na implementaci

Odhady fází v konfigurátoru jsou rozděleny na odhad konfigurace, analýzy, migrace, práv viz sekce 5.7.

K funkčním blokům jsou po fázi výběru přiřazovány odhady, kde je možné vybrat z předdefinovaného maxima a minima člověkohodin na implementaci. Počet člověkohodin se zadává podle náročnosti funkčnosti. Pokud je funkcionality funkčního bloku náročnější, tak se odhad bude blížit k uvedenému

maximu člověkohodin, pokud bude funkcionalita méně náročná, bude se blížit k minimu.

Zprovoznění bloků není nutné definovat u funkčností, které jsou zahrnuty. Je to funkce, kdy se počítá s větší složitostí systému, nebo je vytvořena funkce navíc, která není definována. Ceny jsou v konfigurátoru dopočítány podle zadaných člověkohodin.

■ 5.9 Etapy implementace

Podle hodnot odhadů lze přejít na Etapy, kdy se vychází z odhadů člověkohodin, které se rozdělují do etap podle toho, v jaké etapě na daný funkční blok je dán větší důraz. V této fázi se cena nemění. Dochází k přerozdělení člověkohodin daných odhadů vzhledem k počtu etap.

■ 5.10 Součinnosti

V činnostech u implementace je možné zadat součinnosti, kde lze určit skupiny pracovníků, kteří by se měli na projektu podílet. Pokud bude spolupráce s více dodavateli, nebo potřeba více externích služeb, tak bude potřeba více součinností.

Kapitola 6

Požadavky a diagramy konfigurátoru

Kapitola je věnovaná funkčním požadavkům, návrhům a základním UML diagramům vytvořeným pro návrh konfigurátoru.

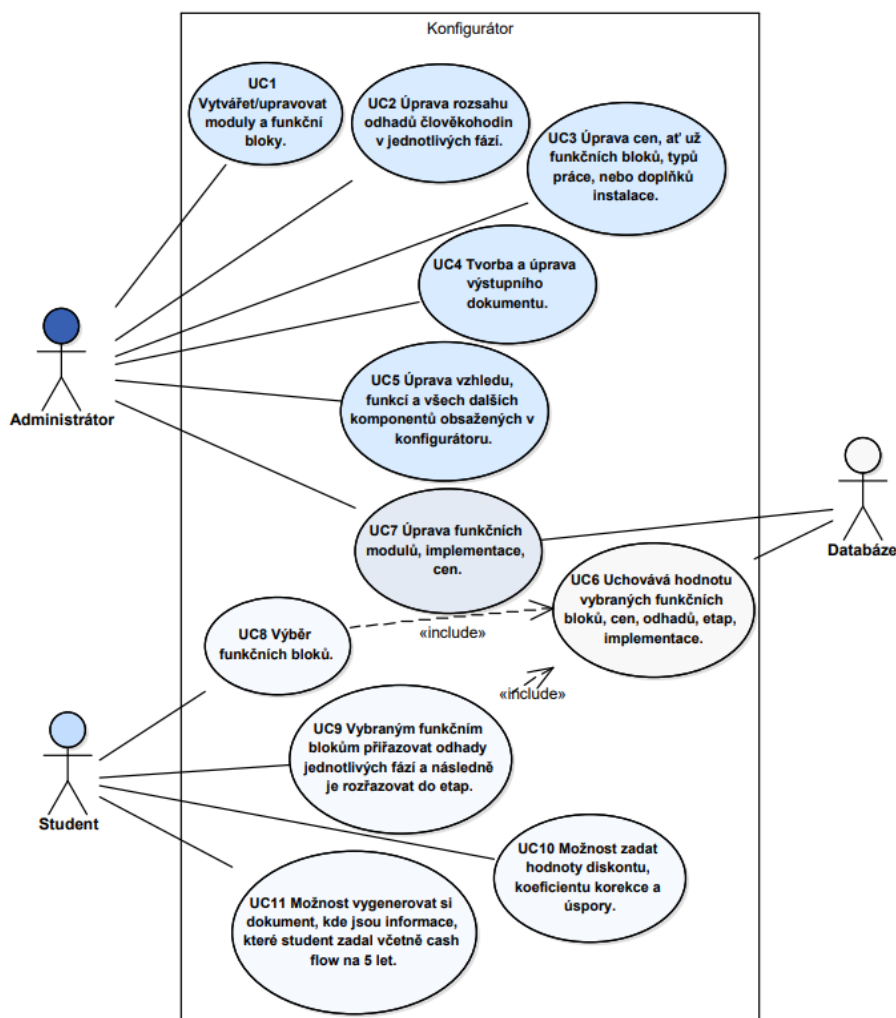
6.1 Funkční požadavky

"Funkční požadavky definují, co systém bude uživatelům umožňovat." [11]
Reflektují návrhy na změnu viz sekce 4.5, a zobecňují, jaké funkce konfigurátor umožňuje.

- Konfigurátor bude umožňovat výběr modulů.
- Konfigurátor bude umožňovat navázat na související vybraný modul, vybrat ho, a přidat do celkové ceny systému (upozornit na to).
- Konfigurátor bude umožňovat nacenění modulů.
- Konfigurátor bude umožňovat výběr počtu uživatelů a následné nacenění.
- Konfigurátor bude umožňovat nacenění prací jednotlivých odhadů.
- Konfigurátor bude umožňovat rozřazení odhadů do etap.
- Konfigurátor bude umožňovat přidání dalších činností v rámci implementace, včetně součinností a jejich nacenění.
- Konfigurátor bude umožňovat přidání doplňkových položek a následné nacenění.
- Konfigurátor bude umožňovat průběžné zobrazování cen a práce.
- V Konfigurátoru bude možnost generovat výsledný dokument, kde budou přehledně všechny informace, které byly zadány do konfigurátoru, včetně cash flow a návratnosti investic.

6.2 Příklad užití

"Případ užití je sada několika akcí, které vedou k dosažení určitého cíle. Ukazuje, jaké funkcionality systém obsahuje a jak (kým) jsou spouštěny." [12] Účelem tohoto diagramu je popis funkcionality systému. Na Obrázku 6.1 jsou vidět tři aktéři, kteří interagují s konfigurátorem a jedenáct případů užití.



Obrázek 6.1: Případy užití

Administrátor má možnost přidávat a upravovat počáteční informace, které konfigurátor obsahuje (definovaná v 5. Kapitole). Dále upravovat design konfigurátoru, manipulovat s jednotlivými ovládacími prvky, upravovat funkce, proměnné viz Kapitola 7. Může měnit dokument výstupů konfigurátoru. Spravuje databázi, kde se uchovávají data o všem, co konfigurátor obsahuje, což je možné vidět na případu užití UC7, kde je propojení s administrátorem a pak s databází.

UC6,8,9 nastiňuje situaci, kdy student v průběhu práce zadává odhady,

vybírání funkčních modulů a ty se pak zapisují do databáze. To zajišťuje možnost skončit s prací a vrátit se k ní po nějaké době. Student má dále možnost zadat doplňující informace, které se u každého projektu mohou lišit. Nakonec má možnost si vygenerovat dokument, který shrnuje všechny informace zadané do konfiguratoru, včetně vyhotoveného cash flow.

6.3 Doménový model

Doménový model reflektuje data z 5. Kapitoly. Jsou zde rozděleny jednotlivé fáze do tříd a několika enum metod viz Obrázek 6.2. Enum metody se v konfiguratoru nemění a zůstávají tak, jak jsou vidět na obrázku. Jsou jimi `Module`, jejich instance jsou všechny moduly zmíněné v 5. Kapitole. Typ práce, kdy jsou jako instance definovány typy práce a jejich cena za člověkohodinu. Doplňky instalace mají jako instance další služby, které je možné pořídit, a jejich hodnoty. Poslední enum metodou je `Phase`, která obsahuje výčet fází na projektu a váže se k implementaci.

Dalšími třídami jsou `FunctionalBlock`, `Estimate`, `Step`, `Implementation`. Funkční bloky uchovávají informace o jednotlivých funkčních blocích. Je zde název funkčního bloku, jestli je obsažen v jádře, jestli má návaznost na jiné moduly a další vlastnosti, které jsou definované v 5. Kapitole. Všechny tyto vlastnosti pomáhají v tvorbě konfiguratoru. V Kapitole 7. jsou tyto hodnoty využity při implementaci. Každý vybraný funkční blok má možnost rozdělení odhadů člověkohodin do jednotlivých fází. Na obrázku si můžeme všimnout, že Funkční bloky si uchovávají informace o odhadech a etapách, to je možné vidět v příloze A.1, kde jsou zobrazeny tabulky. V této příloze jsou u jednotlivých funkčních bloků informace, včetně odhadů a rozřazení do etap. Poslední třída je `Implementation`, která stojí mimo ostatní, jelikož nenavazuje na zmíněné třídy, ale obsahuje další činnosti vhodné zahrnout do projektu.

6.4 Diagram nasazení

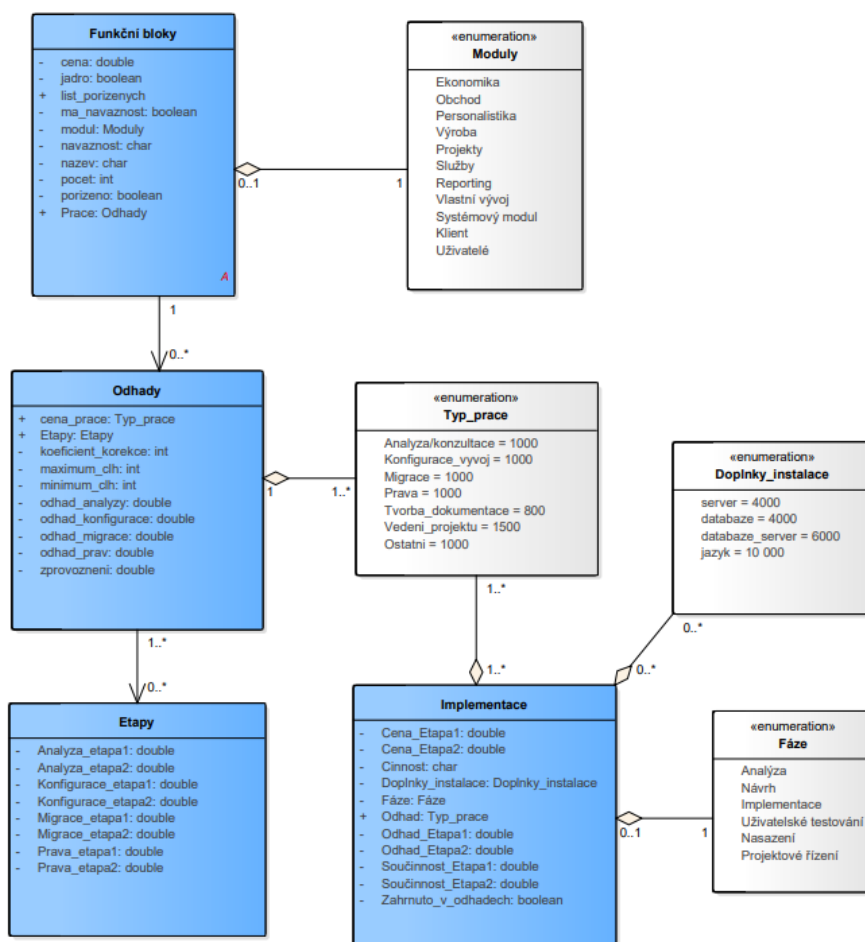
"Diagram nasazení ukazuje rozdělení jednotlivých komponent ať už softwarových nebo hardwarových na zdrojích a uzlech a jejich spolupráci."[13]

V diagramu nasazení jsou propojené dva uzly viz Obrázek 6.3. První je `Office 365`, který obsahuje prostředí `Excelu`, kde jsou uložena data a `Power Apps`, ve kterém je vytvořen konfigurator. Prostředí mezi sebou komunikují prostřednictvím protokolu. Druhým je počítač studenta, který pro spuštění aplikace musí mít přístup k internetu. Více v Kapitole 7.

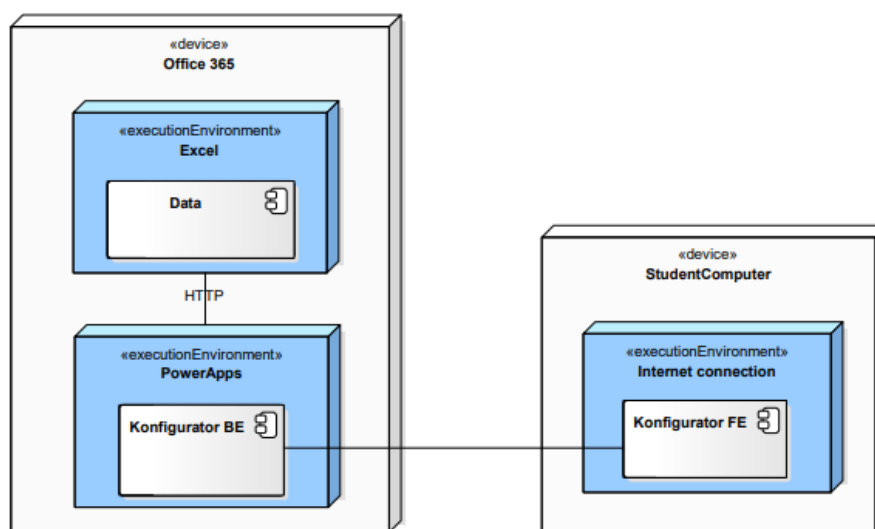
6.5 Návrh obrazovek

Návrhy jsou totožné s finální verzí konfiguratoru. Po dohodě s vedoucím práce jsme rozhodli, že je do této kapitoly nebudeme vkládat. Finální podoba konfiguratoru, včetně rozvržení a funkčností jednotlivých obrazovek, je uvedena v Kapitole 8.

6. Požadavky a diagramy konfigurátoru



Obrázek 6.2: Ukázka domain modelu



Obrázek 6.3: Deployment diagram

Kapitola 7

Power Apps a implementace konfigurátoru

Tato kapitola se zabývá implementací konfigurátoru v Power Apps a nabízí základní návod pro vytváření vlastních aplikací. V této kapitole je popsáno, jak jsem postupovala při tvorbě konfigurátoru. Od výběru prostředí, přes databázi, až po jeho testování. Zmíněné jsou funkce, vlastnosti a ovládací prvky, se kterými jsem pracovala.

Ze začátku práce jsem si zanalyzovala, v jakém prostředí bude bakalářská práce vytvořena. Vedoucí práce mi navrhl, že je možné práci vytvořit v Power Apps, což je platforma od společnosti Microsoft. Power Apps je mocný nástroj, jak pro programátory, tak pro ty, kteří se chtějí něco nového naučit a v programování nejsou tak zbláhli. Tady u nás mi přijde, že to tak rozšířené není. Proto má tato práce sloužit i jako představení prostředí a práce s ním, na co si dát pozor a jak se rychle zorientovat.

Na této platformě lze vytvářet tři typy aplikací podle prostředí. V prostředí plátna lze vytvářet aplikace na základě dat a přetahovat prvky na plátno podle potřeby. Logika se vytváří pomocí kódu, který je podobný funkcím v Excelu a má prvky z programovacích jazyků. Modelem řízené aplikace jsou aplikace zaměřené na součásti a nevyžadují kód. Většina rozvržení je zde stanovená a určují ji součásti, které se do aplikace přidávají. Poslední jsou portály. V tomto prostředí lze vytvářet webové aplikace, které mají možnost přihlašování organizací, prohlížení dat, anonymní procházení obsahu. [14]

Po zjištění všech informací o jednotlivých prostředích jsem se rozhodla začít práci v prostředí portálu. Pro tuto práci mi to přišlo nejlepší, jelikož jsem chtěla webovou aplikaci. Po zkoušce založení projektu jsem ale velmi rychle zjistila, že touto cestou nepůjdu. Bohužel prostředí portálu potřebuje jinou licenci, ke které nemám práva a ani na podpoře ČVUT se mi nepovedlo ji získat. Jelikož modelem řízené aplikace jsou stavěné na součástech (tabulky, relace a další), tak nad touto možností jsem ani nepřemýšlela a rovnou se vrhla na prostředí plátna. Než se však spustí aplikace, je třeba rozmyslet, pro koho je aplikace tvořená viz sekce 7.1.

Po výběru prostředí jsem hledala vhodnou databázi. Po prozkoumání možností jsem nakonec zvolila Excel a tabulky vytvořila zde.

7.1 Uživatelské prostředí

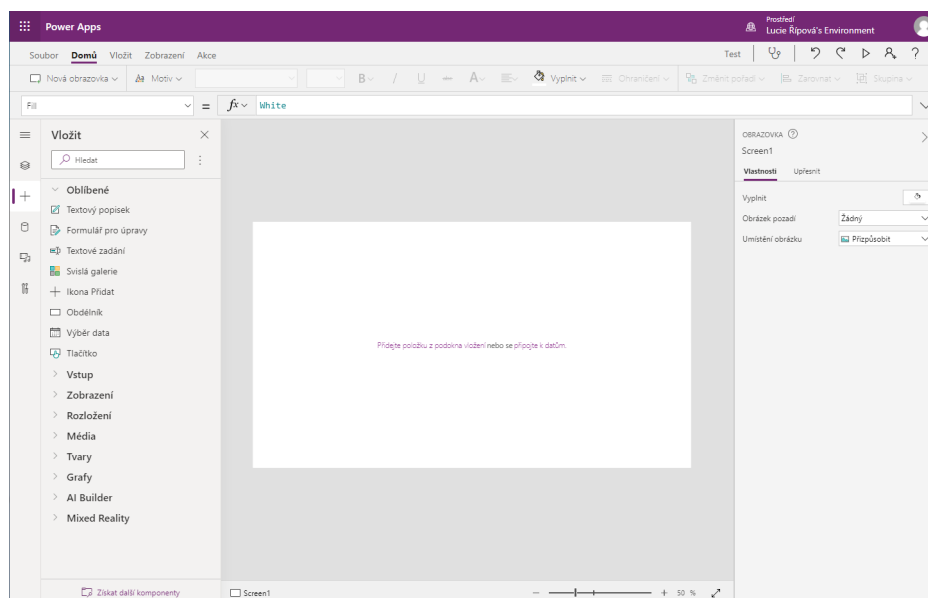
Uživatelské prostředí je něco jiného než prostředí, které je pro tvorbu aplikací. Ke každému účtu jsou primárně dvě. Jedno je prostředí organizace, ve které je vytvořen účet (v mém případě ČVUT), druhé je osobní viz Obrázek 7.1 v pravém horním rohu. V osobním prostředí lze vytvářet aplikace, ale nelze je sdílet s lidmi v prostředí organizace. Já jsem si nejdříve založila aplikaci v osobním prostředí a když jsem ji chtěla sdílet, tak jsem ji musela exportovat a importovat v prostředí organizace, aby k ní měl přístup i někdo další. Proto je důležité se na začátek práce zamyslet, pro koho má aplikace být.

7.2 Prostedí plátna

Prostředí plátna je primárně určené pro prostředí tabletů a mobilů. Aplikaci je ale možné otevírat i z desktopových webových prohlížečů. Prostedí plátna nabízí více možností, jak nový projekt založit. Lze si založit aplikaci plátna od začátku, což znamená, že se vygeneruje prázdná aplikace jen s úvodní plochou. Je také možné použít předem připravené šablony a ty následně upravovat. Pro zjednodušení práce je připravená možnost vygenerovat aplikaci z dat, která máme připravena.

Tvorba aplikací v Power Apps je jednodušší v tom, že není potřeba vše programovat. Aplikace lze navrhovat přetažením prvků z panelu prvků na plátno. Funkčnost jednotlivých prvků se spravuje na liště, k tomu určené. Logiku aplikace obstarávají příkazy podobné programovacím a výrazy, které jsou použity při práci v Excelu. [14]

Na Obrázku číslo 7.1 je možné vidět aplikaci plátna po vygenerování.



Obrázek 7.1: Prostedí plátna

V tomto případě byla vybrána možnost Aplikace plátna od začátku, ze které je vytvořen i konfigurátor. Více viz sekce 7.3. Na středu obrazovky je prostor pro vytvoření aplikace. Vlevo je panel prvků, ze kterého vybíráme ovládací prvky, které je možné použít. V pravém panelu je možné upravovat prvky a jejich vlastnosti.

7.3 Sestavení aplikace

Jedna z možností sestavení aplikace zmíněné výše, je vygenerování aplikace ze zdroje dat. Jako zdroj dat je možné si vybrat mezi Microsoft Excel, seznamy SharePoint, tabulek SQL a mnoha dalších formátů, které se mohou ukládat do cloudových služeb jako OneDrive, Dropbox, SQL Server a další. Po vygenerování je připravena galerie s daty, které byly připojeny.

Další možnost jak vytvořit aplikaci jsou již předpřipravené šablony. Ty stačí jen upravit podle potřeby.

Poslední možností je Aplikace plátna od začátku. Tu jsem využila pro implementaci konfigurátoru. Po vygenerování prostředí lze přidávat a upravovat obrazovky, galerie, formuláře a další ovládací prvky. Lze si přizpůsobovat a měnit data ve zdroji, nebo i rozlišovat zdroje dat.

7.4 Data v Excel souboru

Data vychází z Doménového modelu jednotlivých tříd viz sekce 6.3. Jsou rozdělena do dvou listů. V prvním listu jsou funkční bloky, odhady, etapy. Ve druhém je implementace. Koresponduje to s doménovým modelem, kdy jsou třídy Funkční bloky, Odhady, Etapy provázané a třída Implementace stojí mimo ostatní. V příloze A.1 je vidět rozdělení funkčních bloků do modulů. U každého bloku je zaznamenána cena, návaznost a pomocí true a false je rozlišeno, jestli je blok součástí jádra. Ve sloupci „Pořízeno“ je u funkčního bloku hodnota „isCore“, pokud je blok v jádru, nebo „isFalse“, pokud tam není. Pomocí funkce a hodnoty „isCore“ v databázi, se v aplikaci automaticky zaškrtně checkbox s daným funkčním blokem a není možné ho upravovat. Jakmile je v aplikaci zaškrtnut checkbox s vybraným funkčním blokem, tak se do databáze zapíše „isTrue“, pokud se odškrtně, změní se zpět na výchozí hodnotu „isFalse“.

V příloze A.1 je také možné vidět další sloupce jako odhad konfigurace, analýzy, migrace, práv a jejich rozdělení do etap. Do konfigurátoru se zadávají odhady na základě vybraných funkčních bloků. Ke každému funkčnímu bloku je připravené minimum a maximum člověkohodin. Poté, co je hodnota zadána tak, aby seděla složitosti funkčního bloku, zapíše se do databáze na příslušné políčko odhadu. To samé je u etap, kdy je třeba dát si pozor, že hodnota, která se zadává do etap vychází z odhadů.

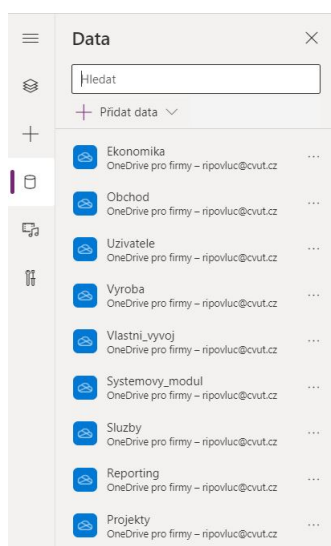
Implementace nabízí možnost zahrnout další práce na projektu viz příloha A.2. Pokud se zadá hodnota do jakýchkoliv políček připravených, tak se zapíše do databáze.

Po připojení do Power Apps je ke každému řádku přiděleno vlastní ID, to je vidět na konci každé tabulky viz příloha A.1 i A.2.

7.5 Připojení k datům

Excel soubor se musí nacházet v cloudovém úložišti Office 365, nebo musí být v počítači synchronizovaný pomocí OneDrive. Excel soubor obsahuje všechny tabulky a data, která chceme použít. Tabulky musí být naformátovány jako tabulka a nesmí obsahovat funkce. V návrhu tabulky pak lze přejmenovat tabulku pro přehlednost, což velmi ulehčí vložení tabulky do databáze a následnou práci v aplikaci. [14]

Pro nahrání tabulek do Power Apps, je nutné se nejdříve připojit do cloudového úložiště v podokně Data, kde je zvolený soubor s tabulkami, poté připojit jednotlivé tabulky. Viz Obrázek 7.2.



Obrázek 7.2: Tabulky v Power apps

Během práce na aplikaci lze data v Excelu upravovat. Nové změny v aplikaci je možné vidět až poté, co se tabulka aktualizuje i v Power Apps.

Pro aktualizaci databáze je potřeba v podokně Data vybrat upravenou tabulku a aktualizovat.

7.6 Návrh aplikace

Prostředí Power Apps je velmi intuitivní. Po vygenerování aplikace, z vybraného způsobu zmíněného v sekci 7.3, je připravena úvodní obrazovka, kam lze skládat prvky z podokna návrháře. Je to podobné jako skládat prezentaci v PowerPointu. Prvky lze mazat, přejmenovávat a kopírovat. Navíc je zde možné prvkům dávat různé funkce a vlastnosti.

Vložení prvku na obrazovku je možné dvěma způsoby. Prvním je výběr prvků v horní liště pod kartou Vložit, druhým je přidání prvku ve vertikální liště, tam je pod ikonou plus. Prvky se zobrazují stromově, v levém navigačním podokně, v pořadí, v jakém byly přidány.

Do aplikace je možné přidávat další obrazovky. Stačí jen vybrat Nová obrazovka a zvolit rozvržení obrazovky z možností. Obrazovka se pak objeví v levém navigačním podokně, kde je uveden hierarchický seznam ovládacích prvků. Je potřeba dávat pozor na pořadí, pokud se nějaké prvky překrývají, tak nemusí správně fungovat. Překrytí je možné kontrolovat podle rámečku u každého prvku. Například, když je vloženo na obrazovku tlačítko a nad něj je dán popis, který bude svým ohraničením zasahovat do tlačítka, tak nemusí tlačítko správně fungovat. Tento problém je možné vyřešit najetím na popis a v možnostech změnit pořadí na "Přenést dál", nebo "Přenést do pozadí".

Nejvíce používaným prvkem, kromě popisku, byly v mé aplikaci galerie. Galerie je seznam položek ze zdroje dat. Při přidání galerie je možné si vybrat, jaká galerie bude použita. Vodorovná, svislá, prázdná, s proměnlivými prvky a jaká data chceme připojit. Pro vložení prvků do galerie a zobrazení dat, je třeba vybrat položku z podokna vložení, nebo kliknout na vstup a vybrat příslušný prvek. Zobrazí se počet prvků tolikrát, kolik je řádků v tabulce, která byla zvolena. Pro zobrazení popisků je třeba do kolonky text zadat „ThisItem:‘Název sloupce, který chceme‘“.

7.7 Funkce

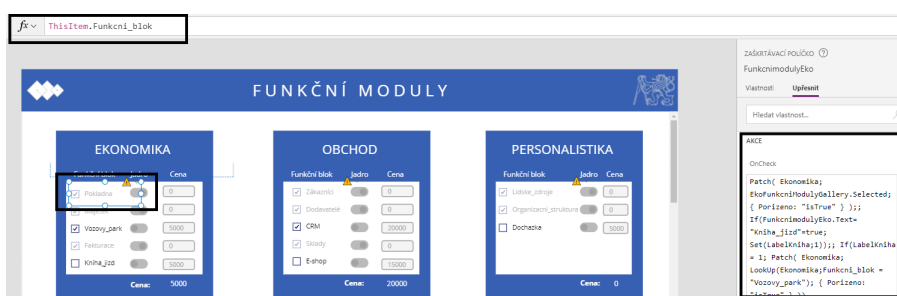
Funkce v Power Apps jsou něco mezi programovacím kódem a excelovskými funkcemi. Funkce pomáhají, aby aplikace nebyla jen plná tlačítek, galerií, obrázků a popisků, ale aby měla nějakou funkčnost.

Funkce se tvoří v návaznosti na vlastnosti, které má každý ovládací prvek, s kterým se pracuje, viz sekce 7.8. Je třeba si vybrat ovládací prvek, pro který má být funkce vytvořena. Dále je třeba vybrat vlastnost, ve které funkce bude a v řádku vzorců, nebo pravém navigačním podokně, pak lze funkci napsat.

Na Obrázku 7.3 je vidět návrh obrazovky Funkční moduly. Je vybrán checkbox obsažen v galerii a data jsou z tabulky Ekonomika. V pravém podokně ovládacího prvku je vybrána vlastnost, na kterou je funkce navázána. V tomto případě je ovládací prvek checkbox, a vlastnost „OnCheck“, což zajišťuje, že při vybrání hodnoty se provede funkce. Funkce zapisuje do databáze hodnotu „isTrue“, pokud je funkční blok vybrán, pokud je odškrtnut, tak „isFalse“.

Jak je možné vidět na obrázku, tak vlastnosti a jejich funkce se mohou zadávat i v horní liště v řádku vzorců. Aktuálně je vybrána vlastnost Text a funkce je `ThisItem.Funkcniblok`, která zajišťuje, že názvy modulů se zobrazí jako ty, co jsou v databázi.

Pokud je potřeba zadat do vlastnosti ovládacího prvku více funkcí, tak je třeba oddělit funkce dvěma středníky. Funkce se provádějí v pořadí, ve



Obrázek 7.3: Funkce

kterém byly napsány.

Funkce mohou obsahovat další funkce. Je třeba pak myslet na to, aby každá měla požadovaný počet hodnot a operátorů.

7.7.1 Nejvíce používané funkce při vývoji konfigurátoru

- **Filter** - vrátí záznamy ze zdroje dat, na základě zadaného kritéria.
 Syntaxe: *Filter(Tabulka; Vzorec1 [; Vzorec2; ...])*
 Tabulka je ze zdroje dat. Podle vzorce se vyhodnocuje záznam v tabulce.
- **If** - vrátí hodnotu true, pokud je podmínka pravdivá, jinak false.
 Syntaxe: *If(Podmínka; PakVýsledek [; VýchozíVýsledek])*
 Podmínka, která se vyhodnocuje, pak výsledek, který je vrácen.
- **Lookup** - na základě kritérií je vybrán záznam.
 Syntaxe: *Lookup(Tabulka; Vzorec [; Redukčnívzorec])*
 Tabulka je ze zdroje dat. Podle vzorce se pak vyhodnocuje záznam. Volitelný redukční vzorec, který se vyhodnotí nad nalezeným záznamem.
- **Patch** - upravuje záznamy ve zdroji dat.
 Syntaxe: *Patch(ZdrojDat, ZákladníZáznam, ZáznamZměny1 [, ZáznamZměny2, ...])*
 Zdroj dat, který se má vyhodnotit. Záznam, který se bude upravovat. Záznam změny jako jeden, nebo více záznamů vlastností, které se upraví.
- **Select** - výběr ovládacího prvku.
 Syntaxe: *Select(Ovládací prvek)*
 Ovládací prvek, který je vybrán.
- **Set** - nastaví proměnnou a její hodnotu.
 Syntaxe: *Set(NázevProměnné; Hodnota proměnné)*
- **Sum** - vypočítá součet výrazů, jako jsme zvyklí v excelu.
 Syntaxe: *Sum(ČíselnýVzorec1; [ČíselnýVzorec2; ...])*

- **Text** - převede libovolnou hodnotu na textový řetězec.

Syntaxe: `Text(AnyValue)`

- **Value** - převede řetězec na číslo.

Syntaxe: `Value(Řetězec [; ZnačkaJazyka])` [14]

Logické operátory

- `=` přiřazení proměnné
- `<` menší
- `<=` menší, nebo rovno
- `>` větší
- `>=` větší, nebo rovno
- `&&` nebo And
- `||` nebo Or
- `!` nebo Not
- **ThisItem** - Vrací záznam aktuální položky. Vztahuje se ke galerii, nebo k formuláři.
ThisItem.FirstName
- **ThisRecord** - Přístup k úplnému záznamu a jednotlivým polím uvnitř funkce.
ThisRecord.FirstName [14]

7.8 Ovládací prvky a jejich vlastnosti

Ovládací prvky mají různé vlastnosti, které jsou vyobrazené v pravém navigačním podokně.

Vlastnosti ovládacích prvků je možné konfigurovat na dvou místech, jak již bylo možné vidět na obrázku 7.3. První je v horní liště na řádku vzorců. Druhá je v pravém navigačním podokně. Na obou místech je to stejné. Osobně jsem si zvykla používat pravé navigační podokno, jelikož tam mohu vidět všechny vlastnosti a snadněji se mezi nimi přesouvat. V horní liště je možné vidět jen aktuální vlastnost, která se upravuje. Každý prvek má různé vlastnosti a upřesnění podle toho, pro co je uzpůsoben.

Pro dobré vyznání se v aplikaci doporučuji správné pojmenování ovládacích prvků, jelikož se s nimi pak pracuje i napříč jinými ovládacími prvky, jejich vlastnostmi, funkcemi a proměnnými.

Zde uvádím mnou nejvíce používané prvky, jejich vlastnosti a funkce, které jsem jim přiřazovala. Další jsou dohledatelné přímo v dokumentaci Power Apps od Microsoftu.

- **Button** - tlačítko.

Tlačítko je často vnímáno, jako prvek, na který když se klikne, provede se akce. V konfigurátoru využívám tlačítka hlavně na přechod mezi obrazovkami, smazání hodnot, nebo generování PDF souboru. Například funkce tlačítka na druhé obrazovce „Navigate(FunkcniBloky)“ ve vlastnosti „OnSelect“ zařídí, že po kliknutí na tlačítko je aplikace ze stávající obrazovky přesunuta na obrazovku Funkčních bloků. Do vlastnosti „Text“ je dán popis tlačítka. V tomto případě "Přejít na Funkční bloky", kde jasně definuji, kam po kliknutí na tlačítko bude uživatel přeměrován.

- **Gallery** - zobrazení seznamu záznamů ze zdroje dat

Galerie jako taková nemá samostatně význam. Slouží pro zobrazování záznamů vzhledem k připojeným datům. Po vybrání galerie je třeba vybrat i ovládací prvek, který bude v galerii zobrazený. Výchozí počet ovládacích prvků, po vybrání, je pět. Po připojení galerie k datům se počet prvků rovná počtu řádků v tabulce. Pomocí vlastností se nastavuje, jaké hodnoty budou zobrazeny. Když je vybrána galerie z panelu prvků, je možné vybrat z připravených šablon, které nám určují, jak se vybrané ovládací prvky zobrazí. Všechny ostatní ovládací prvky, které pracují s daty a databází jsou nejdříve v galerii právě proto, aby se s nimi lépe pracovalo. Jsou navázané na tabulku a pro všechny ovládací prvky platí pak stejné vlastnosti. Když se do prvního prvku zadá, že bude mít data z prvního sloupce, tak do druhého prvku nelze zadat, že jsou ze sloupce druhého.

- **Checkbox** - zaškrtování možností.

Ovládací prvek, kdy při výběru je nastavena hodnota true, nebo naopak false.

Tento ovládací prvek jsem navazovala nejčastěji s prvkem galerie, aby se vytvořilo tolik checkboxů, kolik bylo řádků v tabulce a já s nimi mohla snadněji pracovat. Checkbox zajišťuje výběr funkčních bloků v modulech. Pokud jsou funkční bloky vybrány, mohou se jim přiřazovat hodnoty odhadů. Checkboxům jsem nastavovala vlastnost „Default“ na:

```
ThisItem.Jadro_systemu || ThisItem.Porizeno = "isTrue"
```

Což znamená, že se z databáze přiřadí hodnota true, pokud je funkční blok v jádru systému, už byl vybrán, nebo zůstal checkbox nezaškrtnutý a pak prvek není v jádru, nebo zatím nebyl vybrán.

Důležitá vlastnost checkboxů v konfigurátoru je vlastnost „OnCheck“, „OnUncheck“. Při „OnCheck“, kdy uživatel vybere funkční modul, se do databáze zapíše hodnota „isTrue“ vybranému funkčnímu bloku. Při „OnUncheck“, kdy uživatel odškrtně checkbox se do databáze zapíše "isFalse". Příklad „OnCheck“ u jednoho z checkboxů.

```
Patch(
    Ekonomika ;
```

```

        EkoFunkcniModulyGallery . Selected ;
    {
        Porizeno : "isTrue "
    }
);;

If (FunkcnimodulyEko . Text= "Kniha_jizd"=true ;
Set (LabelKniha ;1));;

If (LabelKniha = 1;
Patch (
    Ekonomika ;

    LookUp(Ekonomika ; Funkcni_blok = "Vozovy_park " );
    {
        Porizeno : "isTrue "
    }
))

```

Funkce Patch zařizuje zápis dat do databáze. Další funkce „Set“ nastavuje proměnnou, pokud je vybrán funkční blok „Kniha jízdy“ který má návaznost. Proměnná se pak nastaví na 1 a vybere se automaticky i navazující funkční blok, jehož hodnota se nastaví na true, pomocí poslední funkce Patch.

Poslední funkce checkboxů, kterou jsem zadávala je „DisplayMode“ kde je možné nastavit hodnoty jako „Edit“ a „Disabled“. Pokud je hodnota v databázi ve sloupci Pořízeno „isCore“, tak checkbox nelze editovat, jelikož funkční bloky, v jádru jsou už automaticky v licenci obsaženy.

```

If (

    ThisItem . Porizeno = "isCore " ;

    DisplayMode . Disabled ;

    DisplayMode . Edit

)

```

Ve vlastnosti „OnCheck“ je možné si ukázat i skládání funkcí. Zde je zobrazena poslední funkce z vlastnosti „OnCheck“ u checkboxu.

```

If (LabelKniha = 1;
Patch (
    Ekonomika ;

```

```

        LookUp(Ekonomika ; Funkcni_blok = "Vozovy_park " );
    {
        Porizeno : "isTrue "
    }
))

```

Vě vlastnosti jsou obsažené 3 funkce. První je „If“, druhá „Patch“ a třetí „LookUp“. Funkce „If“ určuje podmínku, která když je pravdivá, provedou se zbylé funkce. Funkce „LookUp“ vybere navazující funkční blok a „Patch“ zapíše do databáze, že byl blok vybrán. Je třeba dodržet počet operátorů a hodnot, jinak se funkce neprovede.

- **Text input** - zadávání textu, čísel a dalších dat.

Ovládací prvek, který slouží k zadávání textu a čísel. Nejčastěji jsem ho používala s prvkem galerie, abych měla hned napojenou tabulku a počet záznamu v ní. V konfiguratoru slouží pro zadávání cen odhadů, implementace a dalších hodnot. Navazuje na checkboxy, jelikož když není vybrán funkční blok, tak nelze upravovat textinput. Výchozí hodnota je prázdná, ale pokud je v databázi hodnota, tak zobrazí ji a bude s ní počítat.

```

Patch(
    Ekonomika ;
    EkoKonfigGallery . Selected ;
    {
        Odhad_konfigurace : EkoKonfigTextInput . Text
    }
)

```

Funkce slouží k zapsání dat do databáze potom, co uživatel zadá, nebo změní data v textinputu. Další vlastnost, kterou jsem zadávala bylo „DisplayMode“. Jako u checkboxu jsem nastavila možnost zápisu, pokud je funkční blok v jádru, nebo je vybrán, jinak se do něj nedá zapisovat. Slouží k tomu účelu, aby funkční bloky, které vybrány nejsou, nemohly mít hodnoty. Vlastnost „Default“ pak zajišťuje, že data ovládacího prvku se shodují s těmi v tabulce, která byla vybrána v galerii.

- **Label** - popisky, zobrazení textu, čísel a dalších dat.

Ovládací prvek label mám v konfiguratoru hlavně, pro popisy bloků, sloupců, hodnot a všeho, co je potřeba. Label jde také provázat s galerií, proto se dá použít, když se zobrazují data z databáze. Je potřeba mít v galerii správnou tabulku a v popisku správný sloupec, který počítá

námi vybrané hodnoty. Je možné popisek zobrazit, nebo schovat v rámci hodnoty proměnné. Například při vybrání funkčního bloku, který má návaznost, se zobrazí upozornění. Je to dané změnou hodnoty proměnné, a tudíž nastavení vlastnosti „Visibility“ z false na true.

■ **Image** - zobrazení obrázku.

Obrázky v Power Apps je nutné nejdříve nahrát do podokna Média a následně je možné s nimi pracovat. Obrázky mají také vlastnosti, proto je to ovládací prvek. Já jsem je využila například v první obrazovce na zobrazení loga konfigurátoru, nebo na upozornění u cen odhadů a etap. Jako upozornění jsou v konfigurátoru použity, když hodnota člověkohodin neodpovídá rozsahu. Pro to bylo potřeba nejdříve použít galerii, do té dát jako ovládací prvek obrázek a porovnávat hodnoty zadané uživatelem. Podle zadané hodnoty se upozornění zobrazí, pokud uživatel zadá jinou hodnotu, než je očekávána, nebo zůstane skryté.

■ **7.8.1 Společné vlastnosti podle kategorie**

Tyto vlastnosti jsou zahrnuty u ovládacích prvků. Každý ovládací prvek má jiné vlastnosti, ale některé mohou mít společné. Liší se podle toho, jaký mají u každého prvku smysl.

- **Barva a ohraničení** - umožňuje nakonfigurovat barvu a ohraničení ovládacího prvku.
- **Základ** (Default) - počáteční hodnota.
- **Obrázek** (Image) - konfigurace grafických prvků.
- **Velikost a umístění** – jak velký má ovládací prvek být a kde se nachází.
- **Text** - konfigurace zobrazování textu v ovládacích prvcích. Vlastnosti písma, zarovnání.

■ **7.8.2 Základní vlastnosti ovládacích prvků**

Zobrazení ovládacího prvku a vlastností, které se u něj nastavují.

- **Default** - počáteční hodnota ovládacího prvku, než jí uživatel změní.
V konfigurátoru tím nastavuji prvky podle dat z databáze. U funkčních bloků zobrazuji popisky. Pokud jsou funkční bloky obsaženy v jádře, nebo jsou vybrány, tak je výchozí hodnota checkboxu true. U prvků jako je textinput načítám data, která jsou v databázi. Při prvním spuštění konfigurátoru tam hodnota není.
Tuto vlastnost může mít, z výše zmíněných, checkbox, galerie, textinput.
- **DisplayMode** - zda je možné prvek upravovat. Pro tento prvek jsou možné hodnoty Edit, View a Disabled.

V konfigurátoru tuto vlastnost používám v případě, když chci určit, jaký prvek nechci, nebo chci, aby se upravoval. Například u prvku checkbox. Pokud jsou funkční bloky už v jádru, tak není možné je vybrat, protože jsou zahrnuty v licenci. V dalších obrazovkách to pak používám pro prvky textinputu, které se váží na vybrané funkční bloky, pokud funkční blok nebyl vybrán, tak nebude součástí IS, a proto mu nemůže být přiřazena hodnota.

Pro ovládací prvky jako je textinput, button, checkbox, galerie, image, label.

- **OnChange** - reakce prvku na změnu hodnoty.

Vztahuje se hlavně na textinput, kdy při každé změně hodnoty je zapsána změna do databáze. V mém případě je to v odhadech, etapách implementaci i ve financích. Jakmile je hodnota změněna, v jakémkoli textinputu, změní se hodnota i v databázi.

- **OnSelect** - reakce na kliknutí, na ovládací prvek, uživatelem.

Je použita u tlačítek, kdy je při kliknutí přechod na další obrazovku, provedení nějaké změny, nebo generování PDF.

Vlastnost pro ovládací prvky button, checkbox, galerie, label, textinput.

- **Text** - text, který se v ovládacím prvku zobrazuje.

Tato vlastnost je použita v aplikaci často. Pokud chci, aby měly ovládací prvky text, ať už z databáze, nebo takový, jaký chci, je vhodné použít tuto vlastnost.

Pro ovládací prvky button, checkbox, label, textinput

- **Value** - hodnota vstupního ovládacího prvku.

Pro ovládací prvek checkbox, kde se rozlišuje true a false, podle výběru.

- **Visible** - možnost zobrazení nebo skrytí ovládacího prvku.

Pro tuto funkci jsem našla využití hlavně v etapách a implementaci, kdy jsem si skryla všechny hodnoty funkčních bloků, které nebyly zadány, což platí hlavně tam, kde se sčítají hodnoty.

Vztahuje se pro button, checkbox, galerii, image, label, textinput.

■ 7.8.3 Další vlastnosti ovládacích prvků

Další vlastnosti které je možné využít.

- **AllItems** - pro všechny položky v galerii. Používám u funkcí, kdy chci určit, že vybírám všechny hodnoty z galerie, které jsou navázány na tabulku. Hodí se na počítání celkové ceny.

- **OnCheck** - změna při zaškrtnutí políčka na hodnotu true.

Při zaškrtnutí checkboxu funkčního bloku se zapíše do databáze hodnota „isTrue“. Také se vykonají další funkce, které si u této vlastnosti definujeme, při zaškrtnutí tlačítka.

- **OnUncheck** - změna při zaškrtnutí políčka na hodnotu false.

Při odškrtnutí checkboxu funkčního bloku se zapíše hodnota „isFalse“. Stejně jako u „OnCheck“ se vykonají i další vlastnosti definované při odškrtnutí.

- **Selected** - vybrané položky z galerie.

7.9 Proměnné

Proměnné nejsou v Power Apps vyřešené moc dobře, jelikož nejde zadat proměnnou, která tam bude od začátku, jak to známe z programovacích jazyků. Proměnné lze v Power Apps zakomponovat pomocí přidruženého prvku. Například, když se klikne na tlačítko, založí se proměnná. Jinak se s ní pracuje stejně. Stačí zadat jméno proměnné a zakomponovat ji do funkce tak, jak má být použita.

Proměnná se nastavuje použitím funkce „Set“ viz sekce 7.7. Počáteční hodnota proměnné je "Žádná hodnota". Hodnoty proměnných je možné zobrazovat, když se klikne v Power Apps na záložku „Soubor“ a poté „Proměnné“.

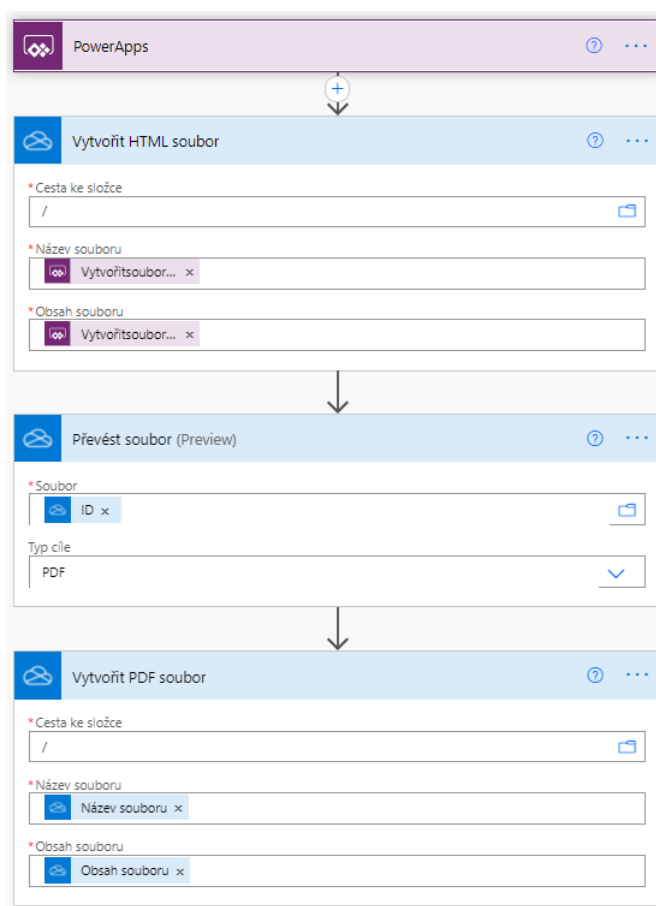
Já jsem proměnné využila k uchovávání hodnoty cen jednotlivých funkčních modulů, odhadů, etap, implementace, i cen celkem. Následně jsem to využila k vyhodnocení projektu ve finanční analýze.

7.10 Power Automate a další služby

"Power Automate je služba, která vám pomůže vytvářet automatizované pracovní postupy mezi vašimi oblíbenými aplikacemi a službami, abyste mohli synchronizovat soubory, dostávat oznámení, shromažďovat data a využívat mnoho dalších možností." [15]

V konfigurátoru využívám tuto službu pro vytvoření PDF, které je souhrnem všech informací, které byly zadány v konfigurátoru. Hlavními prvky jsou v Power Automate toky. Nejprve je potřeba vědět, jaký tok se využije. V mém případě šlo o okamžitý cloudový tok, který vygeneruje dokument do konkrétní složky. Je třeba také definovat, kde se tok aktivuje, což u mě bylo v prostředí Power Apps. Následně se vytvoří tok, kde na prvním místě je možné vidět průběh jeho aktivace. Viz Obrázek 7.4.

Na obrázku je dále možné vidět, že další krok je vytvoření HTML souboru, který následně vygeneruje i soubor PDF. U obou souborů je třeba určit parametry podle toho, jak se soubor bude jmenovat a jaký má být typ souboru, který bude převeden. V tomto případě převedený soubor je soubor HTML. V prvním kroku nelze generovat samotný soubor PDF.



Obrázek 7.4: Tok pro vytváření PDF souboru v Power Automate

Power Automate umožňuje okamžité otestování toku, aby bylo možné vidět, jestli funguje, než bude navázán na aplikaci. Stačí zadat parametry a tok se okamžitě spustí. V podrobnostech je vidět, kdy se tok spustil, jak dlouho trval a jestli proběhl úspěšně.

Navázání toku už proběhlo přímo v konfigurátoru. V záložce Akce v Power Apps je možnost přidat přímo Power Automate. V konfigurátoru je tok navázán na ovládací prvek. Proto jsem vybrala tlačítko a přidala ho do něj. Vygeneroval se mi začátek funkce, do té jsem pak doplnila parametry, které jsem si určila při tvorbě toku.

```
CreatePDF_2.Run(" Konfigurator .html "; HtmlText1.HtmlText)
```

Je funkce doplněná o parametry, pro vytvoření PDF.

Pro vytvoření HTML souboru, který se dal převést do PDF, jsem si musela vytvořit Text HTML. Tento ovládací prvek je podobný popisku, jen s tím rozdílem, že se do něj dávají prvky, jako při vytváření HTML stránky. Text HTML je souborem, který bude převeden. Pro vložení hodnot z konfigurátoru, například proměnných, se musí vždy před danou vloženou hodnotou napsat uvozovka a ampersand.

Vygenerované soubory se nacházejí ve složce která byla definována v toku.

V konfigurátoru je to kořenová složka školního účtu ve OneDrive.

Microsoft nabízí další služby, se kterými je možné propojit vaši aplikaci a dodat jí tak lepší funkčnost. Například *"Power BI, což je kolekce softwarových služeb, aplikací a konektorů, které společně dokáží přeměnit nesouvisající zdroje dat na ucelené, vizuálně poutavé a interaktivní přehledy poznatků."* [16]

7.11 Uložení a publikace

Průběh tvorby aplikace je možné ukládat a k tomu připisovat poznámky ke konkrétní verzi. Navíc probíhá každé 2 minuty automatické ukládání, pokud se aplikace už jednou uložila. Pokud je potřeba tvorba aplikace s někým sdílet, je nutné verzi publikovat. Pak ti, s kterými je aplikace sdílena uvidí provedené změny.

Pro sdílení aplikace je potřeba vybrat konkrétní aplikaci, vybrat Sdílet a v nabídce zadáte jména, nebo emaily uživatelů. Bohužel nelze sdílet aplikaci s nikým jiným než s lidmi z vašeho prostředí (organizace). Prostředí je svázáno s licencí organizace. Pokud je aplikace připojená ke zdroji dat, je třeba sdílet i ten.

Upravovat aplikaci může vždy jen jeden uživatel.

7.12 Sdílení a spuštění aplikace

Uloženou aplikaci lze sdílet s dalšími uživateli, skupinami a určit, kdo aplikaci může používat, nebo spravovat. Aplikaci lze spouštět ve Windows, ve webovém prohlížeči, na zařízeních iOS a Android.

Podporované prohlížeče jsou Google Chrome, Microsoft Edge. [14]

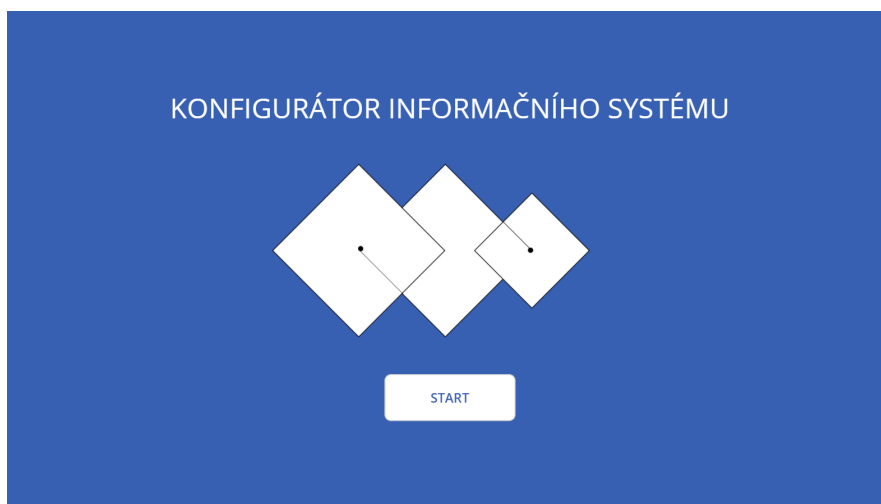
Pro možnost upravovat aplikaci je třeba, aby autor přiřadil práva pro úpravu, nebo aby měl vyexportovanou aplikaci. Vyexportovat aplikaci je možné výběrem aplikace a vybrat export. Exportuje se v zip souboru. Ten je třeba ve svém prostředí importovat.

Kapitola 8

Design a rozvržení konfiguratoru

V této kapitole nastíním vzhled konfiguratoru, rozvržení a funkčnost jednotlivých obrazovek, ze kterých se skládá.

První je zahajovací obrazovka, kde je zobrazeno logo konfiguratoru a tlačítko "START" pro začátek práce. Viz Obrázek 8.1.



Obrázek 8.1: První obrazovka

Dále se skládá z obrazovek informativních, které jsou před každou hlavní obrazovkou a popisují základní informace o tom, co je tvořeno v následující navazující obrazovce. Hlavní obrazovky jsou Funkční moduly, Odhady, Etapy, Implementace, kde uživatel tvoří systém. Poslední obrazovky slouží k vyhodnocení cen a finanční analýze.

Obrazovka Funkční moduly slouží k výběru funkcí, prostředí a počtu licencí pro uživatele. Odhady umožňují rozdělit funkční bloky do fází vývoje a určit jejich složitost. Etapy zajišťují rozdělení odhadů tak, aby byla lepší představa na jakou etapu se má dát větší důraz. Implementace dává možnost vybrat další práce, které by byly možné vynechat. Obrazovka Celkem za implementaci umožňuje shrnutí cen funkčních bloků a jednotlivých etap, které byly zadány. Obrazovka Finanční analýza umí vygenerovat přehled o všem, co se v konfiguratoru vytvořilo, včetně cash flow na pět let.

První hlavní obrazovka je obrazovka funkčních bloků. Skládá se ze tří

částí. Funkční moduly, prostředí a celková cena funkčních bloků, prostředí včetně ceny maintenance. Každý modul byl dán jednotlivě do bloku, aby bylo přehledné, co obsahuje, jaká je jeho cena, jestli je obsažen v jádru a jestli je vybrán. U každého modulu je uvedena celková cena daného modulu. Viz Obrázek 8.2.

Funkční blok	Jadro	Cena
<input checked="" type="checkbox"/> Pokladna	<input type="checkbox"/>	0
<input checked="" type="checkbox"/> Majetek	<input type="checkbox"/>	0
<input type="checkbox"/> Vozovy_park	<input type="checkbox"/>	5000
<input checked="" type="checkbox"/> Fakturace	<input type="checkbox"/>	0
<input type="checkbox"/> Kniha_jezd	<input type="checkbox"/>	5000
Cena: 0		

Funkční blok	Jadro	Cena
<input checked="" type="checkbox"/> Zákazníci	<input type="checkbox"/>	0
<input checked="" type="checkbox"/> Dodavatelé	<input type="checkbox"/>	0
<input type="checkbox"/> CRM	<input type="checkbox"/>	20000
<input checked="" type="checkbox"/> Sklady	<input type="checkbox"/>	0
<input type="checkbox"/> E-shop	<input type="checkbox"/>	15000
Cena: 0		

Funkční blok	Jadro	Cena
<input checked="" type="checkbox"/> Lidske_zdroje	<input type="checkbox"/>	0
<input checked="" type="checkbox"/> Organizacni_struktura	<input type="checkbox"/>	0
<input type="checkbox"/> Dochazka	<input type="checkbox"/>	5000
Cena: 0		

Funkční blok	Jadro	Cena
<input checked="" type="checkbox"/> Zdroje	<input type="checkbox"/>	0
<input type="checkbox"/> Rizeni_vyroby	<input type="checkbox"/>	15000
<input type="checkbox"/> Planovani_vyroby	<input type="checkbox"/>	15000
Cena: 0		

Funkční blok	Jadro	Cena
<input type="checkbox"/> Rizeni_projektu	<input type="checkbox"/>	15000
<input type="checkbox"/> Workflow_Ulohy	<input type="checkbox"/>	20000
<input type="checkbox"/> Sprava_dokumentu	<input type="checkbox"/>	20000
Cena: 0		

Funkční blok	Jadro	Cena
<input type="checkbox"/> Helpdesk	<input type="checkbox"/>	10000
<input type="checkbox"/> Dispecink	<input type="checkbox"/>	10000
<input type="checkbox"/> Servis	<input type="checkbox"/>	10000
Cena: 0		

Obrázek 8.2: Obrazovka funkčních modulů

Druhá hlavní obrazovka je obrazovka odhadů. Viz Obrázek 8.3.

	Zprovoznění	Min čih	Max čih	Odhad konfigurace	Odhad analýzy	Odhad migrace	Odhad práv	Celkem čih	Cena
<input checked="" type="checkbox"/> Pokladna	0	8	16						0
<input checked="" type="checkbox"/> Majetek	0	8	32						0
<input type="checkbox"/> Vozovy_park	0	4	8						0
<input checked="" type="checkbox"/> Fakturace	0	4	8						0
<input type="checkbox"/> Kniha_jezd	0	8	16						0
OBCHOD									
<input checked="" type="checkbox"/> Zákazníci	0	4	8						0
<input checked="" type="checkbox"/> Dodavatelé	0	4	8						0
<input type="checkbox"/> CRM	0	8	32						0
<input checked="" type="checkbox"/> Sklady	0	8	24						0
<input type="checkbox"/> E-shop	8	8	80						0
PERSONALISTIKA									
<input checked="" type="checkbox"/> Lidske_zdroje	0	1	16						0
<input checked="" type="checkbox"/> Organizacni_struktura	0	1	8						0
<input type="checkbox"/> Dochazka	0	8	74						0

Obrázek 8.3: Obrazovka odhadů

Obrazovka Odhady navazuje na obrazovku Funkční moduly. Pokud je funkční blok vybrán, tak se v obrazovce Odhady mohou k jednotlivým funkčním blokům přiřazovat počty člověkohodin odhadů konfigurace, analýzy, migrace, práv. Pokud funkční blok není vybrán, tak není možné mu určit hodnotu odhadu. Pokud není hodnota odhadu v rozpětí minima a maxima, které je uvedeno vedle jednotlivého funkčního bloku, tak je před ovládacím prvkem zadávání červený vykřičník, který je možné odstranit vybráním hod-

noty z požadovaného rozpětí. Na konci obrazovky je vyobrazena celková cena odhadů, včetně ceny celkem za odhady a přepočítané ceny za korekci. Koeficient korekce práce se zadává v předchozí informativní obrazovce a násobí se jím odhady. Čím vyšší nejistota plyne, tím vyšší hodnotu je vhodné zvolit.

Třetí hlavní obrazovka je obrazovka Etapy. Obrazovka Etapy navazuje na obrazovku odhadů a zajišťuje rozložení odhadů do etap. Viz Obrázek 8.4.

	Konfigurace Etapa1	Konfigurace Etapa2	Analýza Etapa1	Analýza Etapa2	Migrace Etapa1	Migrace Etapa2	Práva Etapa1	Práva Etapa2	Celkem Etapa1	Celkem Etapa2	Celkem cih	Celkem cena
EKONOMIKA												0
<input checked="" type="checkbox"/> Pokladna												0
<input checked="" type="checkbox"/> Majetek												0
<input type="checkbox"/> Vozovy_park												0
<input checked="" type="checkbox"/> Fakturace												0
<input type="checkbox"/> Kniha_jezd												0
OBCHOD												0
<input checked="" type="checkbox"/> Zákazníci												0
<input checked="" type="checkbox"/> Dodavatelé												0
<input type="checkbox"/> CRM												0
<input checked="" type="checkbox"/> Sklady												0
<input type="checkbox"/> E-shop												0
PERSONALISTIKA												0
<input checked="" type="checkbox"/> Lidske_zdroje												0
<input checked="" type="checkbox"/> Organizacni_struktura												0
<input type="checkbox"/> Prace_konfig												0

Obrázek 8.4: Obrazovka etap

Další hlavní obrazovka je obrazovka Implementace, která zajišťuje činnosti, jak doplňkové, tak ty, co už obsažené jsou. Viz Obrázek 8.5.

Fáze	Činnost	Zahrnuto v odhadech	Odhad Etapa 1 cih	Cena Etapa 1	Odhad Etapa 2 cih	Cena Etapa 2	Odhad Celkem cih	Cena Celkem	Součinnost Etapa1 cih	Součinnost Etapa2 cih	Součinnost celkem
Analýza	Prezentace systému a jeho možností	<input checked="" type="checkbox"/>									0
	Konzultace s uživateli	<input type="checkbox"/>									
	Soupis požadavků na systém	<input type="checkbox"/>									
	Diskuse a zpětná vazba	<input type="checkbox"/>									
	Požadavky na migraci	<input type="checkbox"/>									
	Požadavky na přístupová práva	<input type="checkbox"/>									
Návrh	Výběr modulů	<input checked="" type="checkbox"/>									
	Příprava konfigurace	<input type="checkbox"/>									
	Definice vývoje nových funkcí	<input type="checkbox"/>									
	Vytvoření dokumentace	<input checked="" type="checkbox"/>									
	Diskuse a zpětná vazba k návrhu	<input checked="" type="checkbox"/>									
	Návrh migrace	<input type="checkbox"/>									
	Návrh přístupových práv	<input type="checkbox"/>									
Implementace	Instalace systému	<input checked="" type="checkbox"/>									
	Konfigurace existujících modulů	<input type="checkbox"/>									
	Vývoj nové funkčnosti	<input type="checkbox"/>									
	Integrace nové funkčnosti do systému	<input type="checkbox"/>									
	Průběžné testování funkčnosti	<input type="checkbox"/>									
	Příprava uživatelské dokumentace	<input checked="" type="checkbox"/>									

Obrázek 8.5: Obrazovka implementace

V obsažených činnostech jsou informace o tom, v jaké fázi se nachází. Doplňkovým činnostem se přiřazují hodnoty a rozdělují se do etap, jako v odhadech. V implementaci se zadávají i součinnosti.

Obrazovka celkové ceny nabízí pohled na celkovou cenu, jak funkčních

bloků, tak jednotlivých etap. Zároveň je zde možné zadat doplňkové činnosti, jako je například instalace databáze, nebo dalšího jazyka. Viz Obrázek 8.6.

Doplňkové položky instalace			
Položka	Cena v Kč	Počet	Celkem
Instalace serveru	4 000	<input type="text"/>	0
Instalace databáze	4 000	<input type="text"/>	0
Instalační balíček databáze + aplikační server	6 000	<input type="text"/>	0
Rozšíření o další jazyk uživatelského rozhraní	10 000	<input type="text"/>	0
Celkem ostatní náklady:			0

Výsledná cena za implementaci	
Položka	Cena (bez DPH)
Licence	3610000
Implementační práce Etapa 1	0
Implementační práce Etapa 2	0
Ostatní náklady	0
Implementace celkem	3610000
Maintanance poplatek (20% z ceny licencí)	722000

Obrázek 8.6: Obrazovka celkové ceny za implementaci

Obrazovka Finanční analýzy shrnuje ceny informačního systému, včetně cash flow. Zadává se zde hodnota diskontní sazby, úspor a umožňuje vygenerovat všechny informace zadané v konfigurátoru. Viz Obrázek 8.7.

Období implementace a dalších 5 let						
Rok	0	1	2	3	4	5
Licence	-3610000					
Etapa 1	0					
Etapa 2	0					
Ostatní náklady	0					
Maintenance	0	0	-722000	-722000	-722000	-722000
Podpora		-200000	-200000	-200000	-200000	-200000
Úspory		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
CF	-3610000	-200000	-922000	-922000	-922000	-922000
DCF	-3610000	-200000	-922000	-922000	-922000	-922000
Náklady celkem (za 5 let)						7498000
Úspory celkem (za 5 let)						-7498000
NPV						
ROI						-1 %

Součinnost	čís
Etapa 1	0
Etapa 2	0
Celkem:	0

Obrázek 8.7: Obrazovka finanční analýzy

Tvorba konfigurátoru vyžadovala spoustu času. Každá hlavní obrazovka se skládá z desítek galerií, do kterých se musely dát ovládací prvky. Desítky popisků, které byly třeba k popsání funkcí. Každý přesun prvku, nebo galerie vyžadoval přesunutí i těch ostatních, aby mezi sebou měly stejné mezery a aby byl konfigurátor vizuálně a funkčně přívětivý.

Kapitola 9

Testování, výsledky a optimalizace

Tato kapitola se věnuje výstupům uživatelského testování, výsledkům a optimalizaci konfigurátoru pro další verze.

9.1 Testování konfigurátoru

Testování probíhalo formou uživatelského testování. Nejdříve bylo nutné sdílet konfigurátor a data se všemi, kteří aplikaci testovali. Zpětná vazba byla kladná. Byly vyzdvihnuty informativní obrazovky, před každou hlavní obrazovkou. Dále vzhled konfigurátoru a možnost exportu finální finanční analýzy do souboru. Konfigurátor byl vyhodnocen jako přehledný a uživatelsky přívětivý. S funkcionalitou problém také nebyl. Bylo jasné, kde se má co zadávat a vyplňovat. Celková práce s konfigurátorem byla velice intuitivní.

Chyby na které jsem byla upozorněna jsem rozdělila na chyby vizuální, chyby ovládacích prvků a chyby databáze. Vizuální chyby byly nejčastěji jen chyby v textu, chybějící čárka, různé fonty a velikost písma. Tyto chyby bylo jednoduché opravit. Proto na Obrázku 9.1 je vidět stejný počet chyb, jako chyb opravených.

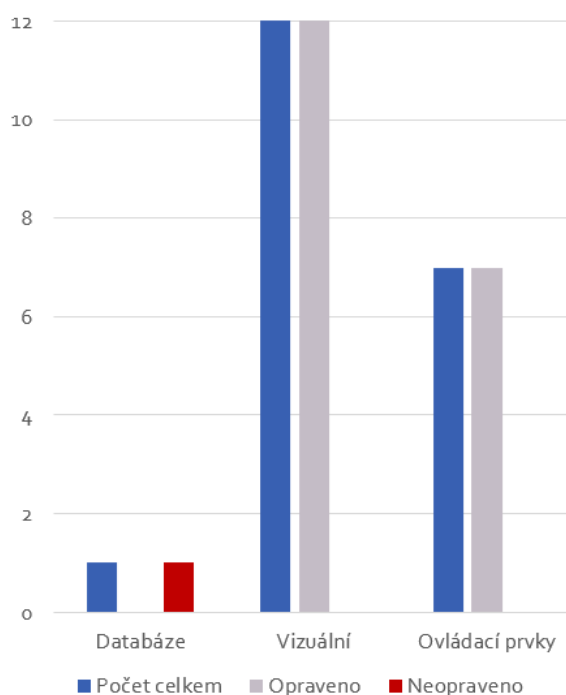
Další byly chyby v ovládacích prvcích, kdy hodnoty přesahovaly do jiných prvků, nebo nefungovali správně. Chyby ovládacích prvků, na které jsem byla upozorněna jsem opravila všechny viz Obrázek 9.1.

Na chyby s databází jsem narazila v každé zpětné vazbě. Data se napříč ovládacími prvky přepisovala, byla dlouhá prodleva v zápisu, nebo se nezapsala vůbec. Nakonec jsem to uvedla jako jednu chybu, jelikož je to problém celé databáze. Na obrázku je tedy vidět počet chyb celkem stejný, jako počet těch neopravených.

9.1.1 Problém se sdílením a chybá databáze

Po nasdílení konfigurátoru jsem si uvědomila, že je zapotřebí nasdílet i databázi v Excelu s daty a všemi tabulkami, do kterých se propisuje veškerá interakce z konfigurátoru. Tím nastal problém při současném přístupu více lidí do konfigurátoru. Uživatelé si navzájem přepisovali data uložená v databázi.

Snažila jsem se to řešit pomocí toků. Vytvořila jsem automatizovaný tok, který měl vytvořit kopii originální databáze ve výchozí složce uživatele,

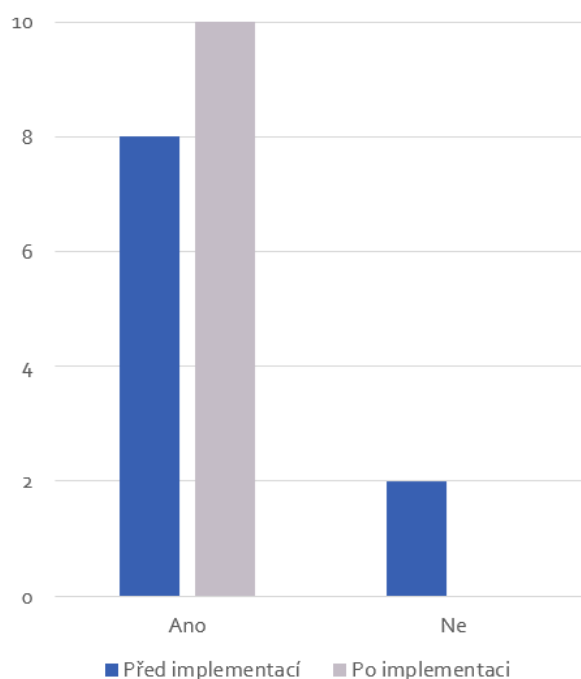


Obrázek 9.1: Analýza chyb

který konfigurační testoval. Nový soubor by uživatel mohl upravovat jen pro sebe a nepřepisoval by hodnoty ostatním. Cesta na nový soubor byla upravena v datech a tok byl navázán na tlačítko "START" na zahajovací obrazovce, které slouží pro spuštění práce na konfiguračnímu. Fungovalo to jen v případě, kdy jsem to testovala u sebe. Po kliknutí na tlačítko se nikomu, kdo konfigurační testoval, databáze ve výchozí složce neobjevila, a proto vyskakovalo upozornění, že tabulky neexistují. Bohužel se mi tento problém nepodařilo vyřešit.

9.2 Výsledky testování

Nejdříve bych chtěla zmínit analýzu, která byla vytvořena ještě před začátkem práce. Analýza se týkala využití konfiguračního při výuce. Kontaktováni byli studenti předmětu Informační systémy, kteří předmět již měli za sebou viz Obrázek 9.2. Tehdy uvedlo 8 studentů z 10, že by konfigurační využili ve svých pracích, zmíněných ve 4. Kapitole. Nutno poznamenat, že se během roku změnila podmínka na studenty vypracovávaný projekt. Studenti navíc vyplňují Excel dokument, který má konfigurační nahradit. Na Obrázku 9.2 je dále možné vidět, jak se studenti staví ke konfiguračnímu před jeho implementací a jak se k němu staví po tom, co ho testovali. Po otestování uvedlo 10 z 10 studentů, že by konfigurační využili při tvorbě svých prací. Sloupeček v grafu na Obrázku 9.2 tedy není vidět, jelikož obsahuje nulové hodnoty.



Obrázek 9.2: Použití konfigurátoru

V návaznosti na zpětnou vazbu z testování a finální analýzy, bylo dokázáno, že by konfigurátor byl přínosem pro výuku v případě změny databáze.

9.3 Návrh budoucích optimalizací

Pro následující verze konfigurátoru by bylo vhodné se zamyslet nad jiným pojetím databáze. Konfigurátor v této verzi funguje tak, že mají všichni stejná data a navzájem si je přepisují. V mém případě by to fungovalo jen za předpokladu, že by všichni zapsaní studenti na předmětu vypracovávali jeden a ten samý projekt.

Dále by se konfigurátor mohl vylepšit přidáním možnosti výběru více etap na projekt. V tomto projektu se pracuje se dvěma, ale v praxi jich může být i více.

Také je možná optimalizace v zadávání prací, kdy ceny prací nebudou přednastavené, ale studenti si je budou volit a následně se budou dopočítávat ceny odhadů v jednotlivých fázích a činnostech implementace.

Kapitola 10

Závěr

Moje bakalářská práce pojednávala o návrhu konfigurátoru IS a jeho implementaci.

Na začátku práce jsem objasnila témata důležitá z oblasti IS. Zaměřila jsem se na definice základních pojmů, které práci provázejí. Zanalyzovala jsem výuku IS včetně toho, jak ji vnímají studenti. Neopomenula jsem problémy, se kterými se studenti nejčastěji setkávají a navrhla změny, které budou usnadňovat jak výuku, tak chápání komplexnosti informačních systémů.

Dále jsem prozkoumala existující IS a zmapovala klíčové oblasti, které jsem zakomponovala do konfigurátoru, aby je studenti mohli využívat ve svých pracích. Do jednotlivých oblastí jsem zařadila funkčnosti tak, aby mohli studenti přiřazovat odhady z fáze implementace a následně je rozřazovat do etap. Nastínila jsem proces implementace a jeho jednotlivé fáze, včetně toho, s čím je možné se v každé fázi setkat. Z průzkumu trhu pak vzešlo nacenění prací, které odpovídá reálným cenám na trhu.

Vytvořila jsem základní UML diagramy a požadavky, které pomohly s počátečním návrhem konfigurátoru.

V praktické části jsem analyzovala prostředí a navrhla konfigurátor, který pomáhá studentům při vytváření jejich IS pochopit ceny a všechny práce, které jsou s tvorbou spjaty. Konfigurátor provádí studenta všemi fázemi vývoje systému a vytvoří finální finanční analýzu z funkčností, odhadů a činností implementace, které student zadal.

Dále jsem popsala tvorbu konfigurátoru v Power Apps. Od výběru prostředí přes funkce, ovládací prvky, proměnné, po sdílení a export. Praktická část provádí implementaci konfigurátoru a nabízí návod pro vytváření vlastních aplikací.

Závěrem jsem provedla zhodnocení všech výsledků, mých poznatků a zpětné vazby od studentů včetně uživatelského testování a navrhla možnou optimalizaci pro další verze.

Všechny stanovené cíle se mi povedlo naplnit, včetně hlavního cíle vytvořit první verzi konfigurátoru. Hlavním přínosem práce je, že konfigurátor pomáhá studentům lépe pochopit tvorbu všech náležitostí IS. Hlavním přínosem pro mě byla získaná zkušenost a informace při tvorbě takto rozsáhlého projektu a seznámení se s novou platformou Power Apps.



Literatura

- [1] D. Chlapek V. Řepa I. Stanovská J. Voříšek T. Bruckner, A. Buchalce-
vová. Tvorba informačních systémů: Principy, metodiky, architektury.
Editio Grada, 2012, ISBN 978-80-247-4153-6.
- [2] Barbora Kodoušková. Informační systémy v kostce: Erp, crm, implemen-
tace[online]. 2020 [cit. 28.12.2020].
Dostupné z:
<https://www.rascasone.com/cs/blog/>.
- [3] Pavel Bezpalec. Co je ict systém [online]. 2020.
Dostupné z:
<https://publi.cz/books/242/01.html>.
- [4] TECHart systems s.r.o. Erp nebo crm systém? mějte jeden informační
systém na všechno... [online]. 2017 [cit. 05.01.2020].
Dostupné z:
[https://wisart.cz/novinky/erp-nebo-crm-system-mejte-jeden-
informacni-system-na-vsechno](https://wisart.cz/novinky/erp-nebo-crm-system-mejte-jeden-informacni-system-na-vsechno).
- [5] Asociace za lepší ICT řešení. Dms systémy – řízení dokumentů[online].
2020.
Dostupné z:
<https://lepsi-reseni.cz/>.
- [6] Kumar Vivek. Co je power apps?[online]. 2020 [cit. 05.01.2020].
Dostupné z:
<https://docs.microsoft.com/cs-cz/powerapps/powerapps-overview>.
- [7] Solitea Vario. Moduly a funkce. 2021.
Dostupné z:
<https://www.vario.cz/>.
- [8] eWay CRM. Moduly. 2021.
Dostupné z:
<https://www.eway-crm.com/cs/moduly/>.
- [9] Ph.D. Ing. Pavel Náplava. Materiály pro výuku.[online]. 2020.

- [10] K2 atmitec s.r.o. Kolik stojí informační systém? 2021.
Dostupné z:
<https://www.k2.cz/cs/kolik-stoji-informacni-system>.
- [11] Ing. Martin Komárek. Specifikace požadavků [online materiály k výuce předmětu smp]. 2016.
- [12] David Čápka. Lekce 2 - uml - use case diagram. 2018.
Dostupné z:
<https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>.
- [13] Petra Rejnková. Diagram nasazení. 2009.
Dostupné z:
http://m.uml.czweb.org/diagram_nasazeni.htm.
- [14] Microsoft. Dokumentace power apps. 2020.
Dostupné z:
<https://docs.microsoft.com/en-us/powerapps/>.
- [15] Microsoft. Dokumentace power automate. 2020.
Dostupné z:
<https://docs.microsoft.com/cs-cz/power-automate/>.
- [16] Microsoft. Dokumentace power bi. 2020.
Dostupné z:
<https://docs.microsoft.com/cs-cz/power-bi/fundamentals/power-bi-overview>.

Příloha A

Tabulky modulů

A.1 Moduly

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
Ekonomika	Polkadna	true	0	false	false	isCore	0	8	16				
	Majetek	true	0	false	false	isCore	0	8	32				
	Vozovy_park	false	5000	false	false	isFalse	0	4	8				
	Fakturace	true	0	false	false	isCore	0	4	8				
	Knihna_jezd	false	5000	Vozovy_park	false	isFalse	0	8	16				

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
	Zakaznici	true	0	false	false	isCore	0	4	8				
	Podavatvile	true	0	false	false	isCore	0	4	8				
	CRM	false	20000	false	false	isFalse	0	8	32				
	Služby	true	0	false	false	isCore	0	8	24				
	E-shop	false	15000	Uzivatelsky_portal	false	isFalse	8	8	80				

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
Personalizika	Uctvke_zdroje	true	0	false	false	isCore	0	1	16				
	Organizacni_struktura	true	0	false	false	isCore	0	1	8				
	Porizacka	false	5000	false	false	isFalse	0	8	24				

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
	Zdroje	true	0	false	false	isCore	0	1	8				
	Rezent_vyraby	false	15000	Planovani_vyraby	false	isFalse	0	8	32				
	Planovani_vyraby	false	15000	false	false	isFalse	0	8	32				

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
	Rezent_projektu	false	15000	false	false	isFalse	0	8	32				
	Workflow/ulohy	false	20000	false	false	isFalse	0	1	16				
	Sprava_dokumentu	false	20000	false	false	isFalse	0	8	16				
	Komunikace	false	10000	false	Zastlani_zprav	isFalse	0	8	16				

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
	Helpdesk	false	10000	false	false	isFalse	0	8	32				
	Dispečerik	false	10000	false	false	isFalse	0	8	32				
	Service	false	10000	false	false	isFalse	0	8	32				

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
Reporting	Business Intelligence	false	50000	false	false	isFalse	16	40	120				
	Uzivatelsky_portal	false	40000	false	Mezkyby_klient	isFalse	8	4	40				
	Topika_reportu	true	0	false	false	isCore	0	1	32				

Funkcni_modul	Funkcni_blok	Jadro_systemu	cena	Vzaba_funkcni_blok	Vzabz_system_modul	Porizeno	Zprovovzneni	min_cih	max_cih	Odhad_konfigurace	Konf_Etapaz1	Konf_Etapaz2	Odhad_analyzy
	Uzivatelsky_vyvoj	false	40000	false	false	isFalse	0	4	100				
	Vyvoze_sluzby	false	5000	false	API	isFalse	0	4	24				
	Export/import	true	0	false	Import/Export	isCore	0	1	24				

Systemovy_modul	Systemovy_blok	Jadro_systemu	Cena	Vazba_funkcni_blok	Portizeno	PowerApps
Systemovy_modul	Identity_management	false	10000	false	isFalse	o_TxscLwVz4
	Zasilani_zprav	false	10000	false	isFalse	Kl6yJY73nv8
	Import/Export	true	0	false	isCore	0k83tRAC68
	API	false	0	Webove_sluzby	isFalse	VccMsdqGt7d

Klient	Klient_blok	Jadro_systemu	Cena	Vazba_funkcni_blok	Portizeno	PowerApps
Klient	Webovy_Klient	false	0	Uzivatel'sky_portal	isFalse	B_gvPRsG9S5
	Aplikace_Java	false	5000	false	isFalse	L6cmhwC2x0
	Aplikace_NET	true	0	false	isCore	6GJU-Yzajno
	Mobilni_Klient_Android	false	5000	false	isFalse	REFU_fm3dws
	Mobilni_Klient_IOS	false	5000	false	isFalse	WDRv4WmlGvK

Uzivatele	Pocet_Uzivatelu	Jadro_systemu	Cena	Vazba_funkcni_blok	Pocet	PowerApps
Uzivatele	Zakladnich_5_uzivatelu	true	0	false		RvCtflxHo
	1_uzivatel	false	10000	false		0Lc-mV8_VE
	Balcek_3_uzivatelu	false	25000	false		9KNPC0IP35g
	Balcek_5_uzivatelu	false	40000	false		YOK_pHPDM
	Balcek_10_uzivatelu	false	70000	false		IIfvcJtaJsk

