

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Tvorba modulu Statistiky pro organizační systém 1CLICK

Bc. Jakub Koláčný

Diplomová práce

2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub Koláčný**
Osobní číslo: **I17206**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Tvorba modulu Statistiky pro organizační systém 1CLICK**
Zadávací katedra: **Katedra softwarových technologií**

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

Teoretická část bude obsahovat tvorbu analýzy a návrh daného modulu obsahující funkční a nefunkční požadavky, případy užití, diagramy aktivit a analytické třídy. Dále bude vytvořen přehled datových položek systému 1CLICK vhodných pro tvorbu statistických výstupů a zvoleny vhodné matematické a statistické funkce pro zpracování dat. Bude navrženo funkční a grafické rozhraní pro definici uživatelsky volitelných statistických výstupů společně s rozhraním pro vizualizaci zpracovaných dat systému. Nedílnou součástí teoretické části bude návrh doporučení jednotlivých řešení a jejich využití. V praktické části bude provedena realizace, validace a implementace modulu, a to jak serverové části pro získávání dat ze systému a jejich další zpracování, tak i webového uživatelského rozhraní určeného pro definici požadavků na statistické výstupy a jejich grafickou prezentaci. Následně bude otestováno výsledné řešení.

Rozsah grafických prací: 10
Rozsah pracovní zprávy: cca 40–50 stran
Forma zpracování diplomové práce: tištěná
Seznam odborné literatury:

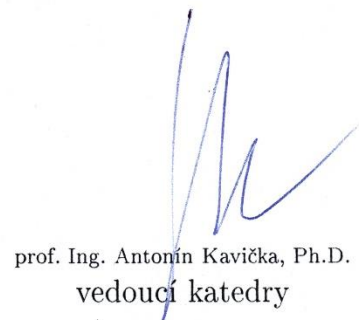
[1] ARLOW, Jim a Ila NEUSTADT. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9. [2] PECINOVSKÝ, Rudolf. Java 8: úvod do objektové architektury pro mírně pokročilé. Praha: Grada Publishing, 2014. Knihovna programátora (Grada). ISBN 978-80-247-4638-8. [3] MASSÉ, Mark. REST API design rulebook. Sebastopol, CA: O'Reilly, 2012. ISBN 978-1-449-31050-9.

Vedoucí diplomové práce: Ing. Josef Skalický
Konzultant diplomové práce: doc. Ing. Michael Bažant, Ph.D.
Katedra softwarových technologií

Datum zadání diplomové práce: 22. října 2018
Termín odevzdání diplomové práce: 18. května 2019



Ing. Zdeněk Němec, Ph.D.
děkan



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

PROHLÁŠENÍ AUTORA

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17. května 2019

Bc. Jakub Koláčný

PODĚKOVÁNÍ

Rád bych poděkoval Ing. Josefu Skalickému za cenné rady a věcné připomínky v celém průběhu tvorby diplomové práce. Mé poděkování patří majitelům společnosti One Click Business Solution s. r. o. za projevenou důvěru, nastínění požadavků a definici představ na daný modul a jeho grafické zpracování. V neposlední řadě patří mé poděkování doc. Ing. Michaelu Bažantovi, Ph.D. za odborné vedení diplomové práce na univerzitní půdě.

ANOTACE

Diplomová práce pojednává o hromadném zpracování dat a možnostech, jak tato data upravovat do manažersky zajímavých statistických výstupů umožňujících zvyšování efektivity práce jednotlivců i týmů ve firmách či společnostech. Hlavním cílem této práce je analýza, návrh a implementace reportního statistického modulu pro systém 1CLICK za použití programovacích jazyků Java a JavaScript. Nedílnou součástí diplomové práce je přehled vybraných druhů podnikových systémů a popis jejich elementárních principů a jejich využití v rámci systému 1CLICK.

KLÍČOVÁ SLOVA

1CLICK, podnikové systémy, zpracování dat, statistiky, report, modul, Java, JavaScript

TITLE

Creation of module Statistics for organizational system 1CLICK

ANNOTATION

This master thesis deals with mass data processing and options for modifying these data into interesting statistical output for managers, because it allows to increase work efficiency of individuals and groups in companies. The main goal of this master thesis is an analysis, a draft and an implementation of module Statistics for system 1CLICK, using programming languages Java and JavaScript. A part of this master thesis is also an overview of selected types of enterprise software's, description of their fundamental principles and their usage within system 1CLICK.

KEYWORDS

1CLICK, business software, processing data, statistics, report, module, Java, JavaScript

OBSAH

0	ÚVOD	14
1	PODNIKOVÉ SYSTÉMY (ENTERPRISE SOFTWARE)	15
1.1	Enterprise Resource Planning – ERP	15
1.2	Content Management System – CMS	15
1.3	Document Management Systems – DMS	16
1.4	Customer Relationship Management – CRM	16
1.5	Human Resources Management – HRM.....	17
1.6	Product Data Management – PDM	17
1.7	Product Lifecycle Management – PLM	17
1.8	Supply Chain Management – SCM.....	18
2	SYSTÉM 1CLICK.....	19
2.1	Společnost One Click Business Solution s. r. o.	19
2.2	Systém 1CLICK.....	19
2.3	Moduly	19
2.3.1	Úkoly	20
2.3.2	E-maily.....	21
2.3.3	Kalendář.....	21
2.3.4	Projekty	21
2.3.5	Procesy.....	22
2.3.6	Organizační struktura.....	23
2.3.7	Dokumenty (DMS)	23
2.3.8	Kontakty (CRM)	23
2.3.9	Doplňkové moduly	23
2.4	Konkurence – jiné systémy	25
2.5	Důvod tvorby modulu Reporty – Statistiky	25
3	STATISTICKÉ ZPRACOVÁNÍ DAT	26

3.1	Popisná statistika.....	26
3.1.1	Základní statistické pojmy.....	26
3.1.2	Charakteristiky polohy.....	27
3.1.3	Charakteristiky variability.....	28
3.2	Rozdělení četností a histogram.....	30
3.3	Časové řady.....	33
4	TECHNOLOGIE POUŽITÉ PRO VÝVOJ.....	35
4.1	Java 8.....	35
4.1.1	Guava: Google Core Libraries for Java.....	37
4.1.2	Apache Commons Mathematics Library 3.....	37
4.1.3	Apache POI.....	38
4.2	JavaScript.....	39
4.2.1	jQuery.....	40
4.2.2	Chart.js.....	42
4.3	REST API.....	43
5	ANALÝZA.....	46
5.1	Požadavky.....	46
5.1.1	Funkční požadavky.....	46
5.1.2	Nefunkční požadavky.....	49
5.2	Případy užití.....	50
5.3	Diagram aktivit.....	52
5.4	Návrh analytických tříd.....	53
6	POPIS FUNKCIONALITY GUI.....	54
6.1	Vyhledávací filtr.....	54
6.1.1	Správa šablon pro přednastavení filtru.....	55
6.1.2	Dialogy.....	56
6.2	Report.....	57

6.2.1	Detail seznamu úkolů.....	58
6.3	Statistika.....	59
6.3.1	Grafy.....	59
6.4	Export do souboru XLSX.....	62
6.4.1	Export souhrnného reportu a statistiky.....	62
6.4.2	Export seznamu úkolů.....	62
7	FUNKČNÍ ČÁST DOPLŇKU.....	63
7.1	Popis balíčků a tříd serverové části doplňku.....	63
7.1.1	Balíček main.....	63
7.1.2	Balíček acl.....	64
7.1.3	Balíček export.....	64
7.1.4	Balíček filter.....	65
7.1.5	Balíček report.....	65
7.1.6	Balíček statistics.....	67
7.1.7	Balíček templates.....	68
7.1.8	Balíček utils.....	69
7.1.9	Balíček web.....	69
7.2	Rozšiřující REST API modulu.....	69
7.3	Volání REST API z klientské části modulu.....	70
7.4	Průběh komunikace mezi uživatelem, klientem a serverem.....	71
7.5	Dynamické zobrazení a generování klientské části.....	71
7.5.1	Popis zdrojových souborů pro webové GUI.....	71
8	TESTOVÁNÍ A VALIDACE.....	73
9	ZÁVĚR.....	74
	POUŽITÁ LITERATURA.....	76
	SEZNAM PŘÍLOH.....	79

SEZNAM ILUSTRACÍ

Obrázek 1 – Modul úkoly v systému 1CLICK	20
Obrázek 2 – Schéma procesu "Faktura přijatá" [10]	22
Obrázek 3 – Rozdělení četností s odlišnou variabilitou [13].....	29
Obrázek 4 – Polygon četností	31
Obrázek 5 – Histogram	33
Obrázek 6 – Dekompozice časové řady $x(n)$ na trendovou složku $T(n)$, cyklickou složku $C(n)$ a náhodnou složku $\varepsilon(n)$ [14]	34
Obrázek 7 – Ukázka grafu polární oblast	43
Obrázek 8 – REST API architektura.....	45
Obrázek 9 – Funkční požadavky: Report	46
Obrázek 10 – Funkční požadavky: Statistika	47
Obrázek 11 – Funkční požadavky: Šablony	48
Obrázek 12 – Funkční požadavky: Export	48
Obrázek 13 – Nefunkční požadavky: Bezpečnost	49
Obrázek 14 – Nefunkční požadavky: Dostupnost	49
Obrázek 15 – Nefunkční požadavky: Logování a Vzhled	49
Obrázek 16 – Diagram případů užití: Modul Reporty – Statistiky.....	50
Obrázek 17 – Matice sledovatelnosti požadavků	51
Obrázek 18 – Diagram aktivit: Komunikace uživatel-klient-server	52
Obrázek 19 – Návrh analytických tříd modulu Reporty – Statistiky.....	53
Obrázek 20 – Vyhledávací filtr.....	54
Obrázek 21 – Nastavení přístupových práv modulu Reporty – Statistiky.....	55
Obrázek 22 – Správa šablon pro přednastavení filtru: Smazání	55
Obrázek 23 – Správa šablon pro přednastavení filtru: Uložení	56
Obrázek 24 – Legenda: Ikony používané v modulu	56
Obrázek 25 – Dialogy ukládání a mazání šablon filtru.....	57
Obrázek 26 – Souhrnný report.....	57
Obrázek 27 – Dialog s detailním seznamem úkolů	58
Obrázek 28 – Navigační menu pro přepínání pohledů mezi statistickými grafy.....	59
Obrázek 29 – Graf počtu všech úkolů uživatelů systému 1CLICK.....	60
Obrázek 30 – Graf vyřešení úkolů v čase	60
Obrázek 31 – Histogram délek trvání úkolů.....	61

Obrázek 32 – Časová řada počtu založených úkolů	61
Obrázek 33 – Přidané tlačítko <i>Reporty – Statistiky</i> v systému 1CLICK	64

SEZNAM TABULEK

Tabulka 1 – Tabulka bodového rozdělení četností	31
Tabulka 2 – Tabulka intervalového rozdělení četností	32

SEZNAM ZKRATEK A ZNAČEK

ACL	Access Control List
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ARES	Administrativní Registr Ekonomických Subjektů
CalDAV	Calendaring Extensions to WebDAV
CDA	Content Delivery Application
CGI	Common Gateway Interface
CMA	Content Management Application
CMS	Content Management System
CRM	Customer Relationship Management
CSS	Cascading Style Sheets
DMS	Document Management Systems
DOM	Document Object Model
ECM	Enterprise Content Management
ERP	Enterprise Resource Planning
GUI	Graphic User Interface
HRM	Human Resources Management
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
ISO	International Organization for Standardization
JDK	Java Development Kit
JS	JavaScript
JSON	JavaScript Object Notation
OOXML	Office Open XML
PDM	Product Data Management
PLM	Product Lifecycle Management
REST	Representational State Transfer
RGB	Red, Green, Blue
RPC	Remote procedure call
SCM	Supply Chain Management
SMTP	Simple Mail Transfer Protocol

SOAP	Simple Object Access Protocol
SQL	Structured Query Language
URI	Uniform Resource Identifier
WCM	Web Content Management
WebDAV	Web Distributed Authoring and Versioning
XML	Extensible Markup Language

0 ÚVOD

Již ze samotné definice informačního systému vyplývá, že jednou z jeho základních funkcí je tvorba prezentace informací pro potřeby uživatelů.

Před třemi lety společnost One Click Business Solutions s. r. o. nabídla autorovi této práce pozici juniorského backendového vývojáře, kterou přijal. Během tříleté praxe dospěl k přesvědčení, že je produkt 1CLICK kvalitním informačním systémem s dobře postavenou filosofií a architekturou umožňující pokrytí všech potřeb jednotlivců, firem i společností v oblasti řízení. Za velký hendikep systému 1CLICK je však považována nedostatečná práce s pořízenými daty, a to zejména v podobě chybějících reportních a statistických výstupů prováděných jak nad konkrétními částmi systému, tak nad celým systémem.

Právě na základě zmíněného faktu je tato diplomová práce zaměřena na tvorbu modulu Statistiky jakožto rozšiřujícího doplňku systému 1CLICK, určeného pro vyhodnocování práce nad úkolovou částí systému. Výstup z tohoto reportu bude sloužit střednímu a vrcholovému managementu ke zvýšení efektivity práce jednotlivců, týmů i celé firmy či společnosti.

Teoretická část diplomové práce nastíní zejména rozdělení nejčastěji se vyskytujících podnikových systémů a jejich popis, obecné seznámení se společností One Click Business Solutions s. r. o. a jejím produktem 1CLICK s detailním rozborem jednotlivých modulů tohoto systému. Nedílnou součástí práce je také shrnutí a vysvětlení základních znalostí pro samotnou tvorbu této diplomové práce z oblastí statistického zpracování dat a popisu technologií potřebných pro vývoj, zejména programovacích jazyků Java a JavaScript.

Praktická část této diplomové práce detailně popisuje jednotlivé kroky vývoje modulu Statistiky. Prvním a základním stavebním kamenem každého vývoje je analýza řešení zpracovaná dle unifikovaného procesu vývoje aplikací včetně návrhu analytických tříd. V následující kapitole čtenář nalezne detailní popis funkcionality grafického uživatelského rozhraní tohoto modulu, který slouží i jako podrobný návod k práci s modulem Statistiky, a to jak z uživatelského pohledu, tak i z pohledu správce systému. Důležitou součástí práce je také deskripce funkční části doplňku, ve které je popsána převážně serverová část tohoto rozšiřujícího modulu pro systém 1CLICK.

1 PODNIKOVÉ SYSTÉMY (ENTERPRISE SOFTWARE)

1.1 Enterprise Resource Planning – ERP

Enterprise Resource Planning v češtině znamená plánování podnikových zdrojů. Jedná se o sadu softwarových nástrojů sloužících pro správu podnikatelských činností, např. řízení projektů, účetnictví, řízení rizik nebo dodržování předpisů a činností v rámci dodavatelských řetězců. Nedílnou součástí ERP je řízení podnikového výkonu, plánování, rozpočtování a reporting výsledků organizace. Klíčovým principem ERP je centrální sběr dat pro širokou distribuci. ERP systémy sbírají a uchovávají sdílená transakční data z více zdrojů organizace, eliminují duplicity dat a poskytují integritu dat s jediným zdrojem pravdy. ERP systémy jsou navrženy kolem společné, definované datové struktury, která má obvykle společnou databázi. To pomáhá poskytovat informace napříč podnikem pomocí normalizovaných dat založených na společných definicích a uživatelských zkušenostech. Dnes jsou ERP systémy klíčové pro řízení tisíců organizací napříč veškerými obory [1].

1.2 Content Management System – CMS

Systém pro správu obsahu (Content Management System) je softwarová aplikace nebo sada souvisejících programů, sloužící k vytváření a správě digitálního obsahu. CMS zahrnuje nástroje pro správu podnikového obsahu (ECM) a správu webového obsahu (WCM). ECM nástroje zajišťují integraci a správu dokumentů, digitálních aktiv a uchovávání záznamů. WCM nástroje usnadňují spolupráci pro tvorbu webových stránek. Systémy pro správu podnikového obsahu se skládají z:

- Aplikace pro správu obsahu (CMA) – grafické uživatelské rozhraní, které umožňuje uživateli kontrolovat vytváření, úpravu a odstraňování obsahu z webových stránek bez nutnosti znalosti HTML.
- Aplikace pro poskytování obsahu (CDA) – komponenta zajišťující služby, které podporují správu a doručování obsahu po jeho vytvoření v CMA.

Za nejvýznačnější vlastnosti CMS jsou považovány indexování, vyhledávání, správa formátů, kontrola revizí a publikování [2].

1.3 Document Management Systems – DMS

Document Management Systems neboli systémy správy dokumentů jsou v podstatě elektronické registratury, které organizace/firma může použít jako základ pro uspořádání všech digitálních dokumentů. DMS umožňují uživatelům zadávat metadata, značky a další popisné informace k daným dokumentům, které lze použít k lepšímu uspořádání a vyhledávání uložených souborů. Většina softwaru pro správu dokumentů obsahuje integrovaný vyhledávací modul, který umožňuje uživatelům rychle najít příslušný soubor i v těch největších knihovnách dokumentů. DMS mají také nastavení přístupových oprávnění, takže pouze příslušní zaměstnanci mají přístup k privilegovaným souborům a informacím. Mezi jedny z nejdůležitějších funkcí DMS patří:

- Uchovávání různých typů souborů, včetně souborů pro zpracování textu, e-mailů, souborů PDF, tabulek a jiných.
- Vyhledávání podle klíčových slov, značek a dalších popisných vlastností.
- Přístupová oprávnění k jednotlivým dokumentům nebo skupinám dokumentů.
- Nástroje pro sledování přístupu uživatelů k daným dokumentům.
- Nástroje pro správu verzí, které sledují úpravy dokumentů a umožňují obnovy starších verzí dokumentů.
- Ovládací prvky umožňující hromadné úpravy dokumentů, např. odstranění zastaralých dokumentů [3].

1.4 Customer Relationship Management – CRM

Softwary určené pro řízení vztahů a interakcí společností se současnými i potencionálními zákazníky se nazývají Customer Relationship Management. Technologie CRM umožňuje firmám shromažďovat a spravovat velké množství zákaznických dat a následně provádět strategie a plánování založené na těchto informacích. Data shromážděná prostřednictvím cílených podnětů, jako jsou různé komunikační kanály, včetně internetových stránek společnosti, telefonátů, e-mailů, online chatů, marketingových materiálů i sociálních sítí, napomáhají firmám řešit konkrétní problémy v průběhu jejich cyklu vztahů se zákazníky. Data také poskytují společnostem důležité pohledy na potřeby a chování zákazníků, což jim umožňuje přizpůsobit např. nabídku produktů cíleným segmentům zákazníků. Informace získané prostřednictvím CRM často generují řešení problémů mimo marketingové funkce společnosti, jako jsou řízení dodavatelského řetězce nebo vývoj nových produktů [4].

1.5 Human Resources Management – HRM

Řízení lidských zdrojů (HRM) zahrnuje principy práce v personální oblasti společností. HRM, označované často také zkráceným pojmem lidské zdroje (HR), zahrnuje:

- nábor zaměstnanců,
- řízení pracovních sil a přiřazení druhu práce,
- dodržování pracovního práva,
- kontrolu výkonnosti,
- vzdělávání a školení zaměstnanců,
- ohodnocení zaměstnanců.

HRM je skutečným řízením zaměstnanců s důrazem na tyto zaměstnance jako na aktiva podniku. V této souvislosti se zaměstnanci někdy označují jako lidský kapitál. Stejně jako u ostatních obchodních aktiv je cílem efektivní využití zaměstnanců, snížení rizika a maximalizace návratnosti investic. Téměř všechny oblasti HRM mají svůj specifický software, který automatizuje různé stupně procesů lidských zdrojů. Nábor uchazečů o zaměstnání například zažil obrovský nárůst počtu softwarových platforem a systémů, které pomáhají jak zaměstnavatelům, tak žadatelům o zaměstnání elektronicky porovnávat organizace a kandidáty, a pak pomáhat při správě pohovorů a náborových procesů [5].

1.6 Product Data Management – PDM

Systémy pro správu produktových dat (PDM) poskytují nástroje pro řízení přístupu a správu datových definic jednotlivých produktů. Uchovávají tzv. metadata o daných produktech. Nástroje PDM poskytují kooperaci s řízením procesů, zejména v oblastech řízení konfigurace nebo řízení technické změny produktů. Správa produktových dat slouží jako centrální úložiště znalostí pro procesní a produktovou historii a podporuje integraci a výměnu dat mezi všemi obchodními uživateli, kteří pracují s produkty včetně projektových manažerů, inženýrů, prodejců, kupujících i týmů zajišťujících kvalitu [6].

1.7 Product Lifecycle Management – PLM

Správa životního cyklu produktu (PLM) je všestranným přístupem pro inovace, vývoj a zavádění nových produktů. Klíčovou součástí PLM je řízení informací od prvotních myšlenek až po konečnou fázi životního cyklu produktu. PLM zahrnuje propojení zainteresovaných osob, sběr a práci s daty, řízení procesů a podnikových systémů a poskytuje informace o výrobcích

pro společnosti a jejich rozšířenou podnikovou síť. Díky své schopnosti integrovat všechna data a procesy související s produkty a eliminovat hranice v hodnotovém řetězci mohou PLM systémy výrazně snížit aktivity bez přidané hodnoty a umožnit zúčastněným stranám spolupracovat v reálném čase pomocí konzistentních informací během celého životního cyklu produktu. V dnešním vysoce konkurenčním, rychle se rozvíjejícím a globálním podnikatelském prostředí mohou dobře navržené a implementované postupy PLM poskytnout společností skutečnou konkurenční výhodu [7].

1.8 Supply Chain Management – SCM

Řízení dodavatelského řetězce (SCM) je sada procesů aktivního řízení činností dodavatelského řetězce s cílem maximalizovat hodnotu zákazníků a dosáhnout udržitelné konkurenční výhody. Představuje vědomou snahu firem rozvíjet a provozovat dodavatelské řetězce co nejúčinnějším způsobem. Činnosti dodavatelského řetězce zahrnují vše od vývoje produktů, zásobování, výroby a logistiky k informačním systémům potřebným pro koordinaci těchto činností.

Organizace, které tvoří dodavatelský řetězec, jsou propojeny prostřednictvím fyzických a informačních toků:

- Fyzické toky zahrnují transformaci, pohyb a skladování zboží a materiálů. Jsou nejviditelnější částí dodavatelského řetězce.
- Informační toky umožňují různým partnerům v dodavatelském řetězci koordinovat své dlouhodobé plány a kontrolovat každodenní tok zboží a materiálů [8].

2 SYSTÉM 1CLICK

Tato kapitola popisuje společnost One Click Business Solution s. r. o., její hlavní vyvíjený produkt – systém 1CLICK a jeho jednotlivé moduly. Kapitola čerpá z [9].

2.1 Společnost One Click Business Solution s. r. o.

Společnost One Click Business Solution s. r. o. vznikla v roce 2011 okolo týmu profesionálů v oblasti zvyšování efektivity a zjednodušování práce vnitrofiremních struktur. Vývoj softwarového nástroje 1CLICK byl založen na dlouholetých zkušenostech z teoretického a praktického managementu jednoho ze zakladatelů a současného jednatele a výkonného ředitele Vladimíra Patzaka. Dnes se jedná o zavedenou společnost působící na českém a slovenském trhu s ambicí průniku na zahraniční trhy.

2.2 Systém 1CLICK

Systém 1CLICK byl uveden na trh v roce 2011 jako nástroj pro úkolování a tvorbu organizační struktury. Postupem času byly vyvíjeny a doplňovány další moduly vedoucí až k dnešní podobě systému, která nabízí ucelené a univerzální řešení pokrývající potřeby širokého spektra trhu v oblasti zvyšování efektivity při řízení týmů, firem i velkých společností. V rámci svého vývoje prošel systém 1CLICK dvěma milníky, a to vydáním verze 2 v roce 2013 a verze 3 v roce 2016. Oba vývojové stupně znamenaly velké změny v designu klientské části systému. V současné době je systém 1CLICK vyspělý nástroj s dynamickým vývojovým růstem.

2.3 Moduly

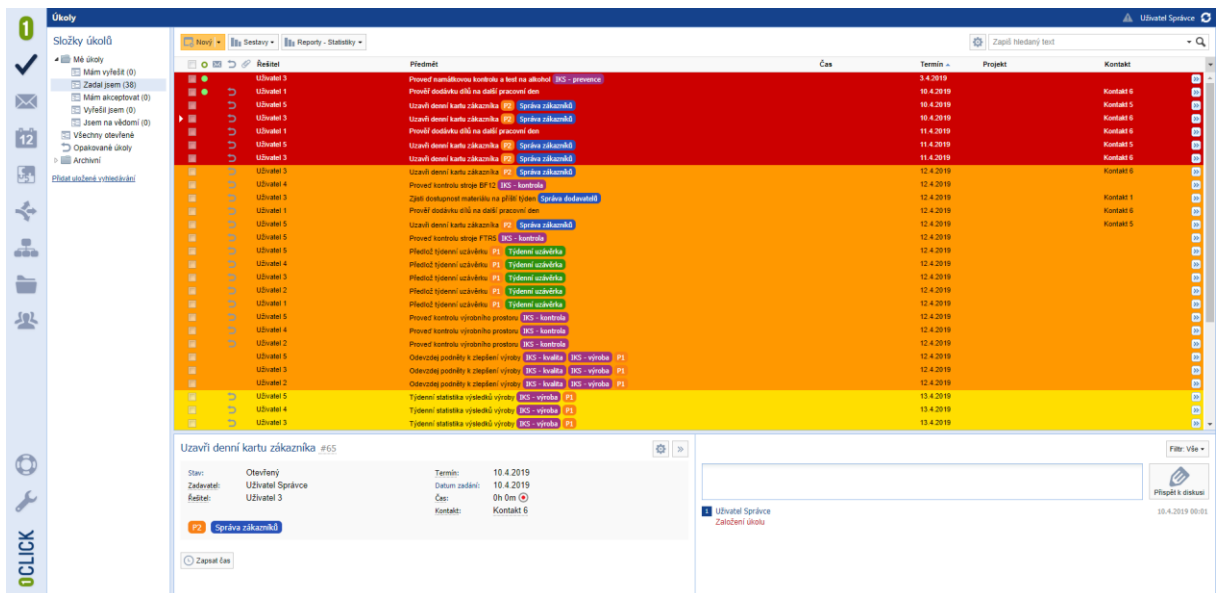
Systém 1CLICK je na trh dodáván ve třech základních verzích lišících se počtem modulů. Verze jsou pečlivě členěny dle rozsahu s použitím pro jednotlivce, malé či střední firmy a velké společnosti. Dále jsou segmentovány dle přístupu k vykonávaným činnostem a způsobu řízení, zejména projektového či procesního nebo jejich kombinací. Existující verze systému 1CLICK jsou:

- 1CLICK MINI – obsahuje moduly Úkoly, Kalendář a E-mailly.
- 1CLICK OPTI – obsahuje moduly Úkoly, Kalendář, E-mailly, Projekty, Dokumenty, Organizační struktura a Kontakty.
- 1CLICK MAXI – obsahuje moduly Úkoly, Kalendář, E-mailly, Projekty, Dokumenty, Organizační struktura, Kontakty a Procesy.

Společnost One Click Business Solution s. r. o. dále nabízí nadstavbové moduly jako je správa produktů, ServiceDesk, personalistika, propojení s účetním systémem Pohoda a další. Nedílnou součástí nabídky je i tvorba individuálních modulů, doplňků a procesů dle požadavků jednotlivých zákazníků.

2.3.1 Úkoly

Úkoly jsou klíčovým modulem celého systému, jelikož zde dochází k interakci mezi uživateli nebo systémem a uživatelem. Úkoly zajišťují aktivní článek řešení projektů, slouží jako nástroj pro tvorbu procesních „workflow“ a využívají se k efektivní práci s e-maily v podobě úkolů založených přímo z e-mailového klienta. Každý úkol obsahuje předmět, zadavatele, řešitele, termín, možnost zadání stráveného času či přidání přílohy. Nedílnou součástí je i diskuze, která slouží jak jako nástroj komunikace nad danou problematikou, tak i k záznamu historie změn daného úkolu. Úkoly jsou intuitivně podbarvené ve vztahu k blížícímu se termínu tak, aby byl uživatel na první pohled o blížícím se termínu informován. U každého úkolu je implementována tzv. zelená tečka indikující změnu na úkolu od posledního zobrazení.



Obrázek 1 – Modul úkoly v systému 1CLICK

K úkolům může být přiřazen také kontakt z modulu Kontakty pro zajištění záznamu do CRM. Projektové úkoly jsou opatřeny informací o projektu, ke kterému jsou přiřazeny. Pro podporu orientace nad úkoly je možno využít barevné štítky (značky), viz Obrázek 1, které lze využít i v ostatních vybraných modulech systému 1CLICK.

2.3.2 E-maily

Modul E-maily je plnohodnotný e-mailový klient, umožňující napojení e-mailových schránek poštovních serverů prostřednictvím protokolů IMAP a SMTP. Díky tomuto modulu je zajištěn vstup dat do systému 1CLICK, a to zejména do modulu Úkoly, modulu Kontakty, stejně tak do modulu Projekty. Dále je možno z příchozích e-mailů startovat přednastavené procesy, kde dochází k využití dat obsažených v e-mailech (odesílatel, předmět, zpráva atd.) a především příloh daných e-mailů, např. spuštěním schvalovacího „workflow“ při doručení e-mailu obsahujícího dokument s přijatou fakturou. Do schránky evidované v nastavení uživatelského účtu jsou zasílány e-mailové notifikace o změnách v systému 1CLICK (zadání nového úkolu, změna události atd.). Velkou přidanou hodnotou je přímá vizuální informovanost uživatele o aktivitách provedených nad daným e-mailem směrem do systému 1CLICK.

2.3.3 Kalendář

Každý uživatel systému 1CLICK má k dispozici svůj personifikovaný kalendář. V rámci nastavení svého kalendáře může povolit přístup jiným uživatelům. Ti si mohou kalendář zobrazit v rámci svého zobrazení kalendářů. Modul Kalendář dále umožňuje definici nepersonifikovaných kalendářů, které jsou určeny vybraným uživatelům jako zdroj pro přehled a plánování nad konkrétními oblastmi, jako jsou mj. sdružování informací o dovolených, rezervace zasedacích místností či služebních vozů. Nedílnou součástí modulu je podpora obousměrné synchronizace pomocí protokolu CalDAV, která umožňuje číst, vytvářet a upravovat události např. z mobilních zařízení.

2.3.4 Projekty

Tento modul je určen k podpoře projektového řízení, tedy jednoho z nejdůležitějších principů v oblasti výkonu zpracování problematik ve firmách. Modul umožňuje zejména definování role projektového manažera a celý projektový tým. Konkrétní projekt je v systému 1CLICK tvořen jako hierarchická struktura etap, které jsou reprezentovány formou úkolů. Etapy se dále rozkládají na tzv. podúkoly, které odpovídají jednotlivým úrovním projektu. Modul umožňuje tvořit libovolnou hloubku rozpadu projektu dle požadované agregační úrovně. Nedílnou součástí modulu je možnost přiložení libovolného počtu dokumentů, souhrnná diskuze nad projektem, volba časového fondu jako času alokovaného na realizaci projektu nebo jeho části s možností sledování postupného vývoje součtu časů strávených nad řešením jednotlivých projektových úkolů. Pro vizualizaci je do systému formou doplňku instalován Ganttův diagram. Dalším doplňkem, který firma One Click Business Solution s. r. o. nabízí, je kalkulace

nákladovosti na projekt, tzn. že sledované množství času je doplněno o finanční hledisko s možností tvorby finančního plánu a jeho porovnání s realitou v každém okamžiku průběhu projektu.

2.3.5 Procesy

Modul Procesy přináší možnost automatizovaného řízení rutinních činností ve firmě. Jedná se například o:

- vyřízení přijatých a vydaných faktur,
- schválení dovolené,
- schválení pracovní cesty,
- a mnohé další.



Obrázek 2 – Schéma procesu "Faktura přijatá" [10]

Primární náplní procesů v systému 1CLICK je automatické zakládání procesních úkolů, které se jednotlivým uživatelům zařazují mezi ostatní úkoly. Procesy též umí automaticky zakládat projekty, startovat a plánovat návazné procesy, archivovat dokumenty, odesílat informační zprávy, vkládat události do kalendáře, nebo vyměňovat data s různými informačními systémy apod.

Každý proces má své vlastní schéma, které definuje, co se má, při každém kroku (procesní fázi) provést. Struktura schémat je podobná formátu JSON. Daná schémata volají procesní funkce definované v systému 1CLICK, popř. přidané dalšími rozšiřujícími doplňky. Tím je umožněno vytváření individuálních procesů pro jednotlivé firmy bez úprav a zásahů do jádra systému.

2.3.6 Organizační struktura

Organizační struktura slouží jako nástroj pro vizualizaci hierarchického členění firmy skládající se z organizačních jednotek a pracovních postů, do nichž jsou zaměstnanci zařazeni. Každý z postů obsahuje podklady pro vykonávání práce (pracovní náplň, produkty, činnosti, dokumenty a jiné). Uživatelé jsou o změnách v postech informováni e-mailovou notifikací či úkolem. Organizační struktura v praxi slouží společností pro podporu řízení standardů ISO.

2.3.7 Dokumenty (DMS)

Systém 1CLICK má implementován vlastní DMS a tím je modul Dokumenty. Slouží pro archivaci a sdílení dokumentů v rámci celého systému. Tím je zajištěna centralizace všech dokumentů a jejich bezztrátovost vůči hmotné papírové podobě. Vložený dokument nelze ze systému smazat, pouze skrýt, tudíž není možné ho nedopatřením odstranit. Modul Dokumenty umožňuje tvorbu stromové adresářové struktury, kde jsou jednotlivé adresáře reprezentovány jako tzv. typy dokumentů. Jednotlivé typy dokumentů jsou modifikovatelné v podobě nastavení popisných vlastností, automatického přejmenování při vložení dokumentu či nastavením práv pro přístup, čtení a editaci dokumentů. Nespornou výhodou je možnost verzování dokumentů se záznamem o všech předchozích verzích a zobrazením celé historie.

2.3.8 Kontakty (CRM)

Jednou z nejdůležitějších součástí podnikových informačních systémů je CRM. V systému 1CLICK je reprezentováno modulem Kontakty. Modul umožňuje k danému kontaktu připojovat úkoly, projekty, procesy, dokumenty, e-maily a mnoho dalšího. Tím je zajištěn sběr dat, informace o historii a aktivitách uživatelů spojených s konkrétním kontaktem. U každého kontaktu je dále možné vést uživatelsky definovatelné záznamy, např. kontaktní osoba, adresa, bankovní účet a jakékoliv jiné, které jsou definovány v systémovém nastavení.

2.3.9 Doplnkové moduly

Pro systém 1CLICK dále existují doplnkové moduly, jež jsou vyvinuté pro řešení konkrétní problematiky firem a společností a využívají kooperace s ostatními kmenovými moduly systému. Doplnkové moduly jsou nabízeny k individuálnímu dokoupení. Mezi doplnkové moduly patří moduly popsané v následujících pododdílech.

2.3.9.1 Personalistika (HR)

Modul HR je koncipován jako volně definovatelný modul pro personální oblast. Modul nemá pevně definované položky, ale umožňuje zákazníkům tvorbu vlastní konfigurace jednotlivých záznamů. Je tedy možné sestavovat strukturu záznamů přesně pro potřeby firem či společností. Každá položka modulu HR má svou dokumentovou část pro možnost přiložení jakéhokoliv dokumentu. Dokumenty jsou přímo vázány na modul Dokumenty. Jednotlivé položky jsou opatřeny historií změn na datech ve složkách. Nejvýznamnější předností je provázání s modulem Procesy. Nad jednotlivými záznamy lze spouštět a sledovat různé procesy souvisejícími s personální oblastí např. schvalování dovolených, služební cesty, schvalování tzv. „sickdays“ nebo lékařské prohlídky. Na modul je možné implementovat proces pro hlídání datových záznamů s automatickým založením předmětného procesu.

2.3.9.2 Produkty

Modul Produkty je založen na stejné filosofii a přístupu jako modul HR. Rozdíl spočívá v pokrývání odlišné oblasti v dané společnosti. K tomu jsou přizpůsobeny položky nabízené pro volnou konfiguraci popisných vlastností produktů. Prvotním záměrem tvorby tohoto modulu bylo pokrytí části trhu v oblasti provádění servisních prací v terénu specializovanými pracovníky. Praxe však ukázala uplatnění i v oblastech PDM nebo např. při správě vozových parků či majetku společností. V kombinaci s dalším doplňkovým modulem *ServiceDesk* (modul pro výkon činnosti jednoho z klíčových procesů společnosti), doplňkem *TemplateForms* (doplňek pro vytváření interaktivních webových formulářů a jejich následného převodu do PDF) a individuálním procesem pro servisní zásahy tvoří ucelený balíček pro vykonávání servisních činností.

2.3.9.3 Plánovací kalendář

Modul Plánovací kalendář je navržen pro potřeby širokého segmentu trhu od stavebnictví po výrobu. Je koncipován jako volně definovatelný modul pro krátkodobé, střednědobé i dlouhodobé plánování pracovního výkonu jednotlivců, i skupin pracovníků. Je vázán na jednotlivé uživatele 1CLICK. Modul je opatřen technologií drag-and-drop pro efektivní práci při plánování činnosti pracovníků. Nedílnou součástí modulu je propracovaná reportní část s možností zasílání plánů zainteresovaným uživatelům, formou ucelených přehledových tabulek či seznamů výkonu práce na nadcházející období. Ty jsou předány zaměstnancům formou e-mailu zasílaného v určený čas.

2.4 Konkurence – jiné systémy

System 1CLICK je založen na velmi specifické unikátní obchodní filosofii. Snaží se pokrýt co největší záběr činností v rámci jednotného uceleného řešení za co nejpříznivější cenu. Z tohoto důvodu je velmi komplikované porovnání s konkurencí, neboť v nabízeném rozsahu neexistuje na trhu ekvivalentní produkt a pokud ano, jedná se o mnohem propracovanější systémy s diametrálně vyšší cenou a odlišnou obchodní politikou. Pokud bychom se podívali na konkurenci z pohledu jednotlivých řešení (modulů) můžeme zmínit zahraniční systém Trello nebo český systém Freeloo pokrývající oblast úkolového managementu. Oba tyto systémy však nemají přesah do ostatních oblastí řízení společnosti jako systém 1CLICK. V oblasti CRM je největším konkurentem produkt společnosti Raynet. Pokud bychom se zaměřili na projektové řízení, je nejbližší řešení EasyProject, který však svou funkcionalitou v této oblasti systém 1CLICK plně převyšuje. Pevný základ systému 1CLICK tvoří dobře propracovaná spolupráce jednotlivých modulů s velkou přidanou hodnotou pro uživatele. Pro vylepšení pozice na trhu je nezbytné postupné doplňování funkcionalit jednotlivých modulů s ohledem na konkurenci daného segmentu.

2.5 Důvod tvorby modulu Reporty – Statistiky

System 1CLICK má v současné době plně zvládnutou problematiku pořizování dat a jejich ukládání a třídění v rámci databází. Velkým hendikepem je však oblast zpracování a vizualizace reportních a statistických výstupů potřebných zejména v oblasti kontroly a manažerského řízení na vyšších úrovních společností. Modul Reporty – Statistiky je koncipován k plnohodnotnému využití a zahrnuje všechny důležité statistické funkce jak pro přímé operativní využití, tak i pro tvorbu dlouhodobých reportů a statistik. Modul zapadá do koncepce vývoje 1CLICK a zajistí udržení dynamického tempa vývoje systému v oblasti podpory zvyšování efektivity práce středních a velkých společností, stejně tak i menších týmů či jednotlivců.

3 STATISTICKÉ ZPRACOVÁNÍ DAT

Tato kapitola se věnuje statistickému zpracování dat. V úvodu popisuje obecné termíny, tj. popisnou statistiku, statistický soubor, statistický znak apod. Dále se věnuje statistickým charakteristikám, tvorbě polygonu četností a histogramu, v poslední části časovým řadám. Kapitola čerpá z [11][12].

3.1 Popisná statistika

Statistika je věda zabývající se získáváním, zpracováním a analýzou dat pro potřeby rozhodování. Statistika se dá v praktickém užití rozdělit na dvě roviny:

- Popisná statistika – uspořádání naměřených dat, výpočet základních číselných charakteristik a zobrazení dat.
- Induktivní statistika – souhrn metod sloužících k odhadům a induktivním úvahám s využitím pravděpodobnosti.

3.1.1 Základní statistické pojmy

3.1.1.1 Statistický soubor a statistická jednotka

Statistický soubor je vymezená množina zkoumaných objektů, které mají z daného hlediska společné vlastnosti (soubor podniků, soubor obyvatelstva, soubor událostí apod.). Soubory, které jsou předmětem zkoumání, označujeme jako *základní soubory*. V praxi se častokrát z rozdílných důvodů nepracuje s celým rozsahem statistického souboru, ale jen se vybranou podmnožinou statistických jednotek neboli s *výběrovým souborem*.

Statistická jednotka je prvek statistického souboru a nositelem vlastností daného souboru.

3.1.1.2 Statistický znak

Statistický znak je měřitelná vlastnost jednotky, která je předmětem zkoumaného zájmu nebo na základě které byl definován základní soubor (rychlost auta, počet zákazníků, pohlaví, známka atd.). V případě, že ve statistickém souboru pracujeme jen s jedním znakem, jedná se o *jednorozměrný soubor*, pokud existuje současně více znaků, jde o dvou-, tří-, resp. obecně *vícerozměrný soubor*. Základním tříděním statistických znaků je rozlišování znaků:

- číselných – kvantitativních a numerických,
- slovních – kvalitativních, alfabetských a kategoriálních.

3.1.2 Charakteristiky polohy

Charakteristiky polohy (úrovně) měří obecnou velikost hodnot znaku ve statistickém souboru. Dělí se na průměry (počítané ze všech dat) a ostatní míry polohy (počítané pouze z vybraných hodnot).

3.1.2.1 Kvantily

Kvantil je hodnota statistického znaku určená tak, že rozděluje seřazený statistický soubor na dvě skupiny hodnot, které jsou menší nebo rovny, resp. větší než tato hodnota. V praxi se používají především tyto kategorie kvantilů:

- Kvartily ($\tilde{x}_{25}, \tilde{x}_{50}, \tilde{x}_{75}$) – rozdělují uspořádaný statistický soubor na čtvrtiny. Dolní kvartil (\tilde{x}_{25}) odděluje 25 % jednotek s nejnižšími hodnotami, prostřední kvartil (\tilde{x}_{50}), též označován jako *medián* (viz Pododdíl 3.1.2.3), rozděluje soubor na 50 % prvků s nízkými hodnotami a 50 % prvků s vysokými hodnotami. Poslední horní kvartil (\tilde{x}_{75}) odděluje 75 % jednotek s nízkými hodnotami od 25 % jednotek s vysokými hodnotami.
- Decily ($\tilde{x}_{10}, \tilde{x}_{20}, \dots, \tilde{x}_{90}$) – rozdělují uspořádaný statistický soubor na 10 stejných částí.
- Percentily ($\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{99}$) – někdy označované jako centily, dělí uspořádaný statistický soubor na setiny.

3.1.2.2 Aritmetický průměr

Je nejnámějším a nejužívanějším druhem průměru a je aplikovatelný všude tam, kde má nějaký informační význam součet hodnot. Pokud by větší význam dával součin hodnot, můžeme využít tzv. geometrický průměr. Aritmetický průměr se nejčastěji značí symbolem \bar{x} . Ze statistického souboru o n hodnotách, označených x_1, x_2, \dots, x_n , získáme aritmetický průměr tím, že sečteme všechna x_i a výsledek vydělíme počtem hodnot:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad \text{nebo} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Tuto podobu aritmetického průměru nazýváme *prostý aritmetický průměr*, jelikož výpočet nepředpokládá žádné předběžné uspořádání hodnot.

Velkou nevýhodou aritmetického průměru je fakt, že je ovlivňován extrémními hodnotami ve statistickém souboru a díky tomu může být výsledek zavádějící. Např. aritmetický průměr ze souboru dat {1, 1, 1, 2, 3, 3, 5, 18} je roven 4,25 přestože šest z osmi hodnot je menších než

tento aritmetický průměr. V podobných situacích je tedy mnohem vhodnější použít pro výpočet *medián*.

3.1.2.3 Medián

Jak bylo řečeno výše, jedná se o padesátiprocentní kvantil označovaný \tilde{x}_{50} resp. pouze symbolem \tilde{x} . Odděluje tedy polovinu menších hodnot od poloviny hodnot větších. To znamená, že medián je roven prostřednímu prvku v uspořádaném statistickém souboru. Pokud statistický soubor obsahuje sudý počet prvků, tzn. neexistuje pouze jedna prostřední hodnota, určí se medián jako aritmetický průměr prostředního prvku x_n a následujícího prvku x_{n+1} .

Na rozdíl od aritmetického průměru není náchylný na extrémní hodnoty, protože závisí pouze na jedné, nejvýše dvou prostředních hodnotách statistického souboru. Pro porovnání s předchozí podkapitolou o aritmetickém průměru, se medián ze souboru dat $\{1, 1, 1, 2, 3, 3, 5, 18\}$ rovná 2,5 což je mnohem více vypovídající o daném souboru než předchozí zkruslená hodnota aritmetického průměru, tzn. že 50 % prvků je menších než 2,5 a zároveň 50 % je větších než tato hodnota.

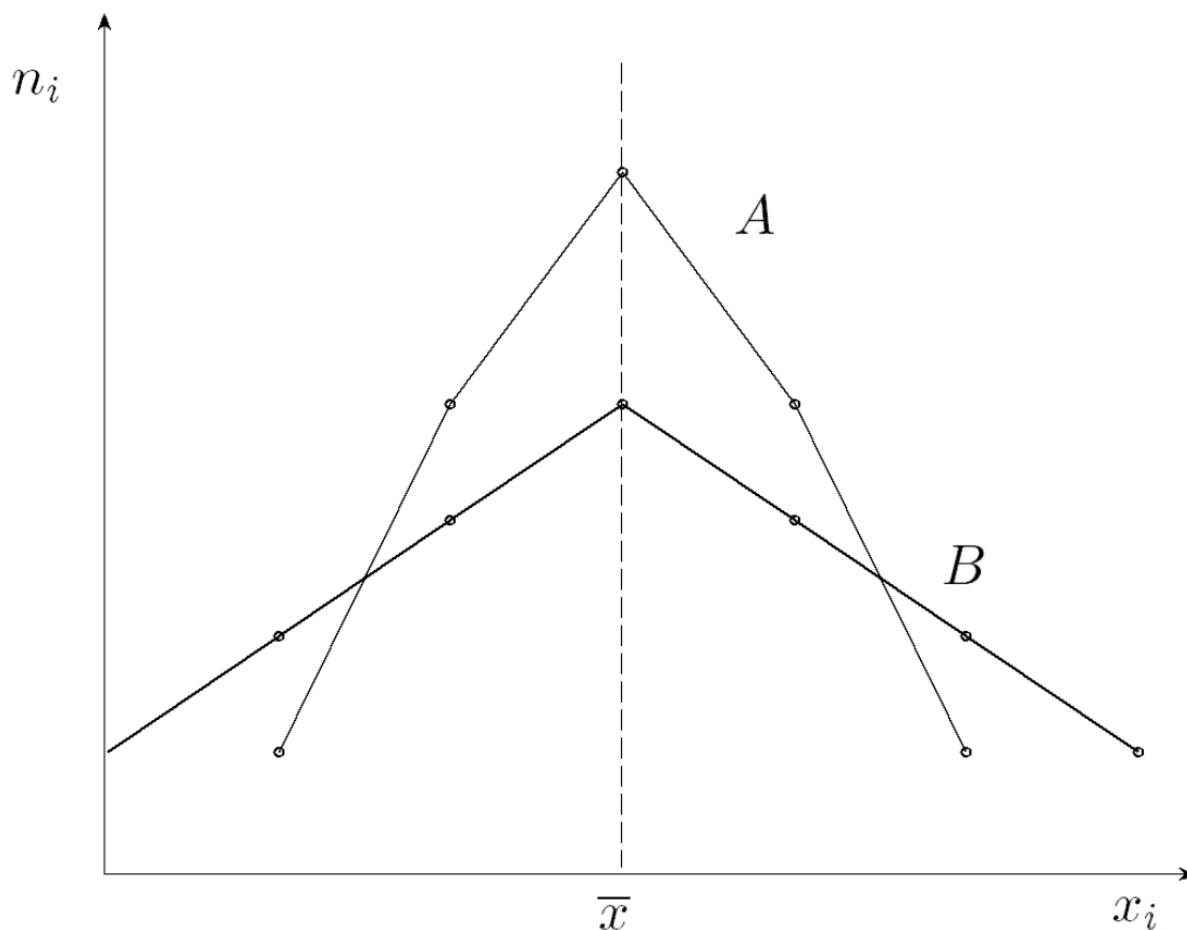
3.1.2.4 Modus

Další statistickou charakteristikou, která není náchylná na extrémní hodnoty, je modus označovaný symbolem \hat{x} . Jedná se o nejtypičtější, resp. nejčetnější hodnotu znaku ve statistickém souboru dat.

Pro názorný příklad vezměme statistický soubor z minulých podkapitol $\{1, 1, 1, 2, 3, 3, 5, 18\}$. Již na první pohled je zřejmé, že modus se v tomto případě bude rovnat 1, neboť četnost této hodnoty je v rámci celého statistického souboru nejvyšší, tj. 3.

3.1.3 Charakteristiky variability

Variabilitou statistického znaku rozumíme kolísání hodnot této veličiny. Pokud statistický soubor obsahuje všechny hodnoty stejné, jedná se o nulovou variabilitu. Dané kolísání hodnot ve statistickém souboru můžeme posuzovat buď jako vzájemnou rozdílnost jednotlivých hodnot sledované veličiny, nebo jako rozdílnost jednotlivých hodnot od aritmetického průměru. Při zpracování dat je možné setkat se s případem, kdy jednotlivá rozdělení četností budou mít shodnou polohu, ale přesto se od sebe budou výrazně lišit (viz Obrázek 3). Hodnoty statistického souboru *A* jsou více koncentrovány okolo aritmetického průměru \tilde{x} než hodnoty souboru *B*. Lze tedy říct, že aritmetický průměr ve statistickém souboru *A* lépe vystihuje míru polohy a má menší variabilitu než soubor druhý.



Obrázek 3 – Rozdělení četností s odlišnou variabilitou [13]

3.1.3.1 Variační rozpětí

Je rychlou, jednoduchou, ale i hrubou charakteristikou variability založenou na informaci o maximální a minimální hodnotě ve statistickém souboru. Udává délku intervalu, v němž se nachází všechny hodnoty prvků daného statistického souboru:

$$R = x_{max} - x_{min}$$

Velkou nevýhodou této míry variability je, že je závislá pouze na extrémních hodnotách, které nemusí být pro sledovaný znak typické.

3.1.3.2 Rozptyl

Patří k nepoužívanějším a nejznámějším mírám variability. Je definován jako aritmetický průměr z čtverců odchylek jednotlivých hodnot znaku od aritmetického průměru:

$$s_n^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Rozptyl má tyto základní vlastnosti:

- Jsou-li jednotlivé hodnoty znaku x_i konstantní, potom je jejich rozptyl roven nule.

$$\frac{1}{n} \sum_{i=1}^n (c - c)^2 = 0$$

- Přičteme-li k jednotlivým hodnotám znaku x_i konstantu c , hodnota rozptylu se nezmění.

$$\frac{1}{n} \sum_{i=1}^n [(x_i + c) - (\bar{x} + c)]^2 = s_n^2$$

- Násobíme-li jednotlivé hodnoty znaku x_i konstantou c , je rozptyl násoben čtvercem této konstanty.

$$\frac{1}{n} \sum_{i=1}^n (x_i * c - \bar{x} * c)^2 = c^2 * s_n^2$$

3.1.3.3 Směrodatná odchylka

Směrodatná odchylka je odmocnina z rozptylu, tzn. jedná se o kvadratický průměr odchylek hodnot znaku od jejich aritmetického průměru. Na rozdíl od rozptylu je vyjádřena ve stejných jednotkách jako sledovaný znak, jelikož rozptyl je dán jednotkami čtverečními.

$$s_n = \sqrt{s_n^2}$$

Směrodatná odchylka je ve statistice často používanou mírou statistické proměnlivosti.

3.2 Rozdělení četností a histogram

Četnost udává počet hodnot daného znaku vyskytujících se v daném statistickém souboru. Rozdělení četností se dělí na *bodové* (nespojité statistický znak a malý počet obměn, grafické zobrazení: tabulka a polygon četností) a *intervalové* (spojitý statistický znak nebo nespojitý znak s velký počtem obměn, grafické zobrazení: tabulka, histogram). Bodové rozdělení četností lze při uspořádaném datovém souboru o rozsahu n prvků rozčlenit na:

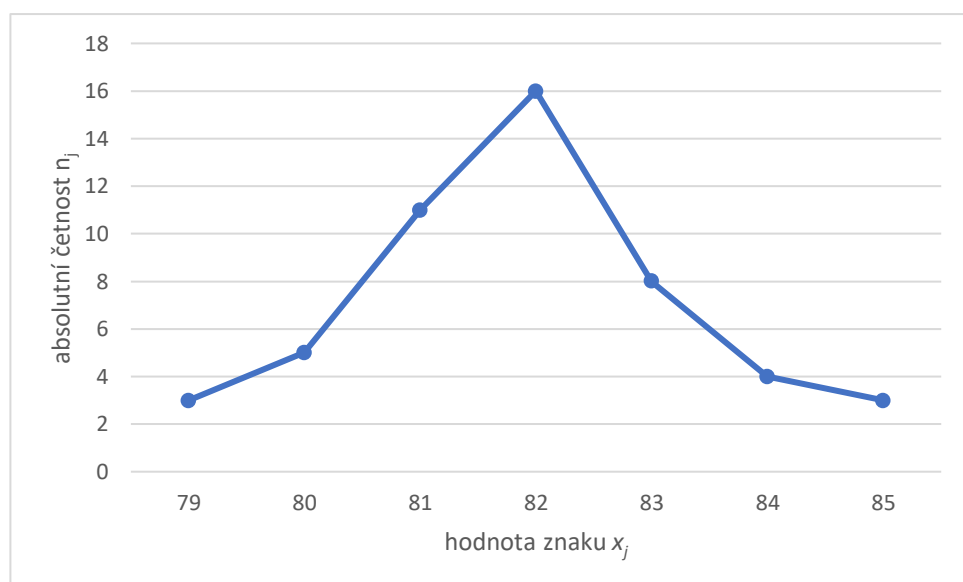
- Absolutní četnost – je dána vztahem $\sum_{j=1}^k n_j = n$, kde n_j představuje počet výskytů varianty x_j v souboru a k je počet variant.
- Relativní četnost – je dána vztahem $\sum_{j=1}^k p_j = 1$, kde $p_j = \frac{n_j}{n}$ představuje podíl výskytů varianty x_j v souboru a k je počet variant.

- Absolutní kumulativní četnost – N_j je dáno vztahem $N_j = n_1 + \dots + n_j$ a udává součet četností všech pozorování, která nepřekračují hodnotu x_j .
- Relativní kumulativní četnost – F_j je dáno vztahem $F_j = \frac{N_j}{n} = p_1 + \dots + p_j$ a udává podíl četností všech pozorování, která nepřekračují hodnotu x_j .

Příklad sestavení tabulky bodového rozdělení četností a polygonu četností ze statistického souboru obsahující data: {83, 85, 81, 82, 84, 82, 79, 84, 80, 81, 82, 82, 80, 82, 80, 82, 83, 84, 82, 79, 83, 82, 83, 82, 82, 82, 81, 80, 82, 82, 83, 80, 82, 85, 81, 83, 81, 81, 83, 82, 81, 85, 83, 79, 81, 81, 81, 84, 81, 82} viz Tabulka 1 a Obrázek 4

Tabulka 1 – Tabulka bodového rozdělení četností

Hodnota znaku x_j	Absolutní četnost n_j	Relativní četnost p_j	Abs. kumulativní četnost N_j	Rel. kumulativní četnost F_j
79	3	0,06	3	0,06
80	5	0,10	8	0,16
81	11	0,22	19	0,38
82	16	0,32	35	0,70
83	8	0,16	43	0,86
84	4	0,08	47	0,94
85	3	0,06	50	1
Σ	50	1	-	-



Obrázek 4 – Polygon četností

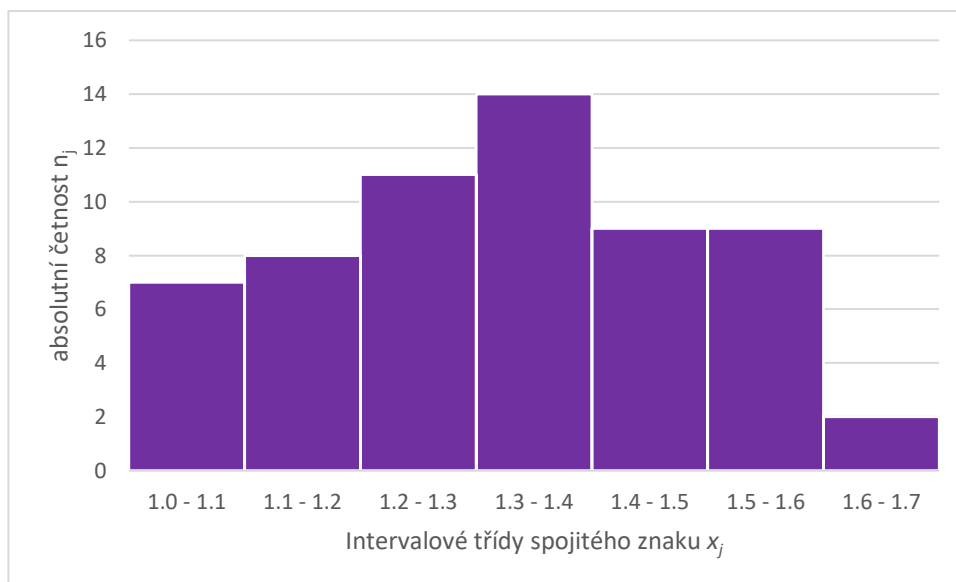
Pro konstrukci histogramu a tabulky intervalového rozdělení četností se musí spojitá data ze statistického souboru nejdřív rozdělit do tzv. tříd. Postup konstrukce tříd (intervalů):

1. Určení variačního rozpětí $R = x_{max} - x_{min}$.
2. Stanovení počtu tříd s využitím pravidel:
 - a. Sturgesovo pravidlo $k \doteq 1 + 3,3 \log n$,
 - b. Yuleovo pravidlo $k \doteq 2,5 \sqrt[4]{n}$,
 - c. další pravidla $k \doteq \sqrt{n}$ nebo $k \doteq 5 \log n$.
3. Stanovení šířky tříd $h = \frac{R}{k}$.
4. Intervaly budou zpravidla polouzavřené zprava.
5. Hranice i středy tříd by měly být vhodně zaokrouhlené.

Příklad sestavení tabulky intervalového rozdělení četností a histogramu ze statistického souboru obsahující data: {1,23; 1,10; 1,54; 1,34; 1,06; 1,09; 1,41; 1,48; 1,52; 1,37; 1,37; 1,63; 1,51; 1,53; 1,31; 1,23; 1,31; 1,27; 1,17; 1,27; 1,34; 1,27; 1,09; 1,01; 1,41; 1,22; 1,27; 1,37; 1,14; 1,22; 1,43; 1,40; 1,41; 1,51; 1,51; 1,47; 1,14; 1,34; 1,16; 1,51; 1,58; 1,33; 1,31; 1,04; 1,58; 1,12; 1,19; 1,17; 1,47; 1,24; 1,45; 1,29; 1,17; 1,63; 1,39; 1,02; 1,38; 1,39; 1,43; 1,28} viz Tabulka 2 a Obrázek 5. Celkový rozsah statistického souboru je tedy roven $n = 60$, nejmenší hodnota $x_{min} = 1,01$ a největší hodnota $x_{max} = 1,63$. Variační rozpětí $R = 1,63 - 1,01 = 0,62$ a pomocí Sturgesova pravidla lze vypočítat počet potřebných tříd, což je po zaokrouhlení výsledku 7 tříd. Šířka intervalu bude $h = \frac{R}{k} = \frac{0,62}{7} \doteq 0,1$.

Tabulka 2 – Tabulka intervalového rozdělení četností

Interval	Střed intervalu x_j	Absolutní četnost n_j	Relativní četnost p_j	Abs. kumulativní četnost N_j	Rel. kumulativní četnost F_j
(1,0;1,1)	1,05	7	0,117	7	0,117
(1,1;1,2)	1,15	8	0,133	15	0,250
(1,2;1,3)	1,25	11	0,183	26	0,433
(1,3;1,4)	1,35	14	0,233	40	0,667
(1,4;1,5)	1,45	9	0,150	49	0,817
(1,5;1,6)	1,55	9	0,150	58	0,967
(1,6;1,7)	1,65	2	0,033	60	1
Σ	-	60	1	-	-



Obrázek 5 – Histogram

3.3 Časové řady

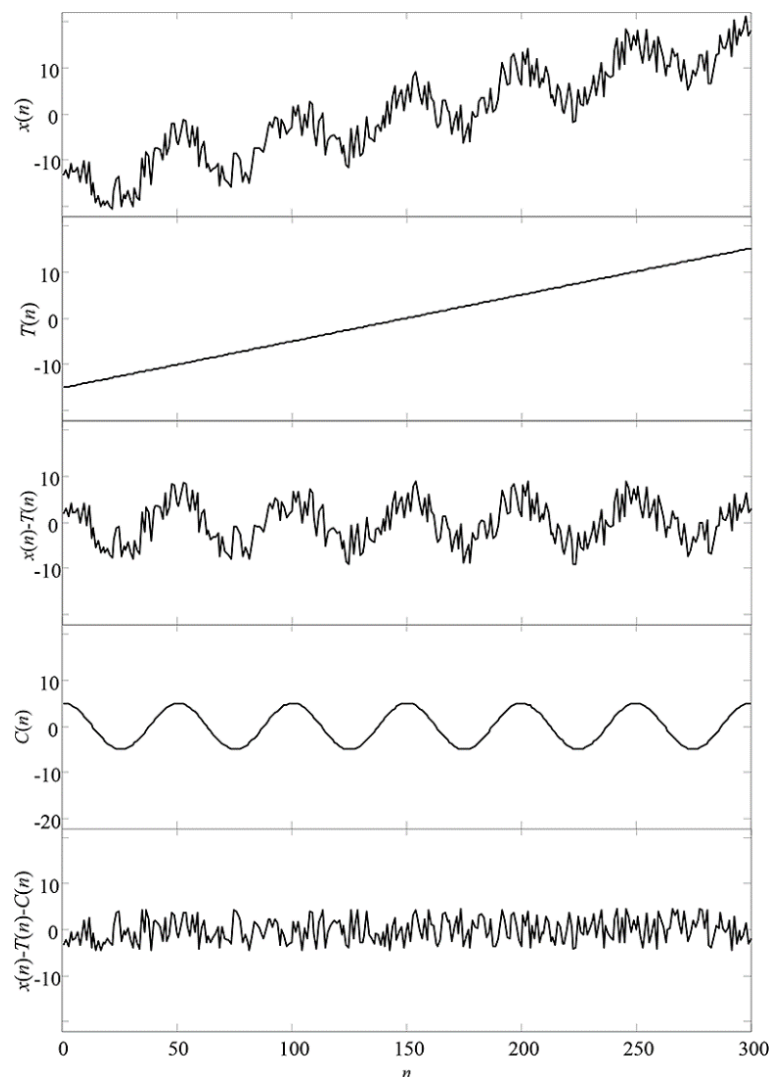
Časová řada je chronologická posloupnost reálných hodnot určitého statistického ukazatele měřeného v určitých časových intervalech. Existuje mnoho způsobů dělení časových řad:

- dlouhodobé / krátkodobé,
- okamžikové / intervalové,
- deterministické / stochastické,
- ekvidistantní / neekvidistantní (konstantní, resp. různá časová vzdálenost mezi jednotlivými hodnotami).

Hlavním cílem analýzy časových řad je porozumění mechanismu generujícího hodnoty sledované veličiny a predikovat její následující vývoj. Volba použité metody analýzy závisí na účelu a cíli analýzy, typu časové řady, dostupném softwaru, teoretickém východisku apod. Prakticky se používají tyto metody:

- Expertní metody – jedná se o kvalitativní metody s uplatněním tam, kde není možné využívat metod kvantitativních.
- Grafická analýza – jednoduchá metoda analýzy časové řady, která je založena na grafickém zobrazení vývoje sledované veličiny. Má spíše subjektivní charakter, ale prospěšné bývá například srovnání grafů různých časových řad mezi sebou.
- Ekonometrické modely – cílem je odhalit příčinné vazby mezi ekonomickými veličinami.

- Spektrální analýza – vychází z předpokladu, že časová řada je směs sinusových a kosinusových křivek s různými frekvencemi a amplitudami, a snaží se odhalit intenzitu zastoupení jednotlivých frekvencí.
- Dekompozice časových řad – vychází z předpokladu, že hodnota sledované veličiny závisí pouze na čase, tzn. lze časovou řadu rozložit na několik nezávislých složek: trend, sezónní, cyklickou a náhodnou složku $Y_t = T_t + S_t + C_t + \varepsilon_t$.
 - Trendová složka – ukazatel rostoucího, resp. klesajícího vývoje zkoumaného jevu.
 - Sezónní složka – pravidelně se opakující odchylka od trendové složky, perioda je menší než celková velikost sledovaného období.
 - Cyklická složka – kolísání okolo trendu v důsledku dlouhodobého (delší než jeden rok) cyklického vývoje.
 - Náhodná složka – zbývá po vyloučení trendu, sezónní a cyklické složky.



Obrázek 6 – Dekompozice časové řady $x(n)$ na trendovou složku $T(n)$, cyklickou složku $C(n)$ a náhodnou složku $\varepsilon(n)$ [14]

4 TECHNOLOGIE POUŽITÉ PRO VÝVOJ

4.1 Java 8

Java je objektově orientovaný interpretovaný programovací jazyk, navržený speciálně tak, aby měl co možná nejméně implementačních závislostí. Zkompilovaný kód může běžet na všech platformách, které podporují tento jazyk bez nutnosti jeho rekompilace. Syntaxe je velmi podobná jazyku C nebo C++. Správa paměti je realizována pomocí automatického algoritmu tzv. garbage collector, který vyhledává již nepoužívané části paměti a uvolňuje je pro další použití. Tato podkapitola vychází z [15][16].

Java byla původně vyvinuta Jamesem Goslingem a vydána v roce 1996 jako základní součást platformy Java společnosti Sun Microsystems, která byla později odkoupena společností Oracle. Implementace Java kompilátorů, virtuálního stroje a knihovny tříd byly původně vydány společností Sun pod soukromými licencemi. Od května 2007 v souladu se specifikacemi Java Community Process přelicencovala společnost Sun většinu svých technologií pod licenci GNU General Public License.

V roce 2018 byla Java společností GitHub vyhlášena jako jeden z nejpopulárnějších programovacích jazyků používaných zejména pro webové aplikace typu klient-server s přibližně nahlášenými 9 000 000 vývojáři.

Během 18 let od vydání první verze jazyka Java bylo vydáno dalších 8 verzí: Java 1.1 (1997), Java 1.2 (1998), Java 1.3 (2000), Java 1.4 (2002), Java 5 (2004), Java 6 (2006), Java 7 (2011) a Java 8 (2014). Osmá verze přinesla velké změny a novinky:

- Lambda výrazy jsou anonymní funkce, které usnadňují a zjednodušují programování. Není nutnost definovat datový typ vstupního parametru. Není potřeba psát kulaté závorky okolo jednoho parametru, ale pro více parametrů jsou již požadovány. V těle výrazu není nutné používat ani složené závorky, pokud obsahuje pouze jeden příkaz. Též zde není nutnost psát klíčové slovo *return* pro navrácení hodnoty, pokud tělo výrazu obsahuje opět jen jeden příkaz. Příklad lambda výrazu sčítající dva vstupní parametry:
 $(x, y) \rightarrow x + y$.
- Reference metod je přidána možnost odkazování na metody podle jejich názvů místo jejich přímého volání. Reference na metodu může být použita pro statické metody,

instanční metody a konstruktory používající operátor *new*. Odkaz na metodu je definován pomocí operátoru „:“: Ukázka textového výpisu prvků kolekce na konzoli:

```
list.forEach(System.out::println)
```

- Výchozí (default) metoda. Dříve mohly mít rozhraní pouze abstraktní metody. Samotná implementace abstraktních metod musí být poskytnuta v jiné třídě implementující toto rozhraní. K překonání tohoto problému zavedla Java 8 koncept výchozích metod, který umožňuje mít v rozhraní metody s implementací, aniž by to ovlivnilo třídy, které toto rozhraní již implementují. Zápis výchozí metody *print()*:

```
public interface example {  
    default void print() {  
        System.out.println("Hello world!");  
    }  
}
```

- Stream API. Datové proudy jsou nová abstraktní vrstva, s jejíž pomocí lze zpracovávat data deklarativním způsobem podobným SQL výrazům. Datový proud představuje posloupnost objektů podporující agregační operace. Vstupním zdrojem dat může být kolekce, pole nebo vstupně-výstupní prostředky. Návrhovým typem většiny proudových funkcí je znovu datový proud, aby mohly být jednotlivé funkce řetězeny za sebe. Konec datového proudu může třeba určovat metoda *collect()*, která transformuje objekty z datového proudu zpět například do kolekce.
- DateTime API. Nové rozhraní data a času odstraňuje nedostatky původního rozhraní. Je vláknově bezpečné, vylepšuje práci s časovými pásmy a zdokonaluje a sjednocuje design např. číslování dnů a měsíců již začíná stejným indexem.
- Třída Optional je velmi podobná třídě Optional z knihovny Guava od společnosti Google, která byla publikována mnohem dříve. Uchovává v sobě *not-null* objekty, a ostatní nahrazuje tzv. chybějící hodnotou. V praxi to znamená, že vývojář nemusí ošetřovat, zda objekt není *null*, ale stačí volat pomocné metody, které určují, zda hodnota je nebo není dostupná.
- Base64. Java 8 nyní obsahuje zabudovaný kodér a dekodér pro kódování Base64. Můžeme použít tři typy kódování:
 - Jednoduchý – výstup je mapován na sadu znaků v rozmezí *A-Za-z0-9+/-*. Kodér nepřidává na výstup odřádkování a dekodér odmítne kterýkoliv znak nepatřící do rozmezí *A-Za-z0-9+/-*.
 - URL – výstup je mapován na sadu znaků, které leží v *A-Za-z0-9+_-*. Je bezpečný pro adresu URL a název souboru.

- MIME – výstupem je známý formát MIME. Výstup je reprezentován řádky, které nemají více než 76 znaků. Pro ukončení řádku je použita kombinace znaků „\r\n“.

Do dnešní doby byly vydány ještě tři verze: Java 9 (2017), Java 10 a 11 (2018) a v roce 2019 se očekává vydání dvanácté verze.

4.1.1 Guava: Google Core Libraries for Java

Guava je sada základních knihoven společnosti Google, které obsahují nové typy kolekcí (např. multimapa nebo multiset), neměnitelné kolekce, knihovnu grafů, funkční typy, I/O, nástroje pro lepší hashování, zpracování řetězců a mnoho dalších. Hlavními autory této open source knihovny pro jazyk Java jsou Google inženýři Kevin Bourrillion and Jared Levy. První verze této knihovny byly vydány okolo roku 2010. Poslední stabilní verze je 27.1, která byla vydána dne 9. března 2019. Čerpáno z [17].

V dubnu roku 2012 byla Guava označena jako 12. nejpopulárnější Java knihovna vedle velmi úspěšných projektů jako Apache Commons a několika dalších. Výzkum prováděný společností GitHub v roce 2013 nad 10000 projekty zjistil, že knihovny naprogramované společností Google, jako jsou například Google Web Toolkit, Guava a další, obsadily 7 míst v hodnocení 100 nejpopulárnějších Java knihoven. V březnu 2018 byla Guava šestým nejoblíbenějším Java projektem na portále GitHub.

4.1.2 Apache Commons Mathematics Library 3

Je knihovna samostatných matematických a statistických komponent, které pomáhají řešit běžné problémy, které nejsou k dispozici přímo v programovacím jazyce Java nebo knihovně Commons Lang, pochází též od neziskové společnosti Apache Software Foundation. 21. března 2016 vyšla knihovna ve verzi 3.6.1. Čerpáno z [18]. Mezi hlavní zásady této knihovny patří:

- Knihovna klade důraz na malé, snadno integrované komponenty, nikoliv na velké knihovny s komplexními závislostmi a konfiguracemi.
- Všechny algoritmy jsou plně zdokumentovány a dodržují obecně uznávané a osvědčené výpočetní postupy.
- Minimální závislosti. Knihovna nemá žádné dodatečné závislosti kromě Apache Commons komponent a základní Java platformy.

Knihovna Apache Commons Mathematics Library tedy nabízí prostředky pro výpočet:

- průměru, mediánu, rozptylu a ostatních ukazatelů popisné statistiky,
- lineární regrese,
- interpolace a polynomů,
- metody nejmenších čtverců,
- lineárních rovnic a matic,
- diferenciálních rovnic,
- testů statistické významnosti,
- faktoriálů,
- generování náhodných čísel s větším omezením (např. distribuce, rozsah), než je možné pomocí základního JDK,
- a mnohé další.

4.1.3 Apache POI

Apache POI je populární rozhraní, které programátorům umožňuje vytvářet, upravovat a zobrazovat soubory Microsoft Office jako jsou Word, PowerPoint a Excel v programech napsaných v jazyce Java. Jedná se o open source knihovnu vyvinutou a distribuovanou společností Apache Software Foundation. První verze byla vydána 30. prosince 2001 a poslední stabilní verze 7. září 2018 (4.0.0). Podkapitola čerpá z [19][20]. Knihovna obsahuje třídy a metody pro dekódování vstupních uživatelských dat nebo souborů do dokumentů Microsoft Office, kterými jsou:

- POIFS (Poor Obfuscation Implementation File System). Tato komponenta je základním prvkem všech ostatních POI složek. Používá se k explicitnímu čtení různých souborů.
- HSSF (Horrible Spreadsheet Format). Používá se ke čtení a zápisu *xls* formátu souborů Microsoft Excel.
- XSSF (XML Spreadsheet Format) se používá pro *xlsx* formát souboru aplikace Microsoft Excel.
- HPSV (Horrible Property Set Format). Používá se k extrahování sad vlastností souborů Microsoft Office.
- HWPF (Horrible Word Processor Format) slouží ke čtení a zápisu *doc* formátu souborů Microsoft Word.

- XWPF (XML Word Processor Format) se používá ke čtení a zápisu *docx* formátu souborů Microsoft Word.
- HSLF (Horrible Slide Layout Format) slouží ke čtení, vytváření a úpravám prezentací aplikace Microsoft PowerPoint.
- HDGF (Horrible DiaGram Format) obsahuje třídy a metody pro binární soubory aplikace Microsoft Visio.
- HPBF (Horrible PuBlisher Format) se používá se ke čtení a zapisování souborů aplikace Microsoft Publisher.

Starší verze knihovny Apache POI podporují pouze starší typy binárních souborů Microsoft Office jako je *doc*, *xls*, *ppt* atd. Od verze 3.5 podporuje knihovna formáty OOXML souborů *docx*, *xlsx*, *pptx* a další.

4.2 JavaScript

JavaScript je interpretovaný, multiplatformní, objektově orientovaný programovací jazyk. Syntaxe je velmi obdobná s jazyky C, C++ nebo Java, ale podobnost s těmito jazyky však končí právě syntaxí, jelikož sémantika jazyka JavaScript je naprosto rozdílná. JavaScript je volně typový jazyk, což znamená, že proměnné nemusí mít zadaný datový typ. Mechanismus dědičnosti v jazyce JavaScript je založen na prototypech, což je zcela odlišné od dědičnosti v jazycích C++ nebo Java, které jsou založené na třídách. Jelikož je jazyk JavaScript interpretovaný, nikoliv kompilovaný, je často považován spíše za skriptovací jazyk místo pravého programovacího jazyka. Nejčastěji se jazyk používá jako součást webových stránek, jejichž implementace umožňuje skriptům komunikovat s uživatelem a provádět dynamické změny na webových stránkách. Celá podkapitola čerpá z [21][22].

Jednou z nejběžnějších, mylně známých představ o JavaScriptu je, že tento jazyk je zjednodušená verze jazyka Java, programovacího jazyka od společnosti Sun Microsystems. Kromě syntaktické podobnosti a faktu, že jazyky Java i JavaScript mohou poskytovat spustitelný obsah ve webových prohlížečích, spolu tyto dva jazyky vůbec nesouvisí. Podobnost názvů je čistě marketingovým trikem společnosti Netscape. Jazyk byl při jeho vzniku v roce 1995 původně nazýván LiveScript, ale název byl následně hned změněn na podobu, jakou známe dnes, pravděpodobně kvůli rozruchu, které v té době bylo okolo jazyka Java.

Obvykle je jazyk JavaScript nazývaný jako *client-side* (na straně klienta), aby bylo zdůrazněno, že veškeré skripty jsou spouštěny a běží na klientském počítači, nikoliv na webovém serveru.

Tato varianta je nejběžnější formou jazyka JavaScript. Daný kód/skript musí být zahrnut v dokumentu HTML nebo na něj odkazovat v hlavičce, aby mohl být interpretován webovým prohlížečem. To znamená, že výsledná webová stránka nebude již pouze statický kód HTML, ale bude obsahovat funkce, které spolupracují s uživatelem, řídí prohlížeč nebo dynamicky vytváří obsah dané webové stránky. Tento *client-side* mechanismus poskytuje mnoho výhod oproti tradičním CGI skriptům na straně serveru. Například lze kontrolovat, zda uživatel zadal platnou e-mailovou adresu do pole formuláře a až následně, pokud jsou všechny položky platné, odeslat data na webový server. Jazyk JavaScript lze také použít k zachycení událostí iniciovaných uživatelem, jako je například zmáčknutí tlačítka myši, stisknutí klávesy a mnoho další akcí, které uživatel uskuteční.

Mezi hlavní výhody jazyka JavaScript patří:

- Menší interakce se serverem. Před odesláním dat na server lze ověřit uživatelský vstup.
- Okamžitá zpětná vazba pro uživatele. Není potřeba čekat na znovunačtení stránky při uživatelských akcích.
- Zvýšená interaktivita. Okamžité reakce na uživatelské události bez nutnosti komunikace s webovým serverem.
- Bohatší rozhraní. Pomocí JavaScriptu lze zahrnout takové položky jako animované posuvníky, drag-and-drop komponenty apod., které návštěvníkům poskytují intuitivnější a pohodlnější používání webových stránek.

S jazykem JavaScript nelze zacházet jako s plnohodnotným programovacím jazykem, jelikož postrádá některé následující podstatné rysy:

- *Client-side* JavaScript nepovoluje čtení nebo zápis souborů na server. To je ponecháno z důvodu bezpečnosti.
- Nelze ho použít pro síťové aplikace, protože tato podpora není k dispozici.
- Nemá podporu pro více vláknové funkce a žádné možnosti pro využití multiprocessoru.

4.2.1 jQuery

Je odlehčená open source JavaScript knihovna, která klade důraz na interakci mezi strukturou dokumentu HTML a funkcionalitou jazyka JavaScript. Knihovna realizuje mnoho běžných činností, které jinak vyžadují napsání mnoha řádků kódu k jejich dosažení, a obaluje je do metod, které lze volat pouze jedním řádkem kódu. jQuery také zjednodušuje mnoho složitých implementací z jazyka JavaScript, například volání AJAX, efekty a animace,

manipulace s modelem DOM nebo úprava CSS. Existuje mnoho frameworků pro jazyk JavaScript, ale jQuery je jedna z nejpopulárnějších, a to dokazuje i její využití na webech velkých společností jako je Google, Microsoft, IBM nebo Netflix. Čerpáno z [23][24].

První verze byla vydána Johnem Resigem v roce 2006 a dnes je knihovna vyvíjena skupinou dobrovolníků nesoucí název The jQuery Team, která v roce 2019 vydala novou stabilní verzi 3.4.0. Jelikož je knihovna světově velmi populární, je možné na internetu nalézt nespočet doplňků, které tuto knihovnu rozšiřují a posouvají její možnosti ještě o kousek dál.

Mezi základní a stěžejní funkce jQuery patří:

- Přístup k prvkům v dokumentu. Bez JavaScript knihoven musí weboví vývojáři často napsat mnoho řádků kódu pro procházení objektového modelu dokumentu (DOM), aby našli konkrétní HTML prvky. S jQuery mají vývojáři k dispozici robustní a efektivní mechanismus selektoru, který usnadňuje získání přesné části dokumentu, s kterou má být manipulováno. Příklad nalezení elementu `<p>` obsaženém v prvku `<div>`, který je definovaný třídou `content`:

```
$('#div.content').find('p');
```

- Změna vzhledu webové stránky. CSS nabízí efektivní techniku vykreslení a stylizování dokumentu, která trochu zaostává, jelikož ne všechny webové prohlížeče podporují stejné standardy. Pomocí jQuery mohou weboví vývojáři překlenout tuto mezeru, a kromě toho, mohou měnit třídy nebo jednotlivé vlastnosti stylů i po vykreslení dané webové stránky. Příklad přidání třídy `active` prvnímu elementu ``:

```
$('#ul > li:first').addClass('active');
```

- Změna obsahu dokumentu. Knihovna není omezena pouze na kosmetické změny, ale může upravovat i obsah samotného dokumentu. Lze měnit text, vložit nebo odebrat obrázky, přeuspořádat seznamy nebo může být celá struktura kódu HTML změněna nebo rozšířena. Příklad přidání nového prvku `<a>` do prvku s id `container`:

```
$('#container').append('<a href="another.html"> Click here!</a>');
```

- Reakce na interakci uživatele. Knihovna jQuery nabízí elegantní způsob, jak zachytit širokou škálu událostí, aniž by bylo nutné znepráhledňovat samotný kód HTML s obslužnými rutinami jednotlivých událostí. Poskytované rozhraní pro zpracování událostí současně odstraňuje nekonzistenci mezi webovými prohlížeči, kterou často musí weboví vývojáři řešit ručně. Příklad registrování události `click`

pro prvek `<button class="details">`, která po zmáčknutí daného tlačítka zobrazí element `<div class="detail">`:

```
$('#button.details').on('click', function() {
    $('#div.detail').show();
});
```

- Animování změn v dokumentu. jQuery poskytuje efektivní implementaci animovaného a interaktivního chování poskytovaného zpět k uživateli. Knihovna obsahuje mnoho předdefinovaných animací pro prvky, ale také sadu nástrojů pro vytvoření nových efektů. Příklad efektu „vysunutí směrem dolů“ pro prvek `<div class="detail">`:

```
$('#div.detail').slideDown();
```

- Načtení informací ze serveru bez aktualizace stránky. Tento návrhový vzor je znám jako AJAX, který původně sloužil k asynchronnímu zpracování webových, ale od té doby byl rozšířen o mnohem větší sadu technologií pro komunikaci mezi klientem a serverem. Knihovna jQuery odstraňuje složitost používání této technologie v rámci odlišných typů webových prohlížečů a umožňuje vývojářům zaměřit se více na funkčnost serveru. Příklad pro načtení dat ze serveru, nalezení prvku s id `content` a vložení jeho obsahu do elementu `<div class="detail">`:

```
$('#div.detail').load('someContent.html #content');
```

- Zjednodušit běžné úlohy JavaScriptu. Kromě všech prvků specifických pro práci s DOM poskytuje knihovna vylepšení i základních složek jazyka JavaScript, například iterace a manipulace s polem a kolekcemi. Příklad cyklu pro sčítání hodnot kolekce:

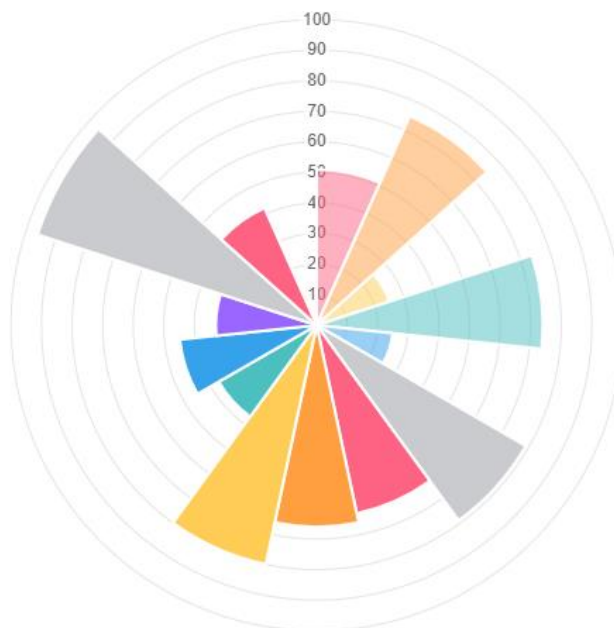
```
$.each(array, function(key, value) {
    total += value;
});
```

4.2.2 Chart.js

Chart.js je komunitně vyvíjená open source knihovna pro JavaScript, která umožňuje vizualizovat data pomocí 8 druhů grafů a jejich vzájemných kombinací. Grafy jsou vykresleny pomocí HTML5 elementu `<canvas>` a díky široké škále atributů a doplňků lze každý z grafů značně graficky přizpůsobit konečným požadavkům, a to i včetně animací. Knihovna nemá žádné jiné vnější závislosti a její velikost je velmi malá (při zmenšení je okolo 11kb). Dalším výhodou této knihovny je, že veškeré grafy jsou responzivní, takže se přizpůsobí jakémukoliv dostupnému místu na webové stránce. Na rozdíl od mnoha jiných knihoven, poskytuje rozsáhlou a přehlednou dokumentaci, která umožňuje snadné a rychlé nasazení této knihovny do dalších aplikací.

Mezi nabízené skupiny grafů patří:

- sloupcové grafy (vertikální, horizontální, skládané...),
- spojnicové grafy (základní, interpolační...),
- plošné a paprskové grafy,
- výsečové a prstencové grafy,
- bodové a bublinové grafy,
- graf radar a polární oblast,
- a další kombinace [25].



Obrázek 7 – Ukázka grafu polární oblast

4.3 REST API

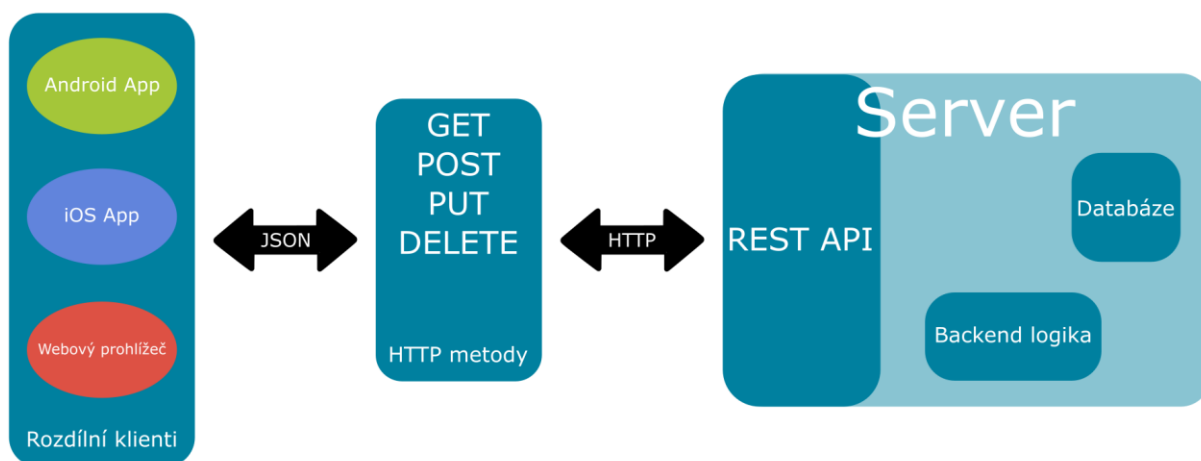
Koncepce REST nebo RESTful API je architektura rozhraní, které je navrženo tak, aby využívalo výhod již existujících protokolů. Je možno jej použít téměř u kteréhokoliv protokolu, ale nejčastěji se využívá s protokolem HTTP při vytváření webových rozhraní. To znamená, že vývojáři nemusí instalovat knihovny nebo další software, aby mohli využít právě tento model rozhraní. REST API bylo definováno Royem Fieldingem (jeden ze spoluautorů protokolu HTTP) v roce 2000 v jeho disertační práci. Celá podkapitola čerpá z [26].

Rozhraní REST je použitelné pro jednotný a snadný přístup ke zdrojům (*resources*). Na rozdíl od SOAP není REST API omezeno pouze na výměnu dat pomocí formátu XML, ale místo toho může vracet XML, JSON nebo jakýkoliv jiný formát v závislosti na tom, co daný klient požaduje. A na rozdíl od RPC nejsou uživatelé povinni znát názvy procedur nebo konkrétní parametry v určitém pořadí, tzn. že je REST rozhraní orientováno datově, nikoliv procedurálně. Fielding ve své disertační práci uvedl šest klíčových omezení, které musí každé správné RESTful rozhraní splňovat:

- Klient-server. Koncept, který odděluje odpovědnosti klienta a serveru a umožňuje je samostatně a nezávisle vyvíjet. Tím pádem mohou být implementovány a nasazovány nezávisle na použitých technologiích a napsány v libovolném jazyce. Jinými slovy, změny provedené v např. mobilní aplikaci by neměly ovlivňovat datovou strukturu nebo návrh databáze na serveru. A zároveň úprava databáze nebo změna v serverové aplikaci by neměla ovlivnit mobilní zařízení.
- Jednotné rozhraní. Toto rozhraní umožňuje klientovi komunikovat se serverem v jednotném jazyce, nezávisle na architektuře jednoho či druhého. Toto rozhraní by mělo poskytovat neměnné se standardizované prostředky komunikace, jako je například použití protokolu HTTP s prostředky URI, operacemi CRUD (Create, Read, Update, Delete) a formátem dat JSON viz Obrázek 8.
- Vrstvený systém. Jedná se o systém tvořený z vrstev, přičemž každá vrstva má specifickou funkčnost a zodpovědnost. Všechny vrstvy jsou oddělené, ale také vzájemně spolupracují jedna s druhou a vytvářejí hierarchii, která pomáhá zformovat více škálovatelné a modulární aplikace. Běžně se využívá pro vynucení bezpečnosti, ukládání do mezipaměti nebo rozložení zátěže.
- Bezstavovost (Stateless). REST rozhraní je bezstavové, což znamená, že jednotlivá volání mohou být provedena nezávisle na sobě a každé volání obsahuje všechna potřebná data k jeho úspěšnému dokončení. To znamená, že se rozhraní nespolehá na data uložená na serveru nebo v relacích, ale pouze na data, která jsou poskytnuta v daném volání. Při komunikaci nejsou na serveru ukládány identifikační informace o klientovi, místo toho každé volání obsahuje potřebná identifikační data, například klíč rozhraní, přístupový token, ID uživatele apod.
- Mezipaměti. Vzhledem k tomu, že je REST API bezstavové, zvyšuje se tím režie zpracování požadavků a celkové zatížení serveru. Ukládáním dat z odpovědí požadavků

do mezipaměti lze snížit latenci, zvýšit celkovou dostupnost a spolehlivost aplikace a snížit zatížení webového serveru.

- Kód na vyžádání (*Code-On-Demand*) umožňuje přenos kódu nebo apletů prostřednictvím rozhraní pro použití v rámci klientské aplikace. V podstatě vytváří inteligentní aplikaci, která již není závislá pouze na vlastní struktuře kódu. Protože klient musí být schopen porozumět a spustit daný kód, je *code-on-demand* jediným omezením, které je považováno za volitelné. Tuto technologii využívají například Java aplety, JavaScript nebo Flash.



Obrázek 8 – REST API architektura

5 ANALÝZA

Obsah této kapitoly se řídí unifikovaným procesem vývoje objektově orientovaných aplikací. Samotné požadavky, analýza i návrh analytických tříd jsou zpracované v softwarovém nástroji pro komplexní modelování, systémovou analýzu a návrh aplikací Enterprise Architect od společnosti Sparx Systems v programové verzi 12.0.1215. Celá kapitola vychází z [27].

5.1 Požadavky

Systémové požadavky jsou základním kamenem vývoje jakéhokoliv systému. Požadavky převážně určují, co by měl systém provádět, nikoliv jak by toho mělo být docíleno. Existuje mnoho podob kategorizace systémových požadavků, ale nejjednodušší a nejpoužívanější je rozdělení na *funkční* a *nefunkční*. Funkční požadavky popisují požadované funkce, tedy to, co by měl systém vykonávat. Nefunkční požadavky jsou omezující podmínky pro daný systém, např. *systém bude uživatelům dostupný 360 dní v roce*. Každý požadavek by měl obsahovat jedinečný identifikátor, název systému, klíčové slovo *bude* a definici funkcionality, tzn. výsledný formát požadavku bude mít podobu: *<id> <systém> bude <funkce>*.

5.1.1 Funkční požadavky

Funkční požadavky pro modul Reporty – Statistiky jsou rozděleny do čtyř logických oblastí zaměřených na konkrétní problematiku: report, statistika, šablony a export.

Oblast Report (viz Obrázek 9) je zaměřena, jak už název napovídá, na část obsahující samotný report, ale i na část nastavení vyhledávacího filtru, díky kterému budou data ze systému 1CLICK vyhledávána a tříděna.

Report
<input checked="" type="checkbox"/> + FR200 - Modul bude v reportu umožňovat filtrování dle datumu založení úkolů
<input checked="" type="checkbox"/> + FR201 - Modul bude v reportu umožňovat filtrování dle datumu termínu úkolů
<input checked="" type="checkbox"/> + FR202 - Modul bude v reportu umožňovat filtrování dle datumu vyřešení úkolů
<input checked="" type="checkbox"/> + FR203 - Modul bude v reportu umožňovat filtrování dle řešitelů úkolů
<input checked="" type="checkbox"/> + FR204 - Modul bude v reportu umožňovat filtrování dle úkolových značek s logickým operátorem A a NEBO
<input checked="" type="checkbox"/> + FR205 - Modul bude v reportu umožňovat filtrování dle projektů
<input checked="" type="checkbox"/> + FR206 - Modul bude v reportu umožňovat filtrování dle kontaktů
<input checked="" type="checkbox"/> + FR207 - Modul bude umožňovat kombinaci daných filtrů
<input checked="" type="checkbox"/> + FR208 - Modul bude v reportu zobrazovat přehled stavů vyřešení úkolů dle zadaných filtrů roztržiděných dle jednotlivých řešitelů
<input checked="" type="checkbox"/> + FR209 - Modul bude v reportu zobrazovat grafický prvek obsahující poměr plnění daných úkolů
<input checked="" type="checkbox"/> + FR210 - Modul bude v reportu zobrazovat detailní přehled jednotlivých úkolů v daných kategoriích pro každého z řešitelů

Obrázek 9 – Funkční požadavky: Report

Pro umožnění vytvoření výsledného reportu je prvním a jediným povinným parametrem filtru datumový rozsah díky kterému budou v systému vyhledány dané úkoly spadající do tohoto rozsahu. Rozšiřujícím nastavením tohoto parametru datumového rozsahu bude určení,

zda se daný rozsah týká založení, vyřešení nebo termínu úkolů. Dalším, teď již volitelným, parametrem filtru bude omezení na konkrétní uživatele (řešitele úkolů). Jak už bylo zmíněno v Podkapitole 2.3.1, každý z úkolů může obsahovat úkolové značky, to znamená, že ve filtru nesmí chybět ani možnost filtrování dle těchto značek. Doplnkovým nastavením u tohoto parametru bude i možnost definice, zda úkoly musí obsahovat všechny zvolené úkolové značky zároveň, nebo stačí, že budou obsahovat pouze nějakou z nich. Poslední dva parametry filtru budou určovat, zda dané úkoly patří do nějakého existujícího projektu nebo jestli se týkají zvolených kontaktů. Nejpodstatnější vlastností filtru bude samozřejmě možnost kombinatoriky daných parametrů mezi sebou pro striktnější omezení při vyhledávání úkolů pro daný report. Samotný report bude znázorňovat plnění daných úkolů, tj. celkový počet úkolů, počet úkolů splněných v termínu, počet úkolů splněných po termínu a počet nesplněných úkolů, a to jak v celkové rovině, tak i roztríděné dle jednotlivých řešitelů. Pro rychlý vizuální přehled, jak si řešitelé stojí v plnění termínů, bude report u každého z nich zobrazovat grafický prvek znázorňující poměr výše zmíněných stavů vyřešení úkolů. Každá z kategorií stavů vyřešení úkolů bude obsahovat seznam úkolů spadajících do této kategorie s detailními informacemi o daném úkolu, podobně jako jsou zobrazeny u úkolu v systému 1CLICK (id, stav, datum zadání/vyřešení/termínu, zadavatele a řešitele, strávený čas, projekt, značky a kontakt).

Další oblastí funkčních požadavků je Statistika, viz Obrázek 10. Veškeré statistické charakteristiky budou zpracovány nad identickou sadou úkolů jako pro report, tzn. podle nastavení filtru, a budou zobrazeny formou vhodných grafů pro danou statistickou charakteristiku.

Statistika	
<input checked="" type="checkbox"/>	+ FR300 - Modul bude zobrazovat všechny statistické výpočty formou grafů
<input checked="" type="checkbox"/>	+ FR301 - Modul bude zpracovávat statistické výpočty nad daty z daného filtru
<input checked="" type="checkbox"/>	+ FR302 - Modul bude počítat průměr a medián počtu úkolů uživatelů v jednotlivých kategoriích
<input checked="" type="checkbox"/>	+ FR303 - Modul bude do grafu shlukovat data vyřešení jednotlivých úkolů v čase (X osa - hodiny/minuty, Y osa - sekundy)
<input checked="" type="checkbox"/>	+ FR304 - Modul bude vytvářet histogram délek úkolů
<input checked="" type="checkbox"/>	+ FR305 - Modul bude vytvářet časovou řadu počtu založení/vyřešení/termínu úkolů v jednotlivých dnech a počítat lineární trend z dané řady

Obrázek 10 – Funkční požadavky: Statistika

Modul Reporty – Statistiky bude obsahovat čtyři manažersky zajímavé grafy (charakteristiky), z čehož jedna z nich bude ještě rozčleněna na další podkategorie. První z grafů bude zobrazovat celkový počet úkolů jednotlivých uživatelů vůči aritmetickému průměru a mediánu celé společnosti. Dále bude možné zobrazit tento stejný graf pro jednotlivé stavy vyřešení úkolů, tj. úkoly splněné v termínu, úkoly splněné po termínu a nesplněné úkoly. Dalším grafem bude bodový graf znázorňující jednotlivé časy vyřešení úkolů, ať už v termínu nebo po něm,

kde na ose x budou vyneseny hodiny a minuty a na ose y sekundy pro lepší vizuální distribuci bodů. Následující statistickou charakteristikou bude sestavení histogramu jednotlivých délek (datum založení až termín) úkolů, a to s proměnlivým počtem a velikostí tříd, které se budou vytvářet až podle počtu vstupních dat. Posledním typem grafu bude časová řada zobrazující počet založení/vyřešení/termínů úkolů v jednotlivých dnech, podle toho, jaký druh časového období bude zvolen ve filtru. Doplňkovou funkcí bude zobrazení lineárního trendu této časové řady i s jeho rovnicí a hodnotou spolehlivosti.

Šablony
<input checked="" type="checkbox"/> + FR400 - Modul bude umět načítat, ukládat a mazat šablony pro přednastavení filtrů
<input checked="" type="checkbox"/> + FR401 - Modul bude obsahovat dva druhy šablon "Systémová" a "Uživatelská"
<input checked="" type="checkbox"/> + FR402 - Modul bude umožňovat správu globálních "Systémových" šablon dle nastavení ACL

Obrázek 11 – Funkční požadavky: Šablony

Šablony jsou následující oblastí funkčních požadavků, viz Obrázek 11. Aby uživatelé při častém používání tohoto modulu nemuseli pokaždé nastavovat složité kombinace parametrů ve filtru, bude modul umožňovat ukládání těchto přednastavených filtrů pod libovolným názvem. Takto uložené šablony filtrů bude možno při opětovném použití načíst pro rychlejší nastavení filtru nebo smazat. Šablony budou dvojího typu: *uživatelská* a *systémová*, přičemž uživatelské šablony budou zobrazeny právě jen uživateli, který je založil a systémové šablony se budou zobrazovat všem uživatelům. Zakládat a celkově spravovat systémové šablony bude umožněno pouze těm uživatelům nebo skupinám uživatelů, kteří k tomu budou mít povolení od administrátora v systému 1CLICK, konkrétněji v nastavení přístupových práv tzv. ACL.

Poslední oblastí je Export, viz Obrázek 12. Celý report, ať už jeho souhrnnou část nebo jednotlivé detailní seznamy úkolů, bude možno vyexportovat do souboru formátu *.xlsx* a všechny statistické grafy bude možno uložit do souboru formátu *.png*.

Export
<input checked="" type="checkbox"/> + FR100 - Modul bude umožňovat export dat do souboru formátu <i>.xlsx</i>
<input checked="" type="checkbox"/> + FR101 - Modul bude umožňovat export z globálního reportu a dat ze statistických grafů
<input checked="" type="checkbox"/> + FR102 - Modul bude umožňovat export jednotlivých částí reportu (detaily skupin úkolů)
<input checked="" type="checkbox"/> + FR103 - Modul bude umožňovat export statistických grafů do souboru formátu <i>.png</i>

Obrázek 12 – Funkční požadavky: Export

5.1.2 Nefunkční požadavky

I nefunkční požadavky jsou rozděleny do čtyř logických oblastí: bezpečnost, dostupnost, logování a vzhled. První oblast (viz Obrázek 13) je zaměřena na bezpečnost. To znamená, že volání funkcí REST API bude umožněno pouze s přiloženým uživatelským tokenem, stejně tak jako je tomu v rámci zbytku systému 1CLICK. Modul bude dostupný dle nastavení přístupových práv zmíněných v předchozí podkapitole.

Bezpečnost
+ NFR100 - Modul bude využívat zabezpečené REST API
+ NFR101 - Modul bude uživatelům přístupný dle nastavení ACL systému 1CLICK

Obrázek 13 – Nefunkční požadavky: Bezpečnost

Modul bude dostupný samozřejmě pouze při běhu systému 1CLICK, jelikož jsou z něho čerpána potřebná data pro report a statistické grafy, a přístup do modulu bude umožněn pouze přihlášeným uživatelům do systému (viz Obrázek 14).

Dostupnost
+ NFR200 - Modul bude dostupný pouze přihlášeným uživatelům do systému 1CLICK
+ NFR201 - Modul bude dostupný vždy při běhu systému 1CLICK

Obrázek 14 – Nefunkční požadavky: Dostupnost

Poslední dvě oblasti nefunkčních požadavků se zabývají logováním a vzhledem (viz Obrázek 15). Veškeré záznamy budou vloženy do centrálního logovacího systému 1CLICK. Design modulu bude již dodržovat standardy a barevné palety mobilní aplikace systému 1CLICK, nikoliv desktopové aplikace tohoto systému.

Logování
+ NFR300 - Modul bude veškeré chyby zaznamenávat do centrálního logu systému 1CLICK

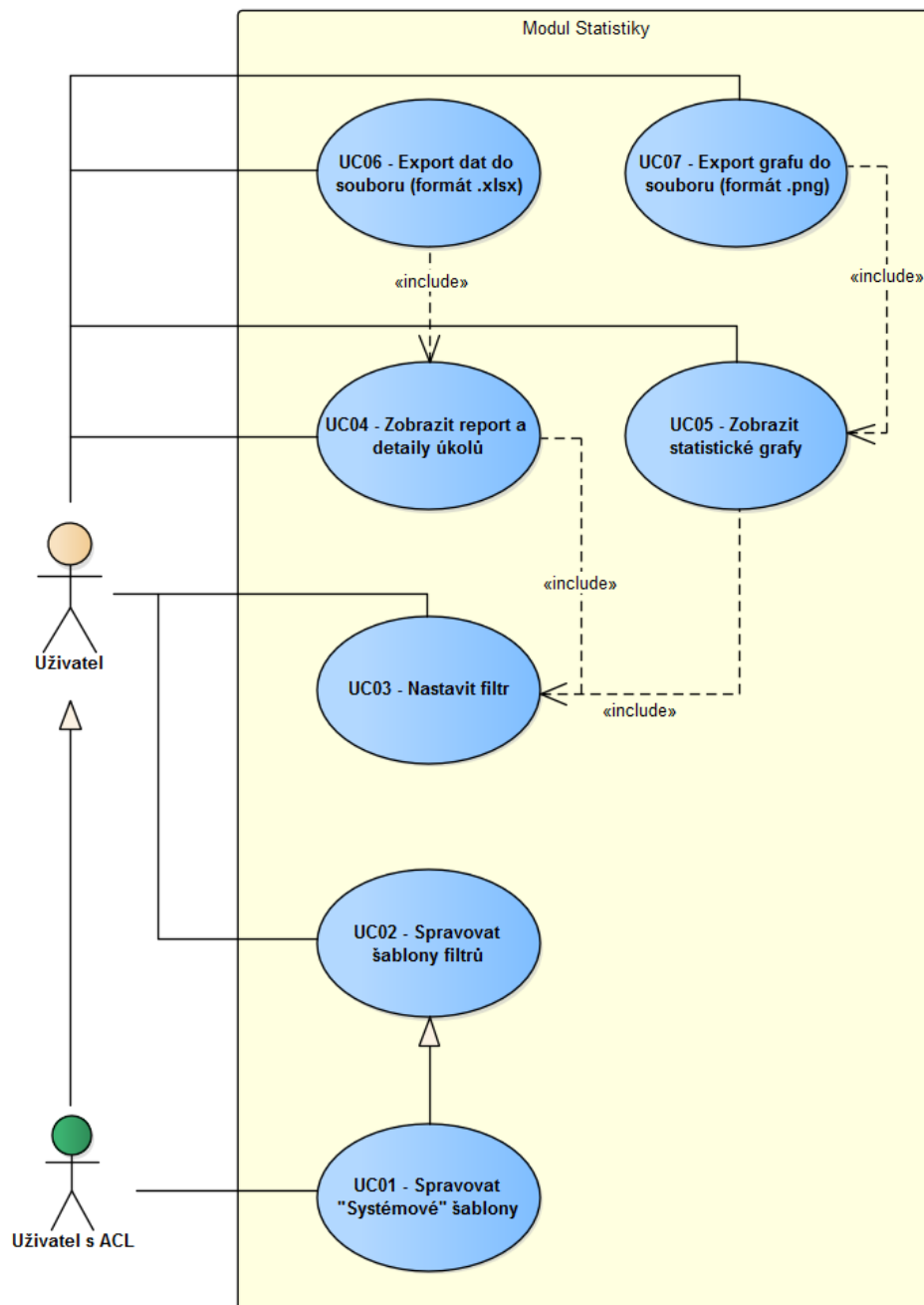
Vzhled
+ NFR400 - Modul bude v grafickém rozhraní dodržovat barvy a standardy mobilní aplikace 1CLICK

Obrázek 15 – Nefunkční požadavky: Logování a Vzhled

5.2 Případy užití

Modelování případů užití je jednou z forem inženýrství požadavků a skládá se z nalezení hranic systému, vyhledání aktérů (osob používajících daný systém) a nalezení případů užití (činností, které mohou aktéři se systémem vykonávat).

Na diagramu případu užití pro model Reporty – Statistiky (Obrázek 16) jsou znázorněny jednotlivé případy užití odrážející funkční požadavky systému, které jsou v přímé interakci s uživatelem, těmi jsou: nastavení filtru, správa šablon filtru, zobrazení reportu a statistických grafů a jejich exporty do souborů.



Obrázek 16 – Diagram případů užití: Modul Reporty – Statistiky

Dále jsou zde viditelní dva aktéři: *uživatel* a *uživatel s ACL*. *Uživatel s ACL* má stejné pravomoci jako *uživatel*, jen s tím rozdílem, že může navíc spravovat systémové šablony filtru. Na diagramu je viditelná posloupnost jednotlivých případů užití a to, že *nastavení filtru* předchází před jednotlivými zobrazeními reportu a statistických grafů a stejně tak je zapotřebí nejdříve report a statistické grafy zobrazit, než budou dané výstupy ukládané do souborů.

Obrázek 17 zobrazuje matici sledovanosti požadavků, která reflektuje správnost návrhu a vytvoření jednotlivých případů užití. To znamená, že každý výše zmíněný funkční požadavek (horizontální řada identifikátorů) v této analýze je realizován jedním případem užití (vertikální řada identifikátorů) a každý případ užití realizuje alespoň jeden funkční požadavek.

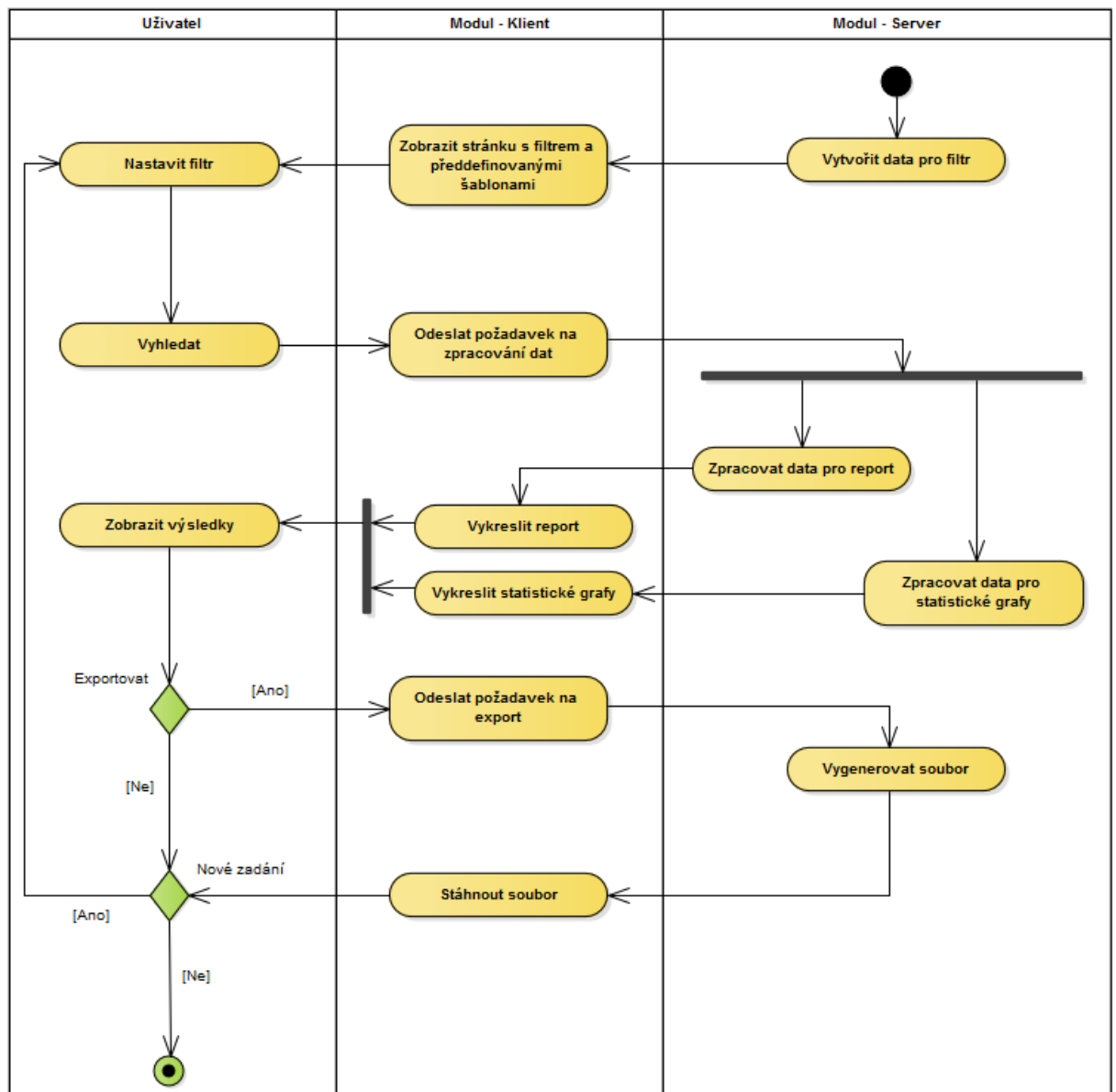
	FR100	FR101	FR102	FR103	FR200	FR201	FR202	FR203	FR204	FR205	FR206	FR207	FR208	FR209	FR210	FR300	FR301	FR302	FR303	FR304	FR305	FR400	FR401	FR402
UC01																							↑	↑
UC02																						↑		
UC03					↑	↑	↑	↑	↑	↑	↑	↑					↑							
UC04													↑	↑	↑									
UC05																↑		↑	↑	↑	↑			
UC06	↑	↑	↑																					
UC07				↑																				

Obrázek 17 – Matice sledovatelnosti požadavků

5.3 Diagram aktivit

Diagramy aktivit jsou objektově orientované vývojové diagramy, díky kterým lze modelovat různé procesy jako kolekce uzlů spojených hranami.

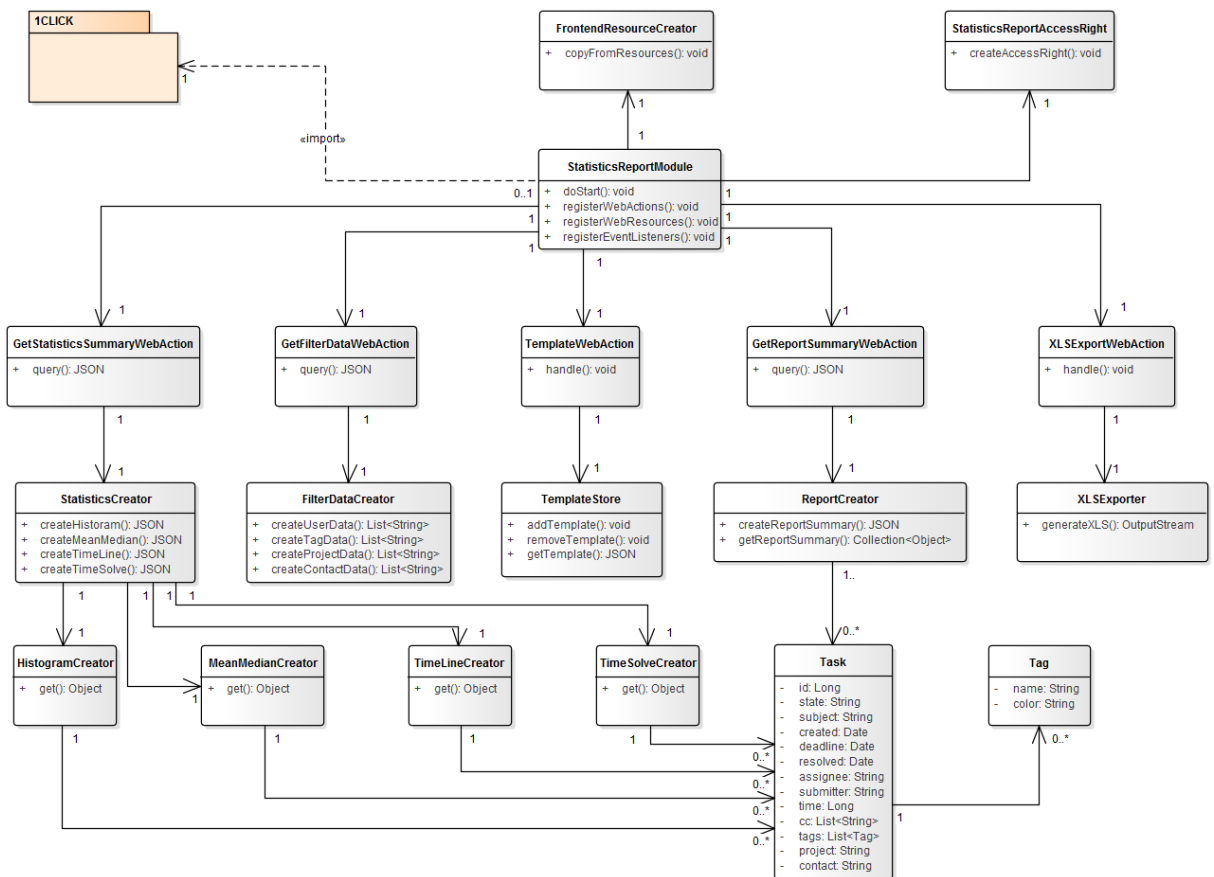
Na diagramu aktivit (Obrázek 18) je znázorněn návrh komunikace a interakce mezi uživatelem a webovým klientem, a webovým klientem a systémem 1CLICK.



Obrázek 18 – Diagram aktivit: Komunikace uživatel-klient-server

5.4 Návrh analytických tříd

Obrázek 19 znázorňuje diagram návrhu analytických tříd pro modul Reporty – Statistika. Návrh obsahuje hlavní třídu *StatisticsReportModule*, která zajišťuje napojení na systém 1CLICK, registraci webových akcí a zdrojů, a realizuje veškeré ostatní závislosti na knihovnách a třídách systému 1CLICK potřebné pro běh tohoto modulu. Dále jsou zde třídy **WebAction* pro definice webových akcí, třída *StatisticsReportAccessRight* rozšiřující ACL systému 1CLICK, třídy **Creator* připravující objekty pro odpovědi na požadavky klienta, třídy *Task* a *Tag* nesoucí informace o jednotlivých úkolech a další.



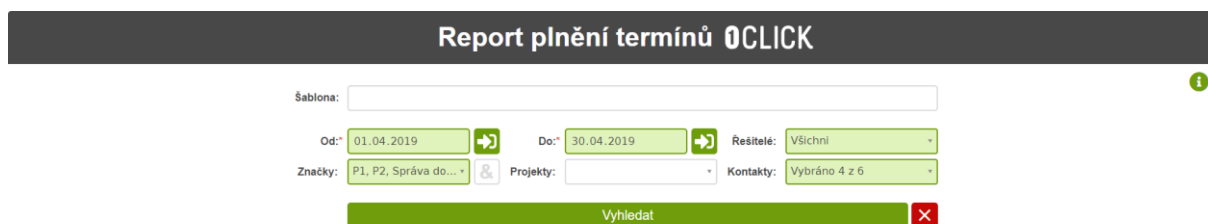
Obrázek 19 – Návrh analytických tříd modulu Reporty – Statistika

6 POPIS FUNKCIONALITY GUI

Veškeré obrázky použité v této kapitole jsou pořízeny z instance systému 1CLICK s názvem *DP_Kolacny*, běžící na vývojovém testovacím serveru společnosti One Click Business Solution s. r. o., obsahující uměle vytvořená testovací data.

6.1 Vyhledávací filtr

Prvním a nejdůležitějším prvkem celého GUI je vyhledávací filtr (Obrázek 20), který se uživateli zobrazí jako první po načtení prostředí obsahující klientskou část modulu. Slouží k výběru úkolů, které mají být zařazeny do reportu či statistických grafů. Aby bylo možné sestavit samotný report nebo statistické grafy, obsahuje filtr dvě povinné položky, tj. *Od* a *Do*, určující datumový rozsah, do kterého jednotlivé úkoly spadají, a bez kterých není možné spustit vyhledávání. S tímto datumovým rozsahem jsou úzce spjaty přepínače definující, k jakému druhu data (založení, termín, vyřešení) úkolu se toto rozmezí vztahuje. Ve výchozím stavu je vždy přepínač nastaven na vztah k založení úkolu.



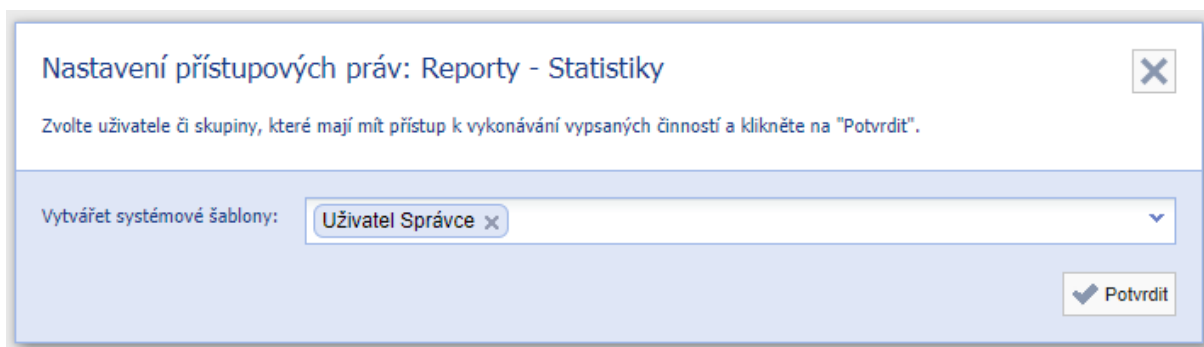
Obrázek 20 – Vyhledávací filtr

Dalším již volitelným omezujícím parametrem filtru je položka *Řešitelé*. Tato vstupní položka umožňuje multi-výběr, což znamená možnost výběru žádného, jednoho, několika nebo všech uživatelů systému 1CLICK. Hlavním a podstatným rozdílem mezi vybráním všech uživatelů a nenastavením tohoto parametru je ten, že při nespecifikování tohoto parametru jsou v reportu zobrazeni pouze ti uživatelé, kteří jsou řešiteli minimálně jednoho úkolu vyhovujícího nastavení filtru. Druhá varianta, tedy zvolení všech uživatelů, zobrazí v reportu všechny uživatele, tzn. i ty, kteří nejsou řešiteli žádného úkolu vyhovujícího zvolenému filtru a budou zobrazeni s nulovými hodnotami. Následující multi-výběrovou položkou vyhledávacího filtru jsou *Značky*. Díky této vstupní položce, jak již název napovídá, lze nastavit omezení vyhledávání na určité úkolové značky. Úkolové značky se odlišují vůči zbytku parametrů filtru tím, že se u konkrétního úkolu může vyskytnout více značek zároveň, oproti řešiteli nebo projektu, kteří mohou mít pouze jeden výskyt. Právě kvůli této odlišnosti a pro větší možnosti kombinatoriky je ke vstupní položce *Značky* přidružen přepínač definující, zda se zvolené

značky budou vyskytovat u daných úkolů všechny zároveň, nebo stačí, že úkol bude obsahovat alespoň jednu z nich. Ve výchozím nastavení je zvolena druhá varianta, tj. úkol obsahuje alespoň jednu ze zvolených značek. Dalším z volitelných parametrů vyhledávacího filtru je multi-výběrová položka *Projekty*. Tento parametr i s parametrem následujícím (multi-výběrová položka *Kontakty*), filtrují vyhledání úkolů pomocí projektu, resp. kontaktu. Samozřejmostí vyhledávacího filtru je, že všechny vstupní parametry je možné mezi sebou kombinovat pro složitější a specifitější pohledy na fungování firmy. Blok vyhledávacího filtru navíc obsahuje funkční prvky pro správu šablon (viz Oddíl 6.1.1) a informační tlačítko zobrazující legendu k ikonám použitých v GUI, (viz Pododdíl 6.1.2.1).

6.1.1 Správa šablon pro přednastavení filtru

Blok vyhledávacího filtru obsahuje interaktivní položku *Šablona*, která umožňuje ukládání, načítání a mazání šablon pro přednastavení filtru. Existují dvě varianty šablon: *uživatelská* a *systémová*. Uživatelské šablony se zobrazují pouze uživatelům, kteří si je pro svou potřebu založili. Systémové šablony jsou zobrazeny všem uživatelům, ale možnost jejich správy je umožněna pouze uživatelům systému 1CLICK, kteří jsou definováni v nastavení přístupových práv pro tento modul. Toto nastavení je přidáno při instalaci modulu do systému 1CLICK do oblasti nastavení systémových ACL, (viz Obrázek 21).



Obrázek 21 – Nastavení přístupových práv modulu Reporty – Statistiky

Položka *Šablona* funguje jako seznam již dříve uložených šablon, rozřazených dle typu, s možností fulltextového vyhledávání. Díky tomu lze pohodlně a bez opakovaného složitého nastavování filtru, načíst dříve nadefinovaný filtr a začít vyhledávat. Tuto možnost budou manažeři firmy využívat převážně pro vytvoření systémových šablon obsahujících přednastavení časových úseků, které budou pro danou firmu stěžejní. Při výběru konkrétní šablony se automaticky vyplní vstupní položky, které daná šablona uchovává, a pokud má



Obrázek 22 – Správa šablon pro přednastavení filtru: Smazání

uživatel dostatečná oprávnění zobrazí se i možnost vymazání této šablony či jen její reset, (viz Obrázek 22). Pokud je zvolena určitá šablona a ručně je změněn jakýkoliv ze vstupních parametrů, daná šablona se z položky *Šablona* odstraní a je umožněno takto změněný filtr uložit jako novou šablonu. Pokud po napsání textu do této položky fulltextové vyhledávání nenalezne shodu s žádným názvem již existující šablony, je umožněno uložit takto nazvanou šablonu jako novou, (viz Obrázek 23).



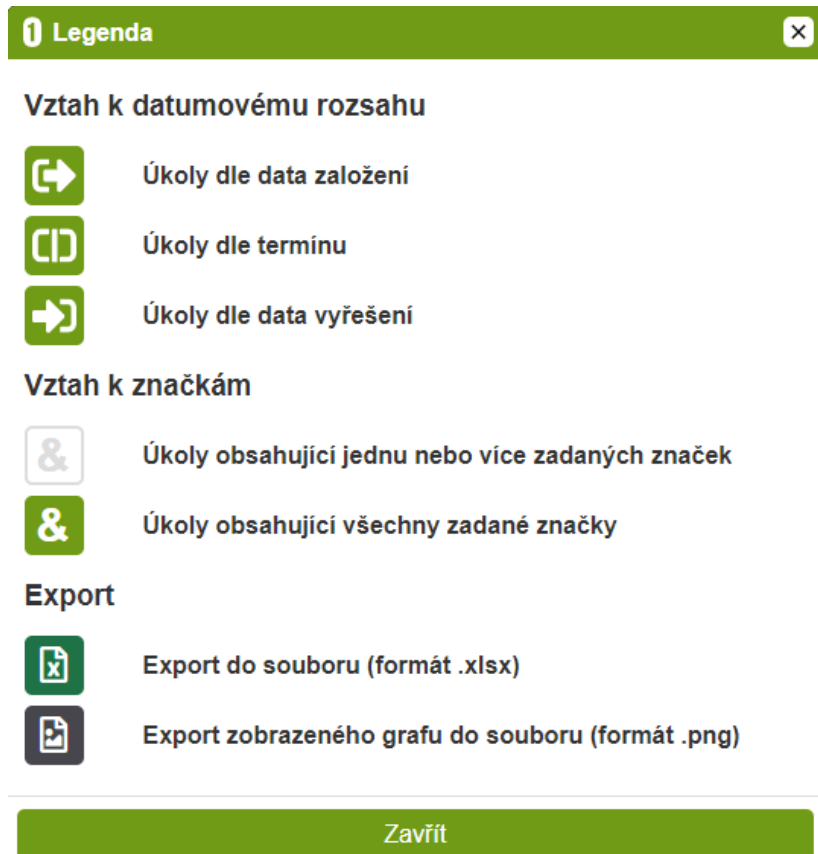
Obrázek 23 – Správa šablon pro přednastavení filtru: Uložení

6.1.2 Dialogy

Jak již bylo zmíněno, blok vyhledávacího filtru ještě zahrnuje funkční prvky pro zobrazení tří dialogů, jak bude vysvětleno v následujících pododdílech.

6.1.2.1 Legenda

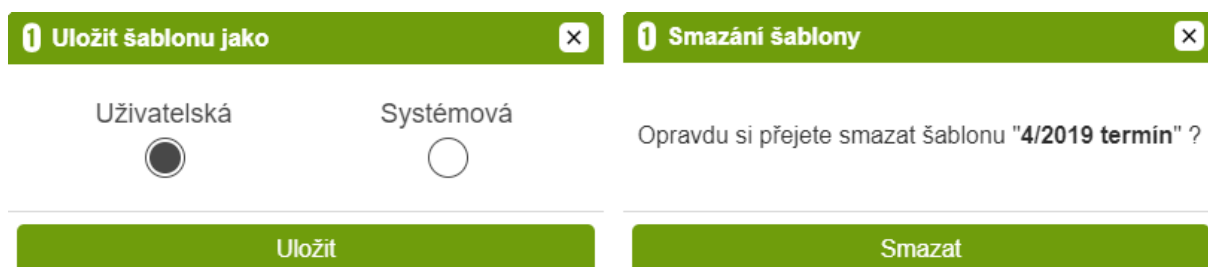
Prvním ze zmíněných dialogů je informativní legenda popisující použité ikony a jejich funkci. Tyto ikony se dají rozčlenit do tří skupin: ikony pro přepínač typu datového rozsahu, ikony pro přepínač operátoru úkolových značek a ikony pro export. Vysvětlení jednotlivých ikon je viditelné na Obrázku 24.



Obrázek 24 – Legenda: Ikony používané v modulu

6.1.2.2 Potvrzovací dialogy pro správu šablon

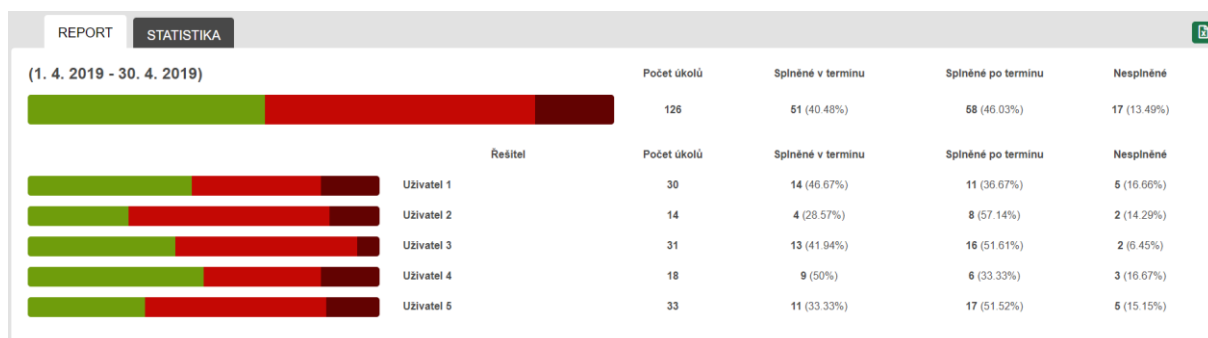
Poslední dva dialogy týkající se bloku vyhledávacího filtru, resp. přímo správy šablon, jsou zobrazeny na Obrázku 25. Jedná se o potvrzovací dialogy ukládání, resp. mazání šablon. Na prvním z nich je vidět možnost volby druhu šablony, která má být uložena. Pokud nemá uživatel dostatečná oprávnění ke správě systémových šablon, je zvolení této varianty vypnuto a uživatel může uložit šablonu pouze jako uživatelskou. Druhý dialog slouží jen jako „ochrana proti neúmyslnému smazání“, jinými slovy potvrzovací dialog pro smazání dané šablony, jejíž název je tučně znázorněný v textu dialogu.



Obrázek 25 – Dialogy ukládání a mazání šablon filtru

6.2 Report

Druhou podstatnou částí celého klientského GUI je samotný report. Ten se zobrazí v záložce *REPORT* až po nastavení vyhledávacího filtru a stisknutí tlačítka *Vyhledat*. Pro report (Obrázek 26) byla použita pouze dvě povinná kritéria, a to parametr *Od* nastavený na datum 1. 4. 2019 a parametr *Do* nastavený na 30. 4. 2019. Souhrnný report zobrazuje definované datumové rozmezí, sumarizované počty úkolů pro celou firmu a rozdělení počtu úkolů mezi jednotlivé uživatele. Jednotlivé úkoly jsou podle stavu rozřazeny do kategorií splnění úkolů: splněné v termínu, splněné po termínu, nesplněné a jejich součet ve sloupci *Počet úkolů*. Každá kategorie plnění úkolů neobsahuje pouze četnost těchto úkolů, ale také procentuální zastoupení této četnosti vůči danému celku. Tato procentuální hodnota je pak využita pro přehledné grafické zobrazení reprezentované barevným „teploměrem“ udávajícím na první pohled poměr

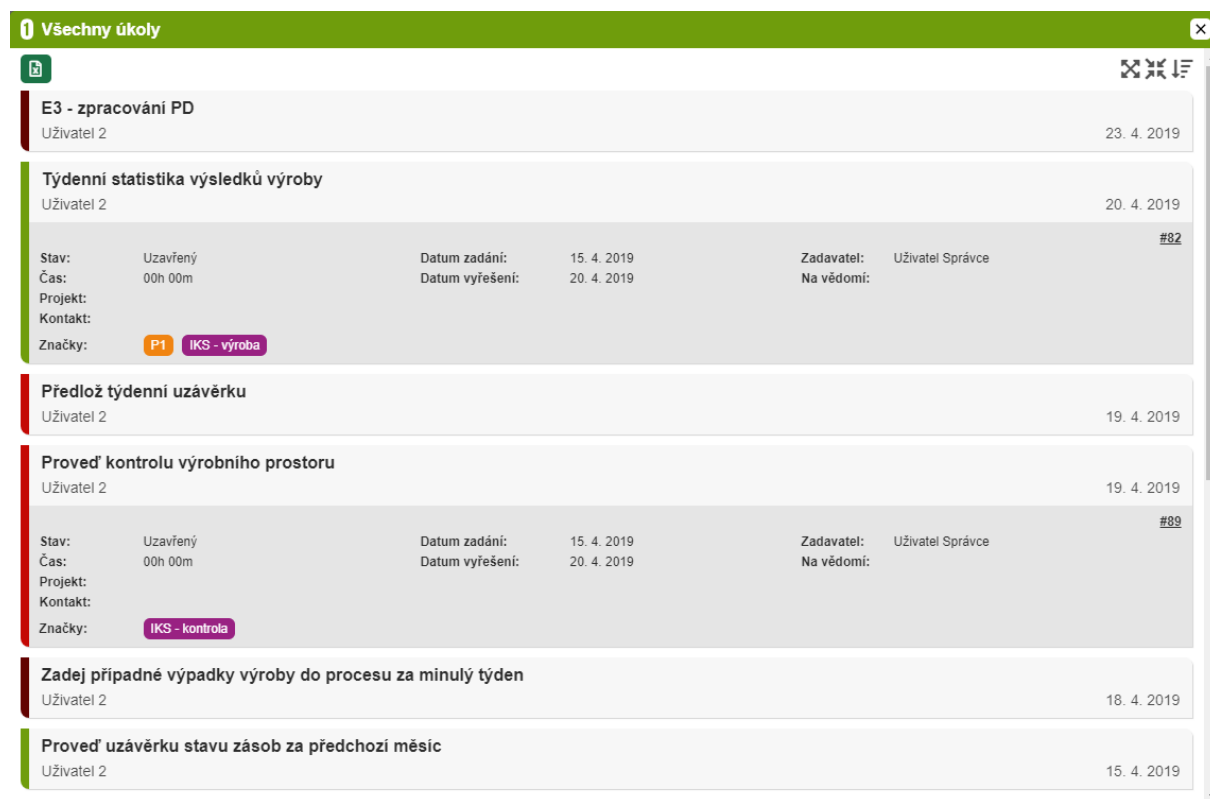


Obrázek 26 – Souhrnný report

plnění úkolů (zelená – splněné v termínu, červená – splněné po termínu, tmavě červená – nesplněné). Tento souhrnný report je možné vyexportovat do souboru XLSX neboli souboru pro Microsoft Excel, (viz Podkapitola 6.4), pomocí zelené ikony exportu v pravém horním rohu tohoto reportu. Na každou hodnotu v jednotlivých kategoriích je také možno kliknout myší pro otevření dialogu obsahujícího seznam detailně popsanych úkolů spadajících právě do této kategorie a hodnoty, (viz následující oddíl).

6.2.1 Detail seznamu úkolů

Jak již bylo zmíněno v předchozí podkapitole, po kliknutí na libovolnou hodnotu z kategorií plnění úkolů se otevře dialog obsahující detailní seznam úkolů spadajících pod tuto hodnotu a kategorii, (viz Obrázek 27). Při prvotním otevření dialogu jsou jednotlivé úkoly minimalizovány, tzn. je zobrazen pouze předmět, řešitel a termín úkolu. Každý úkol je z levé části ohraničen barevným pruhem odrážejícím stav vyřešení tohoto úkolu. Po rozkliknutí daného úkolu se zobrazí doplňkové informace: stav, datum zadání, zadavatel, strávený čas, datum vyřešení, seznam uživatelů na vědomí, projekt, kontakt, úkolové značky a ID úkolu, skrze které je možné přejít na daný úkol přímo do systému 1CLICK pro zobrazení diskuze a dalších dodatečných informací. Tento seznam úkolů je též možné ovládat třemi tlačítky, která jsou umístěna v pravém horním rohu dialogu. Ty slouží k hromadné maximalizaci

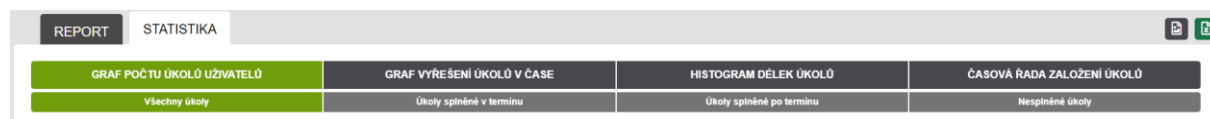


Obrázek 27 – Dialog s detailním seznamem úkolů

a minimalizaci doplňkových informací jednotlivých úkolů a pro seřazení daných úkolů dle data termínu sestupně, resp. vzestupně. I tento seznam úkolů je možné vyexportovat do souboru XLSX, (viz Podkapitola 6.4), pomocí zelené ikony exportu v levé horní části dialogu.

6.3 Statistika

Poslední podstatnou částí celého GUI je záložka *STATISTIKA*. Záložka *STATISTIKA* obsahuje navigační menu pro přepínání pohledů mezi statistickými grafy (Obrázek 28) a samotné zobrazení statistických grafů, (viz Oddíl 6.3.1). Při přepnutí na záložku statistik je doplněno do pravé horní části šedé tlačítko pro uložení právě zobrazeného statistického grafu do souboru PNG. Toto tlačítko je umístěno vedle zeleného tlačítka exportu do souboru XLSX. Navigační menu této záložky je rozdělené na dvě úrovně. První a hlavní tmavě šedá část, obsahující: *GRAF POČTU ÚKOLŮ UŽIVATELŮ*, *GRAF VYŘEŠENÍ ÚKOLŮ V ČASE*, *HISTOGRAM DÉLEK ÚKOLŮ* a *ČASOVÁ ŘADA ZALOŽENÍ ÚKOLŮ*, přepíná mezi jednotlivými typy grafů. Druhá světle šedá část menu je zobrazena pouze při přepnutí hlavního menu na *GRAF POČTU ÚKOLŮ UŽIVATELŮ* a mění zobrazení sloupcového grafu v rámci zvolené konkrétní kategorie plnění úkolů, tj. *Všechny úkoly*, *Úkoly splněné v termínu*, *Úkoly splněné po termínu* a *Nesplněné úkoly*.



Obrázek 28 – Navigační menu pro přepínání pohledů mezi statistickými grafy

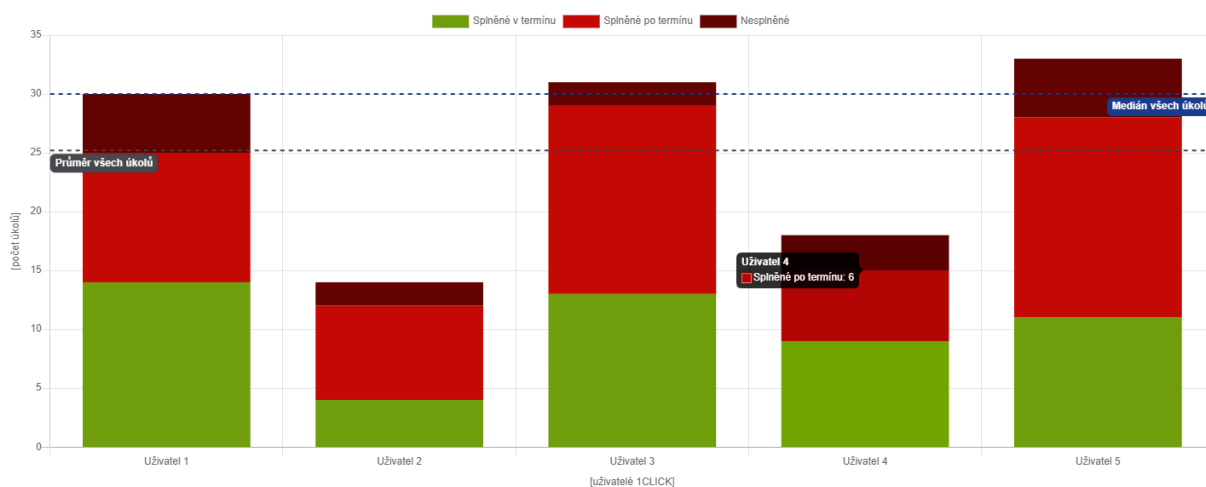
6.3.1 Grafy

Klientské GUI modulu Reporty – Statistiky obsahuje pět druhů grafů (skládaný sloupcový, sloupcový, bodový, histogram a spojnicový s lineárním trendem). První dva typy ještě zahrnují zobrazení hladiny aritmetického průměru a mediánu. Veškeré vykreslené grafy jsou interaktivní a po najetí myši na prvek grafu je zobrazen tzv. tooltip (bublinová nápověda) s detailnějšími informacemi o daném elementu, převážně název a hodnota (osa x a osa y).

6.3.1.1 Graf počtu úkolů uživatelů

Tento graf se v klientské části modulu vyskytuje ve dvou variantách. V prvním případě, (viz Obrázek 29), se jedná o skládaný sloupcový graf, který na ose x zobrazuje uživatele systému 1CLICK a na ose y jsou barevně odlišeny, stejně jako u reportu, počty úkolů jednotlivých skupin plnění úkolů tvořící jeden souhrnný celek (počet všech úkolů daného uživatele). V druhé variantě se jedná pouze o klasický jednoduchý sloupcový graf znázorňující jen jednu z vybraných kategorií plnění úkolů. V obou variantách grafu jsou vyznačeny hladiny

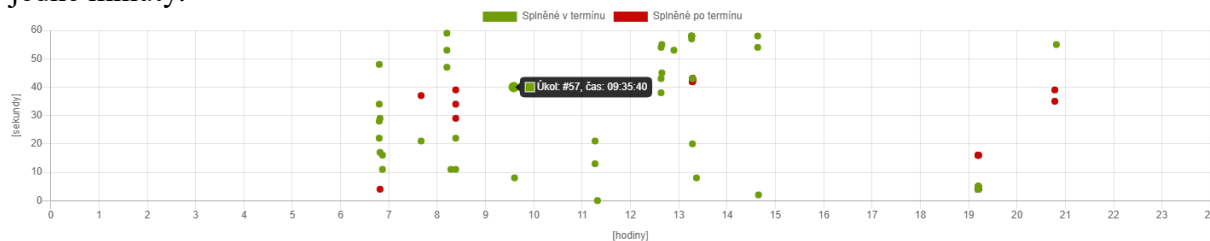
aritmetického průměru a mediánu daných úkolů, které slouží pro porovnání uživatelů vzhledem k těmto statistickým charakteristikám. Jelikož je aritmetický průměr ovlivňován extrémními hodnotami, je zde použita hodnota mediánu, aby výsledné vizuální porovnání uživatelů nebylo zavádějící a zkreslené právě díky těmto hodnotám.



Obrázek 29 – Graf počtu všech úkolů uživatelů systému 1CLICK

6.3.1.2 Graf vyřešení úkolů v čase

Jak již název napovídá, Graf vyřešení úkolů v čase, znázorňuje rozptyl odbavení úkolů v čase pro zjištění, v jaké části dne uživatelé nejčastěji potvrzují vyřešení úkolů. Jedná se o bodový graf (Obrázek 30), ve kterém je každý úkol reprezentován barevným bodem. Barva jednotlivých bodů udává stav vyřešení úkolu v termínu (zelená barva), či po termínu (červená barva). Pro větší vypovídající schopnost grafu, resp. menší překrývání jednotlivých bodů, je graf koncipován tak, že osa x znázorňuje rozsah hodin a minut jednoho dne a osa y rozsah sekund jedné minuty.

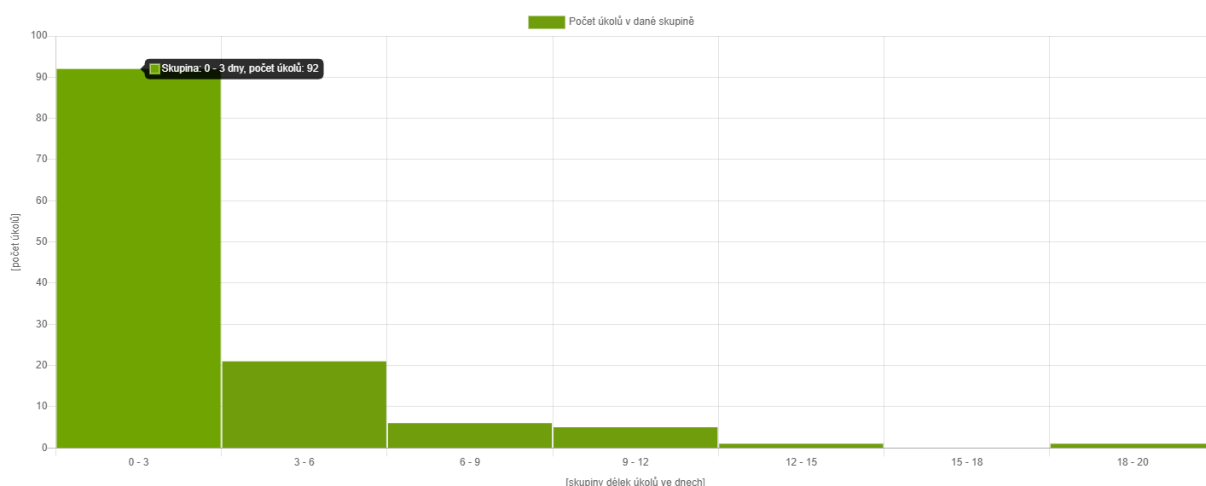


Obrázek 30 – Graf vyřešení úkolů v čase

6.3.1.3 Histogram délek trvání úkolů

Na tomto grafu, resp. histogramu, jsou znázorněny četnosti úkolů spadající do jednotlivých tříd délek úkolů. Délkou úkolu je myšlena délka ve dnech mezi datem založení a termínem úkolu. Tímto grafem lze sledovat, na jak dlouhé období jsou úkoly ve firmě zakládány, zda je firemní politika založena spíše na vytváření krátkodobých, střednědobých či dlouhodobých aktivit.

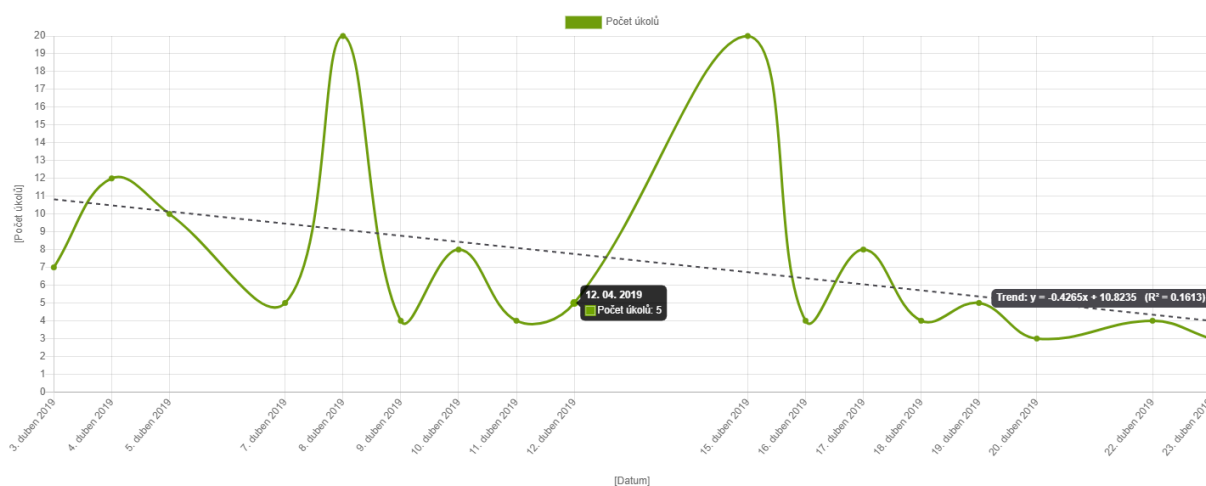
Na Obrázku 31 je patrné, že sledovaná společnost generuje převážně krátkodobé úkoly (0 až 3 dny). Počet a velikost tříd jsou dány množstvím a rozmanitostí vstupních dat, (viz Podkapitola 3.2 týkající se tvorby histogramu).



Obrázek 31 – Histogram délek trvání úkolů

6.3.1.4 Časová řada založení/vyřešení/termínu úkolů a lineární trend

Tento spojnicový graf znázorňuje časovou řadu počtu úkolů ve vztahu k jejich založení (Obrázek 32), vyřešení nebo termínu. To je definováno volbou druhu vztahu k datumovému rozsahu z vyhledávacího filtru zmíněného v Podkapitole 6.1. Na ose x jsou vynesena data dnů, ve kterých došlo k minimálně jednomu výskytu úkolu dle sledované veličiny. Osa y znázorňuje počty těchto úkolů. Nedílnou součástí této časové řady je i křivka lineárního trendu určující dlouhodobý vývoj této veličiny. Součástí křivky je i rovnice tohoto lineárního trendu a hodnota spolehlivosti. Rovnice dle kladného nebo záporného znaménka udává, zda je vývoj rostoucí nebo klesající. Hodnota spolehlivosti je reprezentována číselným rozsahem od 0 do 1. Čím více se blíží tato hodnota k číslu 1, tím je odhad tohoto dlouhodobého vývoje spolehlivější.



Obrázek 32 – Časová řada počtu založených úkolů

6.4 Export do souboru XLSX

Oba níže uvedené druhy exportů obsahují ve vyexportovaném souboru první list s názvem *Nastavení filtru*, který v sobě nese informace o nastavených parametrech filtru, při kterém byl export vykonán. Název vyexportovaného souboru je ve formátu:

```
1CLICK_task_deadline_report_<datum_od>_<datum_do>_<typ_reportu>.xlsx
```

6.4.1 Export souhrnného reportu a statistiky

Dalším listem ve vyexportovaném souboru v tomto druhu exportu je list *Data*. Ten obsahuje data ze souhrnného reportu vložené do sloupců: *Uživatel*, *Počet úkolů*, *Splněné v termínu*, *Splněné v termínu (%)*, *Splněné po termínu*, *Splněné po termínu (%)*, *Nesplněné* a *Nesplněné (%)*. Dalším listem je *Průměr*, *Medián* obsahující v horní části hodnoty aritmetického průměru a mediánu pro všechny kategorie plnění úkolů a hned poté následuje výčet četností úkolů pro jednotlivé uživatele v rámci daných kategorií. Následující list má název *Vyřešení úkolů v čase*, který obsahuje tři sloupce: *ID úkolu*, *Typ* a *Čas*, a reflektuje záznamy z grafu vyřešení úkolů v čase (viz Obrázek 30). Předposledním listem je *Histogram délek úkolů*, který, jak již název napovídá, seskupuje data potřebná pro sestavení histogramu, tedy názvy tříd (sloupec *Délka úkolů (dny)*) a četnosti úkolů (sloupec *Počet úkolů*). Vyexportovaný soubor uzavírá list *Časová řada úkolů*, který na prvním řádku obsahuje rovnici lineárního trendu i s hodnotou spolehlivosti, a na dalších již všechna data časové řady ve sloupcích *Datum* a *Počet úkolů*.

6.4.2 Export seznamu úkolů

Soubor z tohoto druhu exportu na rozdíl od předešlého obsahuje pouze list *Data*, na kterém jsou postupně vypsané dané úkoly ze zvoleného seznamu kategorie splnění úkolů seřazených dle data termínu sestupně. List obsahuje sloupce *Status*, *Termín*, *ID*, *Předmět*, *Řešitel*, *Zadavatel*, *Stav*, *Datum zadání*, *Datum vyřešení*, *Čas*, *Značky*, *Projekt*, *Kontakt* a *Na vědomí*.

7 FUNKČNÍ ČÁST DOPLŇKU

Serverová část modulu Reporty – Statistiky je vytvořena v jazyce Java s využitím vývojového prostředí Apache NetBeans IDE 10.0 od společnosti Apache Software Foundation. Klientská část tohoto modulu je, za pomoci vývojového prostředí PhpStorm 2019.1 od společnosti JetBrains, vytvořena převážně v jazyce JavaScript s využitím knihoven jQuery a Chart.js.

7.1 Popis balíčků a tříd serverové části doplňku

Tato podkapitola se bude zabývat popisem projektu tohoto modulu, tzn. jednotlivými balíčky, jejich třídami a zaměří se na některé významné a zajímavé metody. Celý projekt a veškeré zdrojové kódy jsou k dispozici v Příloze A.

Samotný projekt je klasická Java SE aplikace používající Apache Maven (nástroj pro správu, řízení a automatizaci sestavení aplikací). Následující oddíly se zabývají balíčky pouze z projektového adresáře *src/main/java*.

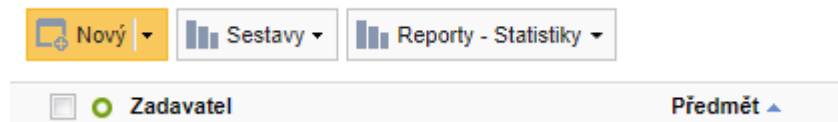
7.1.1 Balíček main

Tento “kořenový” balíček (projektový adresář *src/main/java*) obsahuje třídy potřebné k napojení tohoto modulu na systém 1CLICK a pomocnou třídu *ReportPaths* sloužící k sestavování adres URI pro webové akce a zdrojové adresáře ze zadaných konstant a metody pro jejich získání.

7.1.1.1 Třída *StatisticsReportModule*

StatisticsReportModule je nejdůležitější třídou celého modulu. Zajišťuje samotné napojení na systém 1CLICK a natažení a šíření závislostí na třídy jádra systému. Obsahuje čtyři metody: *doStart*, *registerWebActions*, *registerWebResources* a *registerEventListeners*. Metoda *doStart* je volána při spuštění modulu, resp. při spuštění či restartu systému 1CLICK, a slouží k natažení závislostí tříd jádra systému, vytvoření instancí příslušných tříd a k zavolání registračních metod. První z těchto registračních metod je *registerWebActions*, která registruje všechny nově vytvořené REST API webové akce potřebné ke komunikaci s daným modulem. Další registrační metodou je *registerWebResources*. Ta zajišťuje přidání rozšiřujících zdrojových souborů pro systém 1CLICK, v tomto případě souborů pro přidání tlačítka *Reporty – Statistiky* do okna úkolů přímo v systému (Obrázek 33), které slouží pro zobrazení webového klientského

rozhraní pro tento modul. Poslední registrační metodou této třídy je *registerEventListeners* zajišťující v tomto případě přidání nového typu oprávnění do systémových ACL.



Obrázek 33 – Přidané tlačítko *Reporty – Statistiky* v systému 1CLICK

7.1.2 Balíček *acl*

Balíček *acl* obsahuje pouze jedinou třídu *StatisticsReportAccessRight*, která vytváří novou položku oprávnění pro správu systémových šablon. Tato položka je při spuštění modulu registrována do zbylých systémových ACL pomocí metody *registerEventListeners* ve třídě *StatisticsReportModule*.

7.1.3 Balíček *export*

Tento balíček obsahuje třídy, které se přímo týkají tvorby exportovaného souboru XLSX obsahujícího data z reportu a statistických grafů či seznamu podrobných informací o úkolech.

Výčtový typ *Type* obsahující hodnoty *ALL*, *SOLVED*, *SOLVED_LATE*, *LATE* určuje, o jaký druh kategorie plnění úkolů se jedná.

Třída *XLSFilterData* dědí ze třídy *TaskFilterData*, z balíčku *report*, a rozšiřuje jí o název uživatele, výčtový typ *Type* a název šablony.

7.1.3.1 Třída *XLSExporter*

Třída slouží k samotnému sestavení souboru XLSX. Obsahuje dvě veřejné metody *getXLS* a *getFileName*. Metoda *getXLS* ze vstupních dat (seznam rozříděných úkolů dle kategorií a uživatelů zapouzdřených ve třídě *UserHolder*) a typu exportu (informace obsažené ve třídě *XLSFilterData*) volá příslušné metody pro vygenerování vybrané varianty souboru a ten zapisuje do výstupního datového proudu. Ukázka jedné z privátních metod generující souhrnný report a data ze statistických grafů (privátní metody volané v této metodě generují jednotlivé listy souboru XLSX):

```
private void generateSummaryXLS(List<UserHolder> collection, XLSFilterData
filterData, OutputStream os) throws IOException {
    try (Workbook workbook = new XSSFWorkbook()) {
        createFilterPage(workbook, filterData);
        createDataSummaryPage(workbook, collection);
        createStatistics(workbook, filterData, collection);
        workbook.write(os);
    }
}
```


Druhá veřejná metoda *getFileName* navrácí textový řetězec obsahující výsledný název souboru dle zvolené varianty exportu, který je složen z přednastavených konstant a údajů z filtru.

7.1.4 Balíček filter

Jediná třída *FilterCreator* obsažená v tomto balíčku vytváří data pro položky vyhledávacího filtru z aktuálního stavu systému 1CLICK. Třída poskytuje jednu veřejnou metodu *getFilter*, která vrací interně implementovanou třídu *Entity* podobnou formátu JSON obsahující seznam uživatelů, úkolových značek, projektů, kontaktů a informace o právech ke správě systémových šablon. Přidání seznamů do výstupní třídy *Entity* je docíleno pomocí neveřejných metod: *addUsers*, *addTags*, *addProjects*, *addContacts* a *addTemplateAccess*. Ukázka metody přidání seznamu kontaktů seřazených dle názvu s využitím datových proudů z jazyka Java 8:

```
private void addContacts(Entity entity) {
    entity.set("contacts", ArrayType.ARRAY_OF_STRINGS,
        contactSearcher.getBriefContacts(token, new Entity())
            .getItems()
            .stream()
            .map(contactEntity -> contactEntity.getString("display_name"))
            .sorted()
            .collect(Collectors.toList()));
}
```

7.1.5 Balíček report

Balíček *report* zahrnuje veškeré třídy a výčtové typy, které se zabývají tvorbou reportu. Jedná se tedy o natažení dat ze systému 1CLICK, jejich transformaci a filtrování do podoby, ve které se používají v modulu, a vytvoření výstupní *Entity* z těchto dat pro odpověď na požadavek klienta.

Prvním výčtovým typem je *DateType* s hodnotami *CREATE*, *RESOLVE* a *DEADLINE*. Tento výčtový typ určuje druh vztahu vůči zadanému datumovému rozsahu. Druhým výčtovým typem je *TaskState* určující stav úkolu, tzn. do které kategorie plnění úkolů tento úkol spadá. *TaskState* nabývá hodnot *SOLVED*, *SOLVED_LATE*, *LATE* a *DEFAULT*, které slouží jako příznak pro úkoly, jež nejsou do reportu zařazeny, např. úkoly bez termínu nebo úkoly, kde řešitel (uživatel) je ve stavu *Pasivní* (nemůže se přihlásit do systému, ale lze s ním v systému pracovat) nebo *Neaktivní* (je úplně zakázán).

Dále se v balíčku nachází třída *Tag* nesoucí informace o úkolových značkách, resp. o názvu značky, a její kód barvy RGB. Další třídou je *TaskFilterData* reflektující nastavené parametry z vyhledávacího filtru, tj. datumový rozsah od – do, typ vztahu k datumovému rozsahu (*DateType*), seznam uživatelů, seznam úkolových značek, typ operátoru pro značky, seznam

projektů a seznam kontaktů. Následující třídou v balíčku je *UserHolder*. Tato třída v sobě zapouzdřuje název uživatele a tři seznamy kategorií plnění úkolů (*Task*). Obsahuje veřejné metody pro získání těchto atributů, a navíc metody *getAllTaskCount* (vrací celkový počet úkolů daného uživatele), *addTask* (přidává úkol dle jeho stavu do správného seznamu), *getAllTasks* (vrací seznam všech úkolů uživatele nezávisle na kategorii). *UserHolderCounter* je pomocná třída sloužící k jednoduššímu sčítání počtu úkolů daných kategorií. Obsahuje veřejnou metodu *increase*, která jako vstupní parametr přijímá *UserHolder* a přičítá velikosti seznamů do připravených atributů.

7.1.5.1 Třída Task

Další třídou je *Task*, která transformuje podobu úkolu ze systému 1CLICK do potřebné podoby tohoto modulu. Oproti systémové verzi úkolu je rozšířena o *TaskState* a obsahuje transformované atributy id, systémový stav, stav plnění termínu (*TaskState*), předmět, datum založení/vyřešení/termínu, řešitel, zadavatel, strávený čas, seznam uživatelů na vědomí, značky (seznam tříd *Tag*), projekt a kontakt. Nejzajímavější privátní metodou této třídy je *parseTaskState*, která definuje bez ohledu na čas, do které kategorie plnění úkolů bude tento úkol spadat dle vstupních parametrů data termínu a data vyřešení:

```
private TaskState parseTaskState(Date deadline, Date resolved){
    if (deadline != null) {
        if (resolved != null) {
            if (timeIgnoringComparator.compare(resolved, deadline) <= 0){
                return TaskState.SOLVED;
            } else if (timeIgnoringComparator.compare(resolved, deadline) > 0){
                return TaskState.SOLVED_LATE;
            }
        } else {
            if (timeIgnoringComparator.compare(new Date(), deadline) > 0){
                return TaskState.LATE;
            }
        }
    }
    return TaskState.DEFAULT;
}
```

7.1.5.2 Třída TaskSummary

Toto je nejdůležitější třída celého balíčku. Třída zajišťuje natažení dat o úkolech ze systému 1CLICK, jejich filtraci dle nastaveného vyhledávacího filtru, vytvoření a naplnění instancí *UserHolder* a následné vytvoření výstupní třídy *Entity* pro odpověď na příchozí klientský požadavek. Obsahuje tři veřejné metody *get*, *getCollection*, *getTasks*. Metoda *get* vrací výslednou sestavnou třídu *Entity*. Metoda *getCollection* navrácí seznam *UserHolder* (úkoly roztříděné dle uživatelů a do správných kategorií). A *getTasks* vrácí seznam *Task* (úkolů), které

jsou již očištěné o neplatné úkoly a jsou vyfiltrovány dle parametrů z vyhledávacího filtru. Mezi zajímavější neveřejné metody této třídy patří: *createUserHolders* (vytvoření seznamu *UserHolder* ze vstupního seznamu úkolů), *applyAllFilters* (aplikování kritérií vyhledávacího filtru na vstupní seznam úkolů) a *getTasksFromRange* (vytažení dat úkolů ze systému v zadaném datovém rozsahu).

7.1.6 Balíček statistics

Tento balíček obsahuje třídy zabývající se přípravou dat pro statistické grafy a třídu *StatisticsSummary*, která tato data sdružuje ve veřejné metodě *getStatisticsEntity* do souhrnné třídy *Entity* pro odeslání do klientské části.

7.1.6.1 Třída TimeLinesData

Tato třída připravuje data pro vykreslení časové řady založení/vyřešení/termínu úkolů. Využívá třídu *SimpleRegression* z již zmíněné knihovny Apache Commons Mathematics Library 3, s jejíž pomocí vypočítává rovnici lineárního trendu a hodnotu spolehlivosti. Primární metodou je *createDatasetFromTasks*, která transformuje vstupní seznam úkolů (*Task*) do kolekce četností úkolů k jednotlivým datům ze zvoleného rozsahu. Dále jsou zde pomocné privátní metody upravující tuto kolekci tak, aby byla použitelná pro výpočet lineárního trendu pomocí třídy *SimpleRegression*, která používá poněkud zastaralé dvoudimenzionální pole. Třída obsahuje také metody pro konverzi výsledných zpracovaných hodnot do výstupní třídy *Entity*: *createTimeLinesDataEntity*, *createTaskCountsDataEntity* a *createTrendEntity*.

7.1.6.2 Třída MeanMedianData

Další třídou vytvářející data pro statistické grafy je *MeanMedianData*. Třída zpracovává vstupní kolekci *UserHolder* a pomocí třídy *DescriptiveStatistics* z knihovny Apache Commons Mathematics Library 3 vypočítává aritmetický průměr (metoda *getMean*) a medián (*getPercentile(50)*) z těchto dat. Výsledné zpracování je vráceno zpět v podobě třídy *Entity* ve veřejné metodě *createMeanMedianDataEntity*.

7.1.6.3 Třída HistogramData

Třída *HistogramData* vytváří ze vstupního seznamu úkolů (*Task*) data pro vytvoření histogramu, tzn. výpočet maxima a minima, variačního rozpětí, počtu tříd, velikosti tříd a přiřazení četností hodnot do příslušných tříd. Obsahuje dvě stěžejní metody *getLenghtOfTasks*, která ze seznamu úkolů vytváří seznam jejich délek (datum vytvoření – datum termínu) a metodu *generateDistributionMap* vytvářející samotnou kolekci histogramu, tzn. tříd a jejich četností. Třída využívá z knihovny Apache Commons Mathematics Library 3

třidu *Frequency*, s jejíž pomocí lze jednoduchým způsobem získat četnost hledané hodnoty v určité kolekci hodnot, což je využito právě pro určení četností daných tříd. Ukázka části metody *generateDistributionMap* vytvářející četnostní třídy, tzn. výpočet maxima a minima (proměnné *minValue* a *maxValue*), variační rozpětí (proměnná *R*), počet tříd (proměnná *K*), velikost tříd (proměnná *H*) a spodní a horní hranice tříd (proměnné *lowerBoundary* a *upperBoundary*):

```
Integer minValue = datasetList.stream().findFirst().get();
Integer maxValue = datasetList.stream().skip(
    datasetList.size() - 1).findFirst().get();
Integer R = maxValue - minValue;
Integer K = (int) Math.ceil(1 + 3.3 * Math.log10(datasetList.size()));
Integer H = (int) Math.round(R.doubleValue() / K.doubleValue());
H = (H < 1) ? 1 : H;
for (int i = 0; i < K; i++) {
    Integer lowerBoundary = minValue + (H * i);
    Integer upperBoundary = (minValue.equals(maxValue))
        ? minValue + (H * i)
        : minValue + (H * (i + 1));
    if (upperBoundary >= maxValue) {
        upperBoundary = maxValue;
    }
    if (i == (K - 1) && (lowerBoundary <= maxValue)) {
        upperBoundary = maxValue;
    }
    if (lowerBoundary <= maxValue && upperBoundary <= maxValue) {
        distributionMap.put(
            new ImmutablePair(lowerBoundary, upperBoundary),
            0L);
    }
}
```

7.1.6.4 Třída *TimeTaskSolveData*

Třída *TimeTaskSolveData* vytváří ze vstupního seznamu úkolů potřebná data k vytvoření bodového rozptylového grafu zobrazujícího čas vyřešení jednotlivých úkolů. Třída obsahuje tři privátní metody pro zpracování času: *parseTimeFromDate* (získání celého časového formátu z data), *parseHoursFromDate* (hodnota hodin pro osu x), *parseSecondsFromDate* (hodnota sekund pro osu y). Všechna výsledná zpracovaná data jsou znovu převedena do výstupní třídy *Entity*.

7.1.7 Balíček *templates*

Balíček *templates* obsahuje pouze jedinou třídu *TemplateStore*, která zajišťuje správu uživatelských a systémových šablon pro přednastavení filtru. Tato třída má tři veřejné metody: *addTemplate* (přidání nové systémové nebo uživatelské šablony), *deleteTemplate* (smazání existující šablony) a *getTemplates* (získání seznamu systémových a uživatelských šablon pro daného uživatele).

7.1.8 Balíček *utils*

Balíček *utils* obsahuje užitečné a pomocné třídy. Třída *TimeIgnoringComparator* je upravená varianta *Comparator<Date>* s tím rozdílem, že při porovnávání dat nebere v potaz časovou složku. Třída *TimeUtils* obsahuje statickou veřejnou metodu *removeTimeFromDate*, která vrací datum v milisekundách očištěný od časové složky. Dále balíček obsahuje třídu *BaseTemplateCreator*. Ta má na starost zveřejnění a zpřístupnění souboru *index.html* (základní/hlavní soubor klientské části) a všech ostatních zdrojových souborů, skriptů, obrázků apod. z určeného adresáře systému 1CLICK a tím zamezuje přístup do jiných adresářů, než je určeno. Poslední třídou balíčku je *TemplateResourcesCreator*, která slouží k nahrání zdrojových souborů klienta z modulu do určeného adresáře systému 1CLICK. Toto nahrání má dvě fáze: smazání předchozí verze souborů (metoda *deletePreviousVersion*) a nahrání nových souborů ze zdrojů modulu do adresáře systému (metoda *copyResourcesToTemplateDir*). Třída je zavolána vždy při startu modulu, resp. startu/restartu systému a tím je zajištěna aktuálnost těchto souborů, tzn. že při nechtěném vymazání některých souborů z datové složky systému stačí systém restartovat a soubory se nahrají zpět.

7.1.9 Balíček *web*

Tento balíček obsahuje třídy/obslužné rutiny pro všechny registrované webové akce tohoto modulu. Třídy *AddTemplateAction* (přidání nové šablony), *DeleteTemplateAction* (smazání existující šablony) a *GetTemplates* (získání seznamů šablon uživatele) volají příslušné metody ze třídy *TemplateStore*. Třída *ExportXLSAction* obsluhuje požadavek pro export dat do souboru. *GetFilter* volá metodu ze třídy *FilterCreator* a vrací připravená data pro sestavení vyhledávacího filtru. Třídy *GetTasksSummary* a *GetStatisticsData* odesílají zpracovaná data pro report, resp. statistické grafy. A třídy *QueryAction* a *ResourceAction* zpracovávají požadavek na soubor *index.html* či jiné zdroje a pomocí třídy *BaseTemplateCreator* je odesílají do webového prohlížeče.

7.2 Rozšiřující REST API modulu

Jak již bylo zmíněno v předchozím oddíle, modul Reporty – Statistiky rozšiřuje systém 1CLICK o devět nových akcí API:

- GET `../api/custom/report/tasks/templates` – seznam šablon uživatele,
- POST `../api/custom/report/tasks/templates/add` – přidání nové šablony,

- POST ../api/custom/report/tasks/templates/delete – smazání existující šablony,
- GET ../api/custom/report/tasks/filter – data pro sestavení vyhledávacího filtru,
- POST ../api/custom/report/tasks/summary – data pro report,
- POST ../api/custom/report/tasks/statistics – data pro statistické grafy,
- POST ../api/custom/report/tasks/export/xls – export souboru XLSX,
- GET ../report/tasks – soubor *index.html*,
- GET ../report/tasks/(.+) – další soubory z adresáře *ICLICK/data/report/tasks*.

7.3 Volání REST API z klientské části modulu

Veškeré klientské funkce volající REST API modulu jsou umístěny v souboru *API_calls.js* a využívají objektové rozhraní *XMLHttpRequest*, které je poskytováno javascriptovým prostředím webových prohlížečů a slouží pro zjednodušení komunikace mezi klientem a serverem, převážně u aplikací založených na technologii AJAX. Zde je ukázka použití *XMLHttpRequest* pro získání dat pro report a sestavení reportu pomocí funkce *createPage*, pokud server úspěšně vrátí připravená data:

```
var request = new XMLHttpRequest();
request.open("POST", "../api/custom/report/tasks/summary");
request.setRequestHeader("Content-Type", "application/json");
request.responseType = "json";
request.onload = function () {
    if (request.readyState === request.DONE) {
        if (request.status === 200) {
            createPage(request.response);
        }
    }
};
request.send(JSON.stringify(
    {
        date_from: date_from,
        date_to: date_to,
        projects: projects,
        users: users,
        contacts: contacts,
        tags: tags,
        tagAND: tagAND,
        dateType: dateType
    }
));
```

V horní části ukázky je vidět zvolená metoda HTTP (*POST*), adresa URI pro danou webovou akci, formát požadavku a odpovědi (*JSON*). Následuje definice funkce, která bude zavolána při

úspěšné odpovědi na požadavek a samotné odeslání požadavku s potřebnými daty (parametry vyhledávacího filtru).

7.4 Průběh komunikace mezi uživatelem, klientem a serverem

Chronologický průběh komunikace mezi uživatelem, klientem a serverem je následující:

1. Uživatel spustí klientské GUI modulu pomocí tlačítka v systému 1CLICK směřující na adresu *../report/tasks*.
2. Server odešle do webového prohlížeče soubor *index.html* a další potřebné soubory.
3. Klient odešle požadavek na data pro vyhledávací filtr a po odpovědi serveru je filtr s těmito daty zobrazen a připraven k použití.
4. Uživatel navolí parametry filtru, či je načte z uložených šablon a zmáčkne tlačítko *Vyhledat*.
5. Klient odešle požadavek na data pro report a statistické grafy, které jsou následně vykresleny a zobrazeny uživateli.
6. Uživatel si následně zobrazená data nebo grafy vyexportuje do souboru (další požadavek na server a vyexportovaný soubor je rovnou webovým prohlížečem stažen do počítače uživatele), pouze je zobrazí nebo zvolí jiné parametry vyhledávacího filtru a pokračuje se znovu od bodu 5.

Pro načtení aktualizovaných dat do položek vyhledávacího filtru, např. po přidání nového uživatele do systému, je potřeba načíst klientské GUI znovu (obnovení webového prohlížeče).

7.5 Dynamické zobrazení a generování klientské části

Soubor *index.html* obsahuje jen základní strukturu webové stránky a veškeré prvky reportu, statistických grafů a dialogů jsou přidávány dynamicky za pomoci knihovny jQuery. To znamená, že zdrojové soubory jsou ze serveru nataženy pouze při prvotním načtení klienta a veškeré úpravy a změny vizuální podoby klientského GUI jsou již dynamicky měněny ve webovém prohlížeči bez nutnosti další komunikace se serverem. Vše je tedy řízeno příchozími daty z jednotlivých volání serverového API a úpravy modelu DOM jsou prováděny lokálně na klientském počítači.

7.5.1 Popis zdrojových souborů pro webové GUI

- Soubor *index.html* – základní kostra stránky HTML a napojení CSS a skriptů.
- Adresář *fonts* – fonty používané v rámci GUI.

- Adresář *images* – veškeré obrázky a ikony obsažené v GUI.
- Adresář *styles* – kaskádové styly použité v GUI.
 - Soubor *style.css* – kaskádové styly pro GUI.
 - Soubor *dialog.css* – kaskádové styly pro dialogová okna.
 - Soubor *multiple-select.css* – kaskádové styly pro multi-výběrové položky filtru.
- Adresář *js* – obsahuje soubory JS.
 - Soubor *API_calls.js* – odesílání požadavků na serverové API.
 - Soubor *content.js* – vytvoření a zobrazení elementů pro report a statistické grafy.
 - Soubor *dialog.js* – vytvoření informačních a potvrzovacích dialogů.
 - Soubor *dialog_content.js* – vytvoření obsahu pro detailní seznam úkolů.
 - Soubor *filter.js* – vytvoření a logika vyhledávacího filtru.
 - Soubor *graphs.js* – vytvoření samotných statistických grafů.
 - Soubor *chartjs-plugin-annotation.min.js* – doplněk pro knihovnu Chart.js.
 - Soubor *main.js* – základní nastavení stránky, registrování událostí apod.
 - Soubor *multiple-select.js* – logika multi-výběrových položek vyhledávacího filtru.

8 TESTOVÁNÍ A VALIDACE

Samotný vývoj modulu Reporty – Statistiky probíhal na lokální instanci systému 1CLICK verze 3.1.8.1 a po vytvoření finální verze byl modul nahrán na vývojový server společnosti One Click Business Solution s. r. o. pro otestování na větším a rozmanitějším množství dat.

Na této testovací instanci bylo vytvořeno šest uživatelů: Správce, Uživatel 1, Uživatel 2, Uživatel 3, Uživatel 4, Uživatel 5, šest kontaktů (Kontakt 1 – Kontakt 6), jeden projekt a dvanáct úkolových značek. Během jednoho měsíce byly zakládány periodické úkoly pro jednotlivé řešitele s různým nastavením úkolových značek, kontaktů apod. V průběhu tohoto období zaměstnanci společnosti One Click Business Solution s. r. o. tyto úkoly náhodně odbavovali a testovali projevené změny v reportu či statistických grafech. Za celý měsíc bylo vytvořeno a odbaveno okolo 200 úkolů.

Během uživatelského testování na vývojovém serveru nebyly nalezeny žádné problémy serverové funkcionality, spíše jen drobné textové překlepy v klientském GUI, či špatné zobrazování některých elementů tohoto rozhraní, například při delších názvech úkolů apod., což bylo v průběhu vývoje a testování odstraněno.

Pro ověření správné funkcionality s větším množstvím dat byl modul nasazen na „ostrou“ instanci společnosti One Click Business Solution s. r. o. běžící na reálném komerčním serveru této společnosti. Při tomto testování byly zjištěny určité nedostatky, a to převážně v oblasti získávání dat ze systému 1CLICK. Například operace pro načítání dat do vyhledávacího filtru trvala nepřiměřeně dlouho, jelikož na této instanci existuje mnoho projektů a kontaktů (v řádech tisíců). Musely být tedy zvoleny a použity jiné třídy a přístupy k těmto datům, či dokonce upraveno indexování některých položek v databázi a opraveny metody přímo v jádru systému 1CLICK.

9 ZÁVĚR

Hlavním cílem této diplomové práce bylo vytvoření doplňkového modulu Statistiky pro systém 1CLICK, který má sloužit střednímu a vrcholovému managementu jako nástroj pro zvyšování efektivity práce jednotlivců, týmů i celé firmy či společnosti.

Pro tvorbu této práce autor využíval principy projektového managementu, jelikož proces tvorby tohoto modulu byl pojat jako projekt. V rámci tohoto projektu byl navržen časový harmonogram a byly určeny zásadní milníky. Projekt byl rozdělen na čtyři etapy, a to na analýzu, zpracování funkční části doplňku, návrh a zpracování grafického rozhraní, a testování výsledného řešení, přičemž zpracování funkční části doplňku a návrh a zpracování grafického rozhraní bylo prováděno v paralelním vývoji.

V rámci analytické části byly konzultovány požadavky a představy vedení a zaměstnanců společnosti One Click Business Solutions s. r. o. týkající se na funkcionality a výsledné podoby modulu, které byly následně transformovány do konkrétní podoby analýzy zpracované dle unifikovaného procesu vývoje aplikací.

Pro samotný vývoj modulu Statistiky bylo využito několika odlišných přístupů směřovaných k využití a práci se systémem 1CLICK. V první fázi byl modul vyvíjen za použití lokální instalace systému 1CLICK v rámci jedné počítačové stanice. Po ověření správnosti řešení na lokální úrovni byl modul nasazen na speciální instanci vývojového a testovacího serveru společnosti One Click Business Solutions s. r. o., která obsahovala uměle vytvořené testovací vzorky dat pro ověření správné funkce reportních a statistických výstupů. Aby však bylo možno ukončit vývoj modulu, bylo nutno provést testování na datech z ostrého provozu systému 1CLICK obsahujících reálnou kombinaci a množství dat potřebných pro učinění závěru odpovídající funkcionality modulu dle zadání a požadavku zadavatele. Z tohoto důvodu autor požádal vedení společnosti One Click Business Solutions s. r. o. o možnost nasazení modulu Statistiky na ostrou instanci společnosti, které bylo vyhověno.

V rámci testování modulu na ostrých datech byla však zjištěna nevyhovující délka latence při načítání dat, a to zejména v oblasti práce s projektovou částí systému 1CLICK, která měla za následek potřebu razantních úprav jak samotného modulu, tak i jádra systému 1CLICK. Po implementaci úprav bylo chování modulu Statistiky uvedeno do plně vyhovujících mezí odpovídajících zadání.

Modul Statistiky jako doplňkový modul systému 1CLICK je určen pouze pro jednu z konkrétních problematik systému 1CLICK, a to úkolovou část, a je jen jednou z mnoha možností využití reportů a statistik v rámci celého systému. Autor doporučuje společnosti One Click Business Solutions s. r. o. navázat na tuto práci a rozšířit reportní a statistickou část systému zejména na oblast procesního řízení z důvodu možnosti tvorby reportů a statistik nad nepřehlednou škálou dat pořizovaných v rámci workflow svých zákazníků.

POUŽITÁ LITERATURA

- [1] What Is ERP?. *ORACLE* [online]. [cit. 2019-04-23]. Dostupné z: <https://www.oracle.com/applications/erp/what-is-erp.html>
- [2] ROUSE, Margaret. What is content management system (CMS)?. *TechTarget* [online]. June 2016 [cit. 2019-04-23]. Dostupné z: <https://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS>
- [3] C. UZIALKO, Adam. Document Management Systems. *Business News Daily* [online]. February 20, 2019 [cit. 2019-04-24]. Dostupné z: <https://www.businessnewsdaily.com/8026-choosing-a-document-management-system.html>
- [4] Customer Relationship Management. *Bain & Company* [online]. April 02, 2018 [cit. 2019-04-24]. Dostupné z: <https://www.bain.com/insights/management-tools-customer-relationship-management>
- [5] ROUSE, Margaret. What is human resource management (HRM)?. *TechTarget* [online]. November 2017 [cit. 2019-04-24]. Dostupné z: <https://searchhrsoftware.techtarget.com/definition/human-resource-management-HRM>
- [6] Product Data Management / Product Lifecycle Management. *NPD Solutions* [online]. [cit. 2019-04-24]. Dostupné z: <http://www.npd-solutions.com/pdm.html>
- [7] What is PLM?. *PLM Technology Guide* [online]. [cit. 2019-04-24]. Dostupné z: http://plmtechnologyguide.com/site/?page_id=435
- [8] SCRC, SME. What is Supply Chain Management (SCM)?. *NC State University* [online]. April 2, 2017 [cit. 2019-04-24]. Dostupné z: <https://scm.ncsu.edu/scm-articles/article/what-is-supply-chain-management-scm>
- [9] 1CLICK Software na řízení firmy. *1c2c.cz* [online]. [cit. 2019-04-12]. Dostupné z: <https://www.1c2c.cz/>
- [10] Schéma procesu "Faktura přijatá". In: *1CLICK WIKI* [online]. [cit. 2019-04-13]. Dostupné z: <https://d33v4339jhl8k0.cloudfront.net/docs/assets/58da5eceddd8c8e5c5730e991/images/5b3c93ab2c7d3a099f2e2a40/file-rQZBi-Low5D.png>

- [11] SOUČEK, Eduard. *Statistika pro ekonomy*. Praha: Vysoká škola ekonomie a managementu, 2006. ISBN 80-867-3006-9.
- [12] NEUBAUER, Jiří, Marek SEDLAČÍK a Oldřich KŘÍŽ. *Základy statistiky: aplikace v technických a ekonomických oborech*. Praha: Grada, 2012. ISBN 978-80-247-4273-1.
- [13] NEUBAUER, Jiří. *Charakteristiky variability: Rozdělení lišící se variabilitou*, str. 29 [online]. [cit. 2019-04-21]. Dostupné z: https://k101.unob.cz/~neubauer/pdf/ci-selne_charakteristiky.pdf
- [14] Dekompozice časové řady. In: *Matematická biologie učebnice: Dekompozice časových řad* [online]. [cit. 2019-04-22]. Dostupné z: http://portal.matematickabiologie.cz/res/image/linearni_zpracovani_dat/c02.png
- [15] EVANS, Benjamin J. *Java in a Nutshell*. Sixth edition. Beijing, [2015]. ISBN 978-1-449-37082-4.
- [16] Java 8 Overview. *Tutorials Point* [online]. [cit. 2019-04-14]. Dostupné z: https://www.tutorialspoint.com/java8/java8_overview.htm
- [17] Guava: Google Core Libraries for Java. *GitHub* [online]. 7. 3. 2019 [cit. 2019-04-16]. Dostupné z: <https://github.com/google/guava>
- [18] Commons Math: The Apache Commons Mathematics Library. *Apache Commons* [online]. 28. 7. 2016 [cit. 2019-04-16]. Dostupné z: <https://commons.apache.org/proper/commons-math/>
- [19] Apache POI - the Java API for Microsoft Documents. *Apache Software Foundation* [online]. 4. 9. 2019 [cit. 2019-04-17]. Dostupné z: <https://poi.apache.org/index.html>
- [20] Apache POI Tutorial. *Tutorials Point* [online]. [cit. 2019-04-17]. Dostupné z: https://www.tutorialspoint.com/apache_poi/index.htm
- [21] JavaScript - Overview. *Tutorials Point* [online]. [cit. 2019-04-20]. Dostupné z: https://www.tutorialspoint.com/javascript/javascript_overview.htm
- [22] FLANAGAN, David. *JavaScript: The Definitive Guide*. Fifth edition. Sepastopol, CA: O'Reilly, [2006]. ISBN 978-0-596-10199-2.
- [23] *JQuery* [online]. [cit. 2019-04-18]. Dostupné z: <https://jquery.com/>

- [24] CHAFFER, Jonathan. *Learning jQuery: better interaction, design, and web development with simple JavaScript techniques*. 4th ed. Birmingham: Pack Publishing, [2013]. ISBN 978-1-78216-314-5.
- [25] *Chart.js* [online]. [cit. 2019-04-17]. Dostupné z: <https://www.chartjs.org/>
- [26] MASSÉ, Mark. *REST API Design Rulebook*. Sebastopol, CA: O'Reilly, [2012]. ISBN 978-1-449-31050-9.
- [27] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

SEZNAM PŘÍLOH

Příloha A – <i>Přiložené CD</i>	80
---------------------------------------	----

Příloha A – *Přiložené CD*

Obsahující:

- text práce (.pdf),
- analýza zpracovaná v programu Enterprise Architect (.eap),
- Java projekt daného modulu bez napojení na systém 1CLICK (.zip).