

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## REPORTING - ERP SYSTÉM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MILAN PÁLA

BRNO 2013



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# REPORTING - ERP SYSTÉM

REPORTING - ERP SYSTEM

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. MILAN PÁLA**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. PAVEL OČENÁŠEK, Ph.D.**

BRNO 2013

## **Abstrakt**

Práce se zabývá tvorbou modulu pro ERP systém. Modul by měl umět produkovat průběžná data o výrobě, hlídat produktivnost výroby a upozorňovat na problémy ve výrobě. V práci je zhodnoceno zpracovávání velkého množství dat a ukázán způsob předpočítávání dat. Zabývá se také návrhem na predikci údajů ze známých dat.

## **Abstract**

This work deals with creating a module for existing ERP system. Module should be able to produce data progress of production, monitor productivity of production and warn if some issue will happen. This work evaluates a processing of a large amount of data and it shows different possibilities how to precalculate data. It also deals with a draft how to predict information from known data.

## **Klíčová slova**

erp, produktivnost, statistiky, rozhodovací stromy, regrese, kritická cesta, php, mysql

## **Keywords**

erp, productivity, statistics, decision tree, regression, php, mysql, critical path

## **Citace**

Milan Pála: Reporting - ERP systém, diplomová práce, Brno, FIT VUT v Brně, 2013

# Reporting - ERP systém

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doktora Očenáška

.....

Milan Pála  
21. května 2013

## Poděkování

Chtěl bych poděkovat panu doktoru Očenáškovvi za vedení práce a panu inženýru Strachovi z firmy Agerit s.r.o. za poskytnutá data.

© Milan Pála, 2013.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*



# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Plánování procesů a činností</b>	<b>4</b>
2.1 Operační výzkum	4
2.1.1 Síťová analýza	5
2.1.2 Metoda kritické cesty	5
2.2 Korelační analýza	6
2.3 Závěr	6
<b>3 Analýza požadavků</b>	<b>7</b>
3.1 Popis cílového prostředí	7
3.2 Popis požadavků od zadavatele	7
3.3 Architektura aplikace	8
3.4 Analýza metod hledajících zpoždění	9
3.4.1 Metoda využívající dosavadní průměr	9
3.4.2 Metoda rozhodovacího stromu	9
3.4.3 Lineární regrese	10
3.4.4 Genetické algoritmy	10
3.4.5 Grafické znázornění	10
3.5 Poskytnutá data	10
3.5.1 Komentář k datům	12
3.6 Popis komerčních ERP systémů	12
3.6.1 Firma Asseco Solutions, a.s.	13
3.6.2 KARAT Software a. s.	13
3.7 Závěr	14
<b>4 Specifikace</b>	<b>15</b>
4.1 Architektura aplikace	15
4.2 Požadavky na statistiky výroby	15
4.3 Požadavky na sledování produktivnosti	16
4.4 Hlídní pracovní doby	16
4.5 Požadavky na sledování zpoždění zakázek	17
4.6 Závěr	17
<b>5 Návrh aplikace</b>	<b>18</b>
5.1 Architektura aplikace	18
5.2 Struktura aplikace	18
5.3 Datová vrstva	19

5.4 Akce systému . . . . .	21
5.5 Závěr . . . . .	21
<b>6 Implementace</b>	<b>23</b>
6.1 Architektura aplikace . . . . .	23
6.2 Navigace v aplikaci . . . . .	23
6.3 Dodatky k implementaci . . . . .	26
6.3.1 Názvy tabulek . . . . .	26
6.3.2 Zjednodušení formátování doby trvání . . . . .	26
6.3.3 Nastavení aplikace . . . . .	26
6.4 Způsob ignorování záznamů . . . . .	26
6.5 Implementované pohledy . . . . .	27
6.5.1 Pohled: Všechny zakázky . . . . .	27
6.5.2 Pohled: Detail zakázky . . . . .	28
6.5.3 Pohled: Produktivnost zakázek . . . . .	29
6.5.4 Pohled: Trvání operací . . . . .	30
6.5.5 Pohled: Zpožděné zakázky . . . . .	32
6.5.6 Pohled: Pomalí pracovníci . . . . .	33
6.6 Závěr . . . . .	34
<b>7 Možnosti rozšíření</b>	<b>35</b>
<b>8 Závěr</b>	<b>36</b>
<b>A Obsah CD</b>	<b>38</b>

# Kapitola 1

## Úvod

Tato práce se zabývá analýzou požadavků na doplnění výrobního ERP systému. Konkrétně modulu, který by z dat nabízel vedoucímu pracovníkovi průběžná data a výrobě a produktivitě. Systém tak bude vhodně doplňovat a kompletovat řešení, které firma nabízí svým zákazníkům.

Bude nutné analyzovat způsoby, jakým budou informace z databáze získávány, protože zpracovávat data z výroby znamená pracovat se značně objemnými databázemi. Požadavky pracující v reálném čase můžou při špatném návrhu schématu databáze trvat značnou dobu. Nabízí se některé informace předpočítávat, pokud by to ušetřilo následný výpočetní čas a nepřineslo příliš velkou paměťovou zátěž.

Na základě historických dat je mnohdy možné predikovat některé údaje, což může mít přínos pro uživatele, neboť taková predikce nabízí reálný pohled založený na již proběhlých zakázkách. Nabízí se predikovat předpokládané termíny dokončení zakázek. Systém opatřený takovými vlastnostmi dokáže včas upozornit na špatné rozhodnutí a navrhnout lepší řešení.

## Kapitola 2

# Plánování procesů a činností

### 2.1 Operační výzkum

Přibližně od 40. let 20. století se rozvíjela vědecká disciplína s názvem *operační výzkum*. Nejdříve se metod operačního výzkumu používalo k řešení úloh souvisejících s vojenskými operacemi. V současnosti představují tyto metody soubor nástrojů pro řízení složitých ekonomických systémů. Výhodou je zkvalitnění ekonomického rozhodování, především ve smyslu rychlosti a kvalifikovanosti použitím kvantitativních metod. Jako neoddělitelná součást se prosazuje matematické modelování, zejména tzv. ekonomicko matematické modely.

Kvalitativní analýza problému je východiskem při formulaci tzv. ekonomického modelu. V této první etapě je nutné především vymezit předmět zkoumání, určit nejdůležitější faktory, definovat a klasifikovat hypotézy, proměnné a úkoly. Je nutné stanovit hodnotící kritérium pro posouzení kvality fungování studovaného systému.

Konstrukce matematického modelu spočívá v transformaci ekonomického modelu pomocí matematických výrazových prostředků.

Po sestavení matematického modelu je možné spolu se vstupními daty přejít k řešení úlohy. Po vyhodnocení by měla nastoupit fáze verifikace výsledků, která spočívá v porovnání teoretických hodnot získaného optimálního řešení s dosavadní situací a v porovnání rozdílů vypočtené a skutečné úrovně kritéria kvality fungování systému. Nakonec je možné provést vlastní implementaci metody, tedy realizace výsledků v praxi.

Uplatnění modelů operačního výzkumu jako nástrojů optimálního rozhodování a řízení v relativně stálých ekonomických systémech a procesech je nejčastěji v těchto oblastech:

- dlouhodobé výhledy a koncepce,
- střednědobé a krátkodobé projekty,
- operativní řízení výroby,
- přímé řízení výrobních a technologických procesů.

Pro strategické úlohy je charakteristické, že jejich řešení zahrnuje dlouhý časový horizont. Jsou zaměřeny na celý systém. Problémy řešené v kratších projektech mají charakter taktických dílčích úloh. Jedná se o stanovení cílů pro jeho dílčí subsystémy.

V oblasti operativního řízení výroby je cílem urychlit a zpřesnit pomocí optimalizačního modelu vydávání, kontrolu a evidenci bezprostředních pracovních rozhodnutí. Jde např. o metody operativního řízení subsystémů, reagování na změny okolí systému apod.

Tyto problémy většinou nelze řešit exaktními postupy a proto se používají s osvědčenými heuristickými postupy.

S rozvojem výkonné výpočetní techniky je možné nasazovat metody operačního výzkumu do přímého řízení výrobních procesů. Snahou je dosáhnout optimalizace některých důležitých paramterů.

Možnosti použití operačního výzkumu v řízení rozhodování jsou rozsáhlé. V současné době se metody operačního výzkumu staly nedílnou součástí tzv. systémů pro podporu rozhodování. [14, s. 7–15]

### 2.1.1 Síťová analýza

Mezi jednu z disciplín operačního výzkumu patří *síťová analýza*. Metody síťové analýzy jsou založeny na teorii grafů a teorii pravděpodobnosti. Využívá se jí pro plánování, koordinaci a kontrolu složitých úkolů v oblastech hospodářské činnosti, např. v investiční výstavbě, technickém rozvoji apod. [14, s. 17]

Metody síťové analýzy umožňují stanovit minimální čas potřebný pro realizaci celého projektu a vytipovat činnosti, na jejichž průběhu závisí dodržení termínu stanoveného pro realizaci celého projektu.

K přednostem disciplíny patří například:

- umožňují graficky znázornit průběh akce,
- umožňují propočítat dílčí i konečné termíny jednotlivých operací,
- umožňují stanovit činnosti limitující průběh realizace projektu.

Dílčí změny nejsou vždy podmínkou pro zpoždění realizace projektu a nevyžadují nové přeplánování. K nejznámějším metodám síťové analýzy patří: *metoda kritické cesty* (CPM, Critical Path Method), *technika hodnocení a rozboru programu* (PERT, Program Evaluation and Review Technique) a další. [9, s. 163]

### 2.1.2 Metoda kritické cesty

Metoda CPM umožňuje určit časovou náročnost celého projektu, termíny zahájení a dokončení dílčích operací projektu, posloupnost operací, které jsou pro projekt limitující. Důsledná analýza umožňuje v některých případech i zkrátit dobu realizace projektu. Metoda využívá operace na orientovaném hranově ohodnoceném acyklickém grafu, který musí být souvislý. Takový graf má zároveň alespoň jeden uzel vstupní. Pokud obsahuje právě jeden takový uzel a zároveň jeden uzel koncový, nazývá se takový graf síťový.

Při praktické aplikaci metody se uplatňuje následující postup:

- Proveďte se naplánování postupu jednotlivých částí projektu pomocí síťového grafu.
- Propočte se časová náročnost projektu i dílčích částí.
- Určí se časové rezervy činností.
- Nalezne se kritická cesta a proveďte se její analýza.

Hrana grafu představuje činnost a je ohodnocena časem, po který daná činnost probíhá. Uzel je okamžik zahájení, nebo ukončení a má nulovou dobu trvání.

Pro tvorbu grafu je nutné znát všechny činnosti, které jsou nezbytné pro provedení projektu. Dále posloupnost těchto činností, jejich vzájemnou návaznost a dobu trvání těchto činností. Uvedené informace lze získat dvojím způsobem: *metodou vpřed* a *metodou vzad*. Obě metody vytvoří stejný graf, ale popsán bude postup metodou vzad. Metoda vpřed je vhodnější pro projekty, které známe a opakují se. Metoda vzad je vhodnější pro projekty dosud nevyjasněné a nové.

Všechny činnosti se seřadí do tabulky, každá činnosti na jeden řádek. Pro každou činnost se určí všechny bezprostředně předcházející činnosti do druhého sloupce tabulky. Třetí sloupec tabulky poté obsahuje dobu trvání každé činnosti. Činnosti, které nemají žádné předchůdce se poté znázorní jako hrany vycházející z počátečního uzlu síťového grafu. Dále se uzly a hrany navazují podle svého pořadí, až vznikne celá síť. Všechny hrany grafu jsou orientované směrem od počátečního uzlu ke koncovému (neboli od začátku činnosti do jejího konce). [9, s. 162–212]

Při znalosti síťového grafu je možné nalézt všechny kritické činnosti a jejich spojením kritickou cestu. Pokud se změní doba nějaké činnosti, je snadné určit, zda ovlivní i zpoždění celého projektu.

## 2.2 Korelační analýza

Při posuzování, jak spolu dva atributy souvisí, lze využít korelační analýzu. Korelační analýza určí, jak silně jeden atribut implikuje druhý. Pro numerické atributy lze určit korelaci mezi atributy A a B výpočtem korelačního koeficientu nazývaného také Pearsonův koeficient. Koeficient může nabývat hodnot v intervalu  $(-1; 1)$ . Je-li hodnota kladná, jsou hodnoty obou atributů pozitivně korelované. Čím více se hodnota blíží jedné, tím více je atribut B závislý na atributu A. Pokud je hodnota rovna nule, nemají atributy spolu žádnou návaznost. A pokud je hodnota záporná, jsou atributy negativně korelované. [13, s. 48]

Z těchto hodnot lze poté pro danou aplikaci analýzy odvodit vhodné závěry, např. lze jeden atribut vyřadit z analýzy pro jeho redundantnost, nebo lze objevit souvislost v neznámých datech.

## 2.3 Závěr

V kapitole byla představena historie a současnost vědní disciplíny s názvem *operační výzkum*. Byly zmíněny její metody pro plánování procesů CPM a PERT, kdy byla detailně popsána první metoda, tzv. metoda kritické cesty.

Byla uvedena také metoda korelační analýzy. Její aplikací lze v tabulce hodnot nalézt atributy, které jsou na sobě závislé, respektive spolu nesouvisí.

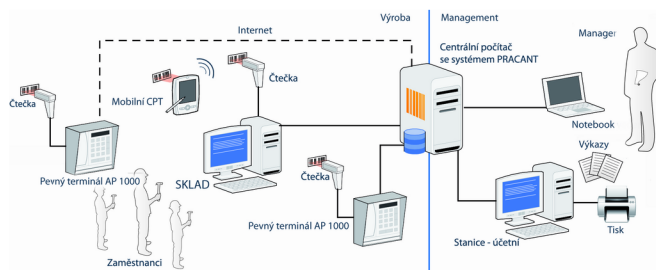
## Kapitola 3

# Analýza požadavků

### 3.1 Popis cílového prostředí

Firma Agerit s.r.o. (zadavatel práce) nasazuje do provozů výrobních podniků svůj ERP systém Pracant ukládající data z výroby. Pracovníci podniků nosí při práci u sebe osobní terminály se čtecím zařízením na čárové kódy. Před začátkem činnosti pracovník prostřednictvím terminálu naskenuje čárový kód označující výrobní zařízení, čímž zaeviduje svou pracovní aktivitu do systému. Díky tomu je k dispozici detailní přehled o čase stráveném na všech fázích výroby. Výrobní stroje jsou prostřednictvím komunikačních rozhraní napojeny na terminály, které také k systému poskytuje zadavatel práce. Pracovníkovi terminály ukazují informace o aktuálním stavu na dílně a zároveň evidují práci strojů a zaznamenávají ji k pracovní náplni zaměstnance.

Všechny informace poté vidí vedoucí pracovník na svém počítači a prostřednictvím ERP systému může nahlížet na docházku, stav skladu, stavy strojů, editovat číselníky atd.



Obrázek 3.1: Schéma systému Pracant, pramen [2]

### 3.2 Popis požadavků od zadavatele

Nyní firma plánuje systém rozšířit o modul, který by dokázal generovat průběžná data o výrobě, produktivitě a na základě daných kritérií hlídal odchylky ukazatelů výroby.

Systém by měl zobrazovat produktivnost výroby, zakázek a zaměstnanců, hlídat dodržování pracovní doby. Systém by měl hlídat časové odchylky od stanovených kritérií, eventuálně umět kritéria sám určit na základě dostupných předchozích zakázek. Pokud možno v reálném čase.

Mělo by být možné do systému vkládat nová pravidla, na základě kterých systém pracuje. Hlášené odchylky by mělo být možné označit jako povolené a na podobné chyby již

neupozorňovat.

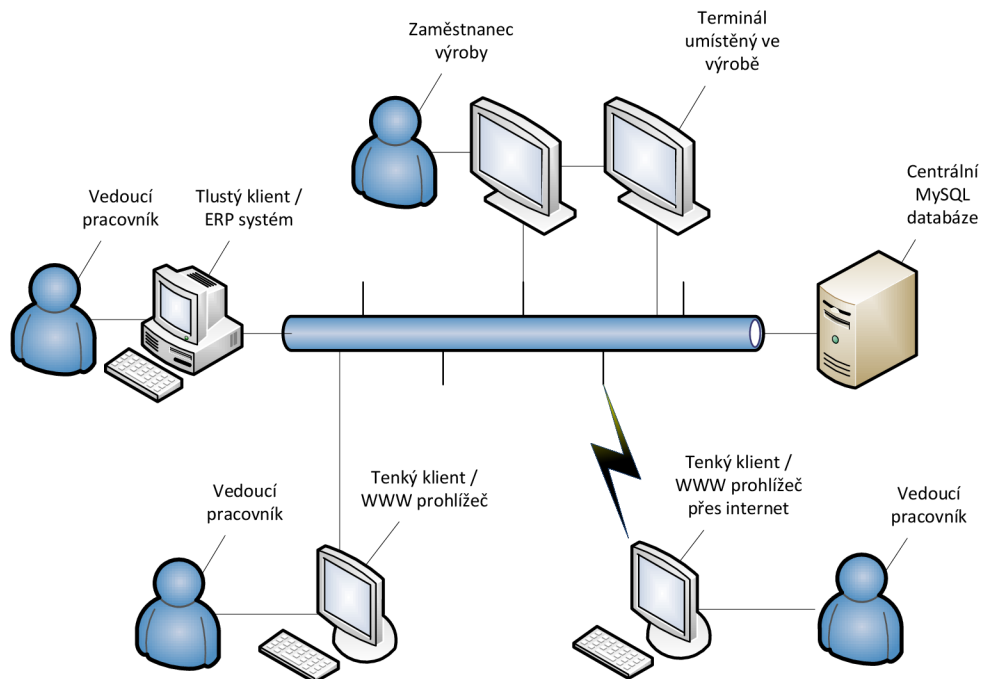
Výstupem by měl být souhrn zobrazující přehledně důležité informace, zejména informace o odchylkách. Systém by měl umožnit zobrazit detaily takovýchto informací a to pravidla, která byla porušena nebo překročena, aby bylo zřetelné, ve které části výroby nastalo zpoždění. Vyhodnocení by mělo být dostupné po zaměstnanci, zakázce, konkrétní operaci. Odchytky by měly být sledovány pro různá časová období (den, měsíc, apod.).

Se systémem bude pracovat vedoucí pracovník výroby.

### 3.3 Architektura aplikace

Dodavatel vyžaduje webovou aplikaci, kterou by mohl spouštět v internetovém prohlížeči. Z ukávek jiných systémů od klienta bylo patrné, že tento způsob fungování informačních systémů preferuje mimo jiné i kvůli možnosti snadného vzdáleného přístupu, která je pro takovéto aplikace typická. Zároveň je možné snadno systém nasadit na téměř jakémkoli počítači ve firmě bez nutnosti instalace na koncový počítač. Z toho plyne i ušetřený čas při zprovožňování systému a s tím související finanční náklady.

Celkový pohled na schéma stávajícího systému včetně implementovaného rozšíření je znázorněn na obrázku 3.2. K centrálnímu serveru a stávajícímu ERP systému přibyl implementovaný systém, který bude dostupný skrze WWW prohlížeč. Implementovaný systém je na obrázku znázorněn ve spodní polovině.



Obrázek 3.2: Schéma architektury systému



## 3.4 Analýza metod hledajících zpoždění

Určení správného termínu vyhotovení projektu (respektive zakázky) představuje důležitý krok. S úspěchem je možné využít metody síťové analýzy zmíněné v předchozích kapitolách. Při samotné realizaci zakázky se poté musí hlídat dodržování naplánovaných termínů a včas odhalit odchylky.

Metoda, kterou provádí pracovník bez dispozice informačního systému s danou funkcionalitou k určení průběhu zakázky zpočívá v tom, že pohledem odhadne, nebo již z praxe zná běžnou dobu, kterou trvá zpracování určitého druhu zakázky a na základě své zkušenosti dále rozhodne, zda u zakázky dochází ke zpoždění. Tato ruční metoda je však nepřesná, neboť si pracovník nemusí všimnout všech nesrovnalostí, které mohou být klíčové. Zároveň může přehlédnout nějaký problém, který nastal ve výrobě a který ovlivní všechny zakázky (např. porucha nějakého stroje). Mimo jiné je tato metoda časově náročná.

Automatická kontrola by měla všechny tyto nedostatky vyřešit a uživateli předat hlášení, ze kterého by snadno rozhodl o dalším postupu.

Pro určení, zda daná zakázka bude dokončena v termínu, nebo dojde ke zpoždění, je nutné důsledně projít všechny fáze zakázky a pro každou určit zpoždění na základě jejich specifických podmínek. Pro některé fáze je možné určit zpoždění přesně na základě aktuální události (např. porucha výrobního zařízení), u jiných např. podle nedostatečných kapacit výroby odhadnout, že nejspíše dojde ke zpoždění. Pokud není možné nebo by bylo složité určit přesné zpoždění, je určitě vhodné uživatele systému upozornit, že zřejmě dojde ke zpoždění a ukázat mu předpokládané důvody proč.

Při znalosti kritické cesty, eventuálně vytvořeném síťovém grafu se dá snadno na základě aktuálních odchylek ukazatelů zjistit, zda dochází k ovlivnění celkové doby zpracování zakázky.

Použitím tohoto přístupu se odhalí možné problémy až v momentě, kdy dojde ke zpoždění ve sledovaném prostředí a tato událost je zanesena do systému.

### 3.4.1 Metoda využívající dosavadní průměr

Pro predikci, jestli bude daná úloha zakázky dokončena včas, se nabízí možnost pro jednotlivé skupiny zakázek určovat ideální průběh určený jako průměr předešlých hodnot. (Z měření by byly vyloučeny zakázky, které se opozdily, aby nebyl ideální čas ovlivněn.) V takovémto případě by se určovala odchylka aktuálně zkoumané zakázky od ideálního průměru a podle nastavené hranice by se mohlo rozhodnout, zda dochází ke zpoždění.

Při využití této metody by byly zvoleny atributy, který dělí zakázky na jednotlivé typy a průměr by se počítal pro skupinu všech zakázek dané kategorie. Předpokladem je, že každá zakázka je parametrizována svou náročností. Náročnost zakázky by se měla započítat do času zakázky, jinak by dělení pouze na druhy zakázek bylo příliš hrubé. Náročnější zakázka bývá zpravidla i lépe finančně ohodnocena a tudíž její delší doba zpracování nevádí.

Parametrizace zakázek bude nutná zřejmě pro všechny ostatní přístupy k odhadování zpoždění.

### 3.4.2 Metoda rozhodovacího stromu

Rozhodnout, zda daná zakázka bude dokončena v řádném termínu je možné na základě rozhodovacího stromu. Díky rozhodovacím stromům vytvořených z dosavadních dat je možné poměrně přesně predikovat některé údaje pro nová data. Např. se nabízí na základě zjištění,

zda dosavadní zakázky byly dokončeny v termínu, rozhodnout, zda i nová zakázka bude dokončena v předpokládaném termínu.

Vytvoření rozhodovacího stromu pro určitou množinu dat vyžaduje tzv. naučení, kdy se z trénovacích dat vytvoří rozhodovací strom. Následně je možné porovnat podle vzniklých pravidel novou položku a výsledkem bude predikovaná hodnota pro tuto položku. Vzniklý strom je možné nechat otestovat částí známých dat, která nebyla použita pro naučení. Díky tomu, že jsou pro ně známy zjišťované hodnoty a data jsou stejně kvalitní, jako trénovací data, lze určit procentuální úspěšnost predikování vytvořeným stromem.

Vstupem pro rozhodovací strom musí být data diskretizovaná a nesmí být zatížena šumem. [7, s. 86] Data v databázi se mohou vyskytovat ve dvou formách, v kategorické a kvantitativní. Jedná se o logické rozdělení, protože fyzicky mohou být i kategorická data uložena číselně a záleží, jak jsou navázána na svůj číselník. Toto znesnadňuje automatické rozdělení takových atributů. V dané doméně se jako lepší možnost jeví využít znalosti experta a ve fázi před tvorbou stromu ručně atributy kategorizovat/diskretizovat. Využití experta je vhodné i pro určení podmnožiny atributů, které se budou pro tvorbu stromu využívat.

### 3.4.3 Lineární regrese

Pro výpočet předem neznámé hodnoty je možné použít metodu lineární regrese. [13, s. 114] Očekává data jsou ve tvaru dvojic, kde první položka značí vstupní atribut a druhý výstupní atribut, při zobrazení pomocí grafu v kartézské soustavě se jedná o hodnoty na osách  $x$  a  $y$ . V případě vypracování zakázek se však nejedná o lineární závislost např. stavu zakázky na aktuálním datu. Metodu je tedy obtížné použít.

### 3.4.4 Genetické algoritmy

Pro predikci neznámých hodnot je možné využít genetické algoritmy [13, s. 113], které dokáží dobře určit předem neznámé hodnoty. V praxi existují komerční systémy, které s takovými algoritmy pracují [11, s. 50], v této práci uvažovány nebudou.

### 3.4.5 Grafické znázornění

Uživateli může být nabídnuto grafické znázornění průběhu práce na zakázkách. Vhodným způsobem je využití Ganttova diagramu. Pro jeho sestavení musí být známá posloupnost činností včetně termínů jednotlivých operací. Tento graf by měl být sestaven už od fáze plánování, jako to nabízí například komerční informační systém KARAT a jeho modul APS (Advanced Planning and Scheduling)<sup>1</sup>. Vzhledem k charakteru grafu se může stát, že při složitějších zakázkách uživatel přehledně nesoulad v návaznosti činností a je tedy vhodné doplnit toto zobrazení dodatečnou informací shrnující neshody.

## 3.5 Poskytnutá data

Od zadavatele práce byla dodána, kromě přístupových údajů k lokální kopii ERP systému Pracant, také data konkrétního výrobního podniku. Data byla zproštěna citlivých údajů a lze je dále použít jako anonymní. Celá databáze obsahovala řadu tabulek, které nebyly

<sup>1</sup>Podle dostupných materiálů od Davida Buly, Senior Account Managera firmy KARAT Software a.s.

přímo určené pro analýzu, ale byly nezbytné pro chod ERP systému. Příložený (na obrázku 3.3) a dále v textu popisovány jsou pouze tabulky nezbytně nutné pro předmět této práce.

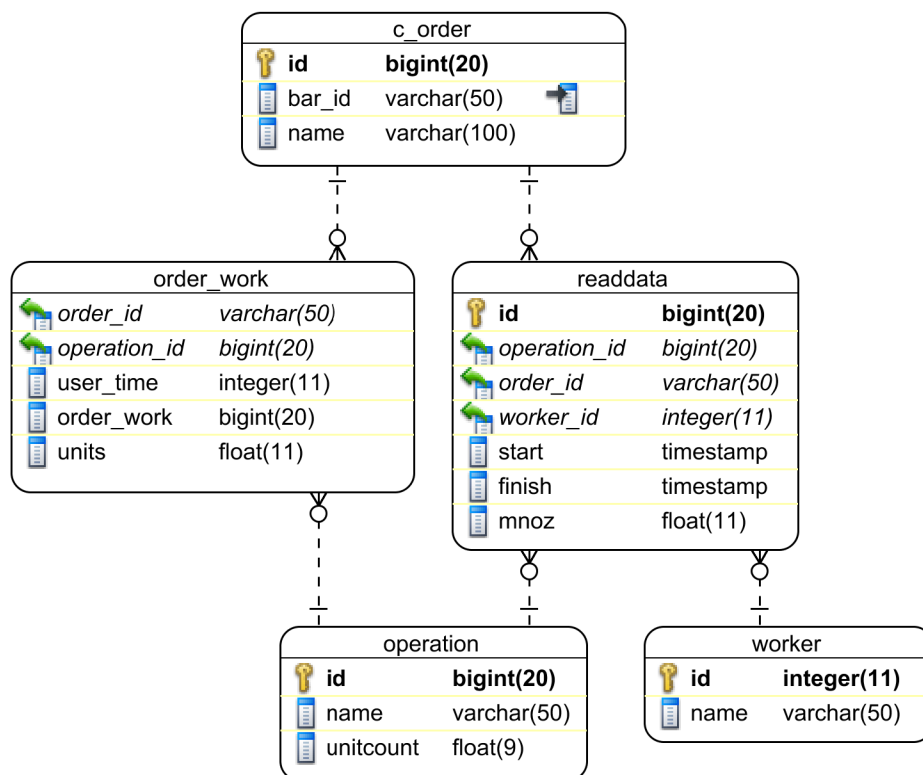
Tabulka `c_order` obsahuje přehled objednávek. Každá objednávka je identifikována atributem `bar_id`, který slouží jako cizí klíč pro přístup z ostatních tabulek. Dále obsahuje, mimo jiné, název objednávky v atributu `name`, začátek a konec práce na zakázce v atributech `start` a `finish`.

Tabulka `operation` obsahuje seznam operací, které je možné v podniku vykonávat. Kromě jiných atributů obsahuje atribut `name` nesoucí název operace. Dále počet jednotek, které daná operace zahrnuje v atributu `unitcount` a typ jednotky `unit`.

Vazební tabulkou mezi objednávkami a operacemi je tabulka `order_work`. Cizími klíči jsou `order_id` a `operation_id`. Pro každou objednávku obsahuje specifický normovaný čas v atributu `user_time` a počet jednotek, které se mají pro danou operaci vykonat (`units`). Tyto údaje jsou předpokládáné a v průběhu zakázky neměnné.

Data, která produkují zaměstnanci svou činností jsou uložena v tabulce `readdata`. Tabulka obsahuje cizí klíče `operation_id` a `order_id`, podobně jako tabulka normovaných operací. Dále identifikuje zaměstnance, který operaci provedl v klíči `worker_id`. Odpracovaná kapacita normy je uvedena v klíči `mnoz`. Zaměstnanec nemusí vždy najednou dokončit plnou normu operace, a tak dvojice klíčů není unikátní. Začátek a konec dílčího plnění uvádí atributy `start` a `finish`.

Další využitou tabulkou je tabulka `workers` nesoucí informace o zaměstnancích. V aplikaci se využívá atribut `name` se jménem zaměstnance.



Obrázek 3.3: Schéma databáze – poskytnutá databáze

### 3.5.1 Komentář k datům

Databáze nevyužívá v případě datumových atributů vestavěné hodnoty NULL pro označení nevyplněných hodnot, ale speciální hodnotu 1899-12-30. Při výpisu dat bude tato hodnota převáděna na hodnotu NULL pomocí konstrukce `IF([readdata].[finish]<>"1899-12-30", [readdata].[finish], NULL) AS [finish]`. Implementační prostředí bývají na využití hodnoty NULL připravena a umí s ní zacházet prostřednictvím standardizovaných funkcí.

Ačkoli se data v ERP systému zobrazují správně, v databázi jsou uložena zřejmě v jiném kódování, než uvedeném `latin1`, protože se některé znaky zobrazují chybně. Jako znaková sada tabulek je použito výchozí kódování `latin1 (latin1_swedish_ci)`. Kódování obsahu zřejmě v tomto kódování není, protože při nastavení tohoto kódování pro připojení do databáze se obsah taktéž deformuje. V aplikaci se tedy zobrazují texty s chybnými znaky. Implementovaný systém žádné sloupce nesoucí text nepoužívá a pokud by došlo k nápravě na straně databáze, budou se správně zobrazovat texty i v systému.<sup>2</sup>

Ne všechny tabulky databáze obsahují primární klíč, např. `order_work`. Dokonce neobsahují ani unikátní klíče, které zabraňují duplicitám a pomáhají udržovat databázi v konzistentním stavu. [12, s. 239] Konkrétně tabulka `order_work` neobsahuje žádnou možnost ověření unikátnosti záznamu. Při zpracování dodaných dat se vycházelo z předpokladu, že jsou všechny údaje v tomto směru v pořádku.

V tabulkách je rovněž využito několik různých datových typů pro určení primárního, respektive cizího klíče. Konkrétně tabulka se zakázkami `c_order` obsahuje umělý primární unikátní klíč, ale ostatní tabulky využívají cizí klíč `bar_id`, což je sloupec datového typu `VARCHAR(50)` indexovaný prostým indexem. Dále některé tabulky obsahující umělý primární klíč odvozený od celého čísla mají datový typ `INT` (celé číslo s paměťovou náročností 4 bajty a rozsahem 2,1, respektive 4,3 miliardy hodnot), jiné tabulky mají klíč založený na datovém typu `BIGINT` (paměťová náročnost 8 bajtů, rozsah oproti `INT` je 18 miliardkrát větší). [3] Pokud by byl ERP systém nasazen po dobu 25 let a obsahoval by všechna data bez archivace, tak by bylo možné do tabulky s primárním klíčem omezeným na bezznačkový celočíselný typ `INT` denně uložit přes 470 tisíc záznamů, což je více, než obsahuje nejmohutnější tabulka `readdata` s 224 tisíci záznamy celkem za několik let. Při využití typu `INT` pro všechny primární klíče je hrubý pokles datové náročnosti podle potřeby datového typu poloviční a rozsah dimenze přesto dostatečný. Zároveň by se sjednotily datové typy primárních klíčů databáze.

Názvy atributů a tabulek rovněž nepoužívají konzistentní názvosloví. Některé názvy jsou psané česky, některé anglicky. Některé názvy využívají podtržítka jako náhradu za mezeru, jiné názvy mezeru mezi slovy ignorují.

## 3.6 Popis komerčních ERP systémů

V rámci analýzy požadavků byly osloveny firmy nabízející komerční ERP programy. Dotaz byl směřován na způsob evidence a řešení případných neshod ve výrobě z pohledu řešení v jejich informačních systémech. Celkem bylo osloveno 5 firem, jejichž seznam byl získán z [1]. Odpověděli dva obchodní zástupci z níže uvedených společností.

<sup>2</sup>Byly provedeny pokusy o převod na správné kódování, ale nebyla nalezena vhodná sada.

### 3.6.1 Firma Asseco Solutions, a.s.

Firma vyrábí podnikové informační systémy HELIOS, které by měly plně pokrývat potřeby velkých, středních i malých firem, ale i dalších institucí, např. z veřejné správy. ERP systém si zakládá na komplexnosti zpracování podnikové agendy od účetnictví až po manažerské rozhodování. Součástí systémů je nabídka oborových řešení pro výrobu, obchod a služby. [5]

Přímo v systému existuje „Evidence vyřazení a typizace neshodných výrobků“. Je určena k evidenci neshodných výrobků vzniklých při výrobě. Doklady o vyřazení vzniknou automaticky podle zadaného množství vyřazených položek na kartě s evidencí odvedené práce. Podle způsobu výroby systém upozorní uživatele na nutnost zadání případných sériových čísel vadných výrobků, případně určení vad z číselníku závad. Stejně tak se určí „cíl nákladů“, neboli způsob, jakým se bude nadále s výrobkem zacházet a jaké finanční náklady to přinese. Určení vady z číselníku se v kontextu systému nazývá „Typizace neshodných výrobků“. Výrobky lze poté dohledávat ve „Skladu neshodných polotovarů“ buď podle útvaru viníka, nebo podle útvaru pracoviště operace. [6]

Jak dodal Ladislav Obrovský, Senior Account Manager ve firmě Asseco Solutions, obecně je neshoda vada přiřazená k materiálu, dílci, případně vzorku, šarži v čase k místu vzniku (středisko), viníku (zaměstnanec) a náležitým popisem příčiny a opatření. Systém HELIOS umožňuje sledovat a evidovat neshody v různých fázích logistického a výrobního procesu jako jsou příjem a uvolnění základního materiálu do výroby, neshody ve výrobě, neshody ve výstupní kontrole, ale i logistické problémy (např. nesplnění termínu dodávky). Systém umožňuje rozlišení typů neshod a jejich sledování v čase a ve výrobě. Standardní kategorie lze navíc při implementaci systému rozšířit o další uživatelské členění. K číselníku neshod lze připojovat externí soubory (např. fotografie s popisem vad nebo instrukcí pro obsluhu).

V případě pozastavení zakázky pracovníkem kvality systém umožní tuto zakázku zablokovat a nedovolí pokračování na další výrobní operaci. Pozastavená zakázka může být znovu vrácena do výrobního procesu až po provedených opatřeních a následné kontrole. Všechny činnosti včetně rozhodnutí o způsobu provedených opatření jsou dokumentovány. Realizace odvedené operace s neshodnými výrobky může být impulzem pro plánování, které zkontroluje potřebu výroby, zda použít kusy z přebytků, nebo navrhnout vystavení nového výrobního příkazu.

Během výrobního procesu může kontrola zjistit i neshodu, která může být řešena schválenou odchylkou. Systém sleduje platnost schválené odchylky a neumožní uvolnění výrobního příkazu do výroby s odchylkou po skončení její platnosti (počet kusů, datum).

Ze systému sledování odchylek lze vytvářet zprávy s vazbou na účetnictví, kde lze sledovat např. sumu nákladů na neshody, dodavatelské neshody (skryté vady atd.), neshody v korunách na pracovníka, pracoviště, nebo výrobek atd.

Ne vždy je využito vestavěných funkcí systému HELIOS, vzhledem k tomu, že podnik již využívá jiný systém na řízení jakosti, obvykle systém Palstat. [4] V tomto případě jsou oba systémy propojeny přes rozhraní pro přenos potřebných dat.

### 3.6.2 KARAT Software a. s.

Informační systém KARAT cílí na malé i velké firmy. Obsahuje několik modulů od účetnictví a personalistiku, přes logistiku, výrobu a prodej až po manažerská rozhodování. Zároveň obsahuje oborová řešení pro výrobu, obchod, služby atd.

IS KARAT obsahuje také modul Neshody. Ten umožňuje sledovat a evidovat zmetky ve výrobě, vypočítat ocenění zmetků ve skutečných nákladech, vyhodnocovat je dle různých



příčin a hledisek, jako jsou viníci, zpětná použitelnost, opravitelnost, dopad do rozpracované výroby, nákladovost, místo vzniku atd.

### **3.7 Závěr**

V této kapitole bylo popsáno cílové prostředí, ve kterém se bude implementovaný systém nacházet. Byly popsány požadavky od zadavatele práce, firmy Agerit s.r.o., tak, jak byly položeny panem Ing. Strachem. Na základě požadavků a osobní návštěvy ve firmě, kde ERP systém běží, byla popsána základní architektura aplikace.

Součástí analýzy také bylo shrnutí způsobů, pomocí kterých by bylo možné na základě dosavadních hodnot určit, zda nové zakázky budou hotovy v čas.

Byla popsána produkční databáze, která byla pro vývojové účely poskytnuta od zadavatele práce. Kromě seznámení s daty a jejich znázorněním byla databáze opatřena kritickým komentářem vzhledem k použitým datovým typům a používání klíčů.

Před začátkem specifikace požadavků proběhlo oslovení komerčních firem, které poskytují své řešení ERP systémů. Odpovědi se dostalo od dvou subjektů, konkrétně firem Asseco Solutions, a.s. a KARAT Software a. s.

# Kapitola 4

## Specifikace

K jednotlivým požadavkům bude následovat bližší rozbor s ohledem na možnosti technické realizace.

### 4.1 Architektura aplikace

Aplikace, kterou dodavatel žádá je klient-server, kdy klienta představuje webový prohlížeč na počítači uživatele systému, serverovou část poté nejčastěji skriptovací jazyk, podle preferencí klienta PHP a databáze MySQL. Tuto kombinaci technologií klient využívá i na jiných projektech. ERP systém Pracant je koncipován jako tlustý klient s databází MySQL. Podrobnosti ohledně verzí jednotlivých částí systému a použitých modulů budou specifikovány v rámci implementační fáze.

### 4.2 Požadavky na statistiky výroby

Z požadavků dodavatele plyne, že chce mít přehled o jednotlivých ukazatelích strukturovaný na zaměstnance, zakázky, procesy výroby apod. Zároveň chce přehledy určovat pro různá časová období. Provádět takovéto agregační dotazy nad rozsáhlou databází čítající řádově tisíce nebo desetitisíce záznamů může být časově náročné, obzvláště pokud není databáze dobře navržena a nevyužívá možnosti indexování, především cizích klíčů. [12, s. 238–240] I v případě, že by databáze byla dobře dimenzována, může výpočet trvat dlouho a zbytečně zatěžovat server. Uživateli se potom data budou zobrazovat s velkým zpožděním a systém nebude mít pro něj takový přínos.

Vzhledem k tomu, že statistiky se pro již uplynulé dny nemění, nabízí se možnost dávkových výpočtů s denní periodou, která by počítala dlouhodobější statistiky a ukládala je do oddělených tabulek. Svým způsobem by se jednalo o odlehčenou verzi přístupu, který se uplatňuje u tvorby a vedení OLAP databází. [8] [13, s. 57] Pro statistiky z delšího časového období (např. měsíční) je tento způsob dostatečný. Pouze denní a např. týdenní statistiky vyžadující aktuální data by se počítaly vždy z produkční databáze. Zde v případě vhodného návrhu DB problémy nehrozí, protože údajů pro výpočet není už mnoho.

Výhoda tohoto hybridního přístupu je v dostatečné aktuálnosti přehledu jak pro nedávná, tak dlouhodobější data, nízkém zatížení databáze při opakovaném zobrazování časově náročných statistik a rychlé odezvě pro uživatele. Ta je pro uživatelský komfort systému důležitá. Nevýhodou je dozajista nutnost dodatečných tabulek (paměťová náročnost)

a související režie s dávkovými výpočty. Ta je ovšem daní za výhody, které tento způsob přináší.

### 4.3 Požadavky na sledování produktivnosti

Klient si přeje, aby systém zobrazoval informace o produktivnosti celé výroby a jednotlivých zaměstnanců a zároveň hlídal dodržování pracovní doby.

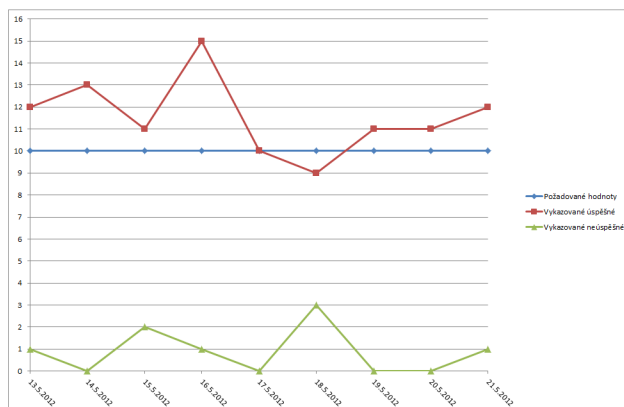
Produktivnost výroby je dána počtem zakázek nebo procesů, které se na daných zakázkách daří v časovém období dokončit a je zajisté důležitým ukazatelem výkonnosti firmy. Pokud klesne produktivnost je důležité včas tento jev odhalit a při tom zjistit, co bylo příčinnou. Je vhodné ji určovat pro různé stupně výroby, kdy je snazší identifikovat slabé místo. Při poklesu výroby by se mělo zobrazit varování, které by nabídlo aktuální a předpokládané hodnoty.

Je důležité správně určit období, za které se bude produktivnost počítat. Aktuální pokles může být vyvážen zrychlením v jiné fázi výroby a je proto důležité najít kompromis mezi aktuálními a včasnými údaji a poplašnými varováními. Nabízí se opět použít podobné časové intervaly, jako v předchozím případě, tedy den, týden, měsíc apod.

Produktivnost zaměstnanců se nabízí sledovat z počtu úkonů, které provádí a porovnávat je s ostatními pracovníky na stejné pozici. Dojde tím k reálnému porovnání, které může být pro uživatele systému také důležité. Systém by měl obsahovat možnost zadat předpokládané hodnoty produktivnosti zaměstnanců, které jsou na ně kladeny.

Pokles výkonnosti zaměstnanců může být způsobena různými faktory. Buď snížením pracovní morálky, ale třeba také technickou závadou, za kterou pracovník nemůže a musí mít kvůli ní prostoje. Tyto krajní situace je nutné také zohlednit.

Na obrázku 4.1 je uveden návrh grafu, na kterém by mohl být uživateli zobrazen náhled produktivnosti zákazníka. Obsahuje jak předpokládané hodnoty od firmy, tak jeho dosahované hodnoty včetně chybovosti.



Obrázek 4.1: Návrh grafu zobrazující produktivnost zaměstnance

### 4.4 Hlídaní pracovní doby

Co se týká hlídání dodržování pracovní doby, stačí kontrolovat, zda zaměstnanci dodržují předepsanou pracovní dobu, správně zohledňovat pracovní přestávky a další nutné záležitosti dané pracovní smlouvou. Varování by mělo přijít vždy při podlimitním stavu, naopak



sledovat nadměrnou pracovní dobu se nemusí jevit tak důležité, ale uživatel systému by měl mít možnost ovlivnit si parametry sledování oběma směry.

## 4.5 Požadavky na sledování zpoždění zakázek

Během výroby zakázky může dojít ke zpoždění důležité fáze, které bude mít za důsledek opoždění celé zakázky. Na základě již zmíněného síťového diagramu a sledování kritické cesty, ale nejen ji, může být snadno včas odhaleno velké množství problémových případů a pracovník může včas rozhodnout o dalším postupu.

Uživateli systému se tedy budou aktivně zobrazovat varování, na kterých uvidí hlášení o nastalém zpoždění v některé fázi zakázky, která má vliv na celkovou délku zpracování zakázky. Vzhledem k tomu, že by některé tyto předpoklady mohly být nepřesné, měl by mít uživatel možnost rozhodnout o správnosti upozornění a případně ji ručně potvrdit, či zamítnout pro daný konkrétní případ, ale i pro budoucí případy.

Všechny ruční zásahy je nutné zaznamenávat, aby bylo možné zpětně dohledat, který pracovník záznam provedl a případně klást důraz podle výše jeho postavení ve firmě, eventuálně penalizovat ho za chybný a úmyslný zásah do systému.

## 4.6 Závěr

V této kapitole byly přiblíženy požadavky na systém s ohledem na možnosti technické realizace. Jednalo se o požadavky na implementační jazyky, způsob použití indexů v databázi, ale probrány byly i požadavky na ukazatele výroby, které si přeje zadavatel práce sledovat.

## Kapitola 5

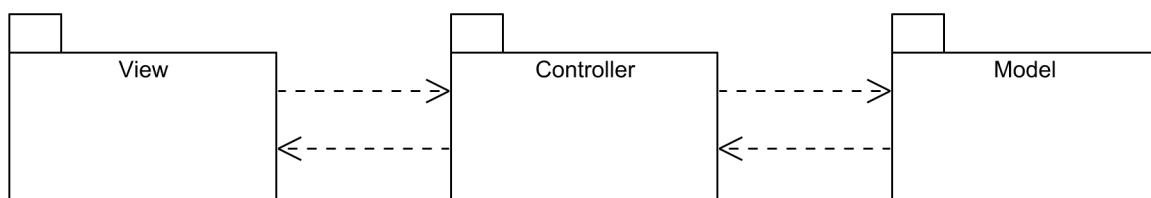
# Návrh aplikace

### 5.1 Architektura aplikace

Aplikace bude postavena jako portálové řešení, respektive jako třívrstvá aplikace typu klient server. To přináší požadavky na implementační jazyky.

Na klientské straně bude využit jazyk HTML. Pro účely prezentování požadovaných informací je dostatečný v kombinaci s dodatečnými technologiemi, především kaskádovými styly na formátování vzhledu. Standardní verze jazyka ovšem neumožňuje složitější generování např. grafů. Pro ty je vhodné zvolit externí knihovnu, které stačí předat zformátovaná data.

Celá aplikace bude navržena podle návrhového vzoru MVC (model-view-controller). Tento návrhový vzor odděluje logiku do tří samostatných celků – modelu, který má na starosti získávání, přípravu a zpracování dat, dále je to vrstva pohledu (view), která se stará o vykreslení dat uživateli. Obsluhu událostí z pohledu, ale i zpracování požadavků webové aplikace obstarává mezivrstva řadiče (controller). Výhodou tohoto přístupu je vhodné oddělení logických částí aplikace a tím i snadná rozšiřitelnost kódu do budoucna. [10, s. 425]

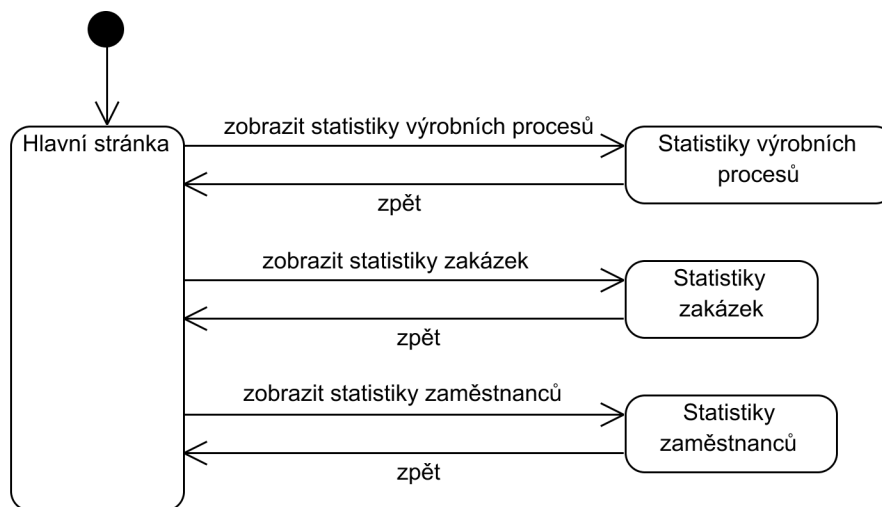


Obrázek 5.1: Ukázka komunikace v MVC návrhovém vzoru ve webovém prostředí

### 5.2 Struktura aplikace

Aplikace bude obsahovat výchozí pohled nabízející přehled základních sumárních charakteristik, které na první pohled prozradí důležité informace o výrobě. Součástí pohledu budou i varování o nalezených odchylkách. Další sadou pohledů bude skupina kolem zaměstnanců, především náhled na pracovní dobu zaměstnanců a produktivitu zaměstnanců. Dále pohledy na zakázky. Obsahovat budou náhled na rozpracování zakázek a produktivitu dokončování zakázek. Obdobnou skupinou bude i náhled na jednotlivé procesy výroby, kdy bude možné

sledovat produktivitu procesů, jejich vytíženost, chybovost atd. Návrh posloupnosti obrazek je zobrazen na schématu 5.2.



Obrázek 5.2: Návrh základní navigace z hlavního menu

Pohledy a akce systému pro editaci záznamů nejsou potřeba, protože ty obstarávají jiné části aplikace. Stejně tak podpora pro přihlašování uživatelů do systému je zahrnuta v základním systému.

Na pohledy navazuje vrstva modelu a řadičů. Každá skupina pohledů bude obsahovat jeden řadič, obsluhující jeho požadavky. Stejně tak bude kopírovat tento styl množina modelů. Návrh jednotlivých tříd je na schématu 5.3.

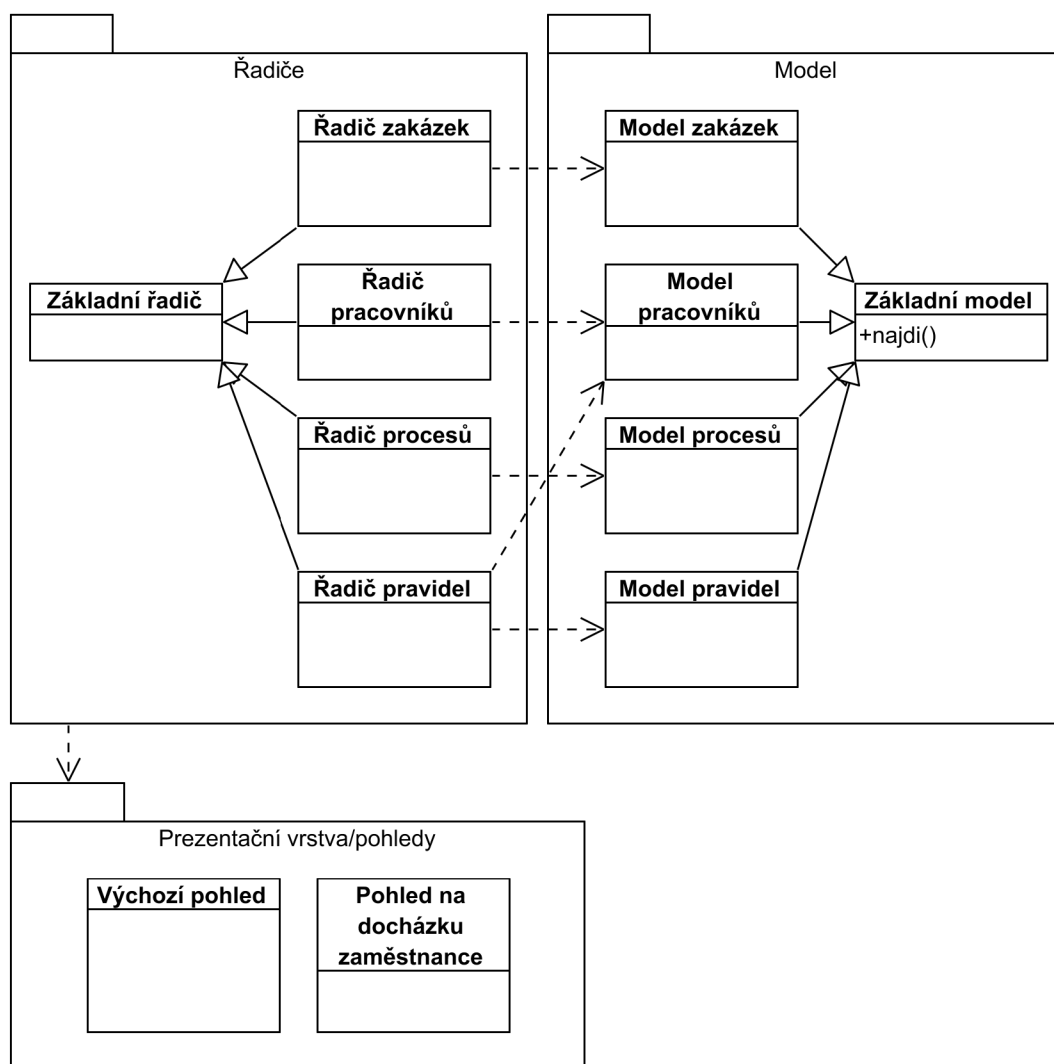
### 5.3 Datová vrstva

Serverové zázemí aplikace požaduje databázovou podporu nutnou pro uchovávání dat. Současná aplikace pracuje s relační databází MySQL, ve které jsou data uložena. I doplňovaná aplikace bude vyžadovat podporu této relační databáze. V případě počítání sumárních dat pro proběhlá období bude nutné vytvořit datový sklad, který bude tato agregovaná data uchovávat.

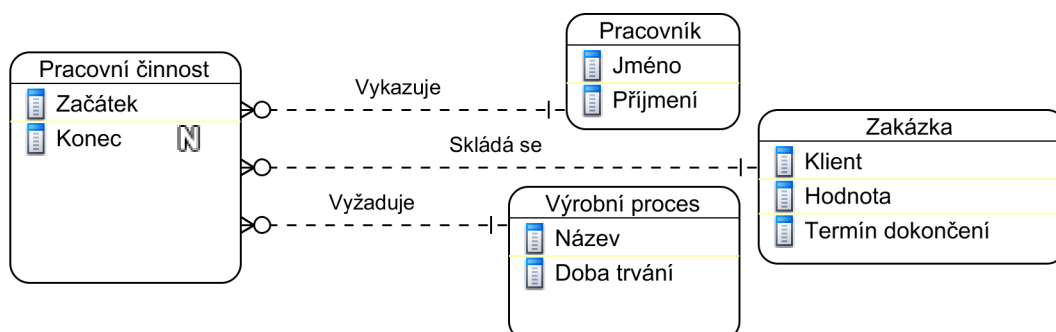
Data o jednotlivých pracovních činnostech jsou uchovávána v následující struktuře: odkaz na zaměstnance, zakázka a druh výrobního procesu, dále datum a čas začátku a konce práce. Viz ER diagram 5.4.

Pro sledování pracovní doby je dostačující využít strukturu, obsahující odkaz na zaměstnance a dále časové razítko příchodu na pracoviště a odchodu z pracoviště.

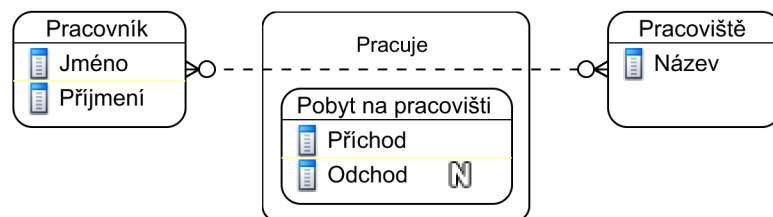
Pokud jde o doby trvání jednotlivých výrobních procesů, ty jsou v produkční databázi uvedeny nepřímou jako datum začátku výrobního procesu a jeho konec. Obsahují odkaz na daného pracovníka a zakázku, pro kterou byly požadovány. Tyto vazby je možné odstínit, stejně tak data je možné převést na časovou jednotku celočíselného typu udávající např. minuty. Interval, podle kterého je možné údaje seskupit by měl mít sám o sobě vypovídající schopnost, aby bylo možné už tato data využít pro přímou analýzu. Bylo by možné počítat přímo pouze průměrnou dobu, ale tím by se přišlo o možnost určit tendenci hodnot. Data tedy bude vhodné seskupit po jednom týdnu (eventuálně měsíci). Z tohoto intervalu bude



Obrázek 5.3: Konceptuální diagram tříd

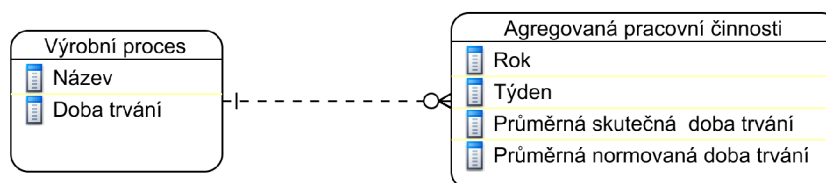


Obrázek 5.4: ER diagram pracovních činností



Obrázek 5.5: ER diagram pobytu pracovníka na pracovišti

možné rychle určit průměrnou dobu za kratší, i delší období (měsíc, rok), určit tendenci dat i celkovou průměrnou dobu.



Obrázek 5.6: ER diagram agregovaných pracovních činností

## 5.4 Akce systému

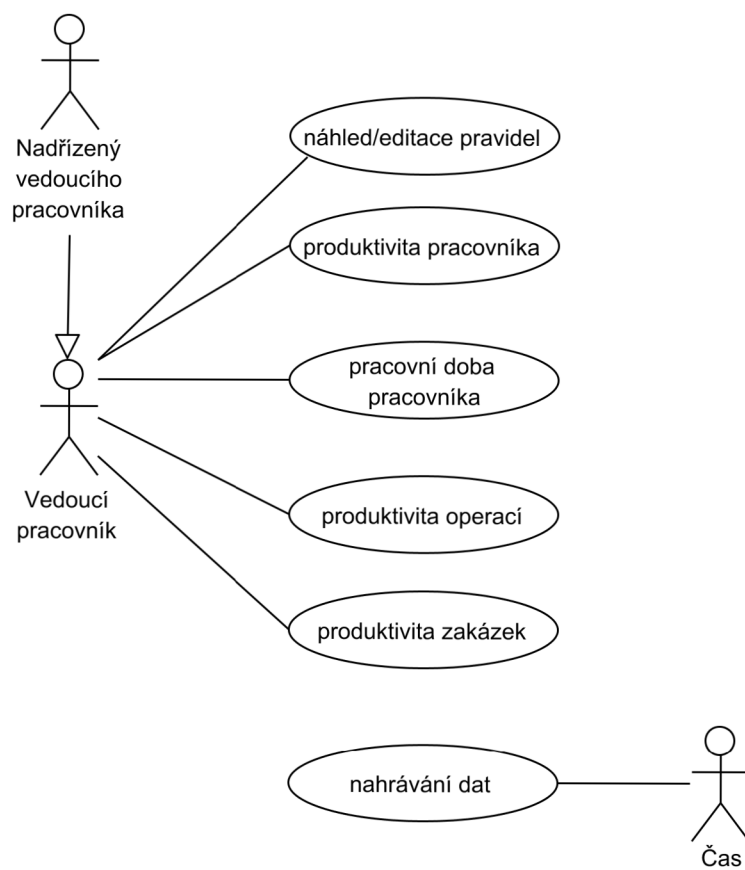
Primárním uživatelem systému bude vedoucí pracovník, který má dohled nad prováděním zakázky. Jeho úlohy v systému jsou nahlížení na stav vyhotovení zakázek, nahlížení na produktivitu pracovníka, operací, zakázek a na nahlížení do pracovní doby pracovníka. Jak bylo zmíněno, bude možné některé akce systému uživatelsky upravovat. Pro tyto případy musí mít do systému možnost přístupu na nadřízený vedoucího pracovníka, který může sledovat, kdo a jak se systémem pracoval.

Dalším aktérem systému je čas, který provádí spouštění naplnění databáze o aktuální data. Jednou za určitý interval bude nutné spustit doplnění analytické části databáze, které bude obsahovat předvypočtené údaje.

Diagram případů užití je uveden na obrázku 5.7.

## 5.5 Závěr

V kapitole byl ukázán návrh aplikace. Představen byl návrhový vzor celého systému, kterým se aplikace řídí. Proběhl návrh navigace v systému a konceptuální návrh tříd. V datovém modelu byly představeny ER diagramy, ze kterých by měla vycházet implementace nových tabulek databáze. Byly pojmenování aktéři v systému a jednotlivé akce a na tomto základě připraven diagram případů užití.



Obrázek 5.7: Diagram případů užití

# Kapitola 6

## Implementace

### 6.1 Architektura aplikace

Jak bylo zanalyzováno v sekci 5.1, aplikace byla navržena jako tenký klient v prostředí webového prohlížeče nad architekturou MVC.

Implementačním prostředím byl zvolen jazyk PHP bez omezení verze ze strany zadavatele. Byla vybrána verze PHP 5.3, která je v současné době sice již nahrazována novější verzí 5.4 a přechod je doporučen.<sup>1</sup> Nicméně kritické opravy na tuto verzi budou nadále vycházet a migrace aplikace na verzi PHP 5.4 je snadná.<sup>2</sup> U vybrané verze lze předpokládat větší pravděpodobnost, že bude použita na produkčním serveru. Databázovým prostředím je MySQL ve verzi 5.5.

Aplikace byla vytvořena v PHP frameworku Nette ve větvi 2.0.x, která je vydána jako stabilní a dále udržovaná.<sup>3</sup> Jako databázová mezivrstva bylo využito Dibi v odpovídající verzi k frameworku Nette.<sup>4</sup>

Všechny verze byly vybrány jako stabilní a vhodné ke komerčnímu nasazení i ve firemním prostředí.

Vzhledem k provozu aplikace v menších firmách bylo jako referenční webový prohlížeč Internet Explorer 8, který byl jako poslední verze nabízen pro Windows XP. Vycházelo se z předpokladu, že firemní prostředí obzvláště v oblasti výroby je konzervativnější a může obsahovat právě zmíněné podmínky. Předpoklad by ověřen osobní návštěvou jedné firmy, kde je ERP systém Pracant zaveden. Aktuální verze ostatních prohlížečů jsou s těmito podmínkami kompatibilní.

### 6.2 Navigace v aplikaci

Navigace byla uzpůsobena tak, aby se nejčastější akce vyskytovaly v první úrovni zanoření. Pokud je možné z nějaké analýzy přejít na detail, je využito zanoření do další úrovně. Pro pracovníka by měla být tedy navigace snadná. Ke všem stránkám existuje tzv. drobečková navigace, tedy zobrazení cesty od úvodní stránky až po aktuální stránku.

---

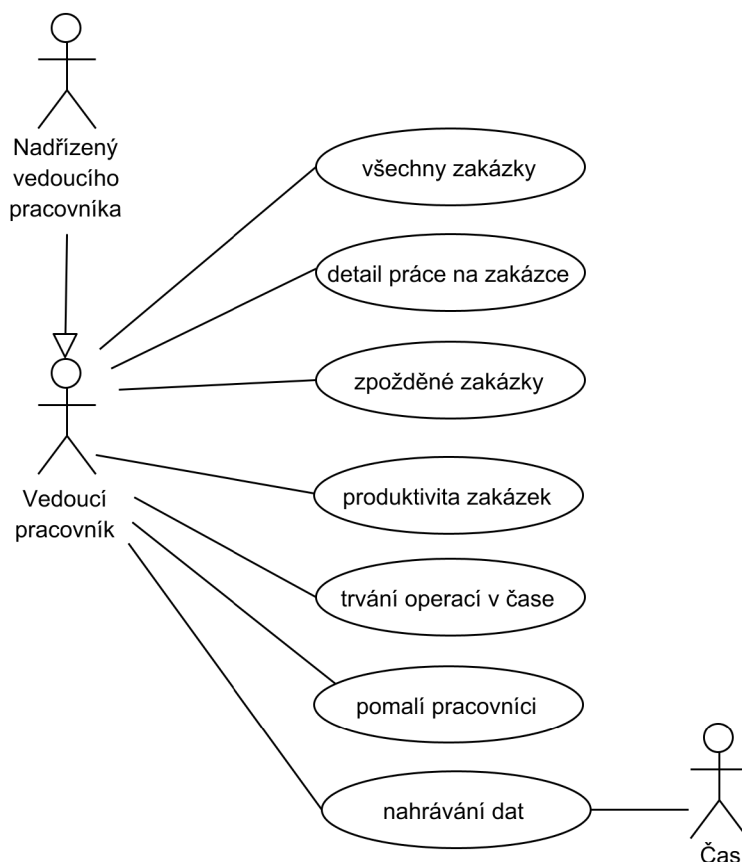
<sup>1</sup>Informace o stavu větve 5.3 doporučeném přechodu na 5.4 lze nalézt v archivu oznámení na oficiální adrese: <http://php.net/archive/2012.php>.

<sup>2</sup>Informace o přechodu mezi verzemi 5.3 a 5.4 lze nalézt na stránce oficiální dokumentace: <http://www.php.net/manual/en/migration54.incompatible.php>

<sup>3</sup>Viz oficiální stránky produktu: <http://nette.org/cs/>.

<sup>4</sup>Viz oficiální stránky produktu: <http://dibiphp.com/>.

Všechny akce systému, které byly implementovány jsou zobrazeny na diagramu užití 6.1.



Obrázek 6.1: Diagram užití – implementované akce systému

Hlavní stránka aplikace zobrazuje pouze odkazy na jednotlivé akce aplikace. Navržena byla i možnost, že by mimo jiné hlavní obrazovka obsahovala přehled varovných ukazatelů. Po zkoušce s dodanými daty bylo zjištěno, že by samotný přehled obsahoval příliš mnoho údajů<sup>5</sup>. Zůstala tedy varianta, kdy je v jednotlivých dlaždicích zobrazen odkaz na cílovou akci společně s údajem o jaký druh analýzy se jedná. Dále obsahuje upozornění na delší čas provádění analýzy a může obsahovat i další údaje. Součástí by mohlo být i varování o počtu sledovaných údajů pro každou akci.

Ohled byl vzat i na odesílání formulářů, kdy při odesílání formuláře pomocí metody POST nelze obnovit stránku, aniž by prohlížeč varoval uživatele o znovu zaslání formáře touto cestou. Všechny operace využívají tedy metodu GET, která také zachovává pro každou operaci jednoznačné URL a usnadňuje tak navigaci v prohlížeči pomocí akcí zpět, respektive vpřed.

Schéma obrazovek systému, které odpovídá celkové navigaci v systému, je zobrazeno na obrázku 6.2.

<sup>5</sup>Způsobeno to bylo spuštěním analýz na dosud používaných datech bez odfiltrování atypických zakázek, zakázek zpožděných kvůli chybám, špatně evidovaných zakázek a podobně. Při nastavení dodatečných filtrů a uzpůsobení aplikace pro konkrétní provoz by se počet varovných ukazatelů měl dostat na úroveň vhodnou pro takovéto zobrazení. Bez znalostí experta z provozu toto nebylo možné provést.





## 6.3 Dodatky k implementaci

### 6.3.1 Názvy tabulek

K dispozici byla ukázková databáze, která obsahovala tabulky používané v ERP systému. Pro zjednodušení byla stejná databáze využita i pro tabulky vzniklé v této práci. Aby se na první pohled odlišily, mají nově vytvořené tabulky předponu `reporting_`. Jejich odstraněním se databáze vrátí do původního stavu.

### 6.3.2 Zjednodušení formátování doby trvání

Framework Nette poskytuje vrámci svého šablonovacího jazyka Latte tzv. „helpery“. Jedná se o zjednodušený způsob volání funkcí v šablonách. Ve frameworku jsou obsaženy některé vestavěné helpery<sup>6</sup>. Syntaxe v šabloně je poté následující: `{$promenna|navezFunkce}`. Jedná se o zápis, kdy se má vypsát hodnota proměnné `$promenna`, jejíž výstup se nejdříve aplikuje jako argument funkce.

V rámci systému byl vytvořeny dva nové helpery, který dobu trvání uloženou jako celé číslo převedou na hodiny, minuty a sekundy. První se jmenuje `secondTime` a převádí čas ze sekund na minuty a sekundy. Druhý se jmenuje `minuteTime` a obdobně převádí čas na hodiny a minuty. Například tedy celé číslo 316 reprezentující dobu v sekundách bude vypasáno jako 5 min 16 s.

Sekundový převod je vhodný v pohledech, které pracují s kratší dobou, např. délky trvání operací převedené na jeden kus. Minutový převod byl použit při znázorňování doby trvání zakázek.

Tyto zápisy urychlí další práci na systému a sjednotí zobrazované údaje.

### 6.3.3 Nastavení aplikace

Veškeré nastavení je k dispozici v jedné třídě `SettingsModel`. Třída obsahuje veřejnou metodu `getSetting`, která pro zadaný klíč vrátí hodnotu nastavení. Seznam možných klíčů je uveden ve stejné třídě jako její konstanty. Nastavení je v této třídě uloženo v asociativním poli. Třída je začleněna řádně do modelu a obsahuje tedy připojení k databázi, aby nebyl do budoucna problém přesunout nastavení do databáze a rozhraní třídy ponechat stejné.

Nastavení obsahuje dvě hodnoty – konstanty, které určují stav zakázky. Konstanta `ORDERS_OK` s hodnotou 105 určuje počet procent, kdy čas plnění zakázky ještě nepřesáhl povolený limit. Konstanta `ORDERS_WARNING` poté obsahuje hodnotu 115, kdy se jedná o hranici, při které je na zakázku pouze upozorněno a kdy se jedná už o chybu v plnění termínu.

## 6.4 Způsob ignorování záznamů

Vzhledem k tomu, že aplikace je navrhována jako volitelné rozšíření stávající aplikace, bylo nezbytné nezasahovat do stávající struktury databáze. Nicméně některé operace pracují přímo nad těmito tabulkami a jedním z požadavků na systém je, aby bylo možné některé údaje do výpočtu nezahrnovat.

Kdyby bylo možné provést úpravu schématu, přidal by se k tabulce požadující nést informaci o vyloučení jeden atribut `exclude`, který by nesl binární logickou hodnotu.

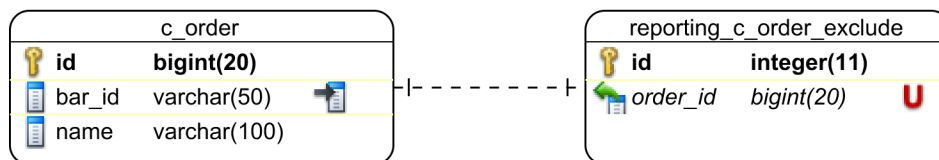
<sup>6</sup>Výchozí helpery šablon. Nette Framework [online]. 2013 [cit. 2013-05-20]. Dostupné z: <http://doc.nette.org/cs/default-helpers>

Zvolena byla simulace tohoto přístupu. Pro každou takovou tabulku byla vytvořena tabulka dodatečná se stejným názvem a prefixem `_exclude`. Nese dva atributy – primární klíč záznamu a unikátní cizí klíč do požadované tabulky. Každý záznam z původní tabulky, který by měl být vyřazen, bude nyní obsahovat záznam ve vyřazovací tabulce. Unikátnost klíče se zajistí, aby nebyly záznamy zbytečně zdvojovány případnou chybou aplikace. Pokud už nebude potřeba záznam vyřazovat, dojde ke smazání celého záznamu.

V SQL dotazu se, kde se operuje s tabulkou obsahující záznamy k vyřazení, se připojí vyřazovací tabulka. V klauzuli `SELECT` se využije příkazu `IF('reporting_c_order_exclude'. 'id' IS NULL, 0, 1) AS 'exclude'` (`reporting_c_order_exclude` je vyřazovací tabulka k tabulce `c_order`.) Všechny záznamy mají primární klíč vyřazovací tabulky nyní roven SQL hodnotě `NULL` kromě těch, které obsahují záznam ve vyřazovací tabulce. Výsledkem tohoto příkazu je hodnota `exclude`, která nabývá hodnoty 1, pokud se má záznam vyřadit, jinak hodnotu 0.

Obdobnou operaci s operátorem `IS NULL`, respektive `IS NOT NULL` lze použít i v klauzuli `WHERE` jakéhokoli dotazu a záznamy tak efektivně vyřazovat.

Tento přístup je výhodný v situaci, kdy větší část záznamů je využita a vyřazena je pouze menší část. Jinak by muselo dojít k obrácení logiky z vyřazování na zařazování.



Obrázek 6.3: Schéma databáze – ukázka tabulky pro ignorování záznamů

## 6.5 Implementované pohledy

### 6.5.1 Pohled: Všechny zakázky

Zobrazením tohoto pohledu se uživateli zobrazí přehled všech zakázek s celkovou dobou trvání (danou součtem časů dílčích operací). Součástí přehledu je kromě celkového normovaného času také čas skutečně strávený zaměstnanci na zakázce a procentuální stav poměru těchto dob. Pro výpočet jsou využity přímo produkční databázové tabulky `c_order` a `order_work` obsahující informace o zakázce a dílčí operace na zakázkách. Dále tabulka `readdata` obsahující data o skutečně odvedené práci.

Tato operace je realizována pomocí jednoho SQL dotazu přímo nad produkčními tabulkami. Vzhledem k celkovému počtu řádků výsledného dotazu před agregací by operace trvala neúměrně dlouho. Bylo proto zvoleno řešení, kdy je uživateli nabídnuto zobrazení výsledků po stránkách. Dotaz je poté prováděn pro podmnožinu zakázek bez nadměrného zpoždění.

Vzhledem k charakteru dotazu nebylo možné použít klauzuli `LIMIT`, která by omezila počet celkových výsledků dotazu. Dotaz by se vždy provedl nad kompletní sadou řádků a poté se omezil. Je tedy nejdříve proveden jeden dotaz nad samotnou tabulkou `c_order` s klauzulí `LIMIT`, který vybere daný počet primárních klíčů této tabulky. Tyto klíče jsou poté využity v hlavním dotazu v klauzuli `WHERE` a operátoru `IN`. Byl tím efektně omezen limit vybíraných řádků se zachováním nízkého času provádění dotazů. První dotaz by mohl

být ještě integrován do hlavního dotazu jako poddotaz, ale použité řešení bylo vhodnější z implementačního hlediska a výkonnostně se od sebe řešení příliš neliší.

Další vlastností vytvořenou speciálně pro potřeby systému je získávání atributu `exclude`, který určuje, zda daná zakázka není vyjmuta z relevantních dat. Více o systému ignorování údajů lze nalézt v kapitole 6.4. U každé zakázky má uživatel možnost ignorovat danou zakázku. Ignorované zakázky jsou znázorněny jako přeskrtnuté a odkaz pro ignorování je změněn na akci „Zahrnout“.

Pro získání doby trvání prací zaměstnanců je použita vestavěná funkce `TIMESTAMPDIFF`<sup>7</sup>. Pro dva zadané datумы vrátí dobu trvání jako celé číslo. Jednotky určuje první argument funkce (`SECOND`, `MINUTE`, `HOURL` atd.). Tyto doby jsou sečteny pomocí agregační funkce `SUM`.

Normovaný čas celé zakázky je pro každou zakázku získán jako poddotaz v klauzuli `SELECT`. Jsou vybrány všechny normované časy všech operací s vyloučením ignorovaných operací a sečteny.

Logika této operace je uložena ve třídě modelu `OrdersModel`, šablony pohledu jsou v sekci `Orders` a vrstvu řadiče zajišťuje třída `OrdersPresenter`.

Součástí stránky je také vyhledávací formulář, kde si uživatel může vybrat jeden sloupec, ve kterém lze vyhledávat a hodnotu, kterou chce vyhledat. Vyhledávání funguje jako filtr, na stejné stránce se mu zobrazí výsledky, které jsou omezeny pomocí zadaných podmínek.

Vybráním jedné zakázky se uživateli zobrazí její detail.

## Všechny objednávky

Navigace: [Úvodní strana](#) / [Přehled objednávek](#)

[Zobrazit pouze zpožděné objednávky](#)

Vyhledat:

Název zakázky	Začátek	Konec	Doba trvání	Normovaný čas	Stav	
<a href="#">Kvadra</a>	2008-03-01	2008-03-31	62 hod 0 min	11 min	33 818,18 %	<a href="#">Ignorovat</a>
<a href="#">botník 4 40 ks</a>	2007-10-15	2007-12-31	125 hod 32 min	30 min	25 106,67 %	<a href="#">Ignorovat</a>
<a href="#">Rošt 16 90 cm</a>	2007-10-12	2007-11-12	72 hod 41 min	37 hod 46 min	192,45 %	<a href="#">Zahrnout</a>
<a href="#">Sendy 17 eela 500 ks</a>	2007-10-12	2007-10-31	680 hod 14 min	636 hod 26 min	106,88 %	<a href="#">Ignorovat</a>
<a href="#">Ema 4 SM</a>	2007-10-12	2007-11-12	57 hod 42 min	28 min	12 364,29 %	<a href="#">Zahrnout</a>
<a href="#">Ema 5 BK</a>	2007-10-12	2007-11-12	65 hod 40 min	4 hod 40 min	1 407,14 %	<a href="#">Ignorovat</a>
<a href="#">Berghen 12 110 ks</a>	2007-10-10	2007-11-10	315 hod 45 min	136 hod 23 min	231,52 %	<a href="#">Ignorovat</a>
<a href="#">Box 7 60 ks</a>	2007-10-10	2007-11-10	196 hod 14 min	175 hod 30 min	111,81 %	<a href="#">Ignorovat</a>

Obrázek 6.4: Ukázka aplikace – přehled všech zakázek

### 6.5.2 Pohled: Detail zakázky

Detail zakázky zobrazí uživateli kompletní rozpis všech operací a prací na dané zakázce. Práce jsou seskupeny podle operací a jsou nabídnuty procentuální stavy vyhotovení jednotlivých prací i operací. Graficky je znázorněno, v jakém stavu je každá práce/operace. Rozmezí 0 až 105 procent času je podbarveno zeleně, 105 až 115 oranžově a vyšší údaje červeně. Hotové množství kusů je znázorněno obdobně.

Každou práci na zakázce, ale i každou celou operaci má uživatel možnost nechat ignorovat a vyloučit ji tak z dílčích i celkových výpočtů. Uživatel tak může ručně označit úkony, které by neměly ovlivňovat statistiky. Automatické napojení na jiné ukazatele (např. automatickou evidenci poruch) není implementováno, protože nebyly dostupné potřebné informace. Ignorování probíhá odkazem s textem „Ignorovat“, který je umístěn u všech

<sup>7</sup><https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html>

relevantních řádků výpisu. Po ignorování se daný řádek s výpisem označí škrtnutím a text odkazu se změní na „Zahrnout“.

Pro započaté práce, které ještě nebyly dokončeny je zobrazována průměrná doba, po kterou v aktuálním měsíci trvá daná operace. Tato hodnota je získána z analytického výpočtu trvání jednotlivých operací. Způsob výpočtu je rozebrán v kapitole 6.5.4.

Uživatel na této stránce přehledně vidí, v jakém termínu byly plněny práce na zakázce. Zároveň vidí dílčí součty operací i celkové za celou zakázku.

### Detailní informace o zakázce

Navigace: [Úvodní strana](#) / [Přehled objednávek](#) / [Detailní informace o objednávce](#)

Operace	Začátek	Konec	Normovaný čas	Délka všech operací	Stav
Box Thomas 4 SM	2008-04-17	2008-08-31	99 hod 13 min	103 hod 26 min	104.25 % <a href="#">Ignorovat</a>

### Průběh prací na zakázce

Operace	Norma		Pracovník	Začátek operace	Konec operace	Hotové množství	Délka operace	
	Čas	Množství						
01 hrubé krác.-bok,pořè.tèè.	4 hod 0 min	480.00	Prchlík Radek	2008-04-23 13:44:43	2008-04-23 14:20:38	31.00	35 min	<a href="#">Ignorovat</a>
			Mikuška Martin	2008-04-24 05:54:47	2008-04-24 08:31:26	250.00	2 hod 36 min	<a href="#">Ignorovat</a>
			Mikuška Martin	2008-04-24 08:44:26	2008-04-24 10:39:10	310.00	1 hod 54 min	<a href="#">Ignorovat</a>
			Mikuška Martin	2008-04-24 12:17:14	2008-04-24 12:45:24	0.00	28-min	<a href="#">Zahrnout</a>
			Součet			123.13 %	591.00	127.08 %

Obrázek 6.5: Ukázka aplikace – detail jedné zakázky

### 6.5.3 Pohled: Produktivnost zakázek

Zobrazením přehledu „Produktivnost zakázek“ získá uživatel přehled o tom, kolik zakázek bylo dokončováno v jakých obdobích. Jedná se analytický ukazatel, který data získává z produkčních tabulek.

Na úvodní stránce přehledu se uživateli zobrazí počet dokončených zakázek v jednotlivých letech (viz obrázek 6.6). Pro každý rok může uživatel provést obdobu operace *drill down*, kdy se dostane na měsíční přehled daného roku (viz obrázek 6.7). Další zanoření již není a odkaz pro každý měsíc vede na stránku se seznamem událostí s nastaveným filtrem právě pro daný měsíc. V případě rozšíření by bylo možné udělat ještě další zanoření na týdny a dny.

Na stránce se zakázkami se správně předvyplní formulář pro vyhledávání, jak je vidět na obrázku 6.8.

K získání počtu dokončených zakázek je využit agregační dotaz, který pro celkový přehled provede seskupení podle roku zakázky. Rok je získán pomocí vestavěné SQL funkce `YEAR()`, která ze zadaného datumu vrátí rok. Výsledek funkce je uložen jako alias, se kterým je možné pracovat v klauzulích `GROUP BY` a `HAVING`. V prvním případě je využito seskupování pouze podle tohoto atributu. Při zobrazení statistik za jeden rok je nejdříve v klauzuli `SELECT` získán obdobným způsobem také měsíc a výsledky jsou seskupeny podle něj. Rok by se nabízel omezit pomocí klauzule `HAVING`, protože v tomto okamžiku je již znám výsledek funkce `YEAR` a je přístupný její alias. Tato klauzule ovšem vybírá data až po seskupení, takže její použití je v tomto případě nevhodné. Byla tedy použita klauzule `WHERE` s omezující

podmínkou `YEAR(finish) = ROK`. Počet řádků je získán pomocí agregační funkce `SUM`.

Díky získávání údajů primárně pouze z jedné tabulky nebylo nutné data předvypočítat a celá operace tak pracuje přímo s produkčními daty v reálném čase.

Dotaz je rozdělen do několika samostatných funkcí. Základní kostra dotazu je obsažena v privátní funkci `findOrdersEnding`, která obsahuje podčást společnou pro seskupování podle roku i podle měsíce. Tuto funkci volají dvě veřejné funkce `findOrdersEndingByYears` a `findOrdersEndingByMonth`. Ty doplňují základní kostru dotazu svými podmínkami.

### Produktivnost zakázek

Navigace: [Úvodní strana](#) / [Přehled objednávek](#) / [Produktivnost zakázek](#)

Rok	Počet dokončených zakázek
2007	85
2008	287
2009	240
2010	211
2011	253
2012	110
2013	1

Obrázek 6.6: Ukázka aplikace – produktivnost všech zakázek

### Produktivnost zakázek v roce 2009

Navigace: [Úvodní strana](#) / [Přehled objednávek](#) / [Produktivnost zakázek](#) / [Produktivnost v roce 2009](#)

Měsíc	Počet dokončených zakázek
1	33
2	19
3	15
4	28
5	25
6	13
7	15
8	16
9	12
10	20
11	22
12	22

Obrázek 6.7: Ukázka aplikace – produktivnost zakázek v jednom roce

#### 6.5.4 Pohled: Trvání operací

Při analýze trvání jednotlivých operací se uživateli nejdříve nabídne přehled všech operací, které jsou v systému evidované. Vyloučeny jsou pouze ty, které obsahují šum, nebo jsou označeny jako ignorované. Výběrem jedné akce se uživateli zobrazí jednoduchý přehled o tom, jak dlouho trvalo vykonávání jednotlivých operací v jednotlivých měsících. Zároveň vidí, jaká byla v daném období navrhovaná norma.

Výpočet těchto údajů v reálném čase byl již pomalý a trval několik desítek sekund. Byl tedy využit přístup předvypočtení údajů do samostatné tabulky. Nyní trvá zobrazení výsledků několik milisekund.

Celá operace je tvořena jedním dotazem. Výsledkem je pět hodnot: operace, rok, měsíc, skutečná doba a normovaná doba. První tři údaje jsou získány triviálně. Skutečná doba



## Všechny objednávky

Navigace: [Úvodní strana](#) / [Přehled objednávek](#)

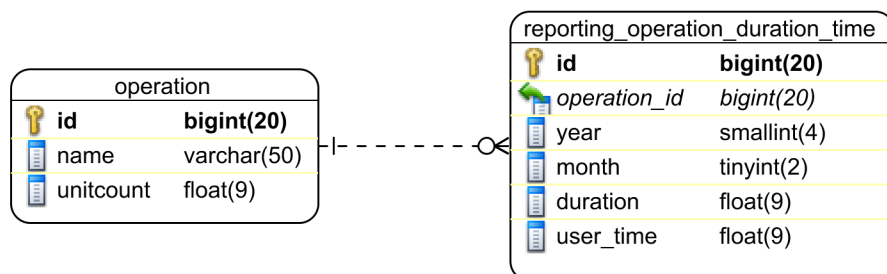
[Zobrazit pouze zpožděné objednávky](#)

Vyhledat:

Název zakázky	Začátek	Konec	Doba trvání	Normovaný čas	Stav	
Cherry 9	2009-09-02	2009-09-30	3 hod 0 min	10 min	1 800,00 %	Ignorovat
Box Thomas 6 SM	2009-09-02	2009-09-30	117 hod 28 min	145 hod 36 min	80,68 %	Ignorovat
Atyp 9	2009-09-02	2009-09-30	83 hod 8 min	10 min	49 880,00 %	Zahrnout
Rošt 90 cm 10	2009-08-26	2009-09-30	67 hod 57 min	85 hod 43 min	79,27 %	Ignorovat
Spojovák 4 SM	2009-08-18	2009-09-30	318 hod 49 min	337 hod 29 min	94,47 %	Ignorovat
Sendy 10 SM	2009-08-18	2009-09-30	500 hod 50 min	629 hod 1 min	79,62 %	Ignorovat
Thomas SM 8	2009-08-07	2009-09-30	212 hod 28 min	190 hod 51 min	111,33 %	Ignorovat
Zábrana 4	2009-07-15	2009-09-30	137 hod 19 min	183 hod 13 min	74,95 %	Ignorovat
Sendy 9 SM	2009-07-09	2009-09-30	653 hod 29 min	686 hod 40 min	95,17 %	Ignorovat
Thomas 7 BK	2009-07-09	2009-09-30	152 hod 55 min	230 hod 24 min	66,37 %	Ignorovat
Nášlapy 4 BK	2009-06-22	2009-09-30	162 hod 24 min	216 hod 16 min	75,09 %	Ignorovat
Ostrov oboustranný 2	2009-05-11	2009-09-30	59 hod 55 min	51 hod 9 min	117,14 %	Ignorovat

Obrázek 6.8: Ukázka aplikace – vybrané zakázky v jednom měsíci

strávená v daném období na zakázkách je spočtena jako součet dob strávených pracovníky na plnění operací. Ty jsou přesně uvedeny v tabulce `readdata`. Pokud operace ještě neskončila, není do analýzy započtena. Součet je dělen počtem jednotek, které pracovníci vyrobili a zprůměrován počtem zakázek. Vznikl tak čas, který udává, jak dlouho trvalo vyrobit jednu jednotku průměrně všem pracovníkům. Z tabulky `order_work` je získána normovaná doba pro všechny operace, která je opět podělena počtem jednotek a počtem zakázek. Opět se získal čas potřebný pro jednu jednotku, ale normovaný.



Obrázek 6.9: Schéma databáze – tabulky pohledu trvání operací

Z analýzy jsou vyloučena zašumělá data, která obsahují špatně evidované údaje, především začátky a konce operací. Také jsou automaticky vyloučeny operace, které nemají uvedenou normu, ty obsahují místo normy pouze číslo jedna a příliš by zkreslovaly údaje. Stejně tak se vyloučily údaje s neevidovaným množstvím, aby nedocházelo k dělení nulou, což by vedlo k chybě.

Nakonec byly vyloučeny údaje, které byly uživatelsky označeny jako ignorované. Jak celé zakázky (tabulka `reporting_c_order_exclude`), tak operace na celé zakázce (tabulka `reporting_orderwork_exclude`), tak dílčí práce na zakázce (`reporting_readdata_exclude`).

Všechny údaje jsou pro celou operaci zobrazeny najednou. Dělení pohledu nejdříve na roky a potom na měsíce je zbytečné, protože výsledných záznamů pro celou operaci nebývá mnoho.

Samotné nahrání dat do tabulky je realizováno pomocí dvou dotazů. Nejdříve dojde

k vyprázdnění původního obsahu (příkazem `TRUNCATE TABLE`). Poté se pomocí příkazu `INSERT INTO ... SELECT ...` vloží data nová. Původní SQL dotaz je uložen v samostatné funkci modelu, aby bylo možné jeho přímé využití nad reálnými daty. Zde je využit na místě klauzule `SELECT`. Díky tomu je celá operace provedena jedním dotazem a není potřeba výsledky procházet programově a postupně vkládat záznam po záznamu.

Nově vypočtená norma má nejen charakter informativní v tomto pohledu, ale využívá se i v dalších pohledech (např. při detailu jedné zakázky).

Nad vypočtená data je možné dále umístit jiný pohled, např. zobrazení pomocí grafu atd.

## Všechny operace

Navigace: [Úvodní strana](#) / [Přehled všech operací](#)

« Předchozí 1 2 3 4 ... 20 ... 40 ... 59 ... 78 Další »

Název operace
<a href="#">0001 názezovna-atyp</a>
<a href="#">0002 strojovna-atyp</a>
<a href="#">0003 montáž-atyp</a>
<a href="#">0004 lakování-atyp</a>
<a href="#">0006 brouš po laku-atyp</a>
<a href="#">0007 balení-atyp</a>
<a href="#">0011 K H P - režie</a>
<a href="#">0013 T.H.P</a>

Obrázek 6.10: Ukázka aplikace – přehled všech operací

## Detaily operace

Navigace: [Úvodní strana](#) / [Přehled všech operací](#) / [Detail operace](#)

### Operace 05 hran.bruska-noha

Rok	Měsíc	Průměrný čas operace	Normovaný čas operace
2007	10	19 s	17 s
2007	11	22 s	19 s
2007	12	19 s	19 s
2008	1	23 s	20 s
2008	2	21 s	21 s
2008	3	22 s	23 s
2008	4	15 s	22 s
2008	5	20 s	20 s
2008	6	20 s	20 s
2008	7	21 s	20 s
2008	8	18 s	20 s
2008	9	19 s	20 s
2008	10	24 s	20 s
2008	11	19 s	20 s
2008	12	18 s	20 s
2009	1	20 s	18 s
2009	6	20 s	18 s

Obrázek 6.11: Ukázka aplikace – detail trvání jedné operace

### 6.5.5 Pohled: Zpožděné zakázky

Pohled na zpožděné zakázky ukáže uživateli všechny zakázky, které byly dokončeny po skončení termínu. Jedná se o analýzu nad produkčními daty v reálném čase.



Základem tohoto dotazu je dotaz na získání všech zakázek. Ten je již seskupený podle unikátního klíče zakázky a obsahuje vypočtenou hodnotu skutečného trvání zakázky `duration` a normovaného času `user_time`. Celý výsledek je omezen klauzulí `HAVING`. Ta kontroluje poměr těchto dvou údajů a vrací pouze ty zakázky, jejichž poměr zpoždění je větší jak 1,15, respektive skutečný čas přesáhl 115 procent normy.

Dotaz obsahuje umělé omezení na maximální zpoždění, které bylo stanoveno na 500 procent. Data obsahují spoustu zakázek, jejichž zpoždění nabývá řádů tisíců procent. Jedná se zřejmě o speciální případy, které by byly po nasazení systému nastaveny jako ignorované.

Stejně jako přehled všech zakázek i tento pohled nabízí volbu ignorovat záznamy, které jsou svojí povahou atypické. Vybráním akce „Ignorovat“ se záznam z toho pohledu skryje. Viditelný je standardně v přehledu všech zakázek, kde se ignorované zakázky zobrazují jako škrtnuté a umožňují tak jejich opětovné zařazení do analýz.

Vzhledem k charakteru dotazu, trvá jeho zpracování několik sekund. Na tuto skutečnost je uživatel upozorněn na dlaždici na úvodní stránce. Zpoždění na vývojovém prostředí dosahovalo přibližně deseti sekund, na dedikovaném webovém serveru přibližně pět sekund.

### Zpožděné zakázky

Navigace: [Úvodní strana](#) / [Přehled objednávek](#)

Název zakázky	Začátek	Konec	Doba trvání	Normovaný čas	Stav
<a href="#">Lehátko BK bez podr. 4</a>	2012-05-12	2012-05-31	12 hod 40 min	4 hod 4 min	311,48 % <a href="#">Ignorovat</a>
<a href="#">Lehátko BK s podr. 4</a>	2012-05-12	2012-05-31	20 hod 0 min	9 hod 0 min	222,22 % <a href="#">Ignorovat</a>
<a href="#">Lehátko ps.BK bez vln. 3</a>	2012-05-12	2012-05-31	28 hod 50 min	21 hod 8 min	136,44 % <a href="#">Ignorovat</a>
<a href="#">Stapak paleta 93x93 cm</a>	2012-05-11	2012-05-31	10 hod 46 min	8 hod 15 min	130,51 % <a href="#">Ignorovat</a>
<a href="#">Spojovák II</a>	2012-05-05	2012-06-05	116 hod 56 min	35 hod 44 min	327,24 % <a href="#">Ignorovat</a>
<a href="#">Pivní set 1</a>	2012-04-30	2012-05-31	104 hod 31 min	43 hod 54 min	238,08 % <a href="#">Ignorovat</a>
<a href="#">Židlička ditská-lakování</a>	2012-04-30	2012-05-31	6 hod 16 min	4 hod 32 min	138,24 % <a href="#">Ignorovat</a>
<a href="#">Paravan-dveře</a>	2012-04-17	2012-04-30	6 hod 31 min	4 hod 22 min	149,24 % <a href="#">Ignorovat</a>
<a href="#">Lehátko BK bez podr. 3</a>	2012-04-10	2012-04-30	94 hod 31 min	80 hod 40 min	117,17 % <a href="#">Ignorovat</a>
<a href="#">Pískovišti s poklopem 2</a>	2012-04-10	2012-04-30	100 hod 10 min	85 hod 31 min	117,13 % <a href="#">Ignorovat</a>
<a href="#">Rošt 90 cm 4</a>	2012-04-05	2012-05-05	121 hod 45 min	50 hod 42 min	240,14 % <a href="#">Ignorovat</a>

Obrázek 6.12: Ukázka aplikace – přehled zpožděných zakázek

### 6.5.6 Pohled: Pomalí pracovníci

Pohled na pomalé pracovníky uživateli nabídne přehled pracovníků, kteří v daných operacích pracovali pod průměrnou měsíční normou vypočítanou při analýze trvání zakázek (viz kapitola 6.5.4).

Dotaz vybírá práce všech zaměstnanců a seskupuje je podle zakázky, operace a pracovníka. Každému pracovníkovi spočte jeho průměrnou dobu, kterou tráví při vyhotovení norem operací. Doba je podělena počtem vyhotovených kusů, aby byla získána průměrná doba za jeden kus. Poddotazem je vždy pro každý takový záznam získána norma, která byla platná v daném měsíci.

V klauzuli `HAVING` je poté nastaveno omezení na celkový poměr zpoždění větší než 115 procent normy. Zároveň je obsaženo umělé omezení na maximální zpoždění 500 procent.

Při výpočtu průměru času, který pracovník stráví na zakázce je využit aritmetický průměr nezohledňující důležitost nebo cenu zakázky. Při určování tohoto faktoru byla použita korelační analýza. Zjišťovala se závislost mezi atributy nákladnosti operace, konkrétně atributy `user_time`, `user_price` a `units` tabulky `order_work`, která přiřazuje jednotlivé operace k zakázkám. Pro výpočet Pearsonova koeficientu byl využit program Microsoft

Excel. Ten obsahuje funkci PEARSON, která pro zadané datové oblasti Pearsonův koeficient spočítá. Výsledek pro podmnožinu 35 záznamů byl v případě porovnání `user_time` a `user_price` 0,97 a pro porovnání `user_time` a `units` 0,96, což značí, že tyto atributy jsou na sobě závislé a to skoro přímo úměrně.

Další možností bylo porovnat atributy, které jsou uvedeny u zakázky samotné v tabulce `c_order`. K těmto atributům nebyla dostupná legenda. Po konzultaci by však bylo možné aritmetický průměr nahradit váženým průměrem a získat tak přesnější hodnoty.

## 6.6 Závěr

V rámci implementace byly vybrány implementační jazyky včetně svých verzí, dále aplikační framework, který splnil požadavky specifikace. Vybrán byl také referenční internetový prohlížeč, ve kterém aplikace bude fungovat.

Byl také diskutován způsob navigace v aplikaci a hierarchie stránek, aby byla práce se systémem co nejjednodušší. Na schématu byla shrnuta kompletní posloupnost všech obrazovek. Zmíněny byly i některé detaily, které v rámci implementace byly vytvořeny.

Dále byly popsány detaily samotné realizace požadavků na sledování produktivnosti a hlídání zpoždění. Konkrétně způsob ignorace záznamů z již existujících tabulek určených pouze pro čtení, pohledy na zakázky, operace atd.

## Kapitola 7

# Možnosti rozšíření

Ve specifikaci požadavků byly zmíněny některé akce, které nakonec implementovány nebyly. Z dodaných dat nebylo možné buď některé operace provést, nebo data byly bez další konzultace nesrozumitelné. Např. síťový graf anebo Ganttův diagram nebylo možné kvůli chybějícím údajům rekonstruovat. Pro tyto diagramy je nutné znát posloupnost jednotlivých kroků výrobního procesu a návaznosti těchto kroků. V dodané databázi tyto informace nebyly k dispozici. Pokud by se databáze rozšířila o takovéto informace, bylo by možné doimplementovat tato rozšíření.

Vzhledem k charakteru dodaných dat, která obsahovala především historické údaje, byly implementovány především analytické pohledy. Akce hlídající aktuálně rozpracované zakázky z těchto důvodů implementovány nebyly, ale mohou být na základě specifikace požadavků a návrhu do systému doplněny.

U kontroly dodržování pracovní doby zaměstnanců nebyly k dispozici údaje o začátku a konci pracovní doby zaměstnanců. Využití možnosti získávání údajů z vykázané práce zaměstnance na zakázkách by bylo teoreticky možné, ale některé zakázky neobsahovaly správně ukončený konec a nebylo jasné, jak s těmito údaji naložit.

V návrhu bylo zmíněno, že hlavní stránka bude obsahovat varování ukazatelů, které se uživateli ihned zobrazí v případě překročení limitů. Po nasazení systému na reálná data bylo zjištěno, že databáze obsahuje příliš mnoho údajů, které jsou mimo hlídané limity. Často to bylo zapříčiněno špatně uloženými údaji, jindy tím, že se jednalo zřejmě o atypické zakázky, které neměly uvedenou normu nebo počet kusů. V ERP systému jsou tato data interpretována uživateli podle jejich zvyklostí, ale pro účely strojového zpracování jsou bez znalostí experta nevhodná. Pro implementaci tohoto rozšíření by musela být databáze očištěna a uvedena do konzistentního stavu.

Všechny údaje jsou vypisovány většinou ve formě tabulek. Z důvodu možných změn a doplňování pohledů nebyla zvolena možnost zobrazení prostřednictvím grafů. Doplnění by spočívalo v analýze, jaké grafy by bylo vhodné použít, a nalezení knihovny, která splňuje vhodné licenční podmínky, eventuálně vytvoření vlastní. Samotné napojení už nebude obtížné i díky využití MVC návrhového vzoru, který je vhodný pro nasazování různých pohledů nad stejná data.

## Kapitola 8

# Závěr

V samém úvodu práce byly představeny metody operačního výzkumu, konkrétně síťová analýza, která se využívá pro optimalizaci projektů a určení správného sledu procesů. Byly zmíněny metody určení předpokládaného termínu zpracování projektu na základě znalosti historických dat.

Práce jako taková se zabývá způsoby řešení požadavků firmy Agerit s.r.o. na aplikaci, která by rozšířila funkce jejich stávajícího ERP systému. Jednotlivé požadavky byly zhodnoceny, co se týká věcné stránky, ale i způsobu implementace a přínosu pro daný systém. Šlo o způsoby, jak z dat systému určit statistiky, které vedoucímu pracovníkovi včas odhalí problémy ve výrobě a pomohou mu učinit vhodné rozhodnutí, jak pokračovat. Bylo zhodnoceno, jak sledovat produktivnost jak výroby, tak zaměstnanců.

Dále bylo nastíněno, jak zjistit, zda zakázky budou dokončeny včas. Byl navržen přístup, který vychází z aktuálních dat a upozorňuje na odchylky ihned, jak je to možné. Byly využity metody z úvodu práce. Tyto metody vyžadují přidanou režii a jejich použití musí být kompenzováno jejich přínosem.

Byl vytvořen základní návrh aplikace, ze kterého se bude dále vycházet a bude doplňován na základě dalších požadavků ze strany zadavatele práce.

Při analýze požadavků byly osloveny komerční společnosti, které nabízejí své produkty ve vyšší cenové relaci. Byly získány informace od dvou společností – Asseco Solutions, a.s. a Karat Software a.s.

Pro implementaci byly vybrány akce systému, které nejlépe odpovídaly povaze dodaných dat. Tedy analytické pohledy zobrazující informace z již dokončených zakázek. Byl implementován pohled analýzy průběhu zakázek, délky trvání operací v závislostech na časovém období, výkonnost zaměstnanců a produktivnost zakázek.

Některé pohledy navržené v úvodu práce nebylo možné provést. Především zobrazení síťového grafu, protože nebyly dostupné informace o napojení jednotlivých operací výroby. Vzhledem k vlastnostem síťového grafu a možnostech, které nabízí pro plánování a kontrolu plnění plánu zakázek by bylo vhodné stávající systém doplnit o tuto funkcionalitu.

Na závěr byly shrnuty další možná rozšíření, která by bylo možné do systému doplnit.

# Literatura

- [1] ERP Systémy. 2013.  
URL [http://www.ekonomickysoftware.com/erp\\_systemy.html](http://www.ekonomickysoftware.com/erp_systemy.html)
- [2] Evidence výroby PRACANT. 2013.  
URL <http://www.evidencnisystem.cz/>
- [3] Integer Types (Exact Value). MySQL 5.5 Reference Manual. 2013.  
URL <http://dev.mysql.com/doc/refman/5.5/en/integer-types.html>
- [4] Produkty systému Palstat CAQ. 2013.  
URL <http://www.palstat.cz/cs/produkty>
- [5] Vyberte si svůj informační systém - HELIOS.eu. 2013.  
URL <http://www.helios.eu/cz/prehled-produktu.html>
- [6] Řízení výroby - Modul Výroba - Helios Green. Helios Green. 2013.  
URL [https://forum.helios.eu/green/doc/cs/index.php?title=Řízení\\_výroby\\_-\\_modul\\_Výroba](https://forum.helios.eu/green/doc/cs/index.php?title=Řízení_výroby_-_modul_Výroba)
- [7] BERKA, P.: *Dobývání znalostí z databází*. Praha: Academia, 2003, ISBN 80-200-1062-9.
- [8] LACKO, L.: *Datové sklady, analýza OLAP a dolování dat*. Brno: Computer Press, 2003, ISBN 80-7226-969-0.
- [9] MORAVCOVÁ, E.; BAŇAŘOVÁ, J.: *Operační výzkum*. Ostrava: VŠB-TECHNICKÁ UNIVERZITA OSTRAVA, Regionální centrum celoživotního vzdělávání, 2003, ISBN 80-248-0365-8.
- [10] PECINOVSKÝ, R.: *Návrhové vzory*. Computer Press, vyd. 1. vydání, 2007.
- [11] SODOMKA, P.; KLČOVÁ, H.: *Informační systémy v podnikové praxi*. 2. aktualiz. a rozš. vyd., Brno: Computer Press, a.s., 2010, ISBN 978-80-251-2878-7.
- [12] VRÁNA, J.: *1001 tipů a triků pro PHP*. Brno: Computer Press, a.s., 2010, ISBN 978-80-251-2940-1.
- [13] ZENDULKA, a. k.: *Získávání znalostí z databází. Studijní opora*. FIT VUT v Brně, 2006.
- [14] ZIMOLA, B.: *Operační výzkum*. Zlín: VUT v Brně, Fakulta managementu a ekonomiky ve Zlíně, 2010, ISBN 80-214-1664-5.

# Příloha A

## Obsah CD

Na přiloženém CD je uložen vypracovaný systém (ve složce `/www/`), vzorová databáze (v souboru `/reporting.db.sql`) a pokyny k instalaci systému (v souboru `/install.txt`). Ve složce `/tex/` jsou uloženy zdrojové soubory této práce. Překlad zdrojových textů se provede pomocí příkazu `make`.