

Návrh a implementace backendové části software pro řízení výroby

Tereza Matulíková

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Tereza Matulíková**
Osobní číslo: **A18060**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Návrh a implementace backendové části software pro řízení výroby**
Téma práce anglicky: **Design And Implementation of the Backend Part of a Production Control Software**

Zásady pro vypracování

1. Nastudujte vybrané ERP systémy a vytvořte návrh aplikace pro řízení výroby.
2. Vytvořte technickou specifikaci jako podklad pro implementaci backendové části aplikace.
3. Implementujte backendovou část aplikace pomocí vhodných technologií.
4. Výsledné API zdokumentujte pomocí vybraného nástroje.
5. Výsledné řešení vhodně otestujte pomocí API testů a prezentujte výsledky.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Phalcon Docs: Phalcon Documentation – Introduction. Phalcon: High Performance PHP Framework – Phalcon Framework [online]. [cit. 2021-12-02]. Dostupné z: <https://docs.phalcon.io/4.0/en/introduction>
2. WUILLEUMIER SALEMME, Isabelle. What is Workflow Management? Pipefy [online]. 12. 11. 2021 [cit. 2021-12-02]. Dostupné z: <https://www.pipefy.com/blog/what-is-workflow-management/>
3. COOPER, Zara. A practical guide to writing technical specs. The OWERFLOW [online]. 6. 4. 2020 [cit. 2021-12-02]. Dostupné z: <https://stackoverflow.blog/2020/04/06/a-practical-guide-to-writing-technical-specs/>
4. The Unified Modeling Language [online]. [cit. 2021-12-02]. Dostupné z: <https://www.uml-diagrams.org>
5. ČÁPKA, David. Lekce 4 – UML – Doménový model. IT Network [online]. [cit. 2021-12-02]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-domenovy-model-diagram>
6. KERŔKOVSKÝ, Miloslav a Ondřej VALSA. Moderní přístupy k řízení výroby. 3., dopl. vyd. V Praze: C.H. Beck, 2012, xxi, 153 s. C.H. Beck pro praxi. ISBN 9788071793199.

Vedoucí bakalářské práce: **Ing. Petr Žáček, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **3. prosince 2021**

Termín odevzdání bakalářské práce: **23. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 24. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářské práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 20. 5. 2022

Tereza Matulíková, v. r.

podpis studenta

ABSTRAKT

Cílem teoretické části bakalářské práce je studium a popis vybraných ERP systémů a jejich modulů pro řízení výroby. Hlavním cílem je pak v praktické části vytvořit návrh jednoduché aplikace pro řízení výroby na základě nastudovaných poznatků a implementovat backendovou část aplikace. Součástí praktické části je dále zdokumentování výsledného API a vhodné otestování pomocí API testů.

Klíčová slova: ERP, HELIOS, ABRA, SAP, API, řízení výroby

ABSTRACT

The aim of the theoretical part of the bachelor thesis is the study and description of selected ERP systems and their modules for production management. The main goal is to design a simple application for production management in the practical part on the basis of the studied knowledge and to implement the backend part of the application. Part of the practical part is also documenting the resulting API and appropriate testing using API tests.

Keywords: ERP, HELIOS, ABRA, SAP, API, production management

Na tomto místě bych ráda poděkovala Ing. Petru Žáčkovi, PhD. za vedení práce, rady a připomínky.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 SEZNAM ERP SYSTÉMŮ	12
1.1 ABRA	12
1.1.1 ABRA Flexi	12
1.1.2 ABRA Gen	12
1.2 HELIOS	12
1.2.1 HELIOS Easy	12
1.2.2 HELIOS iNuvio	13
1.2.3 HELIOS Nephrite	13
1.2.4 HELIOS Red	13
1.3 SAP	13
1.3.1 SAP S/4HANA Cloud	13
1.3.2 SAP Business One	14
1.3.3 SAP Business ByDesign	14
2 VÝROBNÍ MODULY SYSTÉMU HELIOS INUVIO	15
2.1 TECHNICKÁ PŘÍPRAVA VÝROBY - ČÍSELNÍKY	15
2.1.1 Číselník náradí	15
2.1.2 Číselník zakázkových modifikací	15
2.1.3 Číselník pracovišť	16
2.1.4 Číselník strojů	16
2.1.5 Výrobní profese	16
2.1.6 Výrobní personální zdroje	16
2.1.7 Číselník tarifů	16
2.1.8 Číselník změn	16
2.1.9 Plánovací kalendáře	17
2.2 TECHNICKÁ PŘÍPRAVA VÝROBY - VYRÁBĚNÉ DÍLCE	17
2.2.1 Formulář Konstrukce a technologie	18
2.3 TECHNICKÁ PŘÍPRAVA VÝROBY - NAKUPOVANÉ MATERIÁLY	21
2.4 TECHNICKÁ PŘÍPRAVA VÝROBY - VÝROBNÍ A NÁKLADOVÁ STŘEDISKA	22
2.5 ŘÍZENÍ VÝROBY - VÝROBNÍ PLÁN	22
2.5.1 Zaplánování do výroby	22
2.6 ŘÍZENÍ VÝROBY - VÝROBNÍ PŘÍKAZY	22
2.6.1 Řady výrobních příkazů	23

2.7	ŘÍZENÍ VÝROBY - VÝROBNÍ OPERACE	24
2.7.1	Evidence výrobních operací	24
3	VÝROBNÍ MODULY SYSTÉMU ABRA GEN.....	25
3.1	NASTAVENÍ - ČÍSELNÍKY	25
3.1.1	Pracovníci ve výrobě.....	25
3.1.2	Pracoviště.....	25
3.1.3	Tarifní třídy.....	25
3.1.4	Typy technologických postupů	26
3.1.5	Neshody	26
3.1.6	Parametry řad požadavků na výrobu	26
3.1.7	Parametry řad výrobních příkazů	26
3.2	VÝROBA - TPV	27
3.2.1	Kusovníky	27
3.2.2	Technologické postupy.....	27
3.3	VÝROBA - PLÁNOVÁNÍ A ŘÍZENÍ.....	28
3.3.1	Požadavky na výrobu.....	28
3.3.2	Výrobní příkazy	28
3.3.3	Dílenský plán.....	29
3.4	VÝROBA - REALIZACE.....	29
3.4.1	Pracovní lístky	29
3.4.2	Kooperace	30
3.4.3	Dokončené výrobky.....	30
4	SROVNÁNÍ VÝROBNÍCH MODULŮ HELIOS INUVIO A ABRA GEN	32
II	PRAKTICKÁ ČÁST	33
5	NÁVRH APLIKACE.....	34
5.1	ENTITY.....	34
5.1.1	Doménový diagram.....	34
5.1.2	ERD	35
5.2	AKCE.....	36
5.2.1	Vytvoření	37
5.2.2	Smazání	37
5.2.3	Úprava.....	37
5.2.4	Seznam	37
5.2.5	Registrace uživatele	37
5.2.6	Přihlášení uživatele.....	38
5.2.7	Fronta práce	38

5.2.8	Seznam operací výrobního příkazu	38
5.2.9	Přiřazení a odebrání operace výrobního příkazu	38
5.2.10	Přiřazení materiálu a odebrání materiálu přiřazeného k produktu ...	38
5.2.11	Materiál potřebný pro výrobní příkaz	38
5.2.12	Evidence práce na operaci	38
6	IMPLEMENTACE	40
6.1	MODEL	40
6.1.1	Anotace třídy	40
6.1.2	Anotace atributů	40
6.1.3	Anotace metod	42
6.1.4	Generování migrací	42
6.2	DTO	42
6.3	MAPPER	43
6.4	CRUD	43
6.4.1	Vytvoření	43
6.4.2	Úprava	43
6.4.3	Čtení	44
6.4.4	Smazání	45
6.5	TVORBA DOTAZŮ POMOCÍ DOCTRINE QUERYBUILDER	45
7	DOKUMENTACE API	47
8	API TESTY	49
8.1	WORKREPORTCEST	49
8.1.1	Struktura testu	49
8.1.2	Výsledky testů	51
	ZÁVĚR	53
	SEZNAM POUŽITÉ LITERATURY	54
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	60
	SEZNAM OBRÁZKŮ	61
	SEZNAM TABULEK	62
	SEZNAM PŘÍLOH	63

ÚVOD

Jelikož v současné době přechází více a více firem z papírové formy na digitalizovaná řešení, poptávka po ERP systémech roste.[1] ERP systémy představují komplexní řešení, které nabízí širokou škálu funkcionalit pro řízení firmy například v oblasti výroby. Kvůli jejich komplexitě ale nemusí být jejich použití zrovna jednoduché, proto je motivací této práce funkcionalitu výrobních modulů ERP systémů nastudovat, navrhnout aplikaci a nabídnout tak firmám jiné, jednodušší řešení.

Jedním z cílů této bakalářské práce je seznámit se s existujícími ERP systémy. Teoretická část obsahuje podrobný popis součástí výrobních modulů ERP systémů HELIOS a ABRA, jež jsou jedny z nejpoužívanějších systémů vyvinutých českými společnostmi. Závěrečnou kapitolou teoretické části je srovnání modulů pro řízení výroby těchto systémů.

Na základě nastudování výrobních modulů vybraných systémů je v praktické části sepsán návrh aplikace. Aplikace má představovat zjednodušenou verzi modulů pro řízení výroby. V návrhu jsou popsány entity a vztahy mezi nimi pomocí doménového diagramu a datového modelu. Dále obsahuje podkapitoly, které nastíní, jakým způsobem bude řešena funkcionalita převzatá z ERP systémů.

Aplikace je implementována na základě vytvořeného návrhu pomocí vhodných technologií. Způsob implementování tříd a práce s objekty a akce výsledného API jsou demonstrovány na ukázkách kódu. Další součástí je způsob zdokumentování výsledného API a jeho otestování pomocí API testů. Implementace testů je rovněž demonstrována na ukázkách kódu.

I. TEORETICKÁ ČÁST

1 SEZNAM ERP SYSTÉMŮ

V následujících kapitolách jsou popsány jednotlivé druhy tří vybraných informačních systémů. ERP systémy ABRA a HELIOS se řadí mezi nejrozšířenější systémy vyvinuté českými společnostmi [2] [3], řešení společnosti SAP je pak jedním z nejpoužívanějších systémů celosvětově.[4]

1.1 ABRA

Česká společnost ABRA Software nabízí dva ERP systémy, přičemž každý je určen pro jinou velikost firmy.

1.1.1 ABRA Flexi

ABRA Software představuje svůj produkt ABRA Flexi jako chytrý software pro menší byznys. Tento moderní informační systém nabízí dvě varianty, a to aplikaci s webovým rozhraním a desktopovou aplikaci. Mezi těmito variantami je rozdíl ve způsobu používání a v rozsahu funkcí, které podporují. ABRA Flexi disponuje několika moduly, například modulem Prodej, Zboží, Zaměstnanci nebo Účetnictví, neobsahuje však modul pro výrobu.[5]

1.1.2 ABRA Gen

Komplexní informační systém nese název ABRA Gen. Tento produkt je určen pro správu a řízení chodu střední nebo větší firmy. Variabilita tohoto ERP systému mu zaručuje uplatnění ve všech odvětvích. Díky velkému množství modulů pokrývá každou firemní oblast a může tak sloužit jako jediný informační systém ve firmě. Vedle nabízených modulů, do kterých patří například modul Finance a účetnictví, Mzdy a personalistika nebo Sklad, nabízí i modul Výroba.[6]

1.2 HELIOS

Asseco Solutions má na svém kontě širokou škálu informačních systémů pro řízení chodu firmy. V následujících odstavcích jsou popsány jejich vybrané produkty.

1.2.1 HELIOS Easy

Pro malé a začínající firmy přináší řešení firma Asseco Solutions v podobě informačního systému HELIOS Easy. Tento systém nabízí vše potřebné pro kompletní zpracování podnikových agend, ale je možné ho dále rozšiřovat. Mezi obsažené moduly patří Účetnictví, Nákup a prodej, Kniha jízd nebo Výroba.[7]

1.2.2 HELIOS iNuvio

Předchůdce tohoto systému HELIOS Orange patří mezi nejpoužívanější systémy a po desetiletí byl považován za špičku na trhu ERP v rámci středně velkých podniků. Nabízí komplexní pokrytí veškerých firemních procesů od finančního řízení přes personalistiku až po výrobu a tím dokáže pomoci v řízení chodu firmám z absolutní většiny oborů. Nejnovější edice systému HELIOS Orange dostala název iNuvio. Původně se jednalo pouze o vylepšení stávajícího systému HELIOS Orange, tato nástavba se ale stala natolik úspěšnou, že společnost Asseco Solutions dala vzniknout samostatnému produktu, jež by měl plně nahradit svého předchůdce.[8][9]

1.2.3 HELIOS Nephrite

Z produktů vyvíjených společností Asseco Solutions je pro velké a středně velké firmy určen HELIOS Green. Jeho výrobce ho považuje za svou vlajkovou loď. Tento moderní informační systém se přizpůsobí i těm nejnáročnějším požadavkům společnosti. Nedílnou součástí je například modul Ekonomika, Sklady a logistika nebo Výroba. Během několika let chce výrobce nahradit tento systém jeho následovníkem, kterým je HELIOS Nephrite. HELIOS Nephrite cílí na velké firmy pouze na českém a slovenském trhu, jelikož chce Asseco Solutions soustředit veškerou pozornost na specifika místního trhu a podporovat tak lokální firmy.[10][11]

1.2.4 HELIOS Red

Začínající firmy nebo podnikatelé mohou využít pro zpracování podnikových agend HELIOS Red. Tento ekonomický systém není omezen na určitý obor, obsáhne proto širokou škálu podnikání s různým zaměřením. Personalistika a mzdy, Pokladní prodej nebo Skladová evidence, to je výčet jen některých z dostupných modulů v tomto systému.[12]

1.3 SAP

Nabídku ERP systémů od společnosti SAP, jež patří mezi přední světové výrobce softwaru pro řízení obchodních procesů, tvoří následující produkty.

1.3.1 SAP S/4HANA Cloud

Komplexní ERP systém SAP S/4HANA Cloud díky integrovaným inteligentním technologiím, jako je umělá inteligence nebo strojové učení, pomáhá společnostem se přizpůsobit novým obchodním modelům a probíhajícím změnám a používat prediktivní

funkce. Nabízí řešení a postupy pro různá odvětví. Vedle modulu Výroba jsou součástí tohoto systému například moduly Obchod nebo Finance.[13][14][15]

1.3.2 SAP Business One

Pro potřeby malých podniků je navržen software SAP Business One. Jeho úkolem je růst společně s podnikem, být flexibilní, modulární a zjednodušovat procesy a každodenní činnosti. Obsahuje moduly pro finanční řízení, nákup a řízení zásob, ale také rozsáhlé možnosti pro specifické obory, jako například funkce pro maloobchod nebo výrobu.[16][17]

1.3.3 SAP Business ByDesign

Produkt SAP Business ByDesign nabízí podobné možnosti a moduly jako systém SAP Business One. Tento produkt je však více zaměřen na podniky střední velikosti s větším počtem uživatelů. Oproti SAP Business One, který je lehce přizpůsobitelný specifickým potřebám podniku a pro který existují možnosti rozšíření pomocí mnoha doplňků, je SAP Business ByDesign v tomto ohledu ochuzen. V základu nelze procesy tohoto software přizpůsobovat zákaznickovým potřebám, ale lze jej také rozšířit různými doplňky, kterých ale není zdaleka tolik.[18]

2 VÝROBNÍ MODULY SYSTÉMU HELIOS INUVIO

Informační systém HELIOS iNuvio nabízí dva moduly v kategorii Výroba: Technická příprava výroby a Řízení výroby.[19] Modul Technická příprava výroby je určen pro oblast konstrukční a technologické přípravy výroby. Jeho hlavním úkolem je detailní evidence a zpracování dat o konstrukci a technologii výroby a podpora termínovaného změnového řízení pro zachování historie všech úprav těchto dat.[20] Tato data tvoří společnou základnu obou zmíněných modulů a jsou nezbytným vstupem do modulu Řízení výroby. Modul Řízení výroby eviduje rozpracovanou výrobu a obsahuje funkcionalitu pro zadávání výrobního plánu. Na základě výrobního plánu jsou pak vytvořeny výrobní příkazy, které specifikují výrobní dávky daných dílců. Pro představu o kapacitním vytížení jednotlivých zdrojů slouží kapacitní plánování jednotlivých výrobních požadavků.[21]

2.1 Technická příprava výroby - číselníky

Nedílnou součástí modulu jsou číselníky, které slouží tak jako adresáře, ze kterých je možné vybírat položky a odkazovat se na ně při vytváření různých záznamů.

2.1.1 Číselník nářadí

Řadí se zde veškeré nářadí a přípravky potřebné pro výrobu finálního výrobku. Položka nářadí by se zpravidla měla vázat ke konkrétní operaci technologického postupu. Jedná se o kmenovou kartu typu nářadí. Kmenová karta představuje nějakou komoditu neboli zboží. Jednoznačná identifikace je zaručena kombinací skupiny zboží a registračního čísla nářadí.[22]

Do skupiny rozšiřujících atributů se řadí:

- typ nářadí - vratné/nevratné (hodnota ovlivňuje výslednou cenu za opotřebení nářadí),
- typ životnosti - udává, v jakých jednotkách se eviduje životnost nářadí,
- životnost - uvádí se buď v počtech užití (množstevní životnost) nebo v době použití (časová životnost).

2.1.2 Číselník zakázkových modifikací

V případě, že se liší způsob výroby dílce od jeho standardní definice v technické přípravě výroby, je založen Kód zakázkové modifikace. Pod tímto kódem je pak možné provést úpravy v TPV a v případě atypické výroby s nimi pracovat. Bez zadání zakázkové modifikace se pracuje se standardním TPV dílce.[23]

Úpravy pro zakázkovou výrobu je možné provést například v Konstrukci a technologii vyráběného dílce nebo v kalkulační ceně materiálu. Úpravy nepodléhají změnovému řízení, pouze se sleduje informace, zda se jedná o schválený či neschválený záznam.

2.1.3 Číselník pracovišť

Pod pojmem pracoviště si můžeme představit pracovní místo, skupinu strojů nebo konkrétní stroj. Při tvorbě technologického postupu můžeme operaci specifikovat na určité pracoviště. Jednoznačný identifikátor pracoviště se skládá z čísla střediska a označení pracoviště, proto musí mít každé pracoviště určené středisko. Dále se pracovišti definuje například název, počet zaměstnanců, počet strojů nebo kapacita.[24]

2.1.4 Číselník strojů

Pracoviště lze dále dělit na stroje, což jsou pracovní místa s definovanou výrobní kapacitou. Stroje vstupují do kapacitního plánování, v rámci čehož jsou na ně rozděleny výrobní operace.[25]

2.1.5 Výrobní profese

Pokud je potřeba určitá profese pro provedení výrobní operace, uvádí se při definování technologického postupu vyráběného dílce. Profese musí mít jednoznačný identifikátor, dále může obsahovat název a delší popis.[26]

2.1.6 Výrobní personální zdroje

Personální zdroj představuje plánovací jednotku zastupující jednoho člověka. Na personální zdroj může být navázán zaměstnanec z Číselníku zaměstnanců z modulu Mzdy, tato vazba může vzniknout importem vybraných zaměstnanců do tohoto číselníku. Dále může být výrobní personální zdroj přiřazen do jedné a více výrobních profesí. Jednoznačným identifikátorem personálního zdroje je kód.[27]

2.1.7 Číselník tarifů

Výrobní tarifní třídy jsou podkladem pro tvorbu technologických postupů a používají se při výpočtu mzdových nákladů na provedení technologické operace v dané tarifní třídě. Hodnota tarifu vyjadřuje mzdu za jednotku času.[28]

2.1.8 Číselník změn

Číselník změn je jedním z nejdůležitějších číselníků tohoto modulu, představuje totiž evidenci veškerých úprav v rámci TPV. Záznamy v číselníku obsahuje změny v tech-

nické dokumentaci, například opravy kusovníku, opravy v technologickém postupu nebo změny ve výrobní dokumentaci.[29]

Vedle čísla a názvu změny je důležitým atributem Datum platnosti, který určuje, od jakého okamžiku začne daná změna platit. Datum platnosti nelze nastavit v minulosti, lze ale nastavit změnu s Datem platnosti v budoucnosti, od tohoto data se poté změny projeví v navazujícím modulu Řízení výroby. Pokud je změna platná, automaticky obsahuje příznak Platnost. Atribut Řada změny svým Druhem definuje oblast změnování, ve kterých lze tuto změnu použít.

Datum TPV určuje platnost informace, pokud tato informace podléhá změnovému řízení. Při změně nastavení data TPV se zobrazí informace, jejíž hodnota je platná ke zvolenému datu TPV.

2.1.9 Plánovací kalendáře

Tyto kalendáře slouží k definování pracovní doby jednotlivých výrobních zdrojů jako jsou pracoviště, stroje nebo výrobní personální zdroje. Plánovací kalendář, který je označen jako výchozí, je při vytváření výrobního zdroje automaticky přiřazen pro tento nový záznam. Výchozí kalendář je využíván například k rozlišení pracovních a nepracovních dnů při zaplánování výrobního plánu v modulu Řízení výroby.[30]

2.2 Technická příprava výroby - vyráběné dílce

Důležitým prvkem v procesu výroby jsou vyráběné produkty, které v rámci TPV představují vyráběné dílce.

Dílcem se rozumí finální výrobek, součástka nebo polotovár, který je přímo ve firmě vyráběn či kompletován. Nejedná se tedy o nakoupený, dosud neopracovaný materiál ani o nakoupenou součástku bez opracování. V případě vyráběného dílce se jedná o kmenovou kartu s příznakem Dílec.[31]

Vyráběný dílec může obsahovat tyto informace:

- jednoznačný identifikátor - kombinace registračního čísla a skupiny zboží, což je hodnota z číselníku skupin zboží,
- název,
- kmenové středisko - výrobní středisko zodpovědné za výrobu daného dílce,
- jednotka - hodnota z číselníku měrných jednotek, která udává, v jaké jednotce zboží evidujeme,
- rozměry dílce - mohou být využity pro tiskové výstupy nebo do vzorců pro výpočet spotřeby materiálu,

- číslo výkresu,
- označení Režijní položka - dílec se započítává do kalkulací a tiskových výstupů, nevydává se ze skladu do výroby a není na něj generován výrobní příkaz,
- označení Montáž - na dílec není vystaven samostatný výrobní příkaz a také nemá technologický postup, který bývá zpravidla součástí vyšší nadřazené sestavy,
- datum platnosti - datum platné změny v konstrukci a technologii ve vztahu k nastavenému datu TPV,
- různé plánovací parametry.

Následující **plánovací parametry** se využívají při zaplánování výrobního plánu do výroby, jelikož se výrobní plán odkazuje na vyráběný dílec.

- Typ plánovací doby - možné hodnoty: zadaný hrubý čas ve dnech, zadaný čistý čas v hodinách, vypočtený čas z technologického postupu,
- průběžná doba výroby dávky - konstanta ve dnech/hodinách, nebo vypočtený čas
- typ průběžné doby
 - pevná - doba výroby dílce libovolného množství je rovna průběžné době výroby dávky,
 - proměnná - doba výroby plánovací dávky je rovna průběžné době výroby dávky,
- plánovací dávka,
- doba přerušení mezi příkazy - odstup mezi příkazem na daný dílec a příkazem na nadřazený dílec.

K vyráběnému dílci lze přiřadit libovolný počet dokumentů, jako například výkresy nebo náčrty a prohlížení nebo změna údajů daného dílce lze provádět v rámci **formuláře Konstrukce a technologie** .

2.2.1 Formulář Konstrukce a technologie

Formulář se skládá ze sedmi částí:

- kusovníkové vazby,
- technologický postup,

- vazby nářadí,
- vazby OPN,
- vedlejší produkty,
- výrobní dokumentace,
- parametry změny.

Kusovníkové vazby definují materiály a dílce (polotovary) potřebné k výrobě dané součásti, polotovaru či finálního výrobku platné k vybranému datu TPV. Položky, které vstupují do daného dílce, nesou například informaci o kompletovaném dílci, odkaz na položku kusovníku, velikost dávky TPV, množství položky kusovníku ve vztahu k dávce TPV, nebo číslo operace technologického postupu, na kterou je uvedený materiál či polotovar potřeba. Kusovníkové vazby se definují všeobecně na celý dílec nebo přímo na operace. **Kusovník** výrobku obsahuje přehled všech materiálů a polotovarů, které jsou potřebné pro výrobu daného dílce.

Technologický postup slouží k evidenci posloupnosti jednotlivých operací nutných pro výrobu daného dílce. Popisuje způsob výroby finálního výrobku z položek uvedených v kusovníkových vazbách. Operace jsou číslovány podle toho, v jaké posloupnosti jsou při výrobě prováděny. Pro rychlejší tvorbu technologického postupu je možné využít předdefinované typové operace.

Operace jsou rozděleny na tři druhy:

- jednicové operace - klasické operace prováděné ve vlastní výrobě, ohodnocené časem a mzdou, uvádíme u nich číslo střediska či pracoviště
- režijní operace - časové operace, které se většinou do postupu nezahrnují, ovlivňují náklady za režii pracoviště a výpočet kapacity daného střediska či pracoviště,
- kooperace - operace, jež provádí externě jiný dodavatel.

Informace zahrnuté v operaci mohou být následující:

- odkaz na vyráběný dílec,
- typ operace,
- číslo operace,
- název operace,
- konstrukční a technologická dávka,

- text - detailní popis operace,
- odkaz na pracoviště, na kterém se bude daná operace provádět (uvádí se u jednicové či režijní operace),
- odkaz na konkrétní stroj, na kterém se bude daná operace provádět (uvádí se u jednicové či režijní operace),
- tarif - hodnota z číselníku tarifů, dle které se dopočítávají mzdy u jednicové operace,
- různé údaje o čase, například čas na přípravu stroje (seřízení) či obsluhy, celkový čas stroje/obsluhy potřebný pro výrobu dávky TPV daného dílce,
- počet kusů vyráběných jednou operací,
- odkaz na výrobní profesi potřebnou pro provedení operace.

Konstrukční a technologická dávka neboli dávka TPV stanovuje velikost dávky, na kterou je konstrukce a technologie daného výrobku definovaná. Standardně je dávka rovna jedné, v tomto případě se kusovníky, postupy a nářadí definují na jednu evidenční jednotku vyráběného dílce.

U operací existuje možnost vytvoření **alternativ**. Alternativní pracoviště znamená, že se konkrétní operace prováděná na jednom pracovišti může provádět na pracovišti jiném. Dále lze definovat náhradu za základní operaci, tzv. alternativní operaci.[32]

Vazby nářadí definují potřebné přípravy a nářadí pro výrobu daného dílce, z pravidla by se tyto vazby měly vztahovat ke konkrétní operaci technologického postupu.

Informace obsažené ve vazbách:

- odkaz na vyráběný dílec,
- odkaz na nářadí - hodnota z číselníku nářadí,
- číslo operace technologického postupu, na kterou je dané nářadí navázáno,
- počet konkrétních přípravků/nářadí,
- počet užití nebo doba použití daného nářadí - souvisí s životností nářadí, která se uvádí buď v počtech užití (množstevní životnost) nebo v době použití (časová životnost).

Vazby ostatních přímých nákladů neboli **vazby OPN** uvádí náklady ovlivňující výslednou kalkulaci výrobku. Mezi ostatní přímé náklady patří například dodatečné náklady za elektřinu, zapůjčení nářadí, náklady na dopravu aj.

Vedlejší produkty slouží jako evidence výrobních odpadů vzniklých v rámci konkrétní operace technologického postupu výroby dílce. Tyto výrobní odpady mohou dále posloužit ke zpracování jako materiály nebo polotovary. Jedná se o kmenovou kartu s příznakem Vedlejší produkt.[34]

Atributy odlišné od atributů ostatních komodit:

- sklad, na který je vedlejší produkt odveden z výroby,
- nosný dílec pro výrobu - pokud vzniká více hlavních produktů, je jeden zvolen jako nosný dílec pro výrobu a zbytek je zařazen do vedlejších produktů.

Pomocí **výrobní dokumentace** se evidují dokumenty ve výrobě. Oproti standardním dokumentům přiřazeným k výrobnímu dílci jsou výrobní dokumenty změnované, můžeme tedy jejich úpravy sledovat pod změnou z číselníku změn. Zároveň jsou také verzované, jejich úpravy je možné sledovat v rámci jedné změny.[33]

U výrobních dokumentů je možno uvést tyto údaje:

- kategorie - hodnota z číselníku kategorií výrobních dokumentů; kategorie udává, zda se na výrobním dokumentu bude provádět verzování a definuje přístupová práva k dokumentu pro uživatele nebo uživatelské role,
- číslo dokumentu - systém přidělí dokumentu číslo jako jednoznačný identifikátor,
- verze,
- jméno souboru - jméno původního souboru,
- typ dokumentu - přípona původního souboru.

2.3 Technická příprava výroby - nakupované materiály

Nakupovaný materiál je definován jako neopracovaný materiál, který vstupuje do výroby. Řadí se sem všechny nakoupené polotovary, součástky i celky, jež se bez jakékoli úpravy použijí při procesu výroby finálního výrobku. Jedná se o kmenovou kartu s příznakem Materiál.[35]

Stejně jako vyráběný dílec je možné nakupovaný materiál jednoznačně identifikovat pomocí skupiny zboží a registračního čísla. Dále může obsahovat název a jednotku. Od vyráběného dílce se liší například informací o dodavateli daného materiálu, což je hodnota z číselníku organizací.

Do dostupných funkcí patří například přiřazení libovolného počtu dokumentů nebo vytvoření kopie. U nakupovaného materiálu lze také definovat jeho cenu. Informace o ceně podléhá změnovému řízení a ovlivňuje výpočet plánované kalkulace výsledného výrobku.

2.4 Technická příprava výroby - výrobní a nákladová střediska

Patří zde výrobní, montážní a nákladová střediska. Nákladová střediska jsou potřeba pro účtování, číselník výrobních a montážních středisek je nutný pro definování číselníku pracovišť.[36]

Následující nastavení jsou uplatněna při výrobě dílce, jehož kmenovým střediskem je dané středisko: sklad materiálů, výrobní mezisklad, sklad hotových výrobků a nákladové středisko pro potřeby účtování nedokončených výrobků.

2.5 Řízení výroby - výrobní plán

Výrobní plán slouží k evidenci aktuálně požadovaných výrobků i pro plánování výroby do budoucna. Výrobní plán má přiřazenou řadu výrobního plánu. Výrobní řady definují chování výrobního plánu pro určité situace a tím od sebe odlišují různé druhy výroby.

Výrobní plán se odkazuje na vyráběný dílec (většinou jde o finální výrobek), může se odkazovat i na zakázku nebo například na zakázkovou modifikaci vyráběného dílce. Jednoznačným identifikátorem je číslo vygenerované systémem. Dále se na výrobním plánu může vyskytovat informace o požadovaném počtu kusů vyráběného dílce, do jakého data by měla být výroba dílce ukončena a informace o procentuálním plnění výrobního plánu dle stavu operací na navázaných výrobních příkazech.

2.5.1 Zaplánování do výroby

Při procesu zaplánování výrobních plánů do výroby jsou generovány jednotlivé výrobní příkazy na všechny vyráběné dílce potřebné pro výrobu finálních výrobků. Při výpočtu plánovaného termínu zahájení a ukončení výrobního příkazu se pracuje s časovými atributy daného vyráběného dílce a zároveň je brán zřetel na pracovní kalendář firmy, jež je definován jako výchozí v číselníku Plánovací kalendáře.[37]

2.6 Řízení výroby - výrobní příkazy

Výrobní příkaz zastupuje určitou výrobní dávku vyráběného dílce a má přiřazenou řadu výrobního příkazu, která definuje jeho chování. Fáze životního cyklu výrobního příkazu mohou zahrnovat zadání příkazu do výroby, odvedení výrobních operací nebo odvedení hotového výrobku na sklad. K výrobním příkazům lze také přiřadit dokumenty. Při zadávání příkazů do výroby je možné pro tyto příkazy generovat výrobní čísla.[38]

U výrobních příkazů evidujeme stav, který může nabývat následujících hodnot:

- pořízeno - je evidován systémem,
- předzpracováno - je připraven k zadání do výroby,

- zadáno - je zadán ve výrobě,
- pozastaveno,
- ukončeno,
- uzavřeno.

2.6.1 Řady výrobních příkazů

Jak bylo uvedeno, řady výrobních příkazů definují chování výrobních příkazů v různých situacích pomocí nastavení voleb a tím slouží i k odlišení různých druhů výroby. Volby je možné nastavit v několika sekcích, např. předzpracování příkazu, zadání příkazu do výroby, odvedení příkazů, evidence operací, výrobní čísla aj.[39]

Pro zadání příkazu do výroby existují například tyto volby:

- kontrolovat zadání kmenového střediska - při výběru této volby nelze zadat do výroby výrobní příkaz bez uvedeného kmenového střediska,
- kontrolovat zadání zakázky - zvolením této volby nebude do výroby zadán příkaz bez definované zakázky,
- kontrolovat existenci kusovníku - pokud při zvolení této volby dílec na výrobním příkaze neobsahuje v kusovníku alespoň jednu položku, nelze zadat tento příkaz do výroby,
- kontrolovat existenci technologického postupu - je-li tato volba nastavena a technologický postup pro dílec na výrobním příkaze neobsahuje ani jednu operaci, příkaz nelze zadat do výroby.

Sekce evidence kooperací obsahuje následující volby:

- kontrolovat ukončení předchozí operace - nastavením této volby proběhne při odvádění operace kontrola, zda bylo na předchozí operaci zaevidováno minimálně stejné množství dobrých kusů, jako na právě odváděné operaci,
- zdroj nákladového střediska evidence operací - pomocí této volby lze specifikovat, jaké nákladové středisko bude přednastaveno při evidenci výrobních operací. Možnosti přednastavení střediska:

- středisko pracoviště, na kterém probíhá operace,
- středisko útvaru zaměstnance, který provedl danou operaci,
- kmenové středisko z výrobního příkazu.

2.7 Řízení výroby - výrobní operace

Výrobní operace představují generované požadavky na provedení konkrétních úkonů. Výrobní operace obsahují kromě údajů zmíněných v kapitole Vyrábění dílce - Technologický postup další atributy, jako například odkaz na výrobní příkaz, odkaz na zakázku, odkaz na dílec nebo informaci o procentuálním plnění na základě zaevidovaných dobrých kusů. V rámci jednoho příkazu lze operaci jednoznačně identifikovat pomocí dokladu.[40]

2.7.1 Evidence výrobních operací

Při evidenci výrobních operací se uvádí například informace o směně, pracovišti, stroji nebo zaměstnanci, který danou operaci prováděl. Pokud se jedná o kooperační operaci, je možné uvést odkaz na kooperační objednávku. Kooperační objednávka slouží jako evidence požadavků kooperačních operací na dodavatele, kteří zajišťují provedení těchto kooperací.[41]

V rámci tohoto procesu je možné evidovat i míru opotřebení použitého předepsaného nářadí, kdy se odkazuje na požadavek na nářadí daného výrobního příkazu a uvádí se počet užití, nebo doba použití v závislosti na typu životnosti nářadí.

Na výrobních operacích je možné vedle dobrých kusů evidovat i špatné kusy neboli zmetky. Existují tři druhy zmetků:

- interně opravitelné - daná operace se provede znovu,
- opravitelné - je možné opravit nebo přepracovat na jiný výrobek, a to v rámci existujícího nebo nově vytvořeného výrobního příkazu,
- neopravitelné.

3 VÝROBNÍ MODULY SYSTÉMU ABRA GEN

Modul Výroba informačního systému Abra Gen je rozdělen na agendy, které slouží pro evidenci, řízení a plánování výrobních procesů. Agendy výrobního modulu jsou rozděleny do dvou skupin: Výroba a Nastavení. Skupina Výroba - nastavení je nezbytná pro správnou činnost výrobních agend kvůli jejím podpůrným číselníkovým a definičním agendám. Ve skupině Výroba jsou obsaženy agendy, které jsou dále rozděleny podle funkcí do několika okruhů: okruh TPV související s technologickou přípravou výroby, v okruhu Plánování a řízení jsou obsaženy požadavky pro spuštění výroby nebo výrobní příkazy a okruh Realizace zastřešuje samotný průběh výroby.[42]

3.1 Nastavení - Číselníky

Stejně tak jako u systému HELIOS iNuvio jsou nedílnou součástí výrobních modulů i číselníky.

3.1.1 Pracovníci ve výrobě

Obsahem tohoto číselníku je seznam pracovníků ve výrobě s přiřazeným střediskem a tarifní třídou. Zdrojem číselníku je číselník Adresář osob z modulu Adresář. Na pracovníka může být navázán konkrétní uživatel systému ABRA Gen z číselníku Uživatelů z modulu Administrace. Pokud přihlášený uživatel zaeviduje práci na operaci, je tato připsána jemu navázanému pracovníkovi.[43]

3.1.2 Pracoviště

V tomto číselníku jsou definovány pracoviště a stroje, na kterých probíhá výroba. Zde je povinná vazba na středisko z číselníku Střediska z modulu Administrace, dále se uvádí hodinová sazba a průměrná kapacita pracoviště či stroje v hodinách za den.

Pracovištěm může být např. montážní linka, část dílny pro kompletaci nebo skupina strojů se stejným účelem. Odkaz na pracoviště je povinnou položku při vytváření technologických postupů. proto se v případě použití kooperací zavádí fiktivní pracoviště s nulovými sazbami.[44]

3.1.3 Tarifní třídy

Číselník tarifních tříd slouží jako evidence tarifních tříd včetně hodinových sazeb, jež slouží k vyjádření mzdových nákladů na pracovníka nebo operaci. Každý pracovník v číselníku Pracovníci ve výrobě musí mít přiřazenou položku z tohoto číselníku.[45]

3.1.4 Typy technologických postupů

Pomocí typů technologických postupů se odlišují různé druhy výroby, např. výroba kusová, malosériová nebo výroba v kooperaci.[46]

3.1.5 Neshody

V číselníku neshod se evidují typy neshod, jež jsou využívány při vytváření záznamu o nápravě neshody v rámci evidence práce na pracovním lístku.[47]

3.1.6 Parametry řad požadavků na výrobu

V tomto číselníku, jak již plyne z názvu, lze nastavit parametry pro každou řadu požadavků na výrobu podle charakteru výroby. Toto nastavení se zohledňuje při generování výrobního příkazu z požadavku.[?]

Výčet některých parametrů řad požadavků na výrobu:

- schvalování požadováno - hodnota “Ano” způsobí, že před vytvořením výrobního příkazu z požadavku musí být schváleny údaje o výrobní kapacitě, kooperacích, zajištění materiálu a financí,
- hodinová sazba pro ocenění výrobku - pro ocenění výrobku na pracovním lístku se použije hodinová sazba dle pracoviště nebo tarifní třídy,
- tarifní třída pro ocenění výrobku - je-li v předchozím parametru zvolena možnost “Tarifní třída”, v tomto parametru lze definovat, zda se použije tarifní třída uvedená u pracovníka, operace nebo ta třída s vyšší hodnotou z předchozích možností
- firma pro předvyplnění - předvyplní se na požadavku na výrobu jako firma, pro kterou se bude daný výrobek vyrábět.

3.1.7 Parametry řad výrobních příkazů

Obdobně jako u předchozího číselníku lze pomocí tohoto číselníku nastavit parametry řad výrobních příkazů podle druhu výroby. Volby nastavené pomocí těchto parametrů se zohledňují při vytvoření výrobního příkazu.[49]

Výčet některých parametrů řad výrobních příkazů:

- sekce parametrů ovlivňující cenu hotového výrobku
- materiál - hodnota “Ano” znamená, že se bude do ceny výrobku započítávat hodnota spotřebovaného materiálu,

- kooperace - zda se bude do ceny výrobku započítávat hodnota kooperace,
- pevná cena - při volbě hodnoty "Ano" se při ukončení výroby a příjmu hotového výrobku na sklad použije pevná cena uvedená v kusovníku daného výrobku,
- sekce parametrů pro souhrnné pracovní listky:
- parametry pro nastavení, zda se budou souhrnné pracovní listky seskupovat například dle výrobního příkazu, střediska, zakázky nebo projektu
- datum ukončení ve stejný - seskupení v závislosti na datu, ve kterém byly pracovní listky ukončeny (den, týden, měsíc, období).

3.2 Výroba - TPV

Skupina TPV obsahuje agendy pro technologickou přípravu výroby.

3.2.1 Kusovníky

Agenda Kusovníky modulu Výroba slouží k evidenci výrobků a polotovarů vyráběných v rámci firmy. Tyto kusovníky se dále využívají pro vytváření požadavků na výrobu a výrobních příkazů. Kusovníky polotovarů a výrobků obsahují řádky, jež pomocí **skladových karet** materiálu definují, z čeho se daný polotovar či výrobek skládá.[52]

Každá skladová položka má přiřazenou skladovou kartu, která se též nazývá Hlavní skladová karta. Údaje uvedené na této kartě nejsou závislé na konkrétním skladu, tzn. platí pro danou položku a pro všechny sklady. Mezi tyto údaje patří například kód a název zboží, seznam dodavatelů nebo jednotky, ve kterých se daná položka eviduje. Ke každé skladové kartě existují dílčí skladové karty, které se vážou ke konkrétnímu skladu. Jsou zde uvedeny například informace o počtu dané položky na skladě nebo poslední nákupní cena.[51]

Pokud má skladová karta zatrženo parametru Výrobek, značí to, že se jedná o výrobek nebo polotovar vyráběný v rámci vlastní firmy.

Agendy pro správu skladových karet jsou součástí modulu Skladové hospodářství.

3.2.2 Technologické postupy

Význam technologického postupu se neliší od významu uvedeného v kapitole HELIOS iNuvio - Technologický postup. Technologický postup může být přiřazen konkrétní skladové kartě výrobku nebo polotovaru a definovat tak operace prováděné v procesu výroby pro určitou skladovou kartu, nebo může být vytvořen bez této vazby.[53]

Jedné skladové kartě může být přiřazeno více technologických postupů, pokud se tyto postupy liší svým typem, což je hodnota z číselníku Typy technologických postupů.

Dále se přímo na technologickém postupu definuje množství a jednotka nebo odkaz na zakázku z číselníku zakázek.

Na operaci technologického postupu se kromě čísla, názvu a popisu uvádí například tarifní třída, pracoviště/stroj nebo normovaný kusový čas, což je doba potřebná pro provedení operace na jednom výrobku. Operaci lze označit jako ukončující, tzn. po jejím provedení se výrobek považuje za dokončený.

Atribut Dávkově indikuje, zda je operace dávková či kusová. Dávková operace se celá eviduje na jeden pracovní lístek a je u ní uvedeno množství, na jakém byla provedena. Kusová operace se vyskytuje pouze u výrobků se sériovými čísly, pracovní lístek se tedy eviduje pro každé sériové číslo zvlášť.

Materiály pro operaci obsahují odkazy na skladové karty materiálů a informaci o množství a jednotce daného materiálu pro danou operaci.

V neposlední řadě je možné k operaci přiřadit obrázky.

3.3 Výroba - Plánování a řízení

Pokud je potřeba něco vyrobit, nejprve vzniká požadavek na výrobu dané položky. Poté přichází na řadu výrobní příkaz, v rámci kterého se tato položka vyrobí. Proces výroby položky lze plánovat pomocí dílenského plánu.

3.3.1 Požadavky na výrobu

Požadavek na výrobu je doklad, který nese informace o položkách, které se mají vyrobit. Odkazuje se na skladové karty jednotlivých vyráběných položek a obsahuje informaci o jednotce a množství. Ke každé definované vyráběné položce se váže kusovník a technologický postup. V rámci vytváření požadavků se tyto kusovníky a technologické postupy mohou upravit do finálního stavu.[54]

U požadavků na výrobu se uvádí řada dokladu, která definuje chování dokladu, dále se uvádí období, do kterého je tento požadavek zahrnut, sklad pro příjem hotových výrobků nebo firma, pro kterou se daný výrobek vyrábí.

3.3.2 Výrobní příkazy

V této agendě se evidují výrobní příkazy, které se vytváří z požadavků na výrobu. Obsahují vyráběné položky, jejich kusovníky a technologické postupy.

Pomocí výrobních příkazů lze sledovat průběh výroby, například postup prací na operacích, zda jsou dodržovány plánované termíny aj.

Výrobní příkaz musí být spuštěn, aby mohla být zahájena výroba daných vyráběných položek. Po spuštění výrobního příkazu lze evidovat práce na výrobku prostřed-

nictvím pracovních lístků, vydávat materiál a přijímat hotové výrobky na sklad.[55]

3.3.3 Dílenský plán

Agenda Dílenský plán slouží jako evidence dílenských úkonů, přehled plánované práce a také jako podklad pro složitější Kapacitní plánování, což je samostatný modul existující mimo modul Výroba.

Jeden nebo více dílenských úkonů vzniká z konkrétní operace technologického postupu. Při vytváření dílenského úkonu z operace je možné upravovat informace o pracovníkovi, pracovišti, datu nebo směně, jež byly přeneseny z dané operace. Rozdělováním úkonů lze rozkládat zátěž na více pracovišť, případně rozdělovat práci více pracovníkům.

Dílenský úkon nese název operace, ze které byl vytvořen. Pořadí neboli číslo úkonu je generováno v závislosti na stromu operací z technologických postupů tím způsobem, že jsou nejdříve očíslovány dílenské úkony vzniklé ze zanořených operací polotovarů a nakonec z nejsvrchnějších operací výrobku. Naplánované množství na úkonu může být stejné nebo nižší než na dané operaci, pokud byla daná operace rozdělena na více úkonů. Atribut směna je hodnota z číselníku Druhy směn z modulu Administrace, atribut Datum pak určuje, od kdy je dílenský úkol zaplánovaný.[56]

Důležitou informací je stav úkonu, který může nabývat následujících hodnot:

- plánovaný,
- potvrzený - musí být vyplněna informace o datu směně,
- v procesu - začala-li na úkonu probíhat práce,
- odložený,
- dokončený,
- stornovaný.

3.4 Výroba - realizace

V rámci realizace samotné výroby vyráběné položky je potřeba evidovat práci odvedenou na této položce. Na výrobě se mohou podílet i externí dodavatelé v rámci kooperací.

3.4.1 Pracovní lístky

Pracovní lístek vzniká při evidenci práce na operaci, tzn. zahájením nebo ukončením práce na operaci technologického postupu dané vyráběné položky. Práci je možné evi-

dovat pouze na spuštěném výrobním příkazu. Pokud se jedná o operaci kusovou, práce se eviduje přímo pro konkrétní sériové číslo.[57]

Pracovní lístek se odkazuje na výrobní příkaz, operaci, na které je evidována práce, skladovou kartu položky vyráběné v rámci této operace, pracoviště, pracovníka, středisko, tarifní třídu a popřípadě i sériové číslo položky. Dále obsahuje datum a čas zahájení a ukončení dané operace, množství výrobků, pro které byla tato operace dokončena. Podle atributu Ukončení operace se pracovní lístky dělí na ukončené a neukončené.

Při ukládání pracovního lístku se automaticky provádí různé kontroly, například zda byl spuštěn a zároveň nebyl ukončen odpovídající výrobní příkaz nebo zda jsou již ukončeny všechny povinné předcházející operace.

V rámci evidence práce je možné zaevidovat i nápravu určité **neshody**. V tomto případě je vybrán typ neshody z číselníku Neshody, která byla v rámci dané operace napravována.

Skupinu jednoho až několika pracovních lístků zastřešuje **souhrnný pracovní lístek**. Tyto pracovní lístky jsou seskupovány na základě nastavení parametrů řad výrobních příkazů.

3.4.2 Kooperace

Kooperací se rozumí souhrn vybraných vyráběných položek a vybraných operací jejich technologického postupu, které se neprovádí ve vlastní výrobě. Kooperace nemůže existovat samostatně, váže se tedy na konkrétní výrobní příkaz. K jednomu výrobnímu příkazu pak může existovat více než jedna kooperace.

Z vyráběných položek daného výrobního příkazu se vyberou ty, jež budou odeslány do kooperace. Na těchto vyráběných položkách je potřeba dále vybrat operace, které se v rámci kooperace budou provádět. U dávkových operací je zadáno množství, které do kooperace vstupuje. U kusových operací se počet vyráběných položek vstupujících do kooperace odvíjí od počtu vybraných sériových čísel.

Kooperace obsahuje odkaz na firmu, která bude danou kooperaci provádět, a odkaz na středisko. Dále obsahuje sekce Odvezení a Přivezení, které obsahují informace o plánovaném a skutečném datu odvezení/přivezení a uživateli, který odvezl/přivezl položky kooperace. Na konkrétní vyráběné položce je možné zadat množství již vrácené z kooperace, u vyráběné položky se sériovými čísly se datum přivezení eviduje u každého sériového čísla zvlášť.[58]

3.4.3 Dokončené výrobky

Dokončení výrobky jsou ty, u kterých byla provedena ukončující operace technologického postupu daného výrobku. V této agendě lze záznamy o dokončených výrobcích

prohlížet, kontrolovat nebo vytvořit příjemku hotových výrobků. Tyto příjmy jsou pak ukládány do agendy Příjem hotových výrobků v modulu Skladové hospodářství.[59]

4 SROVNÁNÍ VÝROBNÍCH MODULŮ HELIOS INUVIO A ABRA GEN

Jelikož se jedná o moduly určené pro výrobu, zásadní rozdíly mezi funkcionalitami týkající se přímo výroby nenajdeme.

Co se týče číselníků, HELIOS iNuvio disponuje oproti ABRA Gen číselníkem nářadí a důležitým číselníkem změn. Číselník nářadí se uplatňuje ve vazbách nářadí a operací. Tyto vazby představují hodnotnou informaci o nářadí potřebném pro dokončení vyráběného díla. Číselník změn souvisí s nejvýraznější odlišností oblasti výroby obou systémů, a to podporou změnového řízení. HELIOS iNuvio nabízí tuto funkcionalitu, díky níž je možné uchovávat historii různých oprav a úprav v rámci TPV a nastavovat těmto změnám platnost. ABRA Gen obsahuje samostatný modul Sledování změn, kde lze v nastavení vybrat Business objekty, u kterých budou změny sledovány. U těchto objektů se pak ukládají změny položek, které jsou přímo ovlivňovány uživatelem.[60]

Kmenové respektive skladové karty lze v obou systémech pomocí příznaků nebo parametrů rozlišit na výrobky, polotovary nebo materiály. HELIOS iNuvio navíc umožňuje odlišit i vedlejší produkt, který ale v procesu výroby nehraje žádnou významnější roli.

V obou systémech existuje možnost uložit obrázky do číselníku a poté je přiřadit např. k vyráběnému dílci, ke kusovníkům nebo operacím technologického postupu. Co se týče ostatních dokumentů, v systému ABRA Gen existuje samostatný modul Dokumenty a přílohy [61], v HELIOS iNuvio potom samostatný číselník Dokumenty všeobecné [62], který není součástí modulu Výroba. K různým položkám lze pak přiřazovat různé množství těchto dokumentů. Systém HELIOS iNuvio navíc nabízí možnost evidovat zvláště výrobní dokumentaci, která podléhá změnovému řízení.

II. PRAKTICKÁ ČÁST

5 NÁVRH APLIKACE

Aplikace bude vyvíjena jako REST API. Aplikace bude obsahovat zjednodušenou funkcionalitu pro řízení výroby na základě teoretické části. Součástí aplikace bude možnost registrace a přihlášení uživatele. Jednou z hlavních funkcionalit bude získání fronty práce neboli seznamu výrobních příkazů pro konkrétní pracoviště, dále pak získání seznamu operací a seznamu materiálu pro konkrétní výrobní příkaz. Aplikace bude umožňovat uživateli evidovat práci na jednotlivých operacích výrobního příkazu. Automaticky se bude měnit stav operace a výrobního příkazu dle provedené akce a dle počtu odváděných kusů. Pro dotazování API bude zapotřebí autorizace pomocí API klíče uživatele.

5.1 Entity

V aplikaci budou existovat následující entity:

- uživatel (user)
- pracoviště (workplace),
- komodita (commodity),
- produkt (product),
- materiál (material),
- výrobní příkaz (production order),
- operace (operation),
- záznam práce na operaci (work report),
- vazba materiálu k produktu (material to product).

5.1.1 Doménový diagram

Obrázek 5.1 je jednoduchý doménový diagram, který obsahuje entity a znázorňuje, jakým způsobem spolu souvisí. U vazeb je pak uvedena multiplicita.

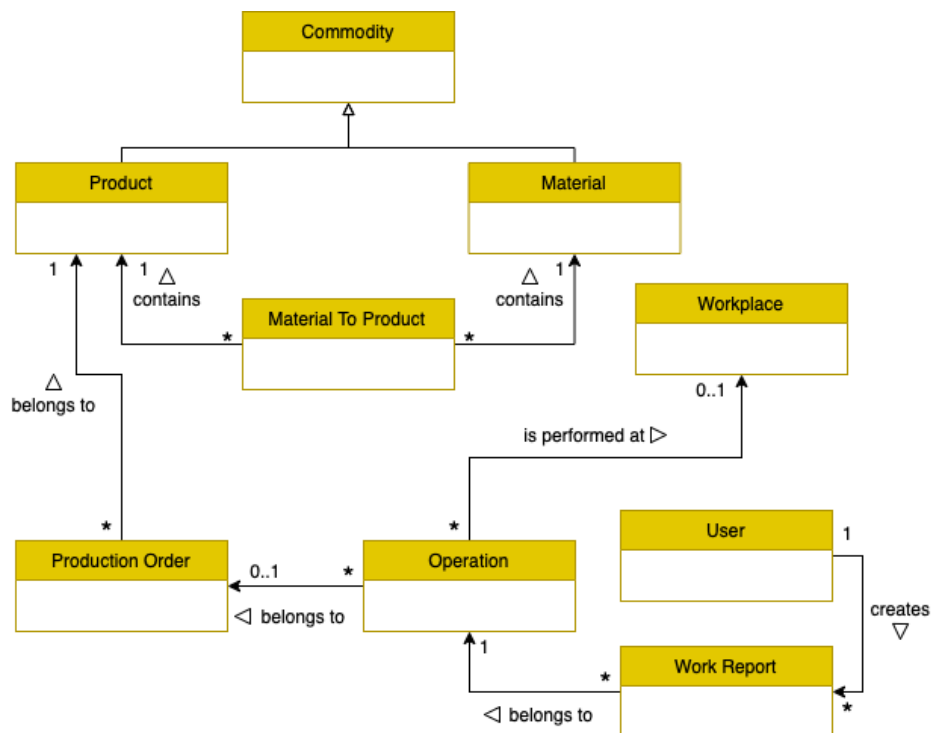
Produkt a materiál dědí z komodity, mezi materiálem a produktem pak existuje vazba M:N, kterou zastupuje entita vazba materiálu k produktu.

Výrobní příkaz patří k produktu, k produktu může patřit více výrobních příkazů, jedná se tedy o vazbu N:1 výrobního příkazu k produktu.

Operace může ale nemusí patřit k výrobnímu příkazu a stejně tak k pracovišti. Zde se jedná o vazbu 1:N výrobního příkazu k operaci a vazbu 1:N pracoviště k operaci.

K operaci může ale nemusí existovat záznam práce na operaci, což definuje vazba 1:N operace k záznamu práce na operaci.

Záznam práce na operaci se musí odkazovat na uživatele, tzn. vzniká vazba 1:N uživatele k záznamu práce na operaci.



Obrázek 5.1 Doménový diagram

5.1.2 ERD

Datový model na obrázku 5.2 obsahuje již atributy entit, jejich typy a omezení jako unikátnost, nebo zda může atribut nabývat nulové hodnoty. Dále jsou znázorněny vazby s multiplicitou pomocí primárních a cizích klíčů.

V tabulce Komodita (commodity) budou uloženy komodity, produkty a materiály. Tyto entity budou rozlišeny pomocí hodnoty ve sloupci typ (type), který může nabývat hodnot *komodita* (commodity), *produkt* (product) a *materiál* (material). Sloupec archivováno (archived) indikuje, zda je daný záznam archivován.

Tabulka pro uložení vazby materiálu k produktu (material_to_product) obsahuje odkaz na produkt a odkaz na materiál, což jsou cizí klíče do tabulky Komodita. Sloupec požadované množství (amount_required) definuje množství materiálu přiřazeného k produktu.

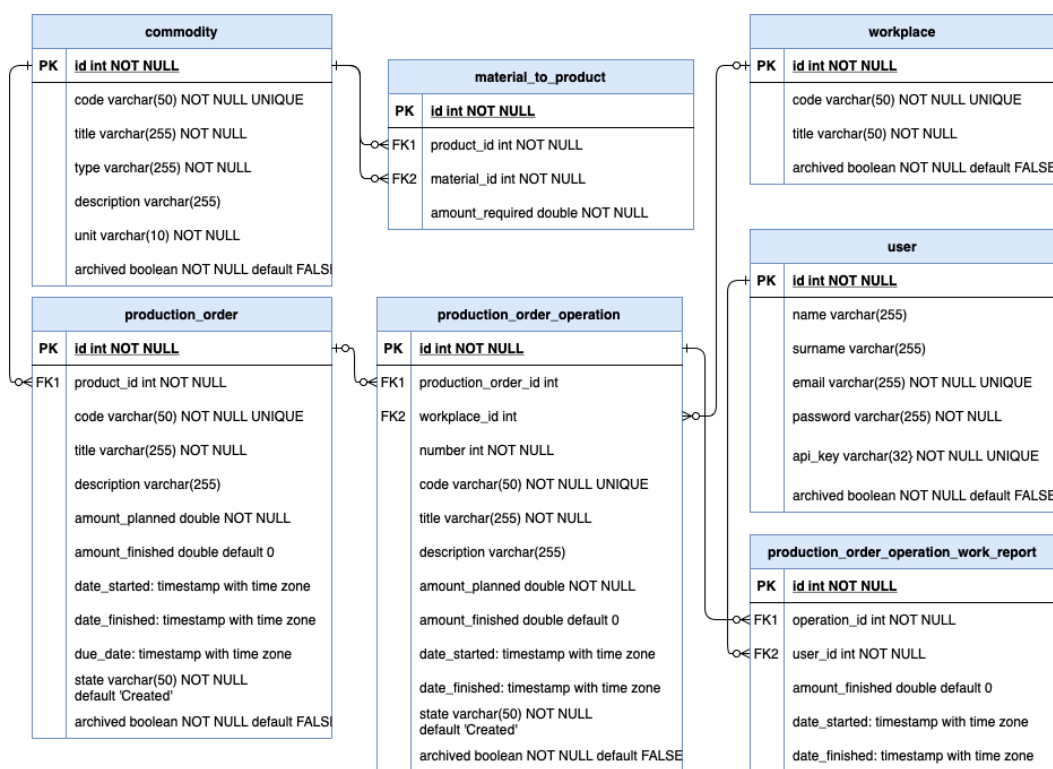
Výrobní příkaz (production_order) obsahuje odkaz na produkt, což je cizí klíč do tabulky Komodita. Stav výrobního příkazu indikuje hodnota ve sloupci stav (state),

který může nabývat hodnot *vytvoreno* (created), *probíhá* (in progress) a *ukončeno* (finished).

Operace výrobního příkazu (`production_order_operation`) se odkazuje na výrobní příkaz, což je cizí klíč do tabulky Výrobní příkaz a na pracoviště, což je cizí klíč do tabulky Pracoviště. Stav operace je indikován opět hodnotou ve sloupci stav (state).

Pro evidenci práce na operaci slouží tabulka Záznam práce na operaci (`production_order_operation_work_report`), která obsahuje odkaz na operaci a uživatele.

Tabulka Uživatel (user) eviduje uživatele. Hash hesla bude uložen ve sloupci heslo (password), vygenerovaný API klíč pak ve sloupci api klíč (api_key).



Obrázek 5.2 ERD

5.2 Akce

V následující tabulce jsou obsaženy entity a základní akce (CRUD). U každé entity je pak zaznačeno, které z těchto akcí jsou na ní dostupné a lze je volat pomocí HTTP dotazů. CRUD je zkratka složená z názvu akce vytvoření (create), čtení (read), úprava (update) a smazání (delete).

Tabulka 5.1 Entity a akce

Entita \ Akce	Create	Read	Update	Delete
User	X	X	X	X
Workplace	X	X	X	X
Commodity	X	X	X	X
Product	X	X	X	X
Material	X	X	X	X
Production Order	X	X	X	X
Operation	X	X	X	X
Work Report				
Material to Product				X

5.2.1 Vytvoření

Aplikace neumožní vytvořit pracoviště, komoditu, výrobní příkaz a operaci s již existujícím kódem, dále pak neumožní vytvořit neboli zaregistrovat uživatele s již existujícím emailem.

5.2.2 Smazání

Při provedení akce smazání uživatele, výrobního příkazu, operace, komodity nebo pracoviště nebude tento záznam odstraněn, pouze se změní hodnota atributu archivováno. Provádění některých akcí na archivovaných entitách bude omezeno (viz dále).

5.2.3 Úprava

Záznamy entit, které jsou archivované, nebude možné upravovat.

5.2.4 Seznam

Při získání seznamu entity uživatele, výrobního příkazu, operace, komodity nebo pracoviště nebudou v základu *archivované záznamy* obsaženy. Toto bude možné změnit nastavením parametru dotazu *archived*, poté se v seznamu budou zobrazovat jak archivované tak nearchivované záznamy.

5.2.5 Registrace uživatele

Při vytvoření neboli registraci uživatele bude vytvořen hash hesla a tento hash pak bude uložen v databázi. Zároveň bude při registraci vygenerován API klíč.

5.2.6 Přihlášení uživatele

Přihlášení uživatele bude možné pomocí emailu a hesla. Pokud je daný email již zaregistrován a daný uživatel není archivován, proběhne kontrola správnosti hesla. Po úspěšném přihlášení je vrácen objekt uživatele s API klíčem. Tento API klíč je pak použit pro dotazování API a uživatel, k němuž patří tento API klíč, je považován za přihlášeného uživatele.

5.2.7 Fronta práce

Fronta práce představuje seznam výrobních příkazů, které nejsou *archivované* a jejichž operace se budou provádět na daném pracovišti. Pokud je pracoviště archivováno, nelze pro toto pracoviště vrátit frontu práce.

5.2.8 Seznam operací výrobního příkazu

Operace výrobního příkazu jsou ty operace, které se odkazují na daný výrobní příkaz.

5.2.9 Přiřazení a odebrání operace výrobního příkazu

Operaci lze přiřadit k výrobnímu příkazu nebo odebrat, pokud nejsou položky na obou stranách vazby archivované. Při přiřazení se kontroluje, zda již daná operace není přiřazena k jinému výrobnímu příkazu.

5.2.10 Přiřazení materiálu a odebrání materiálu přiřazeného k produktu

K produktu lze více materiálů a to i pokud se jedná o stejný materiál. Vždy vznikne nová vazba materiálu k produktu s požadovaným množstvím materiálu. Při odebrání materiálu jsou odstraněny všechny vazby materiálu k produktu, které obsahují daný materiál. Odstranění konkrétní vazby je zaznačeno v tabulce s CRUD akcemi.

5.2.11 Materiál potřebný pro výrobní příkaz

Z existujících vazeb materiálu k produktu bude získán materiál, který je navázaný na produkt, na něhož se odkazuje daný výrobní příkaz. Množství materiálu potřebné na výrobní příkaz bude vypočítáno z množství požadovaného materiálu na daný produkt a plánované množství, které má být vyrobeno daným příkazem. Výsledkem bude seznam, který bude obsahovat toto vypočítané množství a daný materiál.

5.2.12 Evidence práce na operaci

Aplikace bude umožňovat evidovat práci na jednotlivých operacích výrobního příkazu, a to pomocí akce *zahájení práce* a *ukončení práce*.

Zahájení práce bude provedeno vytvořením záznamu práce na operaci s vazbou na danou operaci, počátečním datem a časem a vazbou na přihlášeného uživatele. Pokud se jedná o první zahájení práce na operaci, stav operace bude změněn na *probíhá*. Pokud se zároveň jedná o první operaci výrobního příkazu, bude i jeho stav změněn na *probíhá*.

Práci nebude možné započít, pokud:

- existuje jiná započatá práce na dané operaci,
- je výrobní příkaz dané operace již ukončený,
- existuje předchozí operace a není dokončená,
- je daná operace již ukončená.

Ukončení práce bude provedeno zápisem konečného data a času s počtem odváděných kusů na nalezeném záznamu práce na dané operaci, který není ukončen, tzn. položka konečného data na záznamu práce na operaci není vyplněna.

Práci nebude možné ukončit, pokud:

- neexistuje započatá práce na dané operaci,
- práce nebyla započata právě přihlášeným uživatelem,
- součet odváděných kusů a již dokončených kusů přesáhne počet plánovaných kusů.

Počet odváděných kusů bude přičten k dokončenému množství na dané operaci. Pokud se jedná o poslední operaci výrobního příkazu, tento počet bude přičten k dokončenému množství i na výrobním příkazu. Po odvedení posledního kusu na operaci bude stav operace změněn na *dokončeno*, pokud se jedná o poslední operaci výrobního příkazu, bude i jeho stav změněn na *dokončeno*.

6 IMPLEMENTACE

Aplikace je implementována v jazyce PHP ve frameworku Phalcon. Pro zajištění abstrakce databázové vrstvy a možnosti pracovat s daty jako s objekty je využit ORM framework Doctrine. Pro práci s objekty je použito rozhraní Doctrine - EntityManager, pomocí něhož lze např. získat entitu z databáze nebo naopak entitu uložit.[64] Byl použit přístup *Code first*, kdy jsou nejdříve vytvořeny třídy modelů reprezentující databázové entity, ze kterých se pak pomocí balíčku *Doctrine Migrations* vygenerují soubory s SQL příkazy pro vytvoření tabulek.[65]

6.1 Model

Třídy modelů představují databázové entity. K jejich definici se používají komentářové anotace, jejichž podpora je zaručena použitím balíčku *Doctrine Annotations*. [63]

6.1.1 Anotace třídy

Komentářovou anotací u samotné třídy lze např. specifikovat, že se jedná o entitu nebo název tabulky, jejíž řádek daná entita představuje. Jelikož entita Produkt a Materiál dědí z entity Komodita, je potřeba zde uvést způsob dědičnosti (*InheritanceType*). V případě, že pro tyto entity existuje jedna společná tabulka, je potřeba uvést v anotaci tzv. *DiscriminatorColumn*, což je sloupec, jehož hodnota určuje o jakou entitu se jedná. Pomocí *DiscriminatorMap* se definují hodnoty, kterých tento sloupec nabývá a jaké entity se k daným hodnotám vážou. Pro účely ukázky není uveden celý namespace entit.

```
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity
 * @ORM\Table(name="commodity")
 * @ORM\InheritanceType("SINGLE_TABLE")
 * @ORM\DiscriminatorColumn(name="type", type="string")
 * @ORM\DiscriminatorMap({
 *     "Commodity" = "Commodity",
 *     "Product" = "Commodity\Product",
 *     "Material" = "Commodity\Material"
 * })
 */
class Commodity
```

6.1.2 Anotace atributů

U samotných atributů třídy modelu se definuje sloupec databázové tabulky, který tento atribut představuje. Pomocí *Column* lze uvést datový typ, název, zda má být hodnota unikátní, zda může nabývat nulové hodnoty nebo zda má obsahovat nějakou defaultní

hodnotu. Atribut id je potřeba označit anotací `Id`, aby bylo jasné, že se jedná o primární klíč a uvést strategii pomocí `GeneratedValue`, jakou bude generován.

```
/**
 * @ORM\GeneratedValue(strategy="IDENTITY")
 * @ORM\Id
 * @ORM\Column(type="integer", name="id")
 * @var int|null
 */
protected ?int $id = null;

/**
 * @ORM\Column(type="string", name="code", length=50, unique=true)
 * @var string
 */
protected string $code;

/**
 * @ORM\Column(type="string", name="description", nullable=true)
 * @var string|null
 */
protected ?string $description = null;

/**
 * @ORM\Column(type="boolean", name="archived", options={"default":0})
 * @var bool
 */
protected bool $isArchived = false;
```

Pokud se jedná o atribut typu jiné entity tzn. v tabulce je to sloupec odkazující se do jiné tabulky, definuje se vazba pomocí anotace multiplicity, např. `ManyToOne`, kde uvádíme cílovou entitu (název entity je uveden bez namespace pro účely ukázky). `JoinColumn` slouží k uvedení názvu sloupce cizího klíče a názvu sloupce primárního klíče cílové entity. Na modelu lze vytvořit i inverzní vazbu, tzn. atribut typu entity, která se odkazuje na aktuální entitu (v případě entity výrobního příkazu se jedná o kolekci operací jelikož se operace odkazuje na výrobní příkaz).

```
/**
 * @ORM\ManyToOne(
 *     targetEntity="Product"
 * )
 * @ORM\JoinColumn(
 *     name="product_id",referencedColumnName="id", nullable=false
 * )
 * @var Product
 */
protected Product $product;

/**
 * @ORM\OneToMany(targetEntity="Operation", mappedBy="productionOrder")
 * @ORM\OrderBy({"number"="ASC"})
 * @var Collection|null
 */
protected ?Collection $operations = null;
```

6.1.3 Anotace metod

Komentářovou anotaci lze použít i u metod na modelu, tak je tomu u metody pro vygenerování API klíče a hashování hesla. Anotace `PrePersist` zajistí, že se daná metoda na modelu zavolá před prvním "zavedením" objektu (metoda `persist()` třídy `EntityManager`) viz kapitola CRUD. Pro toto použití je nutné dodat do anotací třídy `HasLifecycleCallbacks`.

```
/**
 * @ORM\PrePersist
 */
public function generateApiKey()
{
    $this->apiKey = (\Nette\Utils\Random::generate(32, '0-9a-zA-Z'));
}

/**
 * @ORM\PrePersist
 */
public function hashPassword()
{
    $this->setPassword(
        password_hash(
            $this->getPassword(), PASSWORD_BCRYPT, ['cost' => 12]
        ));
}
```

Všechny modely pak obsahují metody pro získání a nastavení daných atributů tzv. *getter* a *setter*.

6.1.4 Generování migrací

Jakmile jsou třídy modelů vytvořeny či posléze upraveny, příkazem `migrations:diff` dojde k porovnání aktuálního stavu databáze s aktuálními definicemi vytvoření pomocí Doctrine ORM. Na základě rozdílů se vygeneruje soubor migrace s SQL příkazy, který se aplikuje pomocí příkazu `migrations:execute --up <path to migration file>`.^[65]

6.2 DTO

Pro komunikaci mezi aplikací a rozhraním uživatele jsou využity objekty typu *DTO* alias Data Transfer Object. Pokud představují stejné entity jako modely, obsahují stejné atributy, *getter* a *setter* jako jejich "modelové protějšky". Mohou ale existovat i samostatně, například jako pomocné objekty pro získání nějakého výstupu, který není založen pouze na jedné entitě.

6.3 Mapper

Pro mapování objektů jsou použity třídy, jež obsahují metody pro namapování modelu na DTO a naopak. Gettery a settery jsou použity podle toho, z jakého typu objektu na jaký typ objektu se mapuje. Například při mapování Uživatele z modelu na DTO není namapován atribut obsahující hodnotu hesla.

6.4 CRUD

Na následujících ukázkách kódu bude popsána implementace CRUD akcí na entitě pracoviště.

6.4.1 Vytvoření

```
/**
 * @return WorkplaceDTO
 */
public function createAction(): WorkplaceDTO
{
    $body = $this->request->getJsonRawBody();
    $dto = (new WorkplaceDTO())
        ->setCode($body->code)
        ->setTitle($body->title);
    $this->mapper->mapModel($model = new Workplace(), $dto);
    $this->entityManager->persist($model);
    $this->entityManager->flush();
    $this->mapper->mapDto($dto = new WorkplaceDTO(), $model);
    return $dto;
}
```

Vytvoří se nová instance DTO pro Pracoviště, na které se pomocí setterů nastaví příslušné atributy ze získaného requestu. Poté se pomocí mapperu namapuje DTO na model. Při vytvoření nové instance modelu je tento model pro třídu EntityManager neznámý, proto je potřeba zavolat metodu `$this->entityManager->persist($model)`. Poté je entita uložena do databáze pomocí `$this->entityManager->flush()`. Pro výstup je model namapován na DTO.

6.4.2 Úprava

```
/**
 * @param int $id
 * @return WorkplaceDTO
 * @throws \Exception
 */
public function updateAction(int $id): WorkplaceDTO
{
    $model = $this->repository->find($id);
    if (!$model instanceof Workplace) {
        throw new \Exception(
            'Workplace with id ' . $id . ' not found.',
        );
    }
}
```

```
        );
    }
    if ($model->isArchived()) {
        throw new \Exception(
            'You can not work with an object that is archived.',
            400
        );
    }
    $body = $this->request->getJsonRawBody();
    $dto = (new WorkplaceDTO())
        ->setCode($body->code)
        ->setTitle($body->title);
    $this->mapper->mapModel($model, $dto);
    $this->entityManager->flush();
    $this->mapper->mapDto($dto = new WorkplaceDTO(), $model);
    return $dto;
}
```

Na základě id je nejprve nalezen záznam entity s daným id pomocí `$this->repository->find($id)`, proměnná `repository` představuje repository dané entity získané přes metodu `getRepository(<class name>)` na `EntityManageru`. Pokud záznam s tímto id neexistuje nebo je archivovaný, je vyhozena výjimka s odpovídající zprávou. Vytvoří se nová instance DTO pro Pracoviště, na které se pomocí setterů nastaví příslušné atributy ze získaného requestu. Poté se pomocí mapperu namapuje DTO na již získaný model. Změny jsou opět uloženy voláním `flush()`.

6.4.3 Čtení

Získání konkrétního záznamu a kontrola na existenci probíhá stejně jako v rámci akce upravení, poté je hned model namapován na DTO a poslán na výstup.

Získání seznamu:

```
/**
 * @return WorkplaceDTO []
 */
public function listAction(): array
{
    $archived = $this->request->get('archived') == 1;
    if ($archived) {
        $items = $this->repository->findAll();
    } else {
        $items = $this->repository->findBy(['isArchived' => false]);
    }
    $workplaces = [];
    /**
     * @var Workplace $model
     */
    foreach ($items as $model) {
        $this->mapper->mapDto($dto = new WorkplaceDTO(), $model);
        $workplaces[] = $dto;
    }
    return $workplaces;
}
```

Jak již bylo zmíněno, u seznamu entit, které obsahují atribut `archivováno`, lze pomocí parametru dotazu `archived` ovlivnit, zda budou archivované záznamy do seznamu zahrnuty. Nejdřív tedy proběhne vyhodnocení tohoto parametru a poté se dle jeho hodnoty provede získání všech záznamů voláním metody `findAll()` na repository nebo získání jen těch záznamů, které nejsou archivované, a to voláním `findBy(['isArchived' => false])` na repository. Poté se pole těchto záznamů postupně namapuje na DTO a pole těchto objektů jde na výstup.

6.4.4 Smazání

U akce smazání pracoviště se prvně provede získání záznamu. V případě že existuje a není archivovaný, je zavolán příslušný setter pro nastavení tohoto atributu `$model->setIsArchived(true)`.

6.5 Tvorba dotazů pomocí Doctrine QueryBuilder

Při potřebě složitějšího dotazu do databáze lze využít třídu `QueryBuilder`. Tato třída nabízí metody, díky nimž lze dotaz do databáze postupně poskládat.

Instanci třídy `QueryBuilder` se získá pomocí metody `createQueryBuilder()` na `EntityManager`. Pak už se na této instanci pomocí volání metod definuje například:

- co bude získáno z databáze - `select()`,
- z jaké tabulky - `from()` (zde se uvádí třída databázového modelu a alias),
- mezi jakými tabulkami bude vytvořena relace - `join()` (zde se uvádí opět třída databázového modelu, alias, typ podmínky a podmínka vytvoření relace),
- jaké jsou podmínky pro vyhledání - `where()` (přepíše všechny dosavadní podmínky) nebo `andWhere()` (přidá podmínku k již existujícím podmínkám),
- hodnoty použitých parametrů - `setParameter()`
- podle čeho se má seskupovat - `groupBy()`,
- podle čeho se má řadit - `orderBy()` (není použito v ukázce kódu).

Zkonstruování dotazu a tím vytvoření instance třídy `Query` zajistí metoda `getQuery()`, na které lze pro získání konečného výsledku dotazu zavolat metodu `getResult()`. Na následující ukázce je implementace dotazu pro získání materiálu potřebného pro výrobní příkaz.

```
$productsQuery = $this->entityManager->createQueryBuilder()
    ->select('product')
    ->from(Product::class, 'product')
    ->join(
        ProductionOrder::class,
        'productionOrder',
        \Doctrine\ORM\Query\Expr\Join::WITH,
        'productionOrder.product = product'
    )
    ->where('productionOrder.id = :productionOrderId')
    ->setParameter('productionOrderId', $productionOrderId)
    ->getQuery();

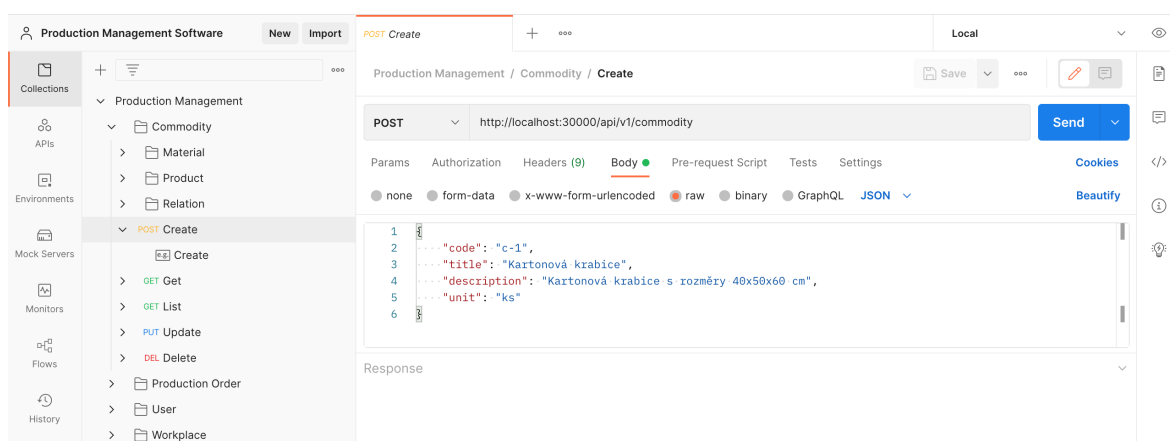
$materialToProductQuery = $this->entityManager->createQueryBuilder()
    ->select('SUM(materialToProduct.amountRequired), material')
    ->from(Material::class, 'material')
    ->join(
        MaterialToProduct::class,
        'materialToProduct',
        \Doctrine\ORM\Query\Expr\Join::WITH,
        'materialToProduct.material = material'
    )
    ->where('materialToProduct.product = :product')
    ->setParameter('product', $productsQuery->getResult())
    ->groupBy('material');
    if (!$archived) {
        $materialToProductQuery->andWhere('material.isArchived = false');
    }

return $materialToProductQuery
    ->getQuery()
    ->getResult();
```

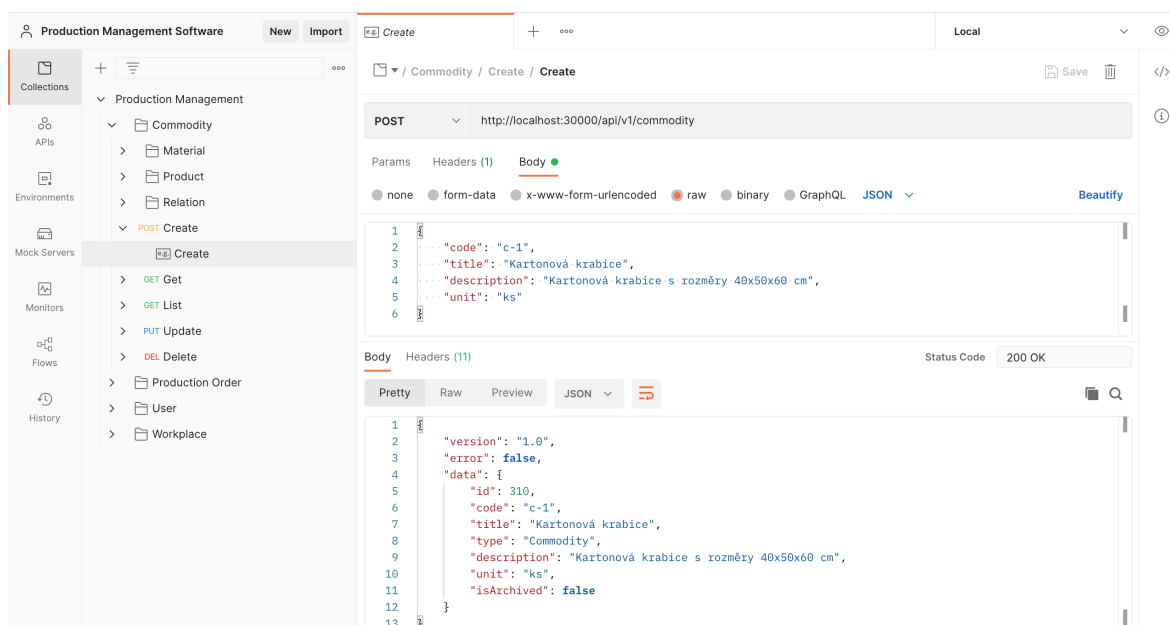
Nejprve je vytvořen dotaz pro získání všech produktů dle id výrobního příkazu, které je uložené v proměnné `$productionOrderId`. Je potřeba získat všechny produkty, vytvořit relaci s výrobními příkazy dle atributu `produkt` na výrobním příkazu a omezit hledání pouze na produkty daného výrobního příkazu. Následuje dotaz pro získání sumy potřebného množství z vazby materiálu k produktu a daný materiál. Vytvoří se relace materiálu a vazby materiálu k produktu, je přidána podmínka, že produkt na vazbě materiálu k produktu je produkt získaný předchozím dotazem, jehož výsledek je předám pomocí parametru. Výsledek bude seskupen dle materiálu, tak aby se ve výsledních stejný materiál objevil pouze jednou se sumou potřebného množství ze všech záznamů.

7 DOKUMENTACE API

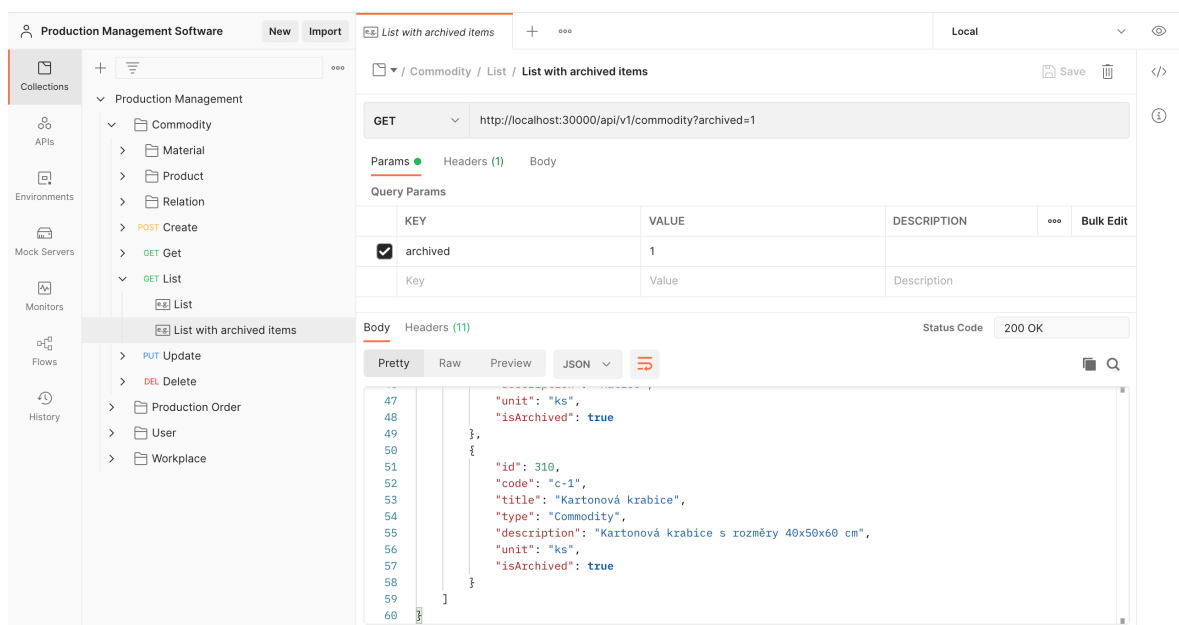
Pro posílání HTTP dotazů a jejich zdokumentování byl využit nástroj Postman. V tomto nástroji lze dotazy strukturovat do kolekcí a složek. U HTTP dotazu se byla využita definice jeho názvu, metody (GET, PUT, POST, DELETE atd.), url, parametru (například *archived*), hlavičky (autorizace pomocí API klíče) a těla dotazu ve formátu JSON. Response jednotlivých HTTP dotazů lze uložit jako *example* k danému dotazu nebo do souboru. Celá kolekce lze vyexportovat do JSON souboru.



Obrázek 7.1 HTTP dotaz pro vytvoření záznamu komodity



Obrázek 7.2 Response HTTP dotazu pro vytvoření záznamu komodity



The screenshot displays the Postman interface for a REST client. The left sidebar shows a collection named "Production Management Software" with a sub-collection "Production Management". Under "Production Management", there is a "GET List" endpoint selected, which is named "List with archived items". The main panel shows the request details for this endpoint:

- Method: GET
- URL: `http://localhost:30000/api/v1/commodity?archived=1`
- Query Params:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> archived	1	
Key	Value	Description
- Status Code: 200 OK

The response body is shown in JSON format (Pretty view):

```
47 |   "unit": "ks",
48 |   "isArchived": true
49 | },
50 | {
51 |   "id": 310,
52 |   "code": "c-1",
53 |   "title": "Kartonová krabice",
54 |   "type": "Commodity",
55 |   "description": "Kartonová krabice s rozměry 40x50x60 cm",
56 |   "unit": "ks",
57 |   "isArchived": true
58 | }
59 | ]
60 |
```

Obrázek 7.3 Response HTTP dotazu pro získání seznamu komodit včetně archivovaných položek

8 API TESTY

Pro API testování byl použit framework *Codeception* a jeho moduly určené pro testování REST API a záznamů v databázi. Před vytvořením testu je potřeba nakonfigurovat url, na kterou se v rámci testů budou posílat HTTP dotazy a přístupové údaje do databáze, aby mohlo být vytvořeno připojení. [66][67]

8.1 WorkReportCest

Pro otestování akcí *zahájení práce* a *dokončení práce* na operaci výrobního příkazu byla vytvořena třída *WorkReportCest*.

8.1.1 Struktura testu

Nepovinnou součástí testu je metoda `_before()`, která se zvolá před každým dílčím testem. Pro účely testování API pomocí HTTP requestů, pro které je potřeba autorizace pomocí API klíče, se nabízí, aby byl každým testem vložen do databáze záznam s uživatelem a definovaným API klíčem a tento API klíč pak nastaven v autorizační hlavičce. API klíč je uložen v proměnné `$apiKey`.

```
public function _before(ApiTester $I)
{
    $this->userId = $I->haveInDatabase('user', [
        'name' => 'Jan',
        'surname' => 'Novak',
        'email' => 'jan.novak@gmail.com',
        'password' => '1234honza',
        'api_key' => $this->apiKey
    ]);
    $I->haveHTTPHeader('Authorization', 'Basic ' . $this->apiKey);
}
```

Všechny testovací metody se volají na instanci `ApiTester`, zde metoda `haveInDatabase()` vytvoří v uvedené tabulce `user` záznam s uvedenými hodnotami a vrátí id tohoto nově vytvořeného záznamu. Každý záznam vytvořený pomocí této metody je po provedení dílčího testu smazán. Id uživatele je uloženo do proměnné pro další účely testování. Pomocí `haveHTTPHeader()` lze nastavit HTTP hlavičku.

Pokud existuje více testů, ve kterých se vkládají záznamy do databáze do stejných tabulek, je vhodné použít tzv. `data provider`. Pro definování sady dat slouží metoda `_dataProvider()`. Na následující ukázce kódu je část definovaných dat použitých pro testování.

```
public function _dataProvider()
{
    return [
        [
```

```

        'product' => [
            'code' => 'p-test',
            'title' => 'TestProduct',
            'type' => 'Product',
            'unit' => 'ks'
        ],
        'productionOrder' => [
            'product_id' => '#pId',
            'code' => 'po-test',
            'title' => 'TestProductionOrder',
            'amount_planned' => 5
        ],
        'operation' => [
            'production_order_id' => '#poId',
            'workplace_id' => '#wId',
            'number' => 1,
            'code' => 'op-test',
            'title' => 'TestOperation',
            'amount_planned' => 5,
        ]
    ]
};
}

```

Pro použití dat je potřeba uvést v komentářové anotaci dílčího testu `dataProvider` s názvem metody, která požadová data vrací. Jako argument dílčího testu se pak uvádí proměnná typu `Example`, do které se data přetransformují a pomocí které pak lze ke konkrétním datům přistupovat.

```

/**
 * @param ApiTester $I
 * @param Example $example
 * @dataProvider _dataProvider
 */
public function workStartActionIsSuccessful(
    ApiTester $I,
    Example $example
) {
    $pId = $I->haveInDatabase('commodity', $example['product']);
    $poId = $I->haveInDatabase(
        'production_order',
        str_replace(['#pId'], [$pId], $example['productionOrder'])
    );
    $wId = $I->haveInDatabase('workplace', $example['workplace']);
    $opId = $I->haveInDatabase(
        'production_order_operation',
        str_replace(
            ['#poId', '#wId'],
            [$poId, $wId],
            $example['operation']
        )
    );
    $I->sendPost(
        '/production-order/operation/' . $opId . '/work-start'
    );
    $I->seeInDatabase(
        'production_order_operation_work_report',
        ['operation_id' => $opId]
    );
}

```

```
    );  
    $I->seeResponseCodeIs(StatusCode::OK);  
}
```

V dílčím testu `workStartActionIsSuccessful` je otestováno úspěšné zahájení práce na operaci. Postupně je do databáze vložen záznam produktu, výrobního příkazu, pracoviště a operace. Získanými id jednotlivých záznamů jsou nahrazeny určité textové řetězce v data setu. Pomocí `sendPost()` je poslán dotaz na danou url. Nakonec proběhne kontrola, zda se vytvořil záznam o provedení práce (`seeInDatabase()`) a zda je správný HTTP kód vráceného response (`seeResponseCodeIs()`).

Pro otestování neúspěšného zahájení práce na operaci, např. pokud daná operace neexistuje, je pozměněn dílčí test tak, že je vynecháno vložení záznamu operace do databáze. Získá se poslední id vložené operace a zvýší se o jedna, s tímto id je pak poslán dotaz na danou url. Kontrola textu vyhozené výjimky a HTTP kódu je provedena následující částí kódu.

```
$I->seeResponseCodeIs(StatusCode::NOT_FOUND);  
$I->seeResponseContains('Operation with id ' . $opId . ' not found');
```

8.1.2 Výsledky testů

Pokud dílčí test projde, je označen jako úspěšný (*PASSED*), pokud neprojde, je označen jako neúspěšný (*FAILED*). Všechny dílčí testy obsažené v testu `WorkReportCest` proběhly na výsledné implementaci úspěšně. Výsledky testů lze uložit do JSONu nebo HTML souboru, kde jsou kroky dílčích testů transformovány do vět.

Codeception Results OK (18.7s)

○

✓ 16

✗ 0

S 0

I 0

Api Tests

- WorkReportCest » Work start action is successful | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 2.79s
 - I have in database "user", {"name": "Jan", "surname": "Novak", "email": "jan.novak@gmail.com", "password": "1234honza", "api_key": "FZ5bR4yLSJdxRXPExOMLNvMKpf7dUY9"}
 - I have http header "Authorization", "Basic FZ5bR4yLSJdxRXPExOMLNvMKpf7dUY9"
 - I have in database "commodity", {"code": "p-test", "title": "TestProduct", "type": "Product", "unit": "ks"}
 - I have in database "production_order", {"product_id": "294", "code": "po-test", "title": "TestProductionOrder", "amount_planned": "5"}
 - I have in database "workplace", {"code": "w-test", "title": "TestWorkplace"}
 - I have in database "production_order_operation", {"production_order_id": "328", "workplace_id": "293", "number": "1", "code": "op-test", "title": "TestOperation", "amount_planned": "5"}
 - I send post "/production-order/operation/340/work-start"
 - I see in database "production_order_operation_work_report", {"operation_id": "340"}
 - I see response code is 200
- + WorkReportCest » Work start action throws exception when operation not found | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 0.85s
- + WorkReportCest » Work start action throws exception when operation does not belong to any production order | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 0.81s
- + WorkReportCest » Work start action changes status and sets date started on production order and operation | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.05s
- + WorkReportCest » Work start action throws exception because of archived operation | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 0.99s
- + WorkReportCest » Work start action throws exception because of unfinished work on operation | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.29s
- + WorkReportCest » Work start action throws exception because of finished production order | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.02s
- + WorkReportCest » Work start action throws exception because of unfinished previous operation | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 0.91s
- + WorkReportCest » Work start action throws exception because of finished operation | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 0.95s
- + WorkReportCest » Work done action is successful | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.16s
- + WorkReportCest » Work done action throws exception because of missing amount finished in request body | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.31s
- + WorkReportCest » Work done action throws exception when there is no work started | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.35s
- + WorkReportCest » Work done action throws exception when amount finished exceeded amount planned | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.03s
- + WorkReportCest » Work done action changes status and sets date finished on operation | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.13s
- + WorkReportCest » Work done action sets amount finished on production order | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1.01s
- + WorkReportCest » Work done action changes status and sets date finished on production order | {"pr..."}, {"pr..."}, {"pr..."}, {"pr..."}, {"nu..."}, {"co..."}, {"pr..."}, {"pr..."}, {"am..."}, {"op..."}, {"co..."} 1s

Summary

Successful scenarios:	16
Failed scenarios:	0
Skipped scenarios:	0
Incomplete scenarios:	0

Obrázek 8.1 Výsledky testů

ZÁVĚR

Jedním z výsledků této bakalářské práce je zpracování nastudovaných poznatků získaných při studiu modulů pro řízení výroby ERP systémů HELIOS a ABRA. Moduly jsou velmi obsáhlé a nabízí širokou škálu možností práce s procesy v rámci řízení výroby. Výsledné zpracování teoretické práce se tedy zaměřuje na nejdůležitější součásti a funkce těchto modulů. Studium vedlo k rozšíření znalostí v oblasti řízení výroby a samotných ERP systémů.

Vědomosti nabyté studiem výrobních modulů těchto systémů pak byly využity v návrhu zjednodušené aplikace pro řízení výroby.

Na základě návrhu byla implementována backendová část software pro řízení výroby. Tento software představuje zjednodušený modul pro řízení výroby a základ pro další vývoj produktu, kdy může být dále rozšiřován tak, aby splňoval další funkce, které nabízí komplexní ERP systémy. Jeho významnou stránkou je jednoduchost použití a možnost se dále rozšiřovat a přizpůsobovat tak požadavkům uživatelů systému.

Součástí praktické části bylo také zdokumentování výsledného API a vhodné otestování pomocí API testů, jejichž výsledky byly prezentovány.

SEZNAM POUŽITÉ LITERATURY

- [1] Automatické řízení firem ERP systémy je v Česku stále populárnější. BusinessINFO [online]. 17.02.2021 [cit. 2022-05-20]. Dostupné z: <https://www.businessinfo.cz/clanky/automaticke-rizeni-firem-erp-systemy-je-v-cesku-stale-popularnejsi/>
- [2] Reference. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: <https://www.helios.eu/reference>
- [3] Profil společnosti. ABRA [online]. [cit. 2022-05-18]. Dostupné z: <https://www.abra.eu/o-firme/>
- [4] Best ERP Software 2022: Top Rated ERP Systems Comparison. Software Testing Help [online]. 3. dubna 2022 [cit. 2022-04-10]. Dostupné z: <https://www.softwaretestinghelp.com/best-erp-software-systems/>
- [5] ABRA Flexi. ABRA [online]. [cit. 2022-02-09]. Dostupné z: <https://www.abra.eu/flexi/>
- [6] ABRA Gen. ABRA [online]. [cit. 2022-02-09]. Dostupné z: <https://www.abra.eu/erp-system-abra-gen/>
- [7] HELIOS Easy. HELIOS [online]. [cit. 2022-02-11]. Dostupné z: <https://www.helios.eu/easy-pro-malou-firmu>
- [8] HELIOS iNuvio. HELIOS [online]. [cit. 2022-02-11]. Dostupné z: <https://www.helios.eu/inuvio-pro-stredni-firmu>
- [9] Na trhu je nová generace oblíbeného systému HELIOS. HELIOS [online]. 20. září 2021 [cit. 2022-02-11]. Dostupné z: <https://www.helios.eu/novinky/na-trhu-je-nova-generace-oblibeneho-systemu-helios>
- [10] HELIOS Nephrite v otázkách a odpovědích. HELIOS [online]. 12. října 2020 [cit. 2022-02-11]. Dostupné z: <https://www.helios.eu/novinky/helios-nephrite-v-otazkach-a-odpovedich>
- [11] Zelený drahokam z rodiny HELIOS. HELIOS [online]. 9. září 2020 [cit. 2022-02-14]. Dostupné z: <https://www.helios.eu/novinky/zeleny-drahokam-z-rodiny-helios>
- [12] HELIOS Red. HELIOS [online]. [cit. 2022-02-14]. Dostupné z: <https://www.helios.eu/red-pro-zivnostniky>

- [13] SAP S/4HANA Cloud: Stručný přehled. SAP [online]. [cit. 2022-02-15]. Dostupné z: <https://www.sap.com/cz/products/s4hana-erp.html>
- [14] SAP S/4HANA Cloud: Capabilities. SAP [online]. [cit. 2022-02-15]. Dostupné z: <https://www.sap.com/cz/products/s4hana-erp/features.html>
- [15] What is SAP?. SAP [online]. [cit. 2022-02-14]. Dostupné z: <https://www.sap.com/cz/about/company/what-is-sap.html>
- [16] SAP Business One: Přehled. SAP [online]. [cit. 2022-02-15]. Dostupné z: <https://www.sap.com/cz/products/business-one.html>
- [17] SAP Business One: Features. SAP [online]. [cit. 2022-02-15]. Dostupné z: <https://www.sap.com/cz/products/business-one/features.html>
- [18] LIM, Kyle. Business One vs. ByDesign: A Comparison Of SAP's ERP Software For SMEs. AFON[online]. 23. dubna 2021 [cit. 2022-02-16]. Dostupné z: <https://www.afon.com.sg/blog/business-one-vs-bydesign-a-comparison-of-sap-s-erp-software-for-smes>
- [19] Úvod - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Úvod_-_Výroba
- [20] Technická příprava výroby - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Technická_př%C3%ADprava_výroby_-_Výroba
- [21] Řízení výroby - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Ř%C3%ADzen%C3%AD_výroby_-_Výroba
- [22] Číselník náradí - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Č%C3%ADseln%C3%ADk_nárad%C3%AD_-_Výroba
- [23] Číselník zakázkových modifikací - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Č%C3%ADseln%C3%ADk_zakázkových_modifikac%C3%AD_-_Výroba
- [24] Číselník pracovišť - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Č%C3%ADseln%C3%ADk_pracovišť_-_Výroba

- [25] Číselník strojů - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Č%3%ADseln%3%ADk_strojů_-_Výroba
- [26] Číselník výrobních profesí - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Č%3%ADseln%3%ADk_výrobn%3%ADch_profes%3%AD_-_Výroba
- [27] Výrobní personální zdroje - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Výrobn%3%AD_personáln%3%AD_zdroje_-_Výroba
- [28] Číselník tarifů - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Č%3%ADseln%3%ADk_tarifů_-_Výroba
- [29] Číselník změn - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Č%3%ADseln%3%ADk_změn_-_Výroba
- [30] Plánovací kalendáře - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Plánovac%3%AD_kalendáře_-_Výroba
- [31] Vyráběné dílce - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Vyráběné_d%3%ADlce_-_Výroba
- [32] Slovník pojmů výroby - Výroba: Alternativy. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Slovn%3%ADk_pojmů_výroby_-_Výroba
- [33] Výrobní dokumentace - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Výrobn%3%AD_dokumentace_-_Výroba
- [34] Vedlejší produkty - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Vedlejš%3%AD_produkty_-_Výroba
- [35] Nakupované materiály - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Nakupované_materiály_-_Výroba

- [36] Výrobní a nákladová střediska - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Výrobn%C3%AD_a_nákladová_střediska_-_Výroba
- [37] Plán - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Plán_-_Výroba
- [38] Výrobní příkazy - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Výrobn%C3%AD_př%C3%ADkazy_-_Výroba
- [39] Řady výrobních příkazů - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Řady_výrobn%C3%ADch_př%C3%ADkazů_-_Výroba
- [40] Výrobní operace - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Výrobn%C3%AD_operac_-_Výroba
- [41] Evidence výrobních operací - Výroba. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Evidence_výrobn%C3%ADch_operac%C3%AD_-_Výroba
- [42] Výroba. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/Part23_Moduly/modul_vyroba.htm
- [43] Pracovníci ve výrobě. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba_nastaveni/pracovnici.htm
- [44] Pracoviště. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba_nastaveni/pracoviste.htm
- [45] Tarifní třídy. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba_nastaveni/tarifni_tridy.htm
- [46] Typy technologických postupů. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba_nastaveni/typy_tech_postupu.htm
- [47] Neshody. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba_nastaveni/neshody.htm
- [48] Parametry řad Požadavků na výrobu. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba_nastaveni/parametry_rad_poz.htm

- [49] Parametry řad Výrobních příkazů. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba_nastaveni/parametry_rad_vp.htm
- [50] Kusovníky. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/kusovniky.htm
- [51] Skladové karty a dílčí skladové karty. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Sklady/vec_obsah_sklad_hlavni_dilci_karty.htm
- [52] Kusovníky. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/kusovniky.htm
- [53] Technologické postupy. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/tech_postupy.htm
- [54] Požadavky na výrobu (POZ). ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/pozadavky.htm
- [55] Výrobní příkazy (VYP). ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/vyr_prikazy.htm
- [56] Dílenský plán. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/dilensky_plan.htm
- [57] Pracovní lístky. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/prac_listky.htm
- [58] Kooperace (KOO). ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/kooperace.htm
- [59] Dokončené výrobky (DKV). ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Vyroba/dokonc_vyrobky.htm
- [60] Sledování změn. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Sledovani_zmen/uvod_sledovani_zmen.htm
- [61] Dokumenty a přílohy. ABRA Gen [online]. [cit. 2022-05-18]. Dostupné z: https://help.abra.eu/cs/19.5/G4/Content/PartS_Dokumenty/uvod_dokumenty.htm
- [62] Dokumenty všeobecné - Číselníky. HELIOS [online]. [cit. 2022-05-18]. Dostupné z: https://public.helios.eu/inuvio/doc/cs/index.php?title=Dokumenty_vseobecné_-_Č%C3%ADseln%C3%ADky

-
- [63] Annotations Reference. Doctrine [online]. [cit. 2022-05-18]. Dostupné z: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.11/reference/annotations-reference.html>
- [64] Getting Started with Doctrine: Obtaining the EntityManager. Doctrine [online]. [cit. 2022-05-18]. Dostupné z: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.12/tutorials/getting-started.html>
- [65] Generating Migrations. Doctrine [online]. [cit. 2022-05-18]. Dostupné z: <https://www.doctrine-project.org/projects/doctrine-migrations/en/3.4/reference/generating-migrations.html>
- [66] REST. Codeception [online]. [cit. 2022-05-19]. Dostupné z: <https://codeception.com/docs/modules/REST>
- [67] DB. Codeception [online]. [cit. 2022-05-19]. Dostupné z: <https://codeception.com/docs/modules/Db>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Application Programming Interface
CRUD	Create Read Update Delete
DTO	Data Transfer Object
ERP	Enterprise Resource Planning
ORM	Object Relational Mapping
REST	Representational State Transfer

SEZNAM OBRÁZKŮ

Obr. 5.1.	Doménový diagram	35
Obr. 5.2.	ERD	36
Obr. 7.1.	HTTP dotaz pro vytvoření záznamu komodity	47
Obr. 7.2.	Response HTTP dotazu pro vytvoření záznamu komodity	47
Obr. 7.3.	Response HTTP dotazu pro získání seznamu komodit včetně archi- vovaných položek	48
Obr. 8.1.	Výsledky testů.....	52

SEZNAM TABULEK

Tab. 5.1. Entity a akce 37

SEZNAM PŘÍLOH

P I. Příložené CD

PŘÍLOHA P I. PŘILOŽENÉ CD

Přiložené CD obsahuje:

- bakalářskou práci ve formátu PDF,
- zdrojové kódy aplikace.