



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTÉM PRO ŘÍZENÍ VÝROBY DEMONSTRÁTORU PRŮMYSLU 4.0

MES FOR INDUSTRY 4.0 DEMONSTRATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Martiník

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Václav Kaczmarczyk, Ph.D.

BRNO 2023

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Tomáš Martiník

ID: 221308

Ročník: 3

Akademický rok: 2022/23

NÁZEV TÉMATU:

Systém pro řízení výroby demonstrátoru Průmyslu 4.0

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout a implementovat systém pro řízení výroby (MES), pomocí kterého bude možné plánovat a rozvrhovat výrobu několika výrobních buněk.

1. Popište funkcionalitu MES systémů. Proveďte rešerši existujících systémů MES a posuďte vhodnost jejich nasazení a případnou nutnost úprav pro zamýšlené použití.
2. Vybraný systém zprovozněte a navrhňte nutné úpravy.
3. Proveďte integraci tohoto systému do SW prostředí demonstrátoru Průmyslu 4.0.
4. Otestujte funkčnost systému.
5. Navrhňte a implementujte doplňující moduly (např. pro výpočet výkonnostních indikátorů či další dashboardy).
6. Celé řešení zadokumentujte.

Termín zadání: 6.2.2023

Termín odevzdání: 22.5.2023

Vedoucí práce: Ing. Václav Kaczmarczyk, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato bakalářská práce se zabývá výrobními systémy, jejich funkcionalitami a rolí v Průmyslu 4.0. Práce se zaměřuje na návrh a implementaci MES řešení pro demonstrátor Průmyslu 4.0. Je provedena rešerše dostupných open-source systémů a na základě požadavků pro systém je jedno řešení vybráno, upraveno pro potřeby demonstrátoru a integrováno do jeho systému.

Klíčová slova

MES systém, Průmysl 4.0, open-source, CRM, ERP, automatizace průmyslové výroby

Abstract

This bachelor's thesis deals with manufacturing systems, their functionalities and their role in Industry 4.0. This thesis focuses on the design and implementation of the MES solution for the Industry 4.0 demonstrator. Research of available open-source systems is conducted and, based on the requirements for the system, one solution is selected, modified for the demonstrator's needs and integrated into its system.

Keywords

MES system, Industry 4.0, open-source, CRM, ERP, automation of industrial production

Bibliografická citace

MARTINÍK, Tomáš. Systém pro řízení výroby demonstrátoru Průmyslu 4.0. Brno, 2023. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/151332>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Václav Kaczmarczyk.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Tomáš Martiník
VUT ID studenta:	221308
Typ práce:	Bakalářská práce
Akademický rok:	2022/23
Téma závěrečné práce:	System pro řízení výroby demonstrátoru Průmyslu 4.0

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 16. května 2023

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Václavu Kaczmarczykovi Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc, trpělivost a další cenné rady při zpracování mé bakalářské práce. Rovněž bych rád vyjádřil svou upřímnou vděčnost svým rodičům, přítelkyni a také svým blízkým kamarádům za jejich neocenitelnou podporu během mého studia a náročných okamžiků. Bez jejich důvěry a podpory bych nebyl schopen překonat obtíže na cestě ke svým cílům.

V Brně dne: 16. května 2023

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. MES SYSTÉMY	12
1.1 FUNKCIONALITY MES SYSTÉMU DLE ISA-95.....	13
1.2 FUNKCIONALITY MES SYSTÉMU MIMO FUNKCE ISA-95	14
1.2.1 <i>Analýza výkonu</i>	14
1.2.2 <i>Sběr a získávání dat</i>	16
1.2.3 <i>Sledování a mapování života výrobku</i>	16
1.2.4 <i>Správa zdrojů pro výrobu</i>	16
1.3 ROLE MES SYSTÉMŮ V PRŮMYSLU 4.0	16
2. DEMONSTRÁTOR VÝROBNÍHO PODNIKU.....	17
2.1 OBECNÝ POPIS.....	17
2.2 PARAMETRY	17
3. STÁVAJÍCÍ ŘEŠENÍ	18
4. REŠERŠE SYSTÉMŮ	19
4.1 PARAMETRY REŠERŠE	19
4.2 ODOO	19
4.2.1 <i>Architektura</i>	19
4.2.2 <i>Moduly Odoo 15 pro výrobní procesy</i>	20
4.2.3 <i>Instalace Odoo pro Windows 10</i>	21
4.3 QCADOO.....	27
4.3.1 <i>Popis architektury</i>	27
4.3.2 <i>Popis pluginů</i>	28
4.3.3 <i>Popis Qcadoo frameworku a jeho jednotlivých pluginů</i>	31
4.3.4 <i>Funkcionality Qcadoo</i>	32
4.3.5 <i>Instalace QCADOO Community pro Windows</i>	33
4.3.6 <i>Spuštění Qcadoo</i>	38
4.4 DOLIBARR	40
4.4.1 <i>Funkcionality v oblasti MES systémů</i>	40
4.4.2 <i>Instalace na Windows 10</i>	40
4.5 ERPNEXT.....	41
4.5.1 <i>Funkcionality v oblasti výroby</i>	41
4.5.2 <i>Instalace na Windows 10</i>	41
4.6 IMES	41
4.6.1 <i>Popis struktury softwaru</i>	42
4.6.2 <i>Funkcionality</i>	42
4.7 ESPOCRM.....	44
4.7.1 <i>Instalace</i>	44
4.7.2 <i>Architektura</i>	44
4.7.3 <i>Backend</i>	47

4.7.4	<i>Shrnutí</i>	50
4.8	SROVNÁNÍ SYSTÉMŮ	51
5.	NÁVRH A IMPLEMENTACE NOVÉHO MES ŘEŠENÍ	53
5.1	NASTAVENÍ A SPUŠTĚNÍ SYSTÉMU	53
5.2	IMPLEMENTACE NOVÝCH MODULŮ.....	55
5.2.1	<i>Správa objednávek</i>	56
5.2.2	<i>Správa vybavení</i>	57
5.2.3	<i>Správa personálu</i>	59
5.2.4	<i>Správa materiálu</i>	62
5.2.5	<i>Produkty a produktové segmenty</i>	65
5.2.6	<i>Procesní segmenty</i>	66
5.2.7	<i>Správa uživatelů, jejich rolí a práv</i>	68
5.3	IMPLEMENTACE EXEKUČNÍHO MODULU	69
5.4	HIERARCHIE CELÉHO SYSTÉMU	71
5.5	MOŽNOSTI DALŠÍ PRÁCE.....	71
5.5.1	<i>Skladový modul</i>	71
5.5.2	<i>Modul pro KPI</i>	71
5.5.3	<i>Využití umělé inteligence</i>	72
5.5.4	<i>Vytvoření či integrace nadřazeného ERP systému</i>	72
6.	ZÁVĚR	73
	LITERATURA	74
	SEZNAM SYMBOLŮ A ZKRATEK	76

SEZNAM OBRÁZKŮ

Obrázek 1: Pyramida ISA-95.....	12
Obrázek 2: Funkcionality ISA-95.....	14
Obrázek 3: model demonstrátoru průmyslové výroby	17
Obrázek 4: stávající MES řešení.....	18
Obrázek 5: Třívrstvá architektura	20
Obrázek 6: Visual Studio Build Tools 2019	22
Obrázek 7: Vytvoření serveru v PostgreSQL (General)	23
Obrázek 8: Vytvoření serveru v PostgreSQL (Connection).....	24
Obrázek 9: Vytvoření Login/Group Role v PostgreSQL (Privileges)	25
Obrázek 10: Moduly pro výrobu v Odoo MES.....	26
Obrázek 11: Vrstvy architektury Qcadoo	28
Obrázek 12: Vnitřní struktura pluginu aplikace Qcadoo	29
Obrázek 13: Přejechy stavů pluginu aplikace Qcadoo.....	30
Obrázek 14: Architektura pluginů Qcadoo	31
Obrázek 15: Přidání SSH klíče do GitHubu	35
Obrázek 16: Seznam SSH klíčů v GitHub účtu	35
Obrázek 17: Apache Maven 3.8.3 ke stažení.....	36
Obrázek 18: Proměnné prostředí Windows 10	36
Obrázek 19: Přidání Apache Maven jako proměnnou prostředí Windows 10	37
Obrázek 20: Kontrola verze Apache Maven v příkazovém řádku	37
Obrázek 21: Chybová hláška při instalaci Qcadoo	39
Obrázek 22: Moduly Dolibarr systému pro výrobu	40
Obrázek 23: IMES	42
Obrázek 24: Příklad úpravy entity ve správě entit v EspoCRM	46
Obrázek 25: Databázová konfigurace EspoCRM	53
Obrázek 26: Zobrazení EspoCRM po prvním spuštění	55
Obrázek 27: Detail objednávky v MES systému	56
Obrázek 28: Databázový model objednávek.....	57
Obrázek 29: Definice vybavení Shaker process cell.....	58
Obrázek 30: Databázový diagram vybavení	59
Obrázek 31: Příklad definice personálu	60
Obrázek 32: Databázový diagram personálu	61
Obrázek 33: Příklad definice vybavení.....	62
Obrázek 34: Databázový diagram materiálu.....	64
Obrázek 35: Příklad segmentu produktu.....	65
Obrázek 36: Databázový diagram produktů a produktových segmentů	66
Obrázek 37: Příklad segmentu procesu.....	67
Obrázek 38: Databázový diagram procesů	68
Obrázek 39: Role a práva v aplikaci EspoCRM	69
Obrázek 40: Repositář exekučního modulu	70
Obrázek 41: Hierarchie celého systému.....	71

SEZNAM TABULEK

Tabulka 1: Srovnání popsaných systémů	52
Tabulka 2: Nastavení PHP pro aplikaci	54
Tabulka 3: Nastavení databáze pro aplikaci	54

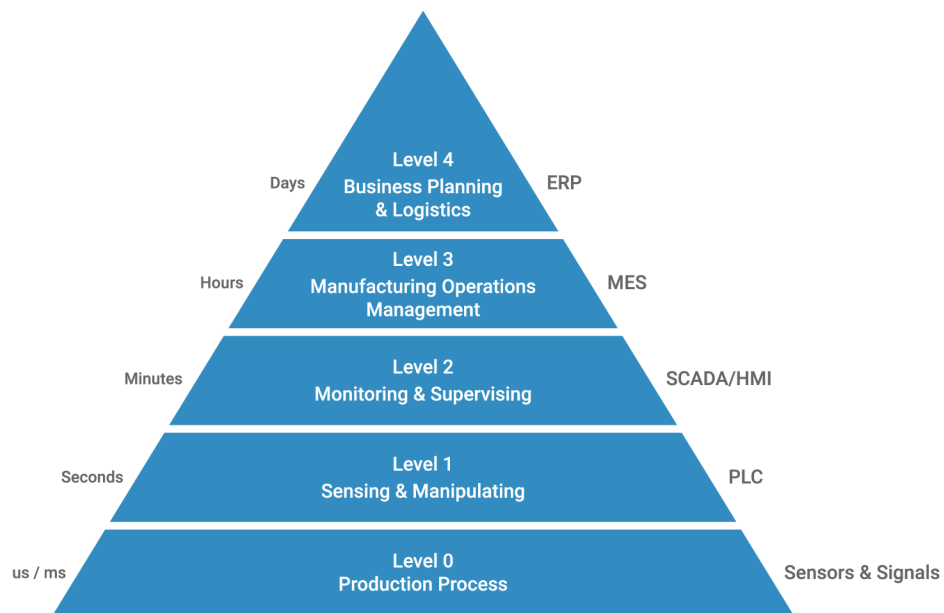
ÚVOD

Průmyslové odvětví se potýká s rostoucími výzvami v oblasti konkurenceschopnosti, efektivity a inovací. S nástupem čtvrté průmyslové revoluce se otevírají nové možnosti pro zefektivnění výrobních procesů. Tento trend je založen na integraci moderních technologií a digitalizaci průmyslových procesů.

Tzv. MES systémy jsou klíčovou součástí Průmyslu 4.0. Slouží totiž k řízení, monitorování a optimalizaci výrobních procesů. Tato bakalářská práce se zaměřuje na řešení dostupných open-source MES řešení a jejich aplikaci na demonstrátoru Průmyslu 4.0. Cílem práce je vybrat vhodné řešení, přizpůsobit jej konkrétním požadavkům demonstrátoru a integrovat ho do výrobního procesu.

1. MES SYSTÉMY

Manufacturing Execution System (MES), známý také jako výrobní informační systém, je softwarové řešení, které má za úkol zvyšovat efektivitu a kvalitu výroby. Tyto systémy pracují v reálném čase a v hierarchii podniku se nachází mezi operátorskou úrovní, kde se operuje s SCADA nebo HMI, a úrovní řízení podniku, kde se vyskytují ERP (celopodnikové) systémy. MES systémy tedy umožňují v reálném čase monitorovat a řídit výrobu, poskytují informace o stavu výrobních procesů, zajišťují sledování kvality výroby a pomáhají s plánováním výroby. [1]



Obrázek 1: Pyramida ISA-95 [2]

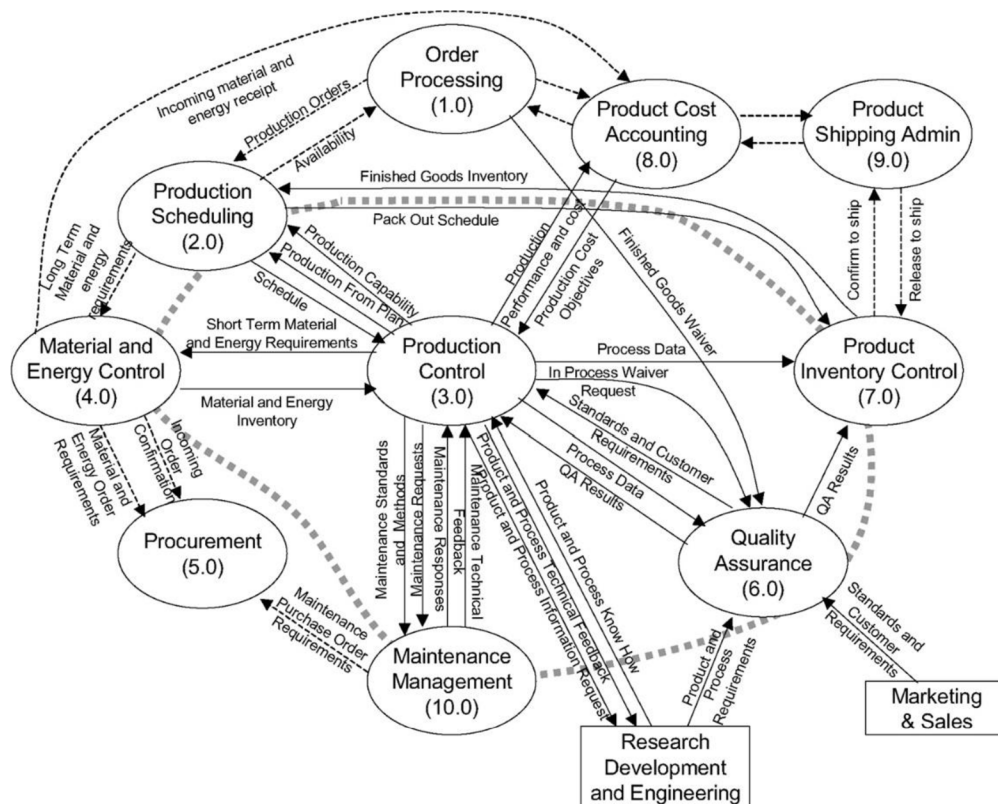
1.1 Funkcionality MES systému dle ISA-95

ISA-95, zkráceně ANSI/ISA-95, je mezinárodní standard pro integraci podnikových a řídicích systémů. Poskytuje abstraktní modely a standardní terminologii pro výměnu informací mezi obchodními systémy a výrobními operacemi ve společnosti.

Cílem standardu je vytvořit efektivní rozhraní mezi ERP systémem a MES. Norma poskytuje model a terminologii, která usnadňuje komunikaci, snižuje náklady a umožňuje bezchybnou integraci. [3] Funkcionality dle tohoto standardu:

- **Zpracování objednávek**
Správa přijatých objednávek, příprava pro zařazení do výrobního procesu, nastavení priorit apod.
- **Plánování výroby**
Jedná se o nastavení pořadí výroby, tzv. fronta práce (co se kdy bude vyrábět, jaká výroba má přednost), cílem je procesy naplánovat tak, aby došlo k co nejvyšší efektivitě, minimálnímu zdržení, snaha o co nejrychlejší či nejméně náročný průběh. [1] [3]
- **Řízení produkce**
Řízení plánovaných procesů, přiřazování materiálu, osob apod., možnost sledování plánovaných procesů a aktuálního stavu výroby, zajišťování materiálu apod.
- **Řízení materiálu a energie**
Definování a sledování materiálu a energie, které jsou spojeny s výrobním procesem.
- **Pořizování (nákup)**
Pořizování potřebného materiálu a komponentů pro výrobu produktů, definování potřeb, výběr vhodných dodavatelů, vyjednávání s nimi ohledně cen a podmínek, vytváření objednávek a sledování dodání zboží.
- **Řízení kvality**
Sledování kvality produktů vycházejících z výroby, tzn. kontrola kvality výrobků během výrobního procesu, testování výrobků a definování standardů.
- **Řízení skladu produktů**
Správa a organizace zásob, přehled o tom, kde se produkty nacházejí, snaha o efektivní využití skladových prostor.
- **Sledování nákladů produktu**
Sledování nákladů spojených s výrobou produktu.
- **Dodání produktu**
Plánování a organizace dopravy, balení a označování produktů, dokumentace pro výdej a příjem, sledování zásilek a komunikace s dopravci a zákazníky.

- **Řízení údržby**
Sledování stavu vybavení, plánování a správa preventivní údržby výrobních zařízení, kroky pro odstranění poruchy vybavení.



Obrázek 2: Funkcionality ISA-95 [56]

1.2 Funkcionality MES systému mimo funkce ISA-95

1.2.1 Analýza výkonu

Jako výkonnostní ukazatel u MES systémů se často využívá OEE (Overall equipment effectiveness), neboli česky celková efektivita zařízení. OEE vyjadřuje procento z celkového času, kdy je výroba produktivní. 100% hodnota OEE vyjadřuje výrobu bez chybných produktů v nejrychlejší možné době a bez časové přestávky. Výsledek OEE tedy znázorňuje poměrně přesný výsledek efektivity výroby. [4]

Zjednodušený výpočet OEE je poměr součinu počtu výrobků s ideální dobou cyklu a plánovaného času výroby. Tento výpočet však nezohledňuje hlavní složky zodpovědné za menší efektivitu výroby.

$$OEE = \frac{\text{počet výrobků} \times \text{ideální doba cyklu}}{\text{plánovaný čas výroby}} \quad (1)$$

Mezi tři „ztrátové“ složky patří dostupnost, výkon a kvalita. Následující vztah počítá právě s těmito zdroji ztrát.

$$OEE = \text{dostupnost} \times \text{výkon} \times \text{kvalita} \quad (2)$$

Dostupnost bere v úvahu všechny časové přestávky, které zdržely výrobu (většinou minimálně v řádu minut) – jak plánované, tak neplánované. Vypočítá se jako podíl doby běhu výroby a plánovaného času výroby. Doba běhu výroby odpovídá rozdílu plánovaného času výroby a součtu časových přestávek. Výsledek udává, jak moc byl plánovaný čas výroby zdržen v reálném procesu.

$$\text{dostupnost} = \frac{\text{plánovaný čas výroby} - \text{časové přestávky}}{\text{plánovaný čas výroby}} \quad (3)$$

$$\text{dostupnost} = \frac{\text{doba běhu výroby}}{\text{plánovaný čas výroby}} \quad (4)$$

Výkon bere v úvahu vše, co zpomaluje rychlost výroby neboli vše, co způsobuje rychlost výroby nižší, než je maximální možná rychlost. Na rozdíl od dostupnosti zohledňuje i velmi krátké časové přestávky (většinou kratší než minuty) a také zdlouhavost cyklů. Vyjadřuje se jako poměr čistého času výroby ku celkovému času výroby.

$$\text{výkon} = \frac{\text{ideální čas cyklu} \times \text{celkový počet výrobků}}{\text{celkový čas výroby}} \quad (5)$$

Ideální čas cyklu je nejrychlejší možný čas, za který může cyklus za optimálních podmínek proběhnout. Po vynásobení ideálního času cyklu celkovým počtem výrobků získáme nejrychlejší možný čistý čas výroby, za který lze zpracovat všechny výrobky při optimálních podmínkách. Výsledek výkonu by neměl přesáhnout hodnotu 100 %. Pokud

hodnota výkonu přesahuje 100 %, pravděpodobně došlo k chybnému určení ideálního času výrobního cyklu.

Kvalita bere v úvahu výrobky, které obsahují vadu či nebyly úspěšně zpracovány na první pokus.

$$kvalita = \frac{\text{počet úspěšně zpracovaných výrobků}}{\text{celkový počet výrobků}} \quad (6)$$

[4]

1.2.2 Sběr a získávání dat

Shromažďování a ukládání dat z výroby pro zpětné využití včetně sledování poruch. Umí zachycovat data v reálném čase, což je značná výhoda oproti zapisování dat ze strojů člověkem. Data jsou často vizualizována a využívána např. k plánování. Ke zpracování dat bývají často využívány analytické nástroje, popř. umělá inteligence. Data mohou být historizována, ukládají se např. data o výkonu stroje, stavy strojů a jejich konfigurace, procesní parametry atd. [1]

1.2.3 Sledování a mapování života výrobku

Informace o zdrojích a výrobním procesu produktu. Tato data mohou být velmi důležitá pro výrobce pro splnění různých legislativních předpisů apod., mapuje vývoj výrobku ve výrobním procesu.

1.2.4 Správa zdrojů pro výrobu

Pro výrobní proces a jeho plánování je třeba mít přehled o zdrojích pro výrobu. Do této skupiny spadá jak materiál, tak personální kapacita, stoje, nástroje či např. energie. MES systém dává informace o dostupnosti, stavu a případnému přidělení zdrojů na konkrétní místo výrobního procesu.

1.3 Role MES systémů v Průmyslu 4.0

Průmysl 4.0 označuje zavádění inovací a moderních technologií do výrobních podniků. Jeho cílem je maximalizovat digitalizaci a efektivitu výrobních procesů využitím tzv. internetu věcí, umělé inteligence, sběru dat, vzdáleného přístupu apod. Pro zefektivnění těchto procesů jsou MES systémy klíčové, protože podnikům umožňují sbírat data v reálném čase, monitorovat a analyzovat je a navrhnout zlepšení.

2. DEMONSTRÁTOR VÝROBNÍHO PODNIKU

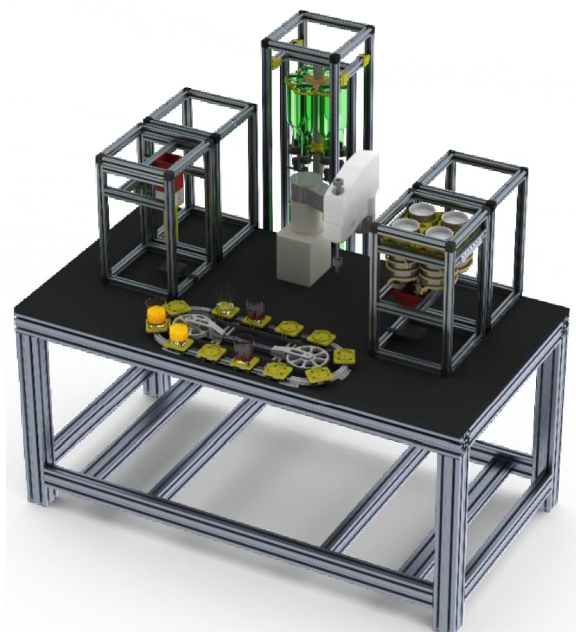
2.1 Obecný popis

Máme k dispozici přípravek, který slouží na Ústavu automatizace a měřicí techniky Fakulty elektrotechniky a komunikačních technologií VUT v Brně pro demonstraci průmyslové výroby na příkladu výroby alkoholických a nealkoholických nápojů. Demonstrátor nabízí studentům možnost vidět, jak probíhá moderní průmyslová výroba, a seznámit se s nejnovějšími technologiemi v oblasti automatizace a řízení výroby.

2.2 Parametry

Základem přípravku je pracovní plocha s rozměry 2000 x 1000 mm, na které se nachází různé výrobní buňky a další zařízení pro vykonání výrobních procesů. Pod pracovní plochou je prostor pro řídicí a podpůrná zařízení. Mezi zařízení demonstrátoru se řadí:

- autonomní buňka zásobníku alkoholu,
- autonomní buňka zásobníku nealkoholických,
- autonomní buňka drtiče ledu,
- autonomní buňka míchače,
- autonomní buňka zásobníku sklenic,
- dopravníkový pás,
- robotický SCARA manipulátor. [5]



Obrázek 3: model demonstrátoru průmyslové výroby [5]

3. STÁVAJÍCÍ ŘEŠENÍ

Pro demonstrátor průmyslové výroby byl navržen jednoduchý MES systém jako webová aplikace, implementovaná pomocí technologií .NET Framework, Windows Communication Foundation a DotVVM Framework. Tento systém obsahuje moduly pro správu materiálu a vybavení dle standardu ISA-95, zároveň modul pro produkty a produktové segmenty, stejně tak jako modul procesních segmentů.

Značnou výhodou tohoto řešení je integrace do systému demonstrátoru. Systém má vlastní exekuční modul, implementovaný v jazyce C#, který umožňuje komunikaci mezi MES systémem a demonstrátorem pomocí S7 protokolu. Exekuční modul má implementované třídy, které odpovídají tabulkám v databázi MES systému. Z této databáze si exekuční modul vyčítá data a následně umožní výrobu produktů z objednávek pomocí vlastního GUI.

Nevýhodou stávajícího řešení jsou chybějící moduly, které se řadí mezi základní funkcionality MES systémů, např. modul pro KPI, reporting, správa personálu, správa skladových zásob apod. Aplikace nedisponuje dokumentací a je tak obtížné a časově náročné vytvořit nové moduly, jelikož je programátor musí vytvořit od začátku až do konce.

Akce	Id	Popis	Třída
<input type="checkbox"/> + [edit] [delete]	2	Rum Republica	Rum
<input type="checkbox"/> + [edit] [delete]	7	Bílý rum	Rum
<input type="checkbox"/> + [edit] [delete]	8	Coca-Cola	Kolové nápoje
<input type="checkbox"/> + [edit] [delete]	9	Vodka broskvová	Vodky
<input type="checkbox"/> + [edit] [delete]	10	Vodka	Vodky
<input type="checkbox"/> + [edit] [delete]	11	Grenadína	Sirup
<input type="checkbox"/> + [edit] [delete]	12	Soda	Voda
<input type="checkbox"/> + [edit] [delete]	13	Led	Voda
<input type="checkbox"/> + [edit] [delete]	14	Soutěžní vzorek č. 1	Soutěžní vzorky
<input type="checkbox"/> + [edit] [delete]	16	Sklenice nízká, typ Whiskey	Sklenice
<input type="checkbox"/> + [edit] [delete]	18	Soutěžní vzorek č. 3	Soutěžní vzorky

Obrázek 4: stávající MES řešení

4. REŠERŠE SYSTÉMŮ

4.1 Parametry rešerše

Na základě nedostatků stávajícího řešení se naskytla úvaha o jeho nahrazení open-source MES systémem dostupným na internetu a jeho následná úprava pro potřeby demonstrátoru a integrace do systému demonstrátoru. Z tohoto důvodu byla provedena rešerše open-source MES systémů, jejíž parametry byly následně poupraveny. Původními požadavky na systém byly:

- dokumentace pro vývojáře
- open-source a zdarma dostupný
- aktivní komunita, např. v podobě fóra
- jednoduchá tvorba nových modulů
- rozšiřitelnost existujících modulů
- moduly pro výrobu (např. skladové zásoby, správa zdrojů, plánování výroby apod.)

Po provedení části rešerše byla rešerše rozšířena o CRM a ERP open-source systémy, které splňují požadavky na systém a daly by se tudíž využít jako systém pro řízení výroby.

4.2 Odoo

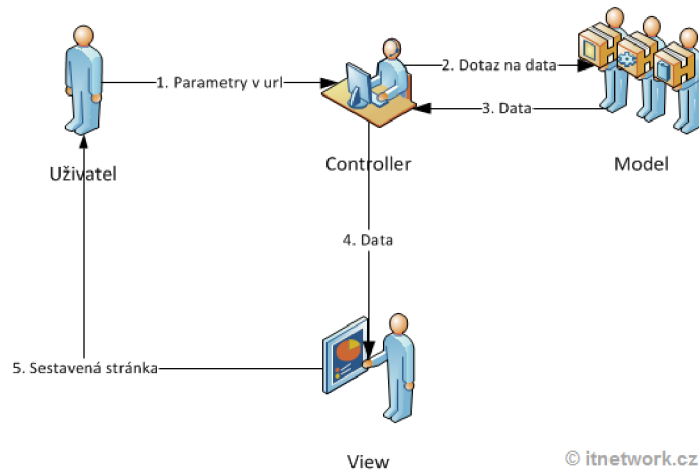
Jedná se o skupinu open-source aplikací pro řízení podniku. V širokém spektru softwarů, které se věnují např. i financím, marketingu či tvorbě webů, se nachází i aplikace Odoo Manufacturing. Tato aplikace se řadí výrobní moduly, její tvůrci při prezentaci kladou důraz na řízení kvality a sledování životního cyklu výrobku. [6]

4.2.1 Architektura

Program má třívrstvou architekturu, prezentační vrstva je složená z HTML 5, CSS a JavaScriptu, slouží pro zobrazení a kontakt s uživatelem. Aplikační vrstva je napsána v jazyce Python a datovou vrstvu tvoří PostgreSQL jako databázový systém. Samotná aplikace se skládá z modulů, každý z těchto modulů reprezentuje část systému, která se zabývá nějakou konkrétní oblastí (Sales, Finance, Product Lifecycle Management, Manufacturing apod.) Každý z modulů je uložen v samostatné složce ve složce modules v projektu. Pro uživatele je k dispozici Odoo Enterprise a Odoo Community, pro účely této práce je použita a popsána druhá zmíněná edice, jedná se o open-source. [6]

Třívrstvá architektura je logická architektura, která rozděluje program na tři vrstvy. První vrstva je vrstva prezentační. Tato vrstva prezentuje výsledky uživateli a přijímá požadavky (vstupy). Druhou vrstvou je vrstva aplikační (aplikační server), která

vykonává operace, výpočty apod. na základě požadavků. Třetí a nejnižší vrstvou je vrstva datová, která slouží k práci s daty. [7]



Obrázek 5: Třívrstvá architektura [8]

4.2.2 Moduly Odoo 15 pro výrobní procesy

MRP

V aplikaci se dají využívat základní funkce MRP systémů, jakými jsou např. správa vyráběných produktů, spuštění produkce výrobků, možnost spravovat opravy či servis položek a také možnost přetavit výsledný produkt zpět na základní komponenty. Odoo také umí skenovat čárový kód (např. sériových čísel) pro urychlení výrobních procesů.

Dále je uživateli umožněno plánovat výrobu (a zpětně plány měnit), vytvářet harmonogramy, program také disponuje plánovačem, který na základě OEE a kapacity pomůže vytvořit harmonogram. Uživatel má přístup k dostupným zdrojům, což usnadňuje plánování do budoucna.

PLM

PLM neboli Product Lifecycle Management se zabývá životním cyklem produktu, jak jeho výrobní částí, tak např. prodejem. [9] Odoo nabízí jednoduché zobrazení změn mezi verzemi produktů a jejich dokumenty, možnost přikládat plány a dokumenty přímo ke kusovníku.

Řízení kvality

„Řízení kvality je především snaha o neustálé zlepšování, jehož výsledkem jsou efektivnější procesy a ve svém důsledku snížené náklady a zvýšená produktivita.“ [10]

Odoo se pyšní automatickým spuštěním kontrol kvality ve výrobě a upozorněním na kontrolu kvality.

Údržba

V aplikaci je možné nastavit automatické údržby, naplánovat je do kalendáře či je spouštět manuálně z ovládacího panelu. Odoo také nabízí výpočet střední doby mezi poruchami (MTBF), což je statistický údaj, který udává čas v hodinách, jak dlouho je schopen např. stroj pracovat bez selhání (přerušení). [11]

Report

Odoo umí vykazovat zprávy z výroby obsahující komponenty využití k výrobě, výpočtu nákladů na základě ceny materiálu a operací a celkovou efektivnost zařízení (OEE). [6]

4.2.3 Instalace Odoo pro Windows 10

Na webových stránkách www.odoo.com je možnost kliknout na odkaz v záhlaví stránky a stáhnout tak instalační balíček pro open-source verzi Odoo 15, která však ani po vykonání všech bodů z návodu autorů na instalaci nelze spustit. Odoo 15 se tedy musí nainstalovat pomocí zdrojového kódu. Pro instalaci a spuštění Odoo open-source se tedy musí vykonat hned několik kroků.

Instalace Python

Python je klíčový programovací jazyk pro Odoo, je v něm napsaná většina zdrojového kódu.

Na webu python.org kliknout na tlačítko *Downloads* v hlavní liště stránky. Následně vybrat možnost *Windows* a stáhnout verzi 3.8.10 (doporučená verze pro tuto popsanou instalaci) souboru *Windows installer*. Následně spustit stažený soubor, po zobrazení instalačního okna vyplnit možnost *Add Python 3.8 to PATH*. Po dokončení instalace vybrat v instalačním okně možnost *Disable path length limit*.

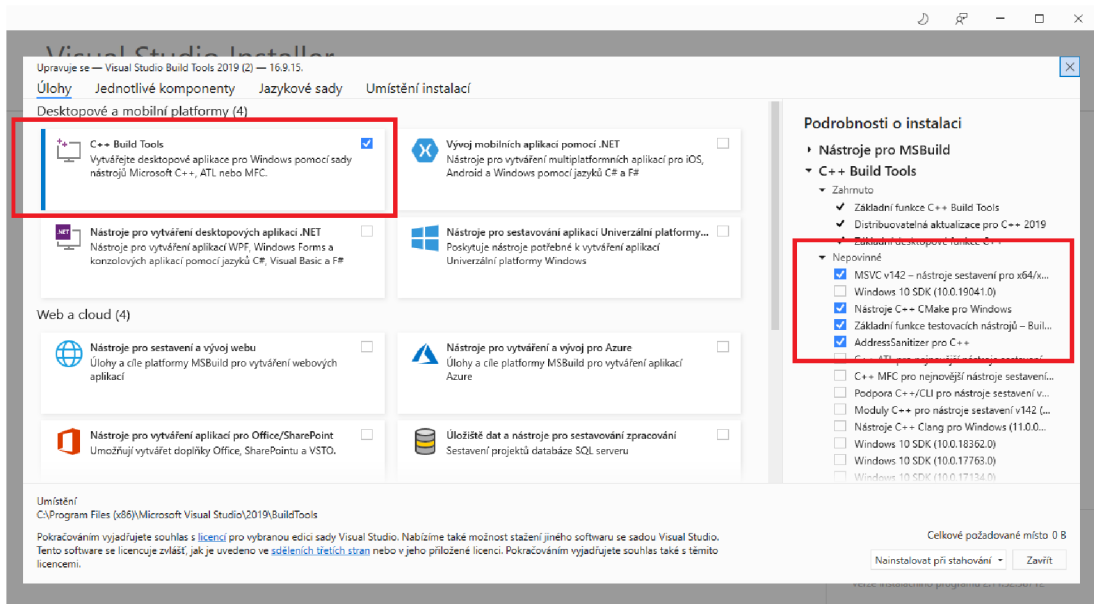
Dalším krokem je update *pip*, což je správce balíčků pythonu (package manager). Do příkazového řádku zadáme následující příkaz:

```
python -m pip install --upgrade pip
```

Instalace Windows Build Tools

Z webové stránky <https://visualstudio.microsoft.com/cs/vs/older-downloads/> rozbalit možnost 2019 a v sekci Visual Studio 2019 a další produkty kliknout na tlačítko stáhnout.

Po přihlášení do Microsoft účtu stáhnout soubor Build Tools for Visual Studio 2019 (version 16.9). Následuje spuštění zmíněného souboru a zvolení (zpravidla první) možnosti C++ Build Tools. V pravém sloupci je možné nevybrat možnost Windows 10 SDK, která není pro instalaci a chod Odoo potřeba.



Obrázek 6: Visual Studio Build Tools 2019

Instalace modulu wkhtmltopdf

Wkhtmltopdf jsou open-source nástroje pro příkazový řádek pro převod formátu HTML do PDF. [12]

Z webové stránky <https://github.com/wkhtmltopdf/wkhtmltopdf/releases/tag/0.12.5> je třeba stáhnout soubor `wkhtmltox-0.12.5-1.msvc2015-win64.exe`, spustit jej a za pomoci instalačního okna nainstalovat.

Instalace git

V případě absence softwaru `git` je třeba jej nainstalovat. Na webové stránce <https://git-scm.com/> je dostupná ke stažení verze pro OS Windows.

Instalace PostgreSQL

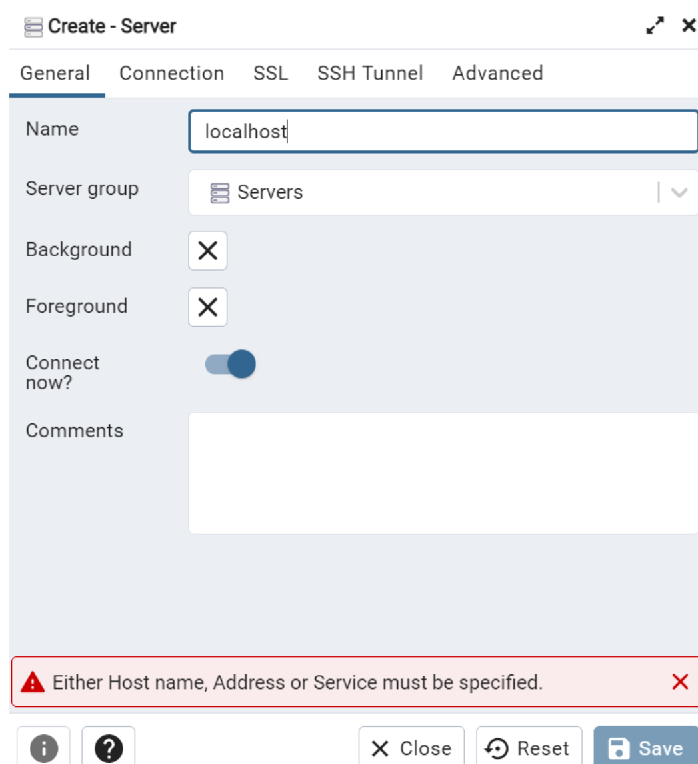
PostgreSQL je open-source objektový databázový systém [13], díky kterému budeme moci vytvořit a využít databázi pro Odoo. Pro instalaci je nutné z oficiálního webu

<https://www.postgresql.org/> vybrat možnost *Download*, následně vybrat verzi OS (pro tento návod se jedná o verzi *Windows*) a následně přes odkaz pod textem *Download the installer* přejít na stránku s instalačními soubory. Pro tento návod je doporučeno stáhnout verzi 13.5.

Po spuštění instalačního souboru volíme přednastavené volby a následně nás instalační okno vyzve o heslo, které vytvoříme a zadáme (heslo pro uživatele). Dokončíme instalaci a přidáme podsložku *bin* mezi proměnné prostředí, např.: `C:\Program Files\PostgreSQL\13\bin`.

Vytvoření serveru v PostgreSQL

Pro vytvoření serveru a databázi pro Odoo je nutné spustit program pgAdmin 4, který byl nainstalován současně s instalací PostgreSQL v předchozím kroku. Po otevření programu je uživatel vyzván k zadání hesla, které bude z důvodu ochrany databázi a serverů vyzván při každém spuštění programu. Následný postup je takový, že z nástěnky pgAdmin se zvolí možnost *Add new server*. V okně pro vytvoření serveru je potřeba vyplnit název serveru a v záložce *Connection* je zase nutné zadat název hostitele (zadáváme „localhost“) a heslo, které bylo vytvořeno v kroku č. 5, který popisoval instalaci PostgreSQL systému.



Obrázek 7: Vytvoření serveru v PostgreSQL (General)

Create - Server ↗ ✕

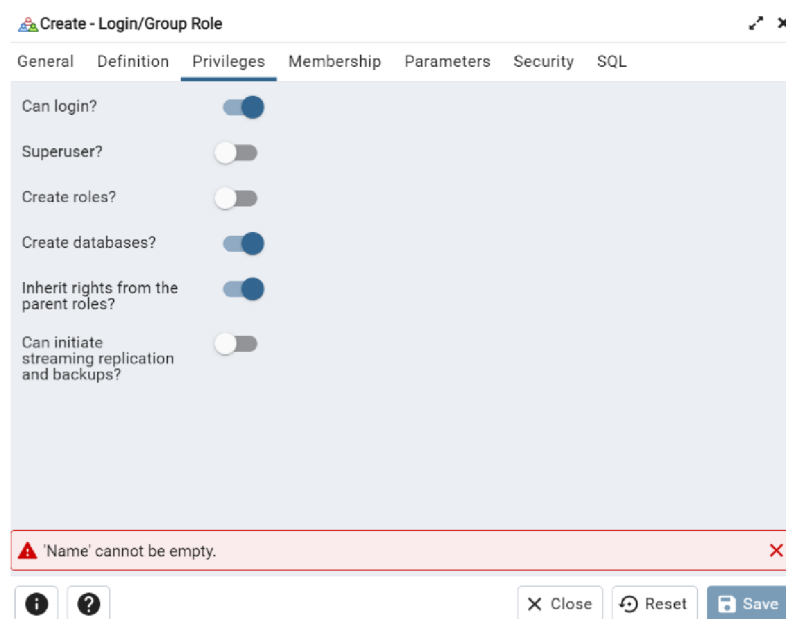
General **Connection** SSL SSH Tunnel Advanced

Host name/address	<input type="text" value="localhost"/>
Port	<input type="text" value="5432"/>
Maintenance database	<input type="text" value="postgres"/>
Username	<input type="text" value="postgres"/>
Kerberos authentication?	<input type="checkbox"/>
Password	<input type="password"/>
Save password?	<input type="checkbox"/>
Role	<input type="text"/>
Service	<input type="text"/>

Obrázek 8: Vytvoření serveru v PostgreSQL (Connection)

Vytvoření přístupu k databázi

Pro umožnění přístupu do databáze je potřeba vytvořit uživatele databáze. V levém sloupci programu se musí kliknout na možnost *PostgreSQL 13* a zadat uživatelské heslo. Následuje kliknutí pravým tlačítkem na *PostgreSQL 13*, v záložce vybrat možnost *Create* a v podzáložce zvolit možnost *Login/Group Role*. V okně v záložce *General* se vyplní jméno (např. *odoo*), v záložce *Definition* se stanoví heslo. V záložce *Privileges* se uživateli přiřazují práva – žádoucí je zatrhnoutí možnosti *Can login?* a *Create databases?*. Zbytek nastavení se nechává ve výchozích hodnotách.



Obrázek 9: Vytvoření Login/Group Role v PostgreSQL (Privileges)

Instalace Odoo

Nejprve se musí stáhnout zdrojový kód. Pro tyto účely se doporučuje vytvoření vlastní složky pro projekt (např. *C:\odoo*). Po kliknutí pravým tlačítkem na vytvořenou složku se zobrazí několik možností, pro získání zdrojového kódu je nutné vybrat možnost *Git Bash Here*, která otevře příkazový řádek pro git. Pro stáhnutí community edice je třeba v dané složce zadat příkaz:

```
git clone https://github.com/odoo/odoo.git -b 15.0.
```

Následně se přejde do běžného příkazového řádku a pomocí příkazu:

```
pip install setuptools wheel
```

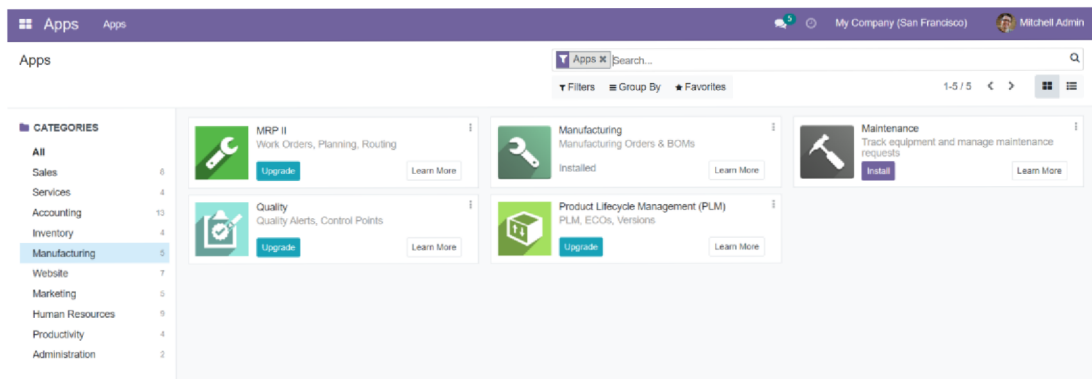
Ve složce projektu se nainstalují setup nástroje pro python. Předposledním úkonem pro instalaci Odoo 15 je instalace modulů nutných pro chod aplikace. Opět v příkazovém řádku ve složce projektu zadáme příkaz:

```
pip install -r requirements.txt
```

Následně už stačí pouze zadat příkaz, který vytvoří databázi a spustí Odoo:

```
python odoo-bin -d odoo15 -r UZIVATEL(odoo) -w HESLO
```

V prohlížeči zadáme adresu localhost:8069, na které se spustí přihlašovací obrazovka do aplikace. Úvodní přihlašovací jméno i heslo je nastaveno na „admin“. Na obrazovce se zobrazí několik možných podaplikací ke stažení do softwaru, pro účely této práce je vhodná položka „Manufacturing“. [6]



Obrázek 10: Moduly pro výrobu v Odoo MES

4.3 QCADOO

Qcadoo je MES řešení vhodné pro malé a střední výrobní podniky. Nabízí dvě verze, první verzí je Qcadoo Commercial version, což je plná verze programu, za kterou se platí společnosti Qcadoo Limited roční poplatky a zahrnuje plnou podporu, REST API, moduly pro integraci s ERP systémem či SCADA, nastavení Saas (Software as a service) a další přídavné funkce. Druhou verzí je Qcadoo Community version, jedná se o open-source verzi, systém funguje jako třívrstvá architektura. [14]

4.3.1 Popis architektury

Databáze

V databázi jsou uložena všechna data aplikace, používá se databázový systém PostgreSQL

Datová vrstva (data layer)

Datová vrstva využívá návrhový vzor DAO (Data Access Object), což je rozhraní, ve kterém se nacházejí definice metod pro manipulaci s databází. Objekty, které potřebují pracovat s databází si vytvoří instanci DAO a volají jeho metody, nevidí však, jak práce s databází funguje. Výhodou tohoto návrhového vzoru je, že při změnách v databázi se aplikační logika nemusí žádným způsobem měnit, provedou se pouze změny související s databází. [20]

Dále je využíván framework Hibernate. Jedná se o nástroj umožňující objektově relační mapování (ORM), což je technika pro převod dat na instance tříd objektově orientovaného jazyka a naopak. [15] Pro výměnu dat mezi datovou a servisní vrstvou je využívána Generic Entity.

Servisní vrstva (service layer)

Servisní vrstva je zodpovědná za veškerou aplikační logiku v Qcadoo. Zároveň pomáhá datové vrstvě s vyhledáváním, stránkováním, tříděním apod. Dále také zajišťuje vytváření a správu modulů a definic, které jsou následně vkládány do pluginů. V servisní vrstvě je využíván Spring, což je open-source framework pro vývoj enterprise Java aplikací. [16]

Prezentační vrstva (view layer)

Prezentační vrstva zobrazuje výstupy aplikace uživateli jako webovou stránku nebo soubory pdf či xls. K zobrazování výstupů používá framework Spring Web MVC, který umožňuje použití objektů jako příkazů nebo form-backing objektů. [16] Dále je využívám Tiles framework pro tvorbu webových šablon [17] a JavaScriptová knihovna jQuery. [18]

Pro přesun dat software využívá DTO (Data Transfer Objects) ve formátu JSON.

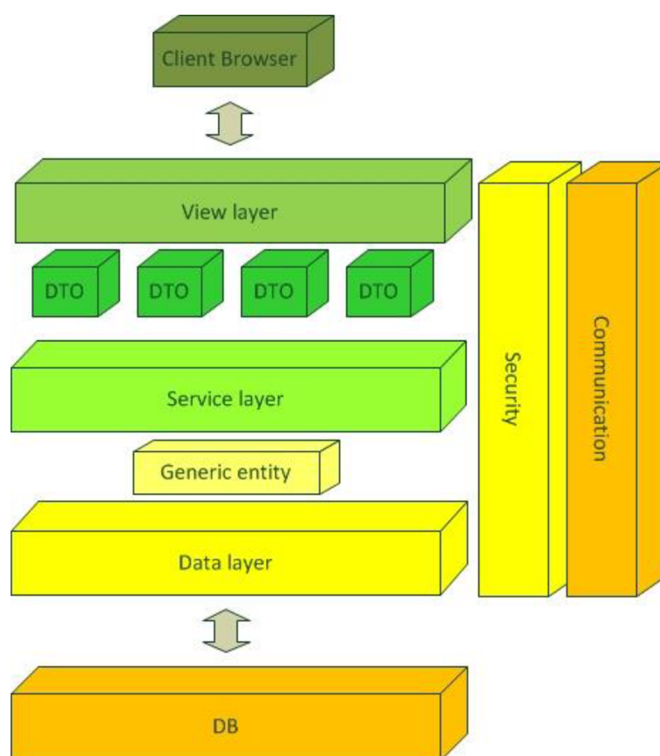
Do prezentační vrstvy se také řadí CRUD Controller, který skládá zobrazení uživateli z komponentů, jakými jsou rozvržení stránky, formuláře, vstupy apod.

Bezpečnost

O bezpečnost se stará autentizační a přístup řídicí framework Spring Security.

Prohlížeč uživatele

O většinu operací na straně prohlížeče se stará knihovna jQuery, prohlížeč uživatele musí podporovat JavaScript, CSS a AJAX.



Obrázek 11: Vrstvy architektury Qcadoo [14]

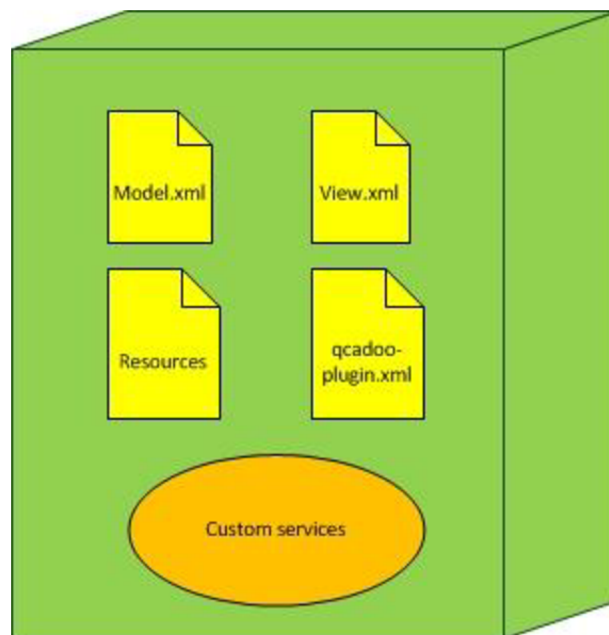
4.3.2 Popis pluginů

Pluginy v aplikaci Qcadoo jsou naprogramovány jako projekty nástroje Apache Maven.

Struktura projektu

- src/main/java/pluginpackage/
- src/main/resources/pluginidentifier/locales/pluginidentifier_en.properties
- src/main/resources/pluginidentifier/model/*.xml
- src/main/resources/pluginidentifier/view/*.xml
- src/main/resources/pluginidentifier/public/css/
- src/main/resources/pluginidentifier/public/img/
- src/main/resources/pluginidentifier/public/js/
- src/main/resources/WEB-INF/jsp/pluginidentifier/
- src/main/resources/root-context.xml
- src/main/resources/web-context.xml
- src/main/resources/qcadoo-plugin.xml
- pom.xml

Vnitřní struktura pluginu



Obrázek 12: Vnitřní struktura pluginu aplikace Qcadoo [14]

Přechody stavů pluginu

Jednotlivé operace musí splnit podmínky v souborech *dependencies*, správnou verzi frameworku a další. V případě, že jsou všechny podmínky splněny, tak se operace

úspěšně vykoná, v opačném případě se vrátíme na příslušný stav.

Download

Operace download stáhne příslušný soubor jar pro konkrétní plugin, uloží jej do složky tmp a vytvoří záznam v databázi se stavem *temporary* (dočasný).

Install

Tato operace přesune stažený jar soubor ze složky tmp do aplikace a změní stav záznamu v databázi z *temporary* (dočasný) na *enabled* (povolen). Po tomto kroku se server restartuje a tím se projeví změny.

Disable

Operace Disable změní stav záznamu v databázi na *disabled* (zakázán).

Enable

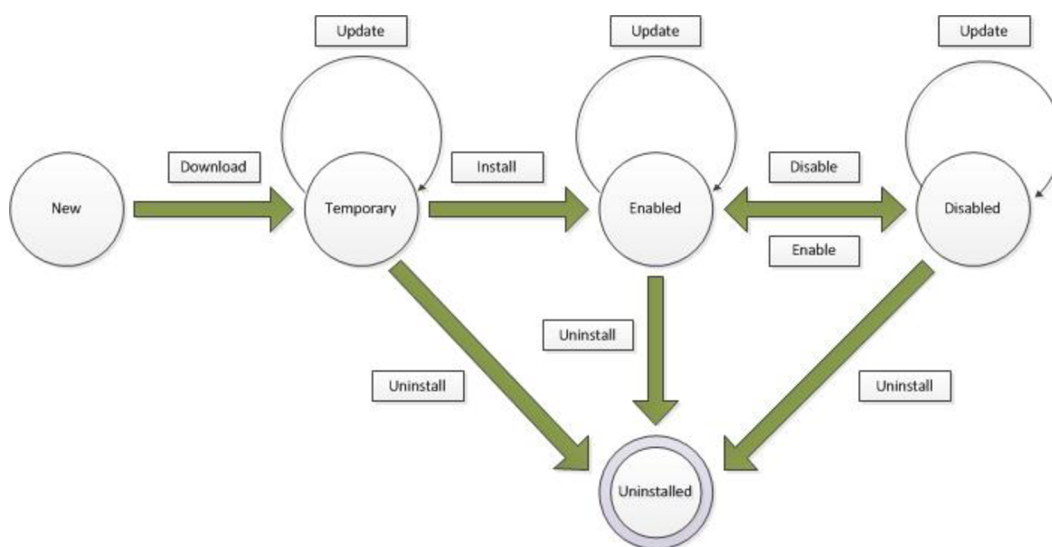
Operace Enable změní stav záznamu v databázi na *enabled* (povolen).

Uninstall

Tato operace odebere soubor jar z aplikace a smaže příslušný záznam v databázi.

Update

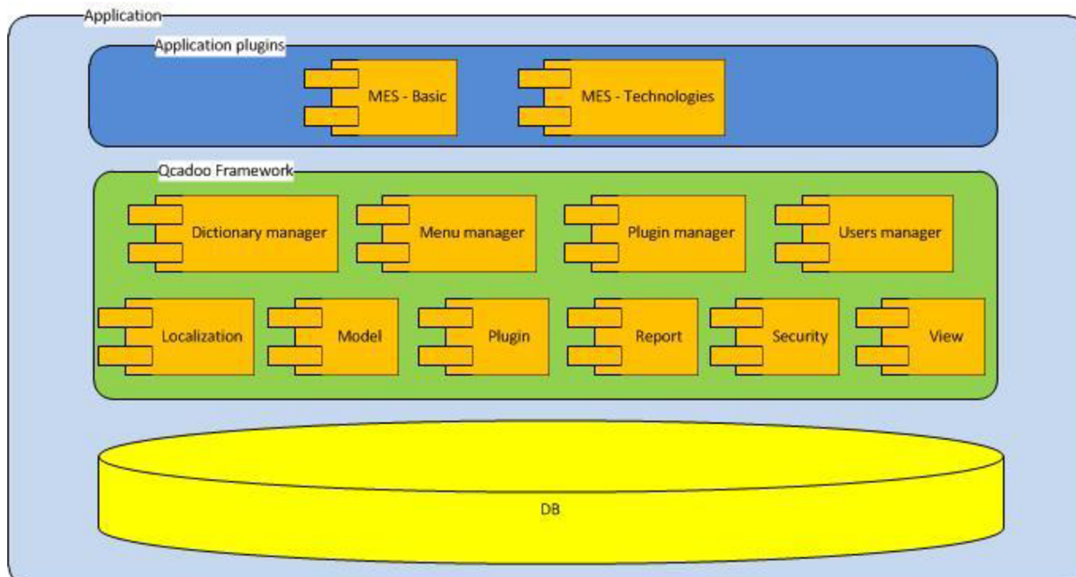
Operace update ověří verzi pluginu. V případě, že je verze pluginu nižší, než je zapsána verze pluginu v souboru jar, plugin může být aktualizován. Po aktualizaci je stará verze souboru jar odebrána z aplikace a nová verze souboru jar je přesunuta do aplikace. Zároveň je aktualizována informace o verzi pluginu v příslušném záznamu v databázi. Stav záznamu v databázi zůstává stejný jako před aktualizací. [14]



Obrázek 13: Přejechy stavů pluginu aplikace Qcadoo [14]

4.3.3 Popis Qcadoo frameworku a jeho jednotlivých pluginů

Qcadoo funguje na principu modulů, jednotlivé funkcionality mohou být přidány pomocí pluginů. Tato architektura funguje díky frameworku Spring, na kterém je Qcadoo systém postaven a který propojuje komponenty tohoto softwaru. Spring také umožňuje zabezpečení komponentů bez duplikace funkcí.



Obrázek 14: Architektura pluginů Qcadoo [14]

Framework

Qcadoo Framework je základní částí systému, poskytuje mj. bezpečnostní mechanismy, transakční zpracování, přístup k datům, správu pluginů a komunikaci mezi nimi.

Pro transakční zpracování a lokalizaci (Qcadoo Localization plugin) využívá Spring Framework.

Qcadoo model je plugin spadající také pod Qcadoo Framework, pracuje s entitami Hibernate a zároveň je společným rozhraním systému pro přístup do databáze.

Qcadoo view je plugin zodpovědný za vytváření a správu obsahu, který je následně vkládán do jednotlivých pluginů.

Qcadoo plugin spravuje stavy jednotlivých pluginů, jejich „tok“ a všech operací s nimi spojenými.

Qcadoo report je plugin pro vytváření a správu reportů dat.

Qcadoo security je zodpovědný za bezpečnostní mechanismy softwaru.

Qcadoo mail je plugin pro odesílání emailů.

Plugin management

Jedná se o plugin pro správu pluginů, umožňuje přidávat, aktualizovat, povolovat, zakazovat a odinstalovat pluginy. Nejedná se o systémový plugin, je tedy možné jej odinstalovat nebo nepovolit.

Menu management

Menu management plugin se stará o správu menu, umožňuje vytvářet, upravovat a mazat jeho položky či měnit jejich pořadí. Opět se nejedná o systémový plugin, je tedy možné jej odinstalovat nebo nepovolit.

User management

Tento plugin je zodpovědný za správu uživatelů, s jeho pomocí je možno vytvářet, upravovat a mazat uživatele. Dále je schopen spravovat role uživatelů. Nejedná se o systémový plugin, je tedy možné jej odinstalovat nebo nepovolit, avšak i v případě odinstalování zůstávají uživatelé a jejich role v systému, využívá jich spring security vrstva, která poskytuje autentizaci a autorizační mechanismy.

Dictionary management

Pomocí tohoto pluginu můžeme vytvářet, upravovat a mazat položky dictionary. Dictionary management není systémový plugin, je jej tudíž možné odinstalovat nebo nepovolit, v případě odinstalování však odstraní pouze funkcionality spojené s tímto pluginem, data zůstanou v systému jako v případě User management pluginu. [14]

4.3.4 Funkcionality Qcadoo

Správa výrobních zakázek

V aplikaci se dají spravovat výrobní zakázky, mít přehled nad náklady s nimi spojenými, pracovat s informacemi o časové náročnosti jejich vyhotovení. Program nabízí plánování výroby, správu zakázek, plánování výroby na jednotlivé směny a definování pracovních instrukcí.

Správa materiálu

Qcadoo je při výrobě a plánování výroby nápomocný mj. v oblasti materiálu. Pro příslušný výrobní proces stanoví typy a kvantitu jednotlivých materiálů, které jsou potřeba. Zároveň umožňuje náhled do informací o materiálu ze skladu a jeho plánovaných zásobách. Mezi funkce spojené se správou materiálu patří plánování materiálových požadavků, sledování zásob ve skladu, generování poptávky materiálu, generování

objednávek a harmonogram plánovaných dodávek materiálu.

Komunikace se subdodavateli

Qcadoo usnadňuje komunikaci se subdodavateli, nabízí správu subdodavatelů, generování požadovaných dokumentů, správu technické dokumentace, portál pro subdodavatele, spravování nákladů v této oblasti a monitorování probíhající práce subdodavatelů.

Rodokmen výrobku

V aplikaci může uživatel registrovat a sledovat vývoj výrobku či součástky, zároveň může zobrazit jejich rodokmen (Genealogy tree).

Plánování výroby

Qcadoo pomáhá naplánovat výrobu s ohledem na časovou náročnost, kapacitu a dostupnost zdrojů, technologie, výkony strojů apod. Nabízí funkce jako grafické plánování, kalendář směn, plánování se signalizací průběhu, změnu zdrojů během procesu, aj.

Správa produkce

Příslušná osoba může sledovat produkci v reálném čase, reagovat na události, evidovat probíhající výrobu apod. Systém umožní registrovat příčiny změn v produkci a zaznamenávat ji různými způsoby (např. grafický terminál či využití čárových kódů).

Správa skladu

Systém poskytuje přehled nad skladovými zásobami, upozorňuje, které zásoby je třeba doplnit, generuje různé typy dokumentací, pomáhá automatizovat proces využití materiálu ze skladu a umožňuje integraci externího finančního či skladového informačního systému. [14]

4.3.5 Instalace QCADOO Community pro Windows

Instalace Git

1. Stažení a instalace

Ze stránky <https://git-scm.com/downloads> stáhnout a nainstalovat Git pro Windows,

2. Nastavení uživatelského jména

Otevřít program Git Bash a nastavit uživatelské jméno příkazem:

```
git config --global user.name "uzivatelske_jmeno"
```

Kontrola:

```
git config --global user.name
```

3. Nastavit email adresu

```
git config --global user.email "email" [35]
```

4. Vygenerovat SSH klíč

```
eval (ssh-agent -s)
```

5. Vložení SSH klíče na Github

Zkopírovat, např.:

```
ssh-add /c/Users/Tomáš/.ssh/id_ed12345  
clip < /c/Users/Tomáš/.ssh/id_ed12345.pub
```

Po přihlášení na github.com jít to záložky Settings -> SSH and GPG keys -> New SSH key. Na této stránce pojmenovat klíč a vložit jej do části „Key“.

SSH keys / Add new

Title

Home Desktop

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'




Add SSH key

Obrázek 15: Přidání SSH klíče do GitHubu

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

	Home Desktop [Redacted]	
	Added on 4 Nov 2021 Never used — Read/write	

Check out our guide to [generating SSH keys](#) or [troubleshoot common SSH problems](#).

Obrázek 16: Seznam SSH klíčů v GitHub účtu

Instalace Apache Maven

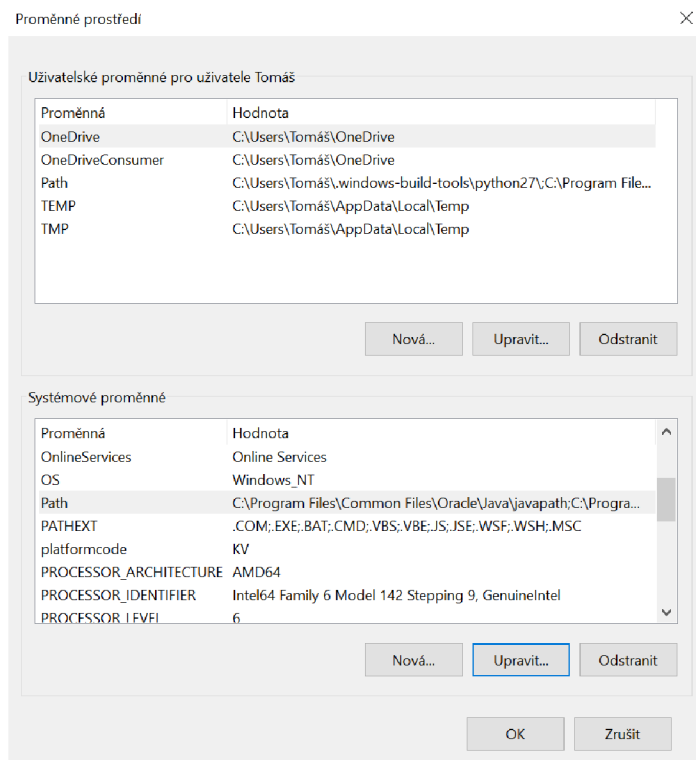
Ze stránky <https://maven.apache.org/download.cgi> stáhnout např. Binary zip archive soubor.

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.8.3-bin.tar.gz	apache-maven-3.8.3-bin.tar.gz.sha512	apache-maven-3.8.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.8.3-bin.zip	apache-maven-3.8.3-bin.zip.sha512	apache-maven-3.8.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.8.3-src.tar.gz	apache-maven-3.8.3-src.tar.gz.sha512	apache-maven-3.8.3-src.tar.gz.asc
Source zip archive	apache-maven-3.8.3-src.zip	apache-maven-3.8.3-src.zip.sha512	apache-maven-3.8.3-src.zip.asc

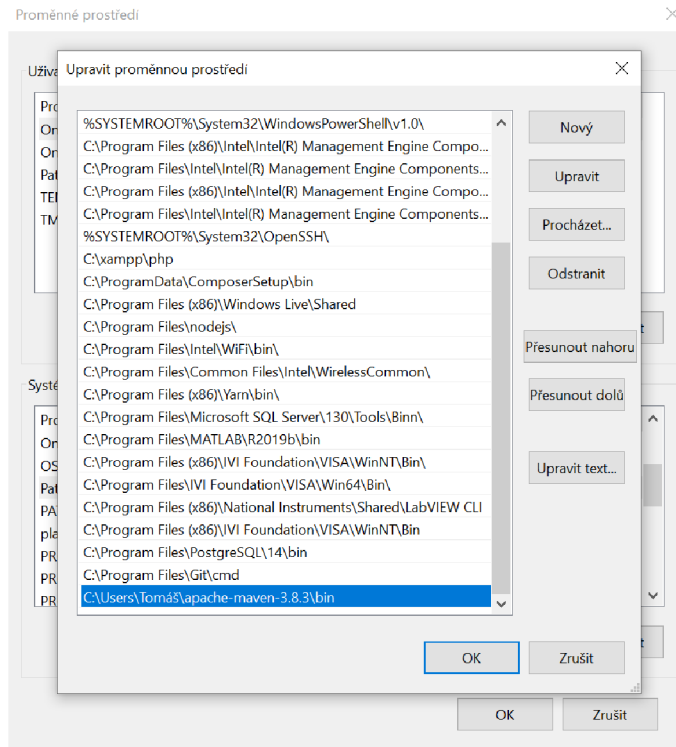
Obrázek 17: Apache Maven 3.8.3 ke stažení

Extrahujeme soubory na lokální disk a přidáme složku bin do proměnných prostředí PATH.

Otevřít Ovládací panely - Systém a zabezpečení - Systém - Upřesnit nastavení systému - Proměnné prostředí.



Obrázek 18: Proměnné prostředí Windows 10



Obrázek 19: Přidání Apache Maven jako proměnnou prostředí Windows 10

Kontrola verze Maven

```
mvn -v
```

[19]

```

Příkazový řádek
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\Tomáš>mvn -v
Apache Maven 3.8.3 (ff8e977a158738155dc465c6a97ffaf31982d739)
Maven home: C:\apache-maven-3.8.3
Java version: 1.8.0-262, vendor: OpenLogic-OpenJDK, runtime: C:\openlogic-openjdk-8u262-b10-win-64\jre
Default locale: cs_CZ, platform encoding: Cp1250
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

```

Obrázek 20: Kontrola verze Apache Maven v příkazovém řádku

4.3.6 Spuštění Qcadoo

Stažení souborů

Pro spuštění Qcadoo je nutné nejprve stáhnout následující čtyři složky se zdrojovými soubory:

1. *qcadoo-super-pom-open* z adresy github.com/qcadoo/qcadoo-super-pom-open,
2. *qcadoo-maven-plugin* z adresy github.com/qcadoo/qcadoo-maven-plugin,
3. *qcadoo* z adresy github.com/qcadoo/qcadoo a
4. *mes* z adresy github.com/qcadoo/mes.

Stažené složky je potřeba uložit do jednoho stejného adresáře, v jehož cestě jsou obsaženy pouze znaky z anglické abecedy.

Instalace jednotlivých částí projektu

Každá ze složek z bodu 1 se instaluje dle návodu pomocí příkazu

```
mvn clean install nazev_slozky -DskipTests=true
```

v příkazovém řádku v dané složce. Je nutné dodržet následující pořadí:

1. *qcadoo-super-pom-open*,
2. *qcadoo-maven-plugin*,
3. *qcadoo*,
4. *mes*.

Při instalaci části *qcadoo* dochází k chybě, která je způsobena chybějící knihovnou *jgraph*. Z adresy <https://mvnrepository.com/artifact/org.jgrapht/jgrapht-core/1.5.1> je potřeba stáhnout soubor *jgrapht-core-1.5.1.jar*, který je třeba uložit do složky *qcadoo/qcadoo-plugin*. Následně je potřeba upravit soubor *pom.xml* v dané složce následujícím způsobem:

1. přidat do části *dependencies* kód:

```
<dependency>  
  <groupId>org.jgrapht</groupId>  
  <artifactId>grapht-core</artifactId>  
  <version>1.5.1</version>  
  <scope>system</scope>  
  <systemPath>{pom.basedir}/jgrapht.jar</systemPath>  
</dependency>  
  
<dependency>  
  <groupId>net.serenity-bdd</groupId>
```

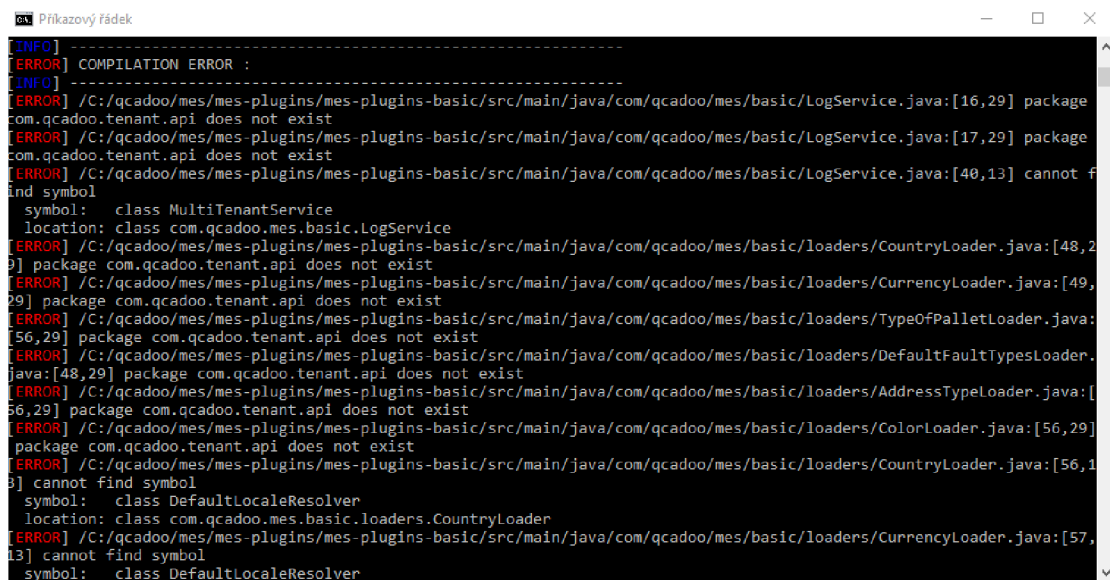
```
<artifactId>serenity-rest-assured</artifactId>
<version>1.1.21</version>
</dependency>
```

2. odebrat z části *dependencies* kód:

```
<dependency>
  <groupId>jgrapht</groupId>
  <artifactId>jgrapht</artifactId>
  <version>0.8.2</version>
</dependency>
```

Pro zjednodušení je možné přejmenovat zmíněný soubor `jgrapht-core-1.5.1.jar` na `jgrapht.jar`, tomuto názvu také odpovídá kód v bodu 1.

Při instalaci poslední části *mes* dochází k chybové hlášce z důvodu chybějících balíčků, bez kterých software není funkční, nejde doinstalovat, a tudíž ani spustit.



```
Příkazový řádek
[INFO] -----
[ERROR] COMPILATION ERROR :
[INFO] -----
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/LogService.java:[16,29] package
com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/LogService.java:[17,29] package
com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/LogService.java:[40,13] cannot f
ind symbol
symbol:   class MultiTenantService
location: class com.qcadoo.mes.basic.LogService
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/CountryLoader.java:[48,2
9] package com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/CurrencyLoader.java:[49,
29] package com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/TypeOfPalletLoader.java:
[56,29] package com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/DefaultFaultTypesLoader.
java:[48,29] package com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/AddressTypeLoader.java:[
56,29] package com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/ColorLoader.java:[56,29]
package com.qcadoo.tenant.api does not exist
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/CountryLoader.java:[56,1
3] cannot find symbol
symbol:   class DefaultLocaleResolver
location: class com.qcadoo.mes.basic.loaders.CountryLoader
[ERROR] /C:/qcadoo/mes/mes-plugins/mes-plugins-basic/src/main/java/com/qcadoo/mes/basic/loaders/CurrencyLoader.java:[57,
13] cannot find symbol
symbol:   class DefaultLocaleResolver
```

Obrázek 21: Chybová hláška při instalaci Qcadoo

4.4 Dolibarr

Dolibarr je open-source ERP a CRM systém, je sestaven z modulů, každý z modulů se stará o jednu konkrétní oblast řízení podniku. Systém obsahuje také moduly v sekci Manufacturing, které poskytují funkcionality pro řízení výrobních procesů. Aplikace využívá architektury klient-server.

Zdrojový kód je napsaný převážně v PHP, pro spuštění programu je potřeba minimálně verze PHP 5.6.0. Jako databázi využívá MariaDB, MySQL či PostgreSQL. [20]

4.4.1 Funkcionality v oblasti MES systémů

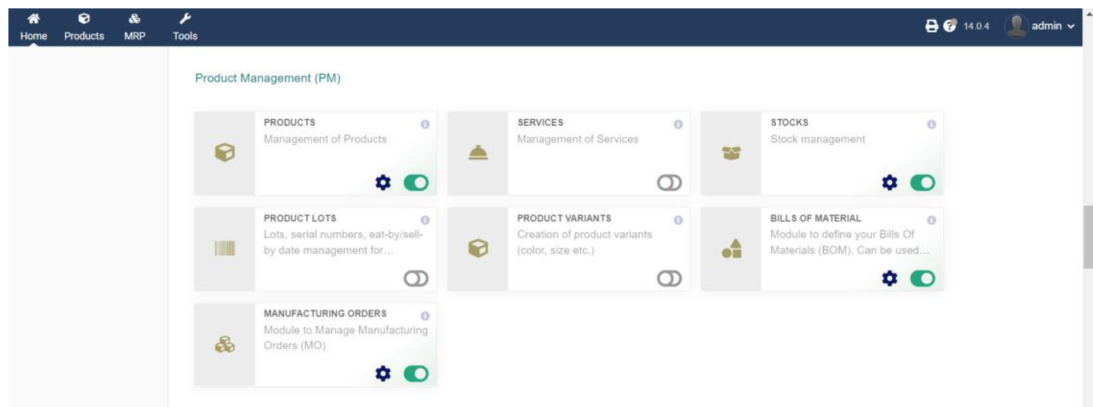
Dolibarr mj. disponuje dvěma moduly, které souvisí s výrobou.

Bills of material

V modulu Bills of material si uživatel může vytvořit všechny výchozí materiály, ze kterých se produkt vyrábí, zadat informace popisující tyto materiály, jejich kvantitu a případně kde jsou naskladněny.

Manufacturing Orders

V modulu Manufacturing Orders může uživatel zadávat systému výrobní zakázky a jejich parametry.



Obrázek 22: Moduly Dolibarr systému pro výrobu

4.4.2 Instalace na Windows 10

Ze stránky <https://sourceforge.net/projects/dolibarr/files/> je potřeba stáhnout soubor DoliWamp.exe. Jedná se o instalační balíček, který vše potřebné (MariaDB, PHP, Apache 2 a následně Dolibarr) nainstaluje a nakonfiguruje. Je možné také nainstalovat manuálně.

4.5 ERPNext

ERPNext je open-source podnikový informační systém (ERP). Je postaven na frameworku Frappe, jenž je určený pro tvorbu full-stack webových aplikací, a jehož zdrojové kódy jsou napsané v jazycích Python a JavaScript. Frappe framework využívá databázový systém MariaDB. [21]

4.5.1 Funkcionality v oblasti výroby

Kusovníky

ERPNext umožňuje vytváření kusovníků a specifikování jejich parametrů.

Subdodavatelé

Díky funkce Subcontracting může uživatel sledovat operace subdodavatelů, dodávkám bude spočítána hodnota na základě ceny zboží a služeb.

Serializovaný inventář

V systému je možné vytvářet sériová označení pro identifikaci položek, možností je také skenování čárových kódů.

Plánování zdrojů

Plánování spotřeby materiálu a zadávání požadavků o dodávky potřebných zdrojů, plánování produkce na základě kapacity a dostupnosti zdrojů.

Správa skladů

Aplikace sleduje množství materiálu ve skladech a porovnává je s kvantitou zdrojů v plánované produkci, vystavuje požadavky na objednávku materiálu a poskytuje reporty o zásobách ve skladech apod. [21]

4.5.2 Instalace na Windows 10

ERPNext a framework Frappe, na kterém je systém postaven, nelze přímo na operačním systému Windows spustit. Autoři nabízí možnost využít např. Oracle VirtualBox pro spuštění instalačního souboru .ova ze stránky <https://erpnext.org/download>, avšak jeho import do VirtualBoxu je nefunkční. [21]

4.6 IMES

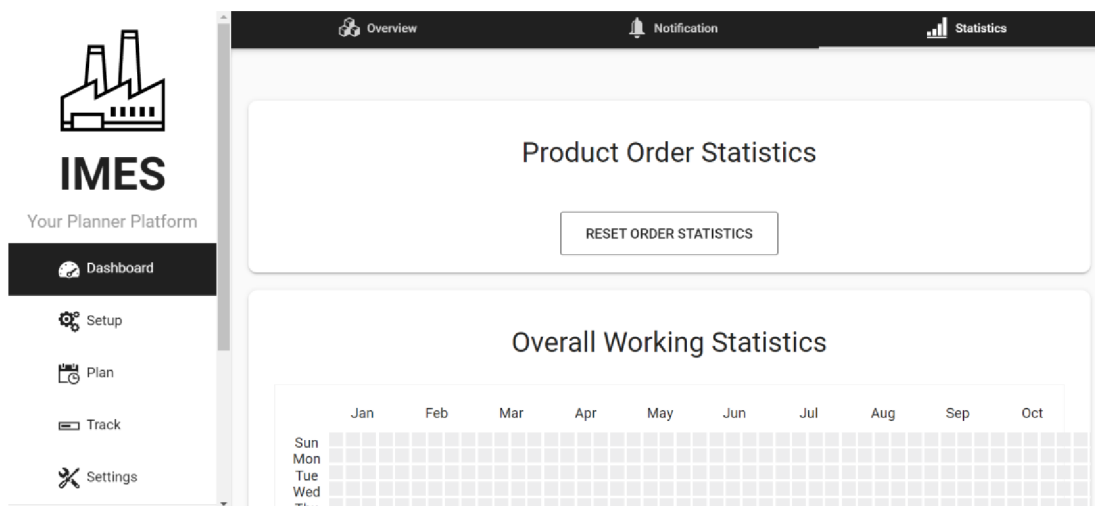
IMES je open-source výrobní informační systém vhodný pro malé a středně velké výrobní podniky. Systém funguje jako webová aplikace, jejíž cílem je usnadnit výrobním

podnikům přechod z průmyslu 2.0 na modernější způsoby výroby, pro které jsou v tuto chvíli příliš drahou záležitostí. Pro jejich stávající řešení existuje pouze omezené množství dostupných nástrojů pro zefektivnění procesů, které jsou např. pro malé podniky příliš nákladné.

4.6.1 Popis struktury softwaru

Aplikace je postavena na JavaScriptové knihovně Polymer, kterou vyvíjí společnost Google. Tato knihovna umožňuje vytváření webů s pomocí html komponentů.

Na daný software se vztahuje licence Apache 2.0, která umožňuje prakticky téměř jakékoliv nakládání se zdrojovým kódem.



Obrázek 23: IMES

4.6.2 Funkcionality

Aplikace vyžaduje internetové připojení a je plně responzivní i pro mobilní prohlížeče. IMES je stále ve stavu vývoje, dle tvůrců je či bude schopen využívat následující procesy a algoritmy:

Lineární programování (LP)

Využívá se k nalezení maxima či minima lineární funkce, pomáhá zvolit nejvíce ideální možné řešení např. z pohledu času výroby.

Celková efektivnost zařízení (OEE)

Zaznamenává výkon výroby, měří zejména čas, během kterého stroje vykonávají

pracovní cykly i s prodlevami, počet vyrobených kusů za časovou jednotku a počet vyprodukovaných vadných dílů.

Původní kvantita (Q0)

Původní kvantita odpovídá kvantitě v objednávce, který obsahuje záložní položky. V případě poškození či chyby je možné využít náhradních položek, které mohou nahradit původní položky a zefektivnit tímto úkonem daný proces.

Algoritmy plánování

Plánování procesů a výroby s ohledem na čas a materiál tak, aby došlo k co nejlepším výsledkům. Program má využívat:

- Earliest Due Date (EDD) – nejprve třídí a dosadí do procesu nejstarší položky.
- Shortest Path Tree (SPT) – položky s nejkratším procesem budou mít prioritu.

[22]

4.7 EspoCRM

EspoCRM je open-source CRM systém vydaný v roce 2014 ukrajinskou společností Letrium LTD. Základní verze systému je volně dostupná, k dispozici jsou i různá oficiální rozšíření (např. Sales Pack či Advanced Pack), která jsou však zpoplatněna. EspoCRM má na poměry open-source systémů pro správu podniků poměrně kvalitní vývojářskou dokumentaci a velmi rozsáhlou komunitu, tudíž je možné najít spoustu diskuzí na fóru na stránkách vydavatele softwaru.

Systém vyžaduje verzi PHP 7.4 a vyšší v tomto jazyce je napsán backend softwaru. Dále je potřeba databáze, EspoCRM využívá MySQL (verzi 5.7 a vyšší) nebo MariaDB (10.1 a vyšší). Frontend systému je psán v jazyce JavaScript. [23]

4.7.1 Instalace

Instalace systému je triviální. Z webových stránek vydavatele uživatel stáhne instalační soubor, který následně nahraje na vlastní server. Dále je potřeba vytvořit databázi (MySQL), se kterou systém bude komunikovat. Poté stačí otevřít adresu, která odpovídá místě na serveru, kam byl nahrán instalační soubor. Na této adrese se spustí instalační okno, do kterého uživatel zadá potřebné údaje (zejména přihlašovací a databázové), čímž se instalace dokončí a systém je připraven.

4.7.2 Architektura

V tomto open-source systému je využíváno MVC softwarové architektury. Základní stavební jednotkou systému EspoCRM je tzv. Entita. Každá entita sdružuje záznamy stejného charakteru, např. vydané faktury, objednávky, společnosti, zaměstnanci apod. Každé entitě odpovídá tabulka v databázi. Systém disponuje nástrojem pro zjednodušení správy entit, pomocí kterého může vytvářet a upravovat vlastní entity i běžný uživatel, lze jej totiž využít jak v kódu, tak v administraci systému. Jedná se o nástroj Entity Manager, který na základě požadavků uživatele edituje příslušné soubory s koncovkou .json, ve kterých jsou uloženy informace o jednotlivých entitách (jejich vazby na jiné entity, rozložení a pole), a příslušnou tabulku v databázi.

Entita může mít několik typů, dle kterých se vygenerují výchozí políčka (typy Base, Base Plus, Event, Person a Company). V administraci lze u entity upravovat následující parametry:

Popisky

Lze nastavit popisky entity, které se budou uživateli zobrazovat místo jména entity, Je vhodné nazývat entitu bez diakritiky, mezer, speciálních znaků apod., zatímco pro účel popisku můžeme psát jakkoliv uživatelsky přívětivý text. Popisky lze nastavit pro jednotné a množné číslo, např. „Vydaná faktura“ a „Vydané faktury“. Systém sám zobrazí příslušný popisek na vhodném místě.

Ikona a její barva

Pro doplnění názvu entity v bočním menu systému, do kterého uživatel přidává entity pro rychlý přístup, parametry se nacházejí v metadatech ve složce clientDefs.

Disabled

Po zaškrtnutí nastaví do příslušného .json souboru ve scopes parametr disabled na „true“ a tím se entita pro uživatele skryje (nelze ji v systému otevřít)

Text Filter Fields

Zde uživatel vybírá pole entity, ve kterých poté systém vyhledává při vyhledávání v entitě či systému, dochází k editaci ve složce entityDefs v metadatech řazení (směr a pole) – výběr pole a směru, dle kterého se mají pro uživatele v entitě zobrazit záznamy, dochází k editaci ve složce entityDefs v metadatech

Status Field

Zde může uživatel vybrat pole, které bude v entitě označeno jako stavové pole. Lze vybrat ze všech polí typu enum v entitě, po vybrání se zobrazí možnost zobrazení kanban, který se bude řídit na základě vybraného pole.

Kanban view

Při zaškrtnutí pole přibude v entitě možnost zobrazení záznamů jako tzv. kanban board. Pro zobrazení této možnosti je třeba zvolit stavové pole Status Field. Tato možnost edituje příslušný soubor s koncovkou .json ve složce metadata/clientDefs

Disable record count

V entitě nebude zobrazen počet záznamů, dle dokumentace může tato možnost zrychlit načítání dat z velké databázové tabulky. Tato možnost edituje příslušný soubor s koncovkou .json ve složce metadata/entityDefs

Optimistic Concurrency Control

Pokud během úpravy záznamu dojde k jeho změně jiným uživatelem, systém uživateli nabídne možnosti vyřešení konfliktu (která data mají být v záznamu uložena) [<https://docs.espocrm.com/user-guide/optimistic-concurrency-control/>] Tato možnost edituje příslušný soubor s koncovkou .json ve složce metadata/entityDefs

Full-Text Search

Umožňuje rychlejší a více komplexní textové vyhledávání. [49] [<https://docs.espocrm.com/user-guide/text-search/>] Tato možnost edituje příslušný soubor s koncovkou .json ve složce metadata/entityDefs

Ignored groups in Kanban view K

Které stavy ze stavového pole mají být ignorovány v zobrazení kanban. Tato možnost se zobrazí pouze při zaškrtnutí pole Kanban view. Tato možnost edituje příslušný soubor s koncovkou .json ve složce metadata/scopes.

The screenshot shows the 'Upravit' (Edit) page for the 'Kontakt' entity in the EspoCRM administration interface. The page is titled 'Administrace > Správa entit > Kontakt > Upravit'. At the top, there are three buttons: 'Uložit' (Save), 'Zrušit' (Cancel), and 'Reset to Default'. The main content area is divided into two columns of settings:

- Jméno:** Contact
- Typ:** -
- Popisek jednotného čísla *:** Kontakt
- Popisek množného čísla *:** Kontakty
- Icon:** [Icon]
- Color:** #ffffff
- Disabled:**
- Stream:**
- Výchozí řazení (pole):** Vytvořeno
- Výchozí řazení (směr):** Sestupně
- Text Filter Fields:** Název, Email
- Status Field:** ---
- Full-Text Search:**
- Disable record count:**
- Optimistic concurrency control:**

Obrázek 24: Příklad úpravy entity ve správě entit v EspoCRM

Příklad části příslušného souboru entityDefs:

```
"collection": {
  "orderBy": "createdAt",
  "order": "desc",
  "textFilterFields": [
    "name"
  ],
  "fullTextSearch": false,
  "countDisabled": false
},
...
```

Entitu tvoří různá pole (např. název, popis, adresa apod.). Každé pole má svůj typ (např. varchar, number, url apod.), dle kterého se ukládá do databáze a zobrazuje

uživateli. Pole lze vytvářet a editovat v administraci ve Správci entit (Entity manager) či v příslušných json souborech.

Dále lze ve správci entit vytvářet a editovat vazby mezi entitami, dle kterých se následně upraví tabulka v databázi (případně vytvoří vazební tabulka). Pro každou vazbu se v entitě vytvoří pole.

Pro uživatelské zobrazení záznamů a jejich polí v entitách se v administraci edituje sekce Layouts, ve které můžeme nastavit která pole uživatel uvidí ve kterém zobrazení, jejich pozici např. v detailu. Můžeme také přidávat spodní a boční panely.

4.7.3 Backend

Backendová část aplikace je psána v jazyce php, je využíváno také json souborů. Aplikace je rozdělena na moduly, pro programátora je důležitý modul (složka) Custom, která obsahuje kód vytvořený uživatelem. Složka Custom je „odolná“ vůči aktualizaci systému, její data jsou zachována, zatímco kdyby došlo ke změně zdrojového kódu v jiných složkách, mohlo by při aktualizaci dojít k jeho ztrátě.

Metadata

Metadata v systému EspoCRM slouží k uchování dat o prvcích systému, např. o entitách. Metadata se dělí na podsložky, ve kterých jsou uloženy odpovídající soubory k entitám a dalším prvkům systému, které definují např. pole, vazby mezi entitami, rozvržení stránky s entitou apod.

Podsložky metadat:

- scopes
- entityDefs
- aclDefs
- selectDefs
- recordDefs
- clientDefs
- entityAcl
- pdfDefs
- notificationDefs
- streamDefs
- fields
- dashlets
- authenticationMethods
- integrations

[23]

Složka scopes obsahuje soubory json, které obsahují obecné informace o entitě (např. zda se vůbec o entitu jedná, zda eviduje její změny, jakého je případně

entita typu, zda má kanban, layouts apod) a obecné definice, které patří v backendu do části scope.

Příklad příslušného souboru scopes:

```
{
  "entity": true,
  "layouts": true,
  "tab": true,
  "acl": true,
  "aclPortal": true,
  "aclPortalLevelList": [
    "all",
    "account",
    "contact",
    "own",
    "no"
  ],
  "customizable": true,
  "importable": true,
  "notifications": true,
  "stream": false,
  "disabled": false,
  "type": "Base",
  "module": "Custom",
  "object": true,
  "isCustom": true,
  "statusField": "test",
  "kanbanStatusIgnoreList": [
    "1"
  ]
}
```

Ve složce entityDefs v metadatech jsou uchovávána zejména data o polích a vazbách entity. Můžeme tam také definovat databázové indexy, které mohou výrazně zlepšit vyhledávání dat v databázi.

Složka clientDefs slouží k uchování parametrů důležitých pro front end systému, tzn. jakým způsobem se entity budou zobrazovat uživateli v prohlížeči. Do jednotlivých souborů této složky můžeme přidávat vlastní tlačítka pro entitu, panely, editovat ikonku, její barvu, spravovat zobrazení kanban nástěnky atd. [23]

Hooky

Hooky nám umožňují zasahovat do systému a vázat logiku např. na změnu dat v entitě bez toho, aniž bychom museli editovat původní zdrojový kód systému. Typickým příkladem je změna hodnoty jednoho pole (např. v jiné entitě) na základě hodnoty pole druhého. Jednotlivé hooky se nachází ve složce Hooks v podsložkách, které mají každá název odpovídající názvu entity, ve které se hoko vykonává. Lze také vytvořit hooky pro všechny entity, tzv. Global hooks, které se definují ve složce Hooks v podsložce Common.

Nejčastější metody v hooku:

- beforeSave – hook se vykoná před uložením dat entity do databáze
- afterSave – hook se vykoná po uložením dat entity do databáze
- beforeRemove
- afterRemove
- afterRelate
- afterUnrelate
- afterMassRelate

[23]

Příklad hooku:

```
<?php
namespace Espo\Custom\Hooks\EquipmentDefinitions;
use Espo\ORM\Entity;
use Espo\Core\ORM\EntityManager;
class UpdateName
{
    public function __construct(
        protected EntityManager $entityManager,
    ) {}

    public function afterSave(Entity $entity, array $options = []): void
    {
        if ($entity->isNew() && !$entity->get('name')) {
            $entity->set('name', 'Název');
        }
    }
}
```

Scheduled jobs

Scheduled jobs jsou funkce, které se vykonávají v pozadí periodicky v určitém intervalu. Jedná se typicky o stahování emailů či kontrolu měsíce či roku, např. pro evidování těchto údajů v číslech faktur. Vytvářejí se ve složce Jobs jako soubory s koncovkou php a v souboru Resources/metadata/app/scheduledJobs.json se následně zapisují.

Příklad vytvoření vlastní naplánované činnosti (scheduler job):

```
{
  "MyJob": {
    "jobClassName": "Espo\\Custom\\Jobs\\MyJob"
  }
}
```

```
<?php
namespace Espo\Custom\Jobs;
use Espo\Core\Job\JobDataLess;
class YourJobName implements JobDataLess
{
    // Pass dependencies through the constructor using DI.
    public function __construct()
    {
    }

    public function run(): void
    {
        // Write your logic here.
    }
}
```

Následně se v Administraci v systému v sekci „Naplánované činnosti“ (Scheduled jobs) naplánuje opakování pomocí syntaxu tzv. cronu, dle kterého bude systém definovanou činností periodicky vykonávat. [23]

4.7.4 Shrnutí

EspoCRM přináší celou řadu výhod, jak pro programátora, tak pro uživatele. Mezi značné výhody patří jednoduchá úprava mnoha částí systému v administraci (zejména vytváření entit), kvalitní dokumentace na webových stránkách vydavatele, aktivní diskusní fórum a jednoduchá instalace v rámci několika minut. Systém programátorovi umožňuje, díky složce Custom, vytvářet vlastní funkce, skripty apod. bez toho, aniž by musel zasahovat do jádra systému.

4.8 Srovnání systémů

Jedním z popsaných řešení je verze 15.0 systému Odoo, který je souborem různých aplikací pro řízení podniku včetně aplikace pro výrobu. Systém je spustitelný v prohlížeči, byl úspěšně nainstalován, jeho výhodou je poměrně kvalitní dokumentace pro uživatele i vývojáře a diskusní fórum. Část jeho využití je však v open-source verzi omezená, např. část API, které autoři aplikace neposkytují zdarma.

Dalším popsaným řešením je Qcadoo MES systém, jenž mj. poskytuje také open-source verzi. Je k dispozici dokumentace, avšak obsahuje množství chyb. Disponuje kvalitním popisem API, ale open-source verze programu nebylo možné dle návodu v dokumentaci nainstalovat a spustit.

Třetím popsaným systémem je ERP systém Dolibarr, který stejně jako Odoo obsahuje modul pro řízení výroby. Disponuje dokumentací a oficiálním diskuzním fórem. Systém se podařilo spustit, jeho nevýhodou je však malý počet funkcí pro výrobní procesy.

Dalším popsaným řešením je ERP systém ERPNext obsahující modul Manufacturing pro řízení výroby. Systém není určen primárně pro operační systém Windows, existuje však možnost, jak jej nainstalovat pomocí programu VirtualBox. Instalační balíček však nebylo možné spustit, a tudíž systém nainstalovat. Pro uživatele je k dispozici dokumentace a oficiální fórum, které je v rámci celého systému značně využíváno.

Pátým popsaným systémem je IMES, což je webová aplikace, která je sice v prohlížeči spustitelná, ale projekt je stále ve vývoji. Z toho důvodu není systém velmi robustní, obsahuje minimum funkcí a dokumentace pro vývojáře prakticky neexistuje, pouze technický popis systému.

Poslední popsaným řešením je EspoCRM, které sice není zaměřeno na výrobu (čistě CRM), avšak díky správci entit lze jednoduše vytvořit a spravovat jakoukoliv entitu, což je značná výhoda. Dále disponuje dokumentací pro vývojáře, aktivním diskuzním forem a lze nainstalovat během několika minut. Systém je modulární a umožňuje vývojáři jednoduše zasahovat do funkcí systému.

Tabulka 1: Srovnání popsaných systémů

Prvek / Název systému	Odoo	Qcadoo	Dolibarr	ERPNext	IMES	EspoCRM
Jednoduchá instalace	Ano	Ne	Ano	Ne ³	Ano	Ano
Diskuzní forum	Ano	Ne	Ano	Ano	Ne	Ano
Vývojářská dokumentace	Ano	Ano	Ano	Ano	Ne ⁴	Ano
Správce entit	Ne	Ne	Ne ²	Ne	Ne	Ano
Modul pro výrobu	Ano	Ano	Ne ¹	Ano	Ano	Ne
Modul pro KPI	Ano	Ano	Ne (pouze placené)	Ne	Ano	Ne (pouze placené)
Skladový modul	Ano	Ano	Ne (pouze placené)	Ano	Ne	Ne (pouze placené)
Globální vyhledávání	Ne	Ne	Ano	Ano	Ne	Ano

¹ Disponuje modulem BOM, ale pro další moduly pro výrobu chybí.

² Na internetu je možné najít modul, který lze doinstalovat a umožní vytvářet entity.

³ Pro Windows se mi nepodařilo nainstalovat

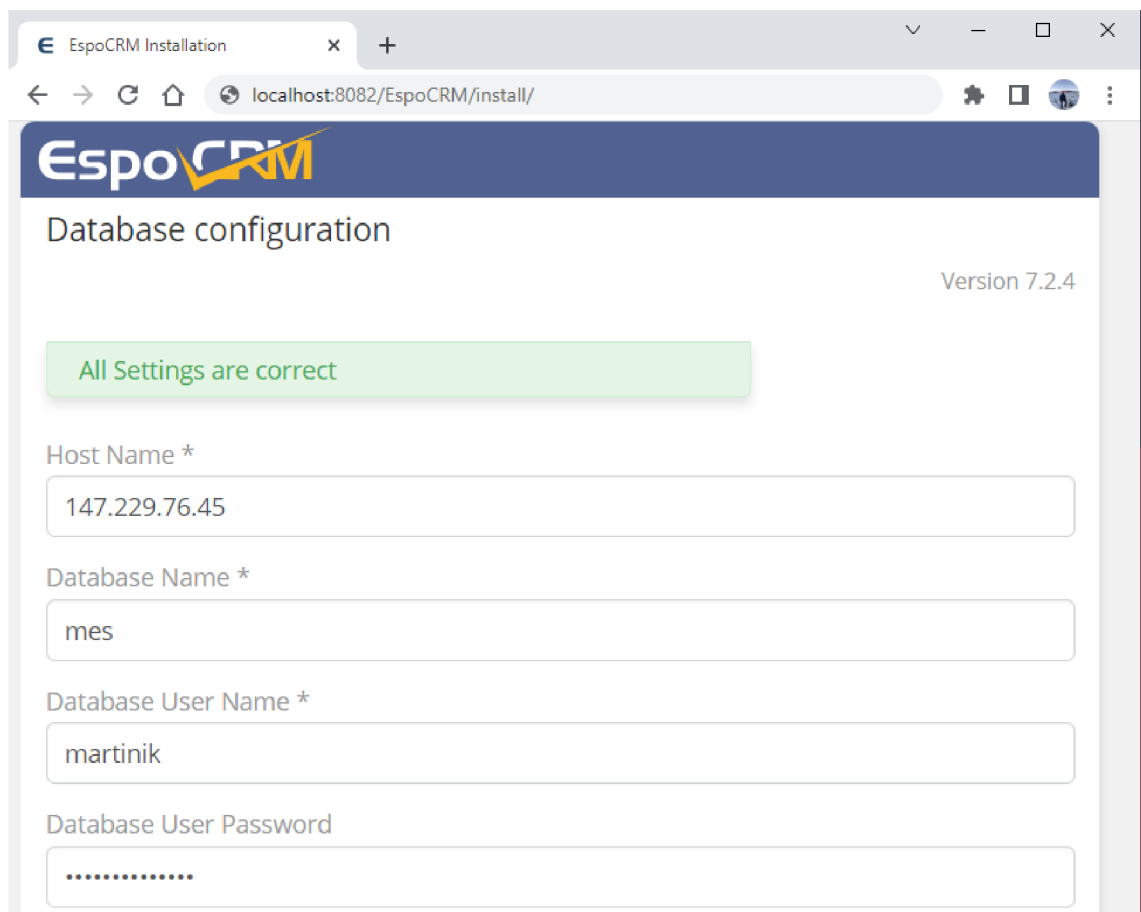
⁴ Existuje technická dokumentace, ale ta spíše popisuje, jak byl systém „postaven“

5. NÁVRH A IMPLEMENTACE NOVÉHO MES ŘEŠENÍ

5.1 Nastavení a spuštění systému

V rámci implementace nového MES řešení bylo v první řadě nutné stáhnout zdrojový kód open-source aplikace EspoCRM, která byla vybrána na základě provedené rešerše. Zdrojový kód byl stažen z oficiálního odkazu na webových stránkách vydavatele softwaru (<https://github.com/espocrm/espocrm>). Dle dokumentace je potřeba nastavit databázi a mít nainstalovaný jazyk PHP o minimální verzi 8.0.

Pro aplikaci byl připraven server s MySQL databází a PHP 8.1.12. Po prvním spuštění aplikace se otevře stránka s výběrem jazyka a následně formulář pro nastavení komunikace s databází.



The screenshot shows a web browser window titled "EspoCRM Installation" with the address bar showing "localhost:8082/EspoCRM/install/". The page content includes the EspoCRM logo, the title "Database configuration", and the version "Version 7.2.4". A green message box states "All Settings are correct". Below this, there are four input fields for database configuration: "Host Name *" with the value "147.229.76.45", "Database Name *" with the value "mes", "Database User Name *" with the value "martinik", and "Database User Password" which is currently masked with dots.

Obrázek 25: Databázová konfigurace EspoCRM

Po správném nastavení databáze v systému došlo k přesměrování stránky na tabulku systémových požadavků. Bylo potřeba upravit některé parametry v konfiguračním souboru jazyka PHP php.ini dle tabulky č.2.

Tabulka 2: Nastavení PHP pro aplikaci

Parametr	Hodnota	Stav
PHP Version	8.1.12	Success
json	On	Success
openssl	On	Success
pdo_mysql	On	Success
mbstring	On	Success
zip	On	Success
gd	On	Success
iconv	On	Success
curl	On	Success
xml	On	Success
xmlwriter	On	Success
exif	On	Success
max_execution_time	180	Success
max_input_time	180	Success
memory_limit	512M	Success
post_max_size	50M	Success
upload_max_filesize	50M	Success

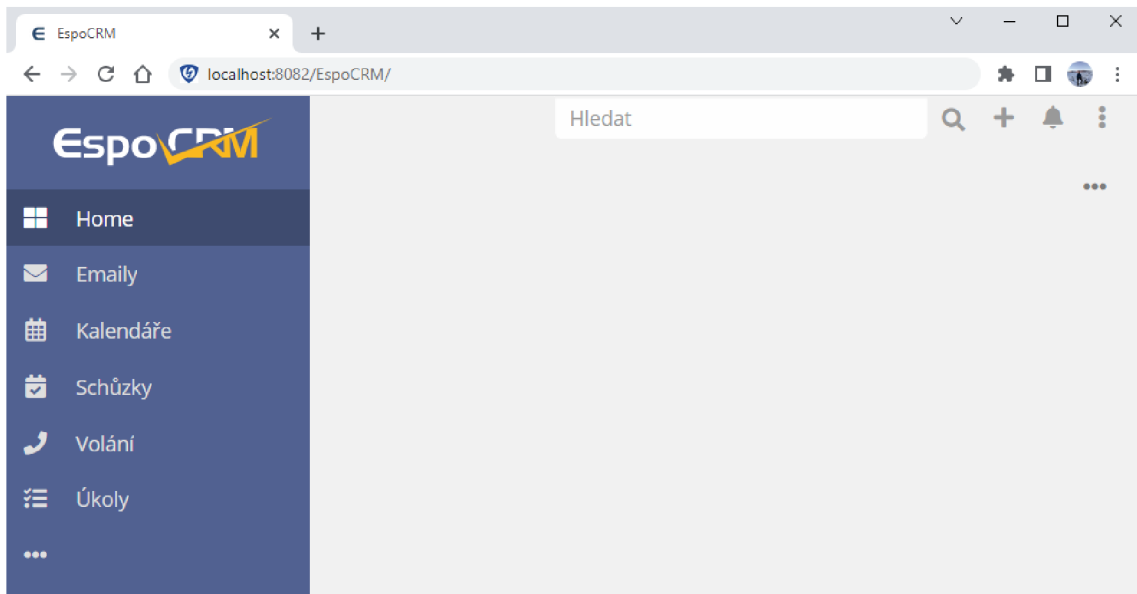
Po úspěšné konfiguraci byla aplikace nainstalována a byly vytvořeny přihlašovací údaje pro administrátora systému. Následně aplikace zobrazila další dialogová okna s nastavením např. názvu aplikace, pravidla pro zobrazování dat, SMTP nastavení apod. Tyto formuláře však nebylo třeba vyplňovat a došlo k dokončení instalace systému. V administraci bylo možné ověřit databázové připojení, viz. Tabulka č.3. Aplikace byla spuštěna a připravena k úpravě na MES systém.

Tabulka 3: Nastavení databáze pro aplikaci

Parametr	Hodnota	Stav
MySQL Version	8.0.28	Success
Host Name	147.229.76.45	Success
Database Name	mes	Success
User Name	martinik	Success

5.2 Implementace nových modulů

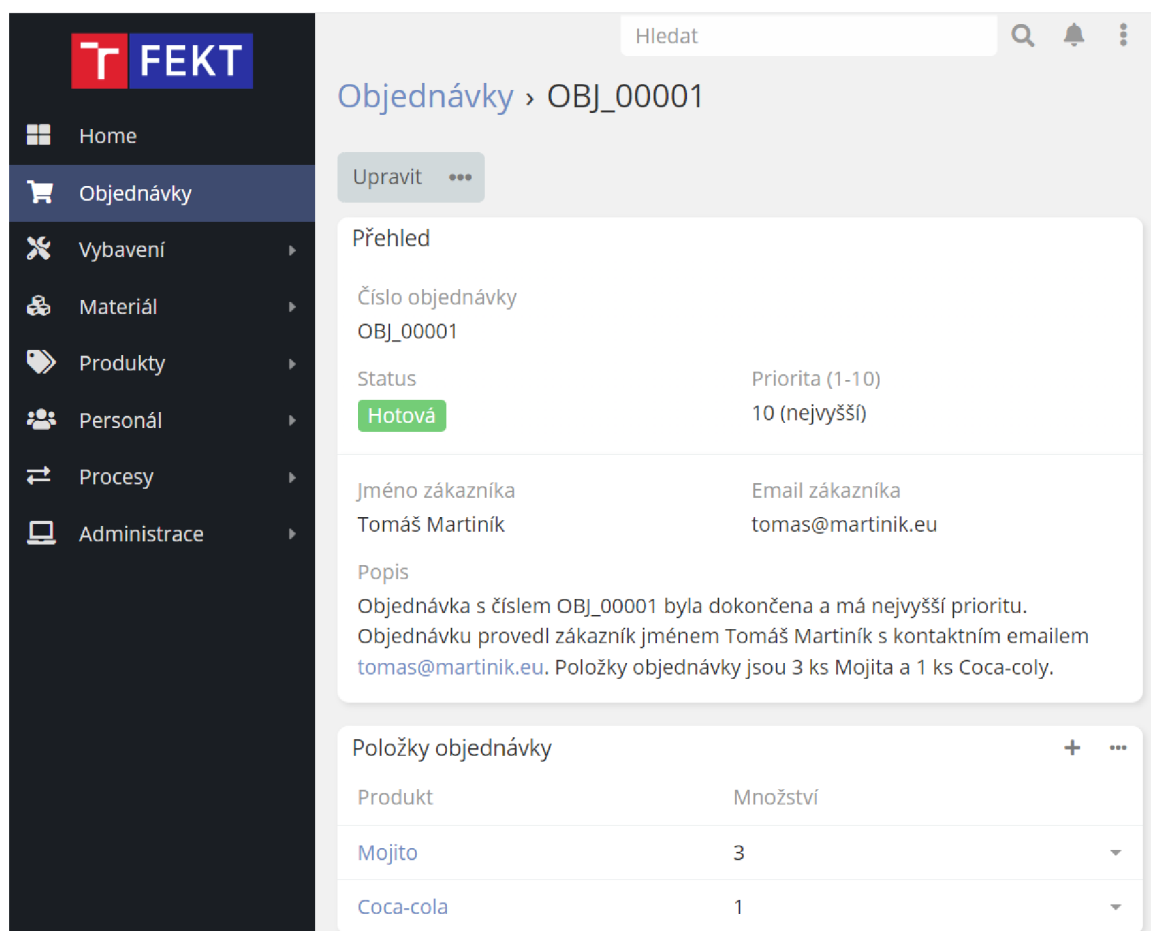
Aplikace po prvním spuštění obsahovala určité moduly (entity) vhodné pro CRM systém. Po vzoru těchto modulů je možné vytvářet moduly nové, čehož bylo využito pro tvorbu MES systému. Cílem bylo, aby tabulky jednotlivých modulů v databázi odpovídaly tabulkám v databázi předchozího MES řešení, což totiž následně usnadní implementaci exekučního modulu pro integraci řešení do systému demonstrátoru.



Obrázek 26: Zobrazení EspoCRM po prvním spuštění

5.2.1 Správa objednávek

Modul pro objednávky byl vytvořen pro jednoduchou správu a evidenci objednávek. V databázi byla vytvořena tabulka orders, což je tabulka se základními údaji objednávky, jako např. číslo, priorita, stav, popis, jméno a identifikátor zákazníka apod. Priorita byla určena na výběr čísla od 1 do 10 (vzestupná důležitost). Obsahuje také časové údaje o objednávce, tedy kdy byla vytvořena, kdy byla dokončena, kdy výroba objednávky začala apod.



The screenshot shows the 'Objednávky' (Orders) module in the FEKT MES system. The breadcrumb navigation is 'Objednávky > OBJ_00001'. There is an 'Upravit' (Edit) button. The 'Přehled' (Overview) section contains the following information:

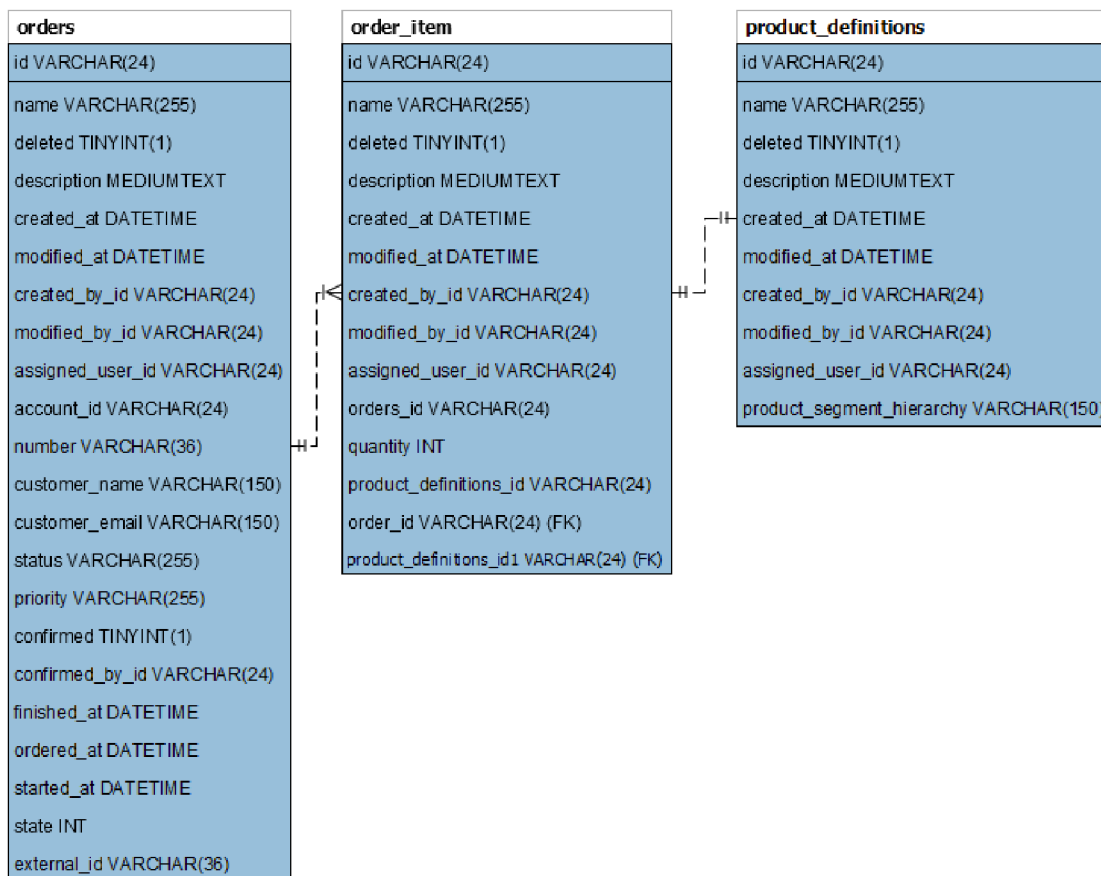
- Číslo objednávky: OBJ_00001
- Status: Hotová
- Priorita (1-10): 10 (nejvyšší)
- Jméno zákazníka: Tomáš Martiník
- Email zákazníka: tomas@martinik.eu
- Popis: Objednávka s číslem OBJ_00001 byla dokončena a má nejvyšší prioritu. Objednávku provedl zákazník jménem Tomáš Martiník s kontaktním emailem tomas@martinik.eu. Položky objednávky jsou 3 ks Mojita a 1 ks Coca-coly.

The 'Položky objednávky' (Order Items) section contains a table with the following data:

Produkt	Množství
Mojito	3
Coca-cola	1

Obrázek 27: Detail objednávky v MES systému

Tabulka orders má vazbu 1:N na tabulku order_item. Tato tabulka existuje pro vyčítání položek objednávky, tedy produktů, které byly objednány, a jejich množství. Položka objednávky má zároveň vazbu N:1 na tabulku product_definitions, která obsahuje informace o definicích konkrétních produktů. Jinými slovy, pomocí této vazby přiřazujeme konkrétní produkt k položce objednávky. Položek pak může mít objednávka libovolný počet.



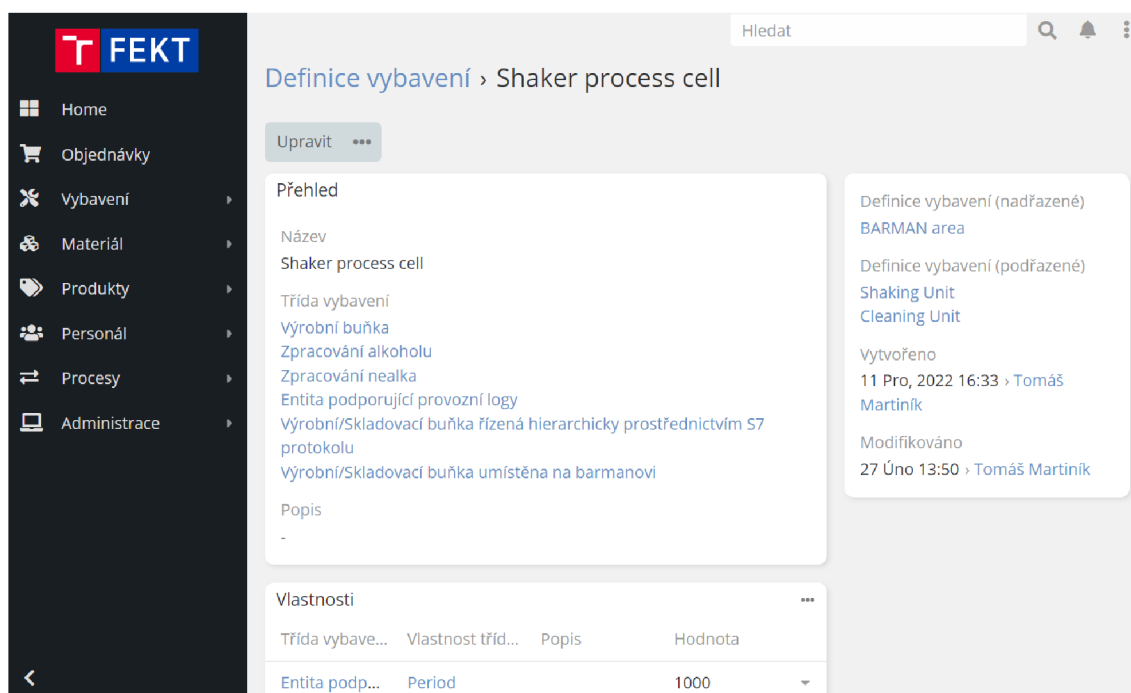
Obrázek 28: Databázový model objednávek

5.2.2 Správa vybavení

Model vybavení se řídí definicí objektových modelů standardu ISA-95. Každé konkrétní vybavení (zařízení, přístroje apod.) má svou definici a každé definici lze přiřadit třídu. Třídy mají své vlastnosti, které je definují a definici se po přiřazení třídy vygenerují vlastnosti odpovídající přiřazeným třídám. Do políček u vlastností přísluší hodnota konkrétní vlastnosti. Každé definici zároveň mohou být přiřazeny nadřazená a podřazené definice, čímž se tvoří jakási hierarchie definic vybavení.

Tento objektový model lze demonstrovat na příkladu definice vybavení Shaker process cell. Jedná se o procesní buňku, která je zodpovědná za protřepání kapaliny. Tato definice má přiřazených několik tříd, např. Výrobní buňka, Zpracování alkoholu nebo Entita podporující provozní logy. V těchto třídách jsou definovány vlastnosti a ty jsou vygenerovány u definice vybavení Shaking Unit, např. z třídy Entita podporující provozní logy je vygenerována vlastnost Period, která odpovídá periodě logování. U této vlastnosti je v této konkrétní definici přiřazena hodnota 1000 ms. Definice má také nadřazenou

položku BARMAN area, což je definice celkové plochy demonstrátoru, a podřazené položky Shaking Unit a Cleaning Unit.



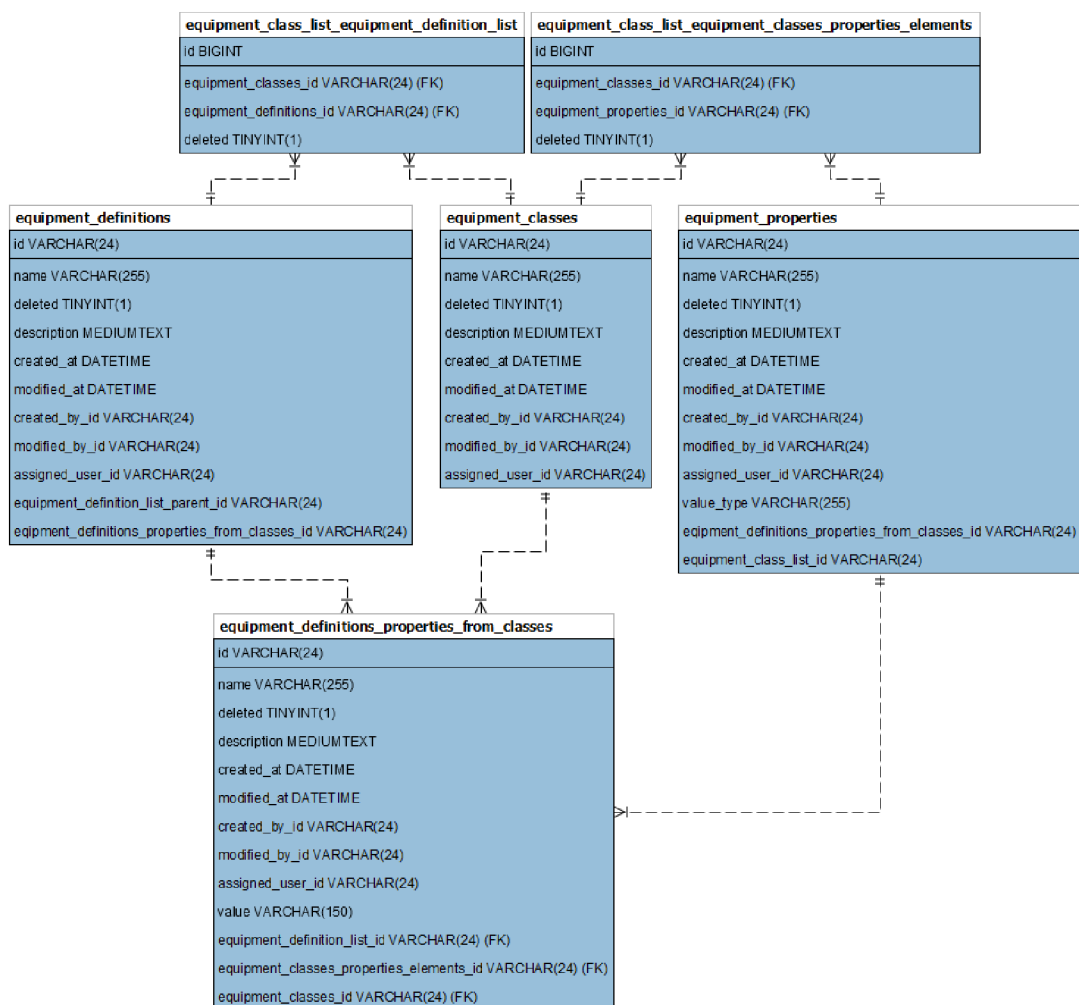
Obrázek 29: Definice vybavení Shaker process cell

Generování tříd u konkrétní definice vybavení je naprogramováno pomocí vlastního skriptu. Ve zdrojovém kódu se nachází v:

```
custom/Espo/Custom/Hooks/EquipmentDefinitions/UpdateFields.php.
```

Tento hook je napsaný v jazyce PHP. Obsahuje třídu UpdateFields, která pomocí metody afterSave která se spouští po uložení záznamu entity (definice vybavení). V této metodě je naprogramována kontrola, zda došlo ke změně v seznamu přiřazených tříd (atribut equipmentClassListsIds). Pokud ke změně došlo, jsou získány přiřazené třídy. Následuje kontrola, které třídy jsou nově přiřazené, které jsou nezměněny a které byly odstraněny. Pro nezměněné třídy se nic nevykonává, zatímco pro třídy nově přiřazené je vytvořen záznam v entitě EquipmentDefinitionsPropertiesFromClasses, který odpovídá položce vlastnosti, tzn. položka s políčky vlastnost, nadřazená třída a hodnota. Pro odebrané třídy ze seznamu jsou záznamy (položky) odstraněny. Pro uživatele se tato projeví tím, že při každém uložení entity (což odpovídá možnosti reorganizovat třídy) se synchronizují vlastnosti přiřazených tříd v uživatelském rozhraní.

Databázový model se skládá ze tří tabulek určených pro definici, třídy a vlastnosti vybavení. Tyto tabulky jsou navzájem propojeny pomocí vazebních tabulek tak, aby bylo možné modelovat vztahy M:N a zároveň tvořit položky (např. pro vlastnosti definic) s dalšími parametry vztahu.



Obrázek 30: Databázový diagram vybavení

5.2.3 Správa personálu

Model personálu se řídí definicí objektových modelů standardu ISA-95. Každý konkrétní personál má svou definici a každé definici lze přiřadit třídu. Třídy mají své vlastnosti, které je definují a definici se po přiřazení třídy vygenerují vlastnosti odpovídající přiřazeným třídám. Do políček u vlastností přísluší hodnota konkrétní vlastnosti.

Tento objektový model lze demonstrovat na příkladu definice personálu Karel Novák. Tato definice má přiřazenou jednu třídu, jedná se o třídu Student VUT. V této třídě jsou definovány vlastnosti a ty jsou vygenerovány u definice personálu, tzn. z třídy Student VUT je vygenerována vlastnost ID studenta U této vlastnosti je v této konkrétní definici přiřazena hodnota 123456.

Generování tříd u konkrétní definice personálu je naprogramováno pomocí hook skriptu. Ve zdrojovém kódu se nachází v:

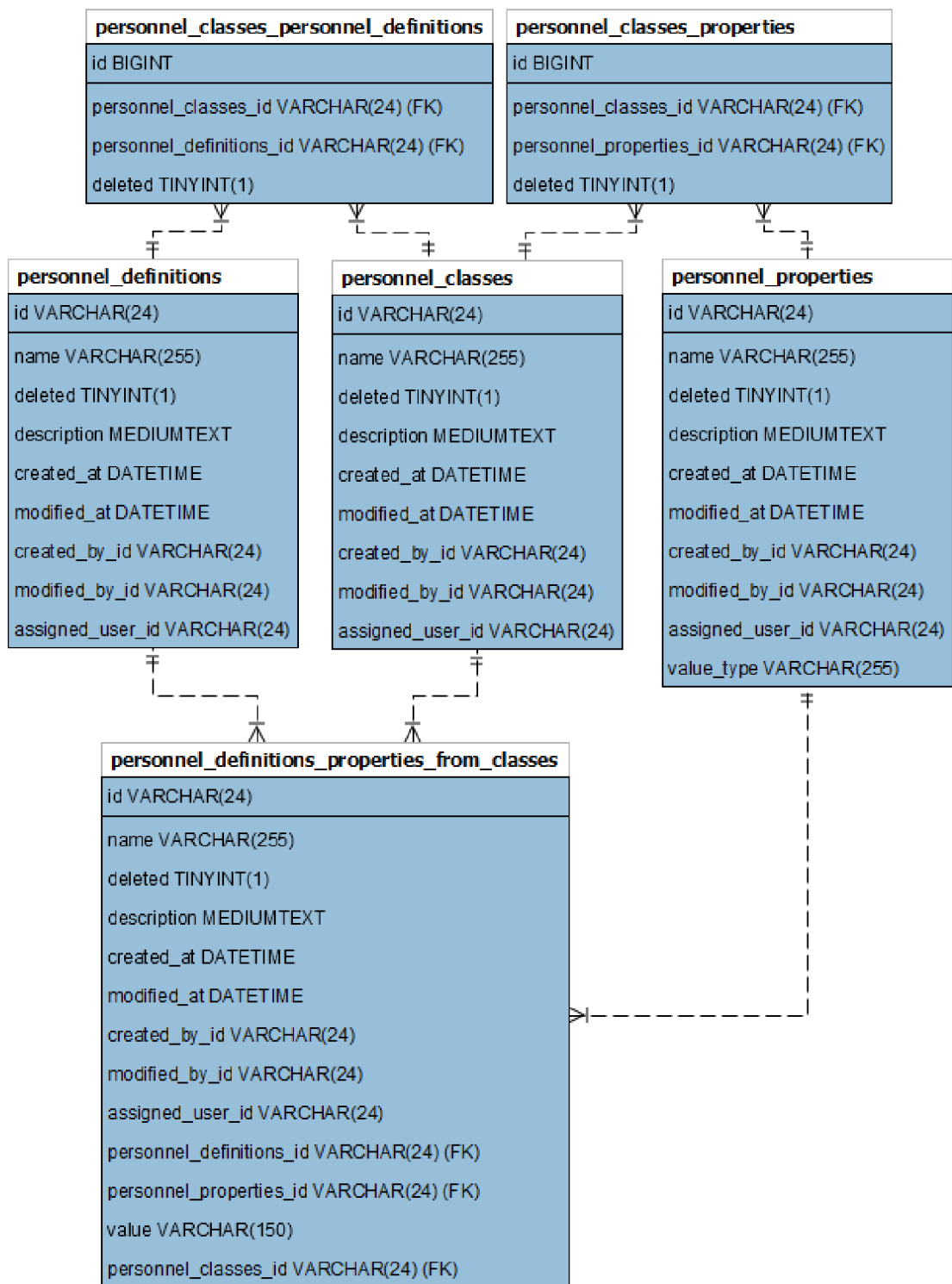
custom/Espo/Custom/Hooks/PersonnelDefinitions/UpdateFields.php.

Tento hook je napsaný v jazyce PHP. Obsahuje třídu UpdateFields, která pomocí metody afterSave která se spouští po uložení záznamu entity (definice personálu). V této metodě je naprogramována kontrola, zda došlo ke změně v seznamu přiřazených tříd (atribut personnelClassesIds). Pokud ke změně došlo, jsou získány přiřazené třídy. Následuje kontrola, které třídy jsou nově přiřazené, které jsou nezměněny a které byly odstraněny. Pro nezměněné třídy se nic nevykonává, zatímco pro třídy nově přiřazené je vytvořen záznam v entitě PersonnelDefinitionsPropertiesFromClasses, který odpovídá položce vlastnosti, tzn. položka s políčky vlastnost, nadřazená třída a hodnota. Pro odebrané třídy ze seznamu jsou záznamy (položky) odstraněny. Pro uživatele se tato projeví tím, že při každém uložení entity (což odpovídá možnosti reorganizovat třídy) se synchronizují vlastnosti přiřazených tříd v uživatelském rozhraní.

Vlastnost personálu	Třída personálu	Popis	Hodnota
ID studenta	Student VUT	6 místné číslo	123456
Název studijního oboru	Student VUT	Zkratka	BPC-AMT
Název fakulty	Student VUT	Zkratka	FEKT

Obrázek 31: Příklad definice personálu

Databázový model se skládá ze tří tabulek určených pro definici, třídy a vlastnosti personálu. Tyto tabulky jsou navzájem propojeny pomocí vazebních tabulek tak, aby bylo možné modelovat vztahy M:N a zároveň tvořit položky (např. pro vlastnosti definic) s dalšími parametry vztahu.



Obrázek 32: Databázový diagram personálu

5.2.4 Správa materiálu

Model materiálu se řídí definicí objektových modelů standardu ISA-95. Každý konkrétní materiál má svou definici a každé definici lze přiřadit třídu. Třídy mají své vlastnosti, které je definují a definici se po přiřazení třídy vygenerují vlastnosti odpovídající přiřazeným třídám. Do políček u vlastností přísluší hodnota konkrétní vlastnosti. Definice má také přiřazenou dávku materiálu, což je entita, v níž jsou obsaženy záznamy s definicí materiálu, množstvím a jednotkou.

Tento objektový model lze demonstrovat na příkladu definice materiálu Rum Republica. Tato definice má přiřazené dvě třídy, jedná se o třídy Rum a Lze zakoupit v obchodě. V těchto třídách jsou definovány vlastnosti a ty jsou vygenerovány u definice materiálu, např. z třídy Rum je vygenerována vlastnost Obsah alkoholu (%), která odpovídá procentnímu obsahu alkoholu v materiálu. U této vlastnosti je v této konkrétní definici přiřazena hodnota 38. Definice má také svoji dávku materiálu, která odpovídá hodnotě 1 l.

Vlastnost materiálu	Třída materiálu	Popis	Hodnota
Cena	Lze zakoupit v obchodě		500
Barva (popis)	Rum		Rumová
Obsah alkoholu (%)	Rum		38
Chuf (popis)	Rum		Ok

Název	Definice materiálu	Množství	Jednotka
Dávka Rum Republica	Rum Republica	1	l

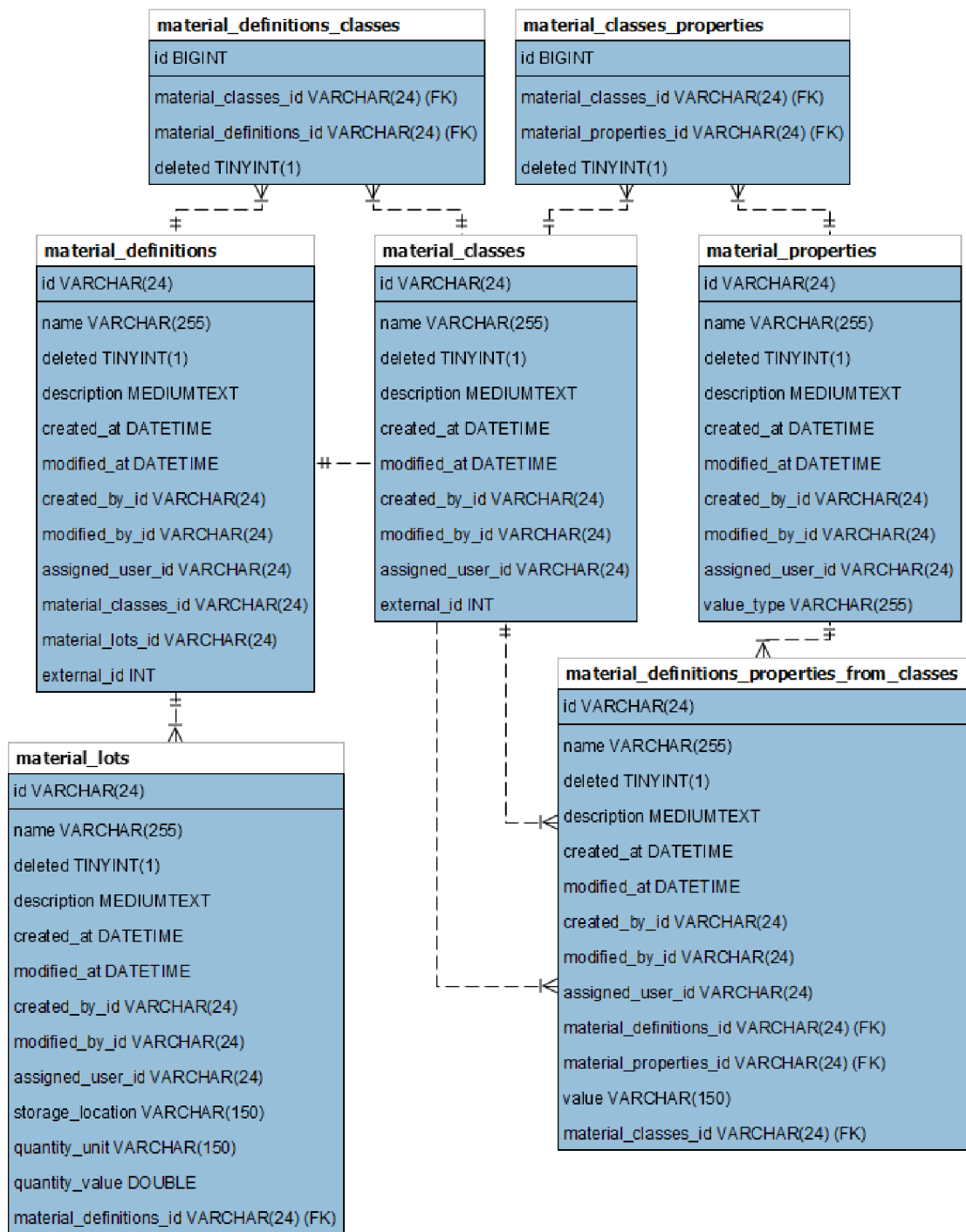
Obrázek 33: Příklad definice vybavení

Generování tříd u konkrétní definice materiálu je naprogramováno pomocí hook skriptu. Ve zdrojovém kódu se nachází v:

```
custom/Espo/Custom/Hooks/MaterialDefinitions/UpdateFields.php.
```

Tento hook je napsaný v jazyce PHP. Obsahuje třídu UpdateFields, která pomocí metody afterSave která se spouští po uložení záznamu entity (definice materiálu). V této metodě je naprogramována kontrola, zda došlo ke změně v seznamu přiřazených tříd (atribut materialClassesIds). Pokud ke změně došlo, jsou získány přiřazené třídy. Následuje kontrola, které třídy jsou nově přiřazené, které jsou nezměněny a které byly odstraněny. Pro nezměněné třídy se nic nevykonává, zatímco pro třídy nově přiřazené je vytvořen záznam v entitě MaterialDefinitionsPropertiesFromClasses, který odpovídá položce vlastnosti, tzn. položka s políčky vlastnost, nadřazená třída a hodnota. Pro odebrané třídy ze seznamu jsou záznamy (položky) odstraněny. Pro uživatele se tato projeví tím, že při každém uložení entity (což odpovídá možnosti reorganizovat třídy) se synchronizují vlastnosti přiřazených tříd v uživatelském rozhraní.

Databázový model se skládá ze čtyř tabulek určených pro definici, třídy, vlastnosti a dávky materiálu. Tyto tabulky jsou navzájem propojeny pomocí vazebních tabulek tak, aby bylo možné modelovat vztahy M:N a zároveň tvořit položky (např. pro vlastnosti definic) s dalšími parametry vztahu.



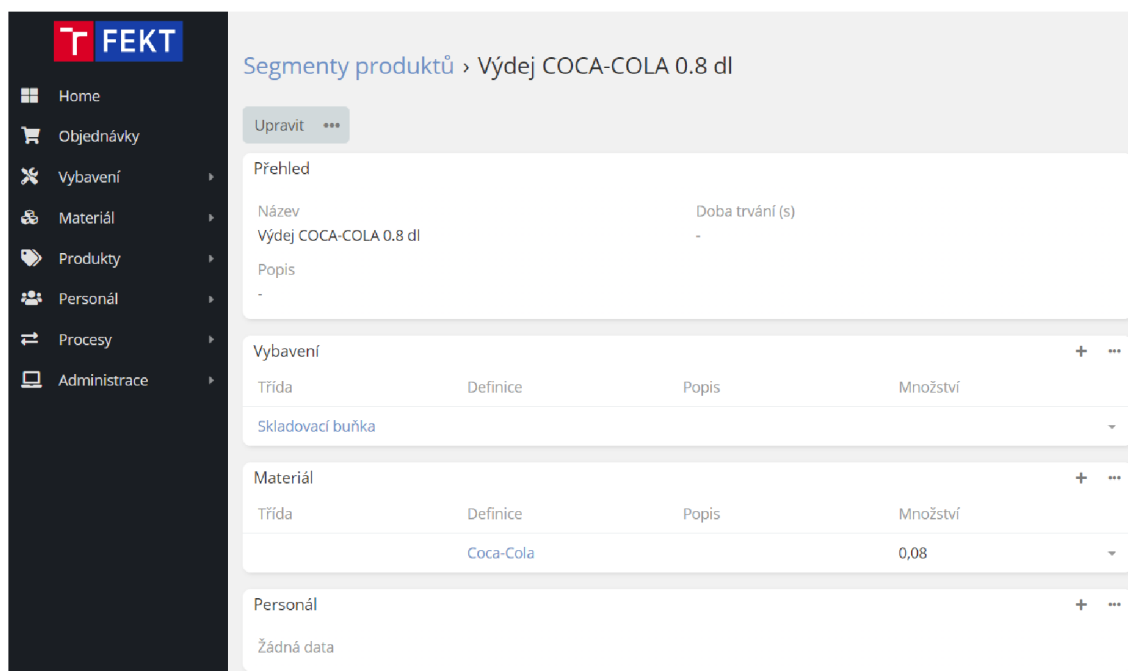
Obrázek 34: Databázový diagram materiálu

5.2.5 Produkty a produktové segmenty

Každý konkrétní produkt má v systému svoji definici. K této definici lze přiřazovat produktové segmenty, což jsou kombinace vybavení, materiálu a personálu potřebné k vyrobení konkrétního produktu. Tyto produktové segmenty lze v systému libovolně vytvářet a skládat tak zmíněné kombinace, ve kterých si lze vybrat, zda bude stanovena konkrétní definice či pouze třída.

Produkty a jejich segmenty lze demonstrovat na příkladu produktu Cuba Libre. Tento produkt má svou definici, ke které jsou přiřazeny čtyři segmenty. Jedná se o Mixování, Výdej COCA-COLA 0.8 dl, Výdej sklenice a Výdej Rum Republica 0.4 dl. Po rozkliknutí segmentu Výdej COCA-COLA 0.8 dl lze vidět, že je k němu přiřazeno vybavení třídy Skladovací buňka a definice materiálu Coca-Cola s množstvím 0.08.

Databázový model se skládá z tabulky pro definici produktového segmentu, který má vazbu na definici produktu. K segmentu je zároveň přes vazební tabulky navázán materiál, vybavení a personál.



FEKT

Segmenty produktů > Výdej COCA-COLA 0.8 dl

Upravit ...

Přehled

Název	Doba trvání (s)
Výdej COCA-COLA 0.8 dl	-

Popis
-

Vybavení

Třída	Definice	Popis	Množství
Skladovací buňka			

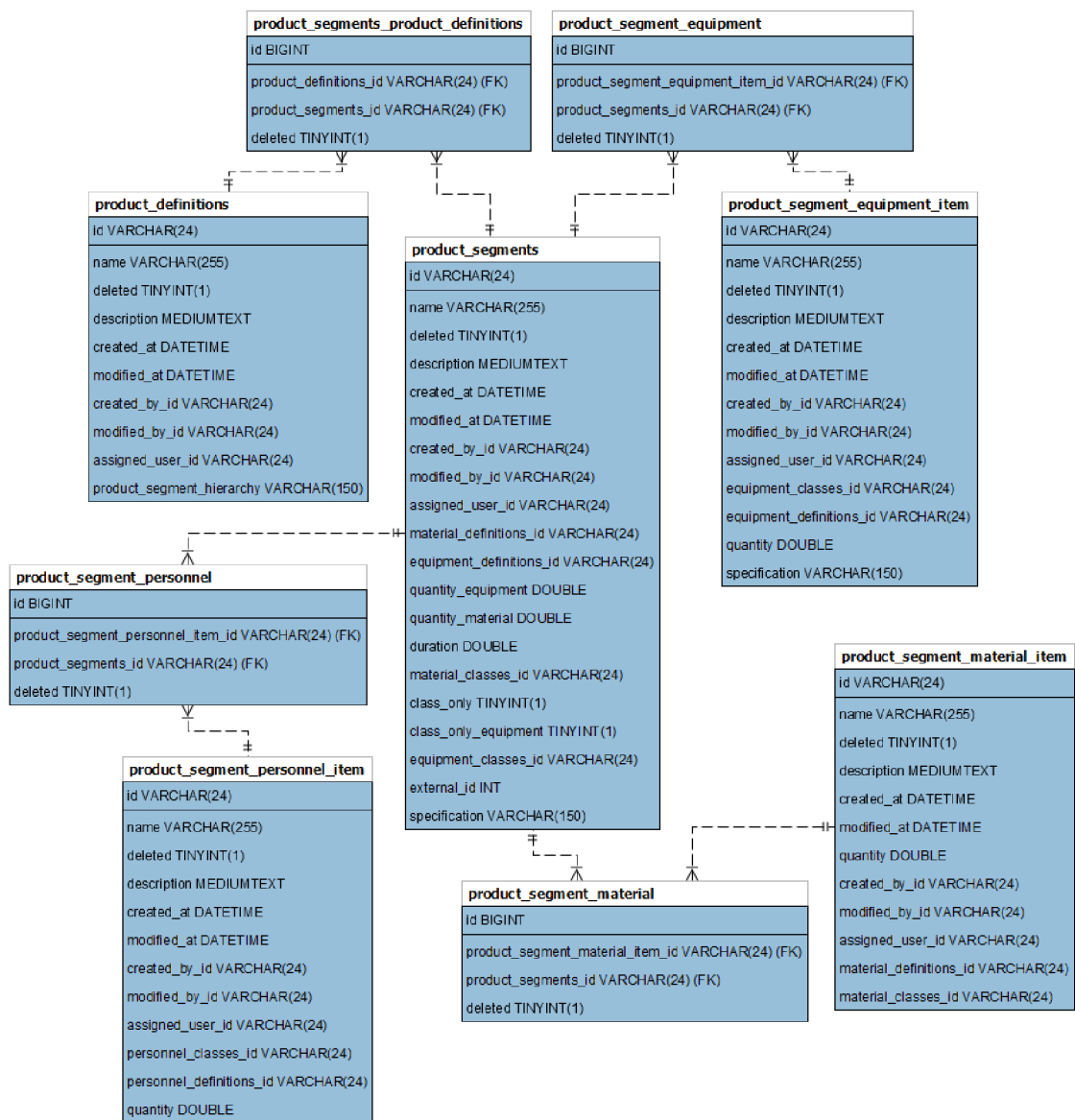
Materiál

Třída	Definice	Popis	Množství
	Coca-Cola		0,08

Personál

Žádná data

Obrázek 35: Příklad segmentu produktu

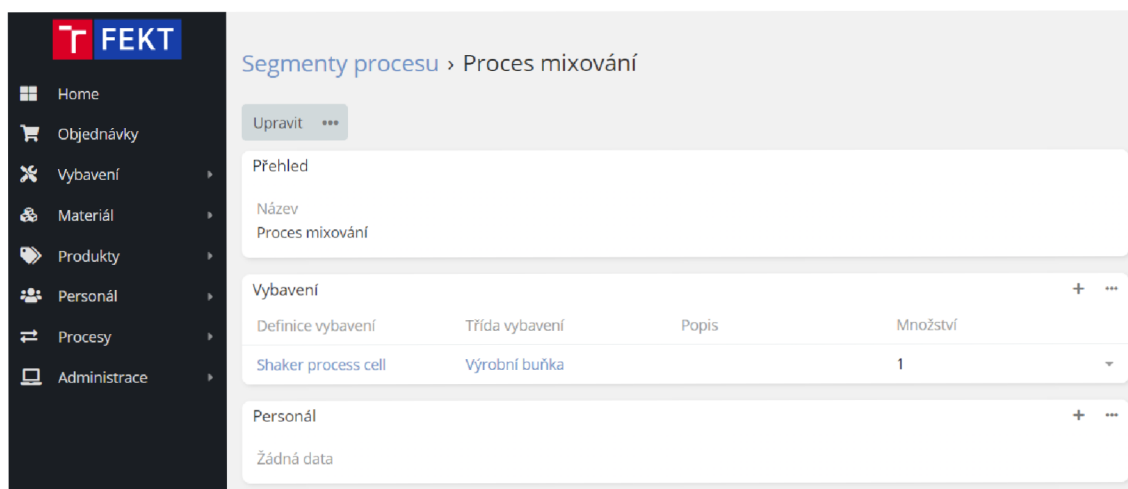


Obrázek 36: Databázový diagram produktů a produktových segmentů

5.2.6 Procesní segmenty

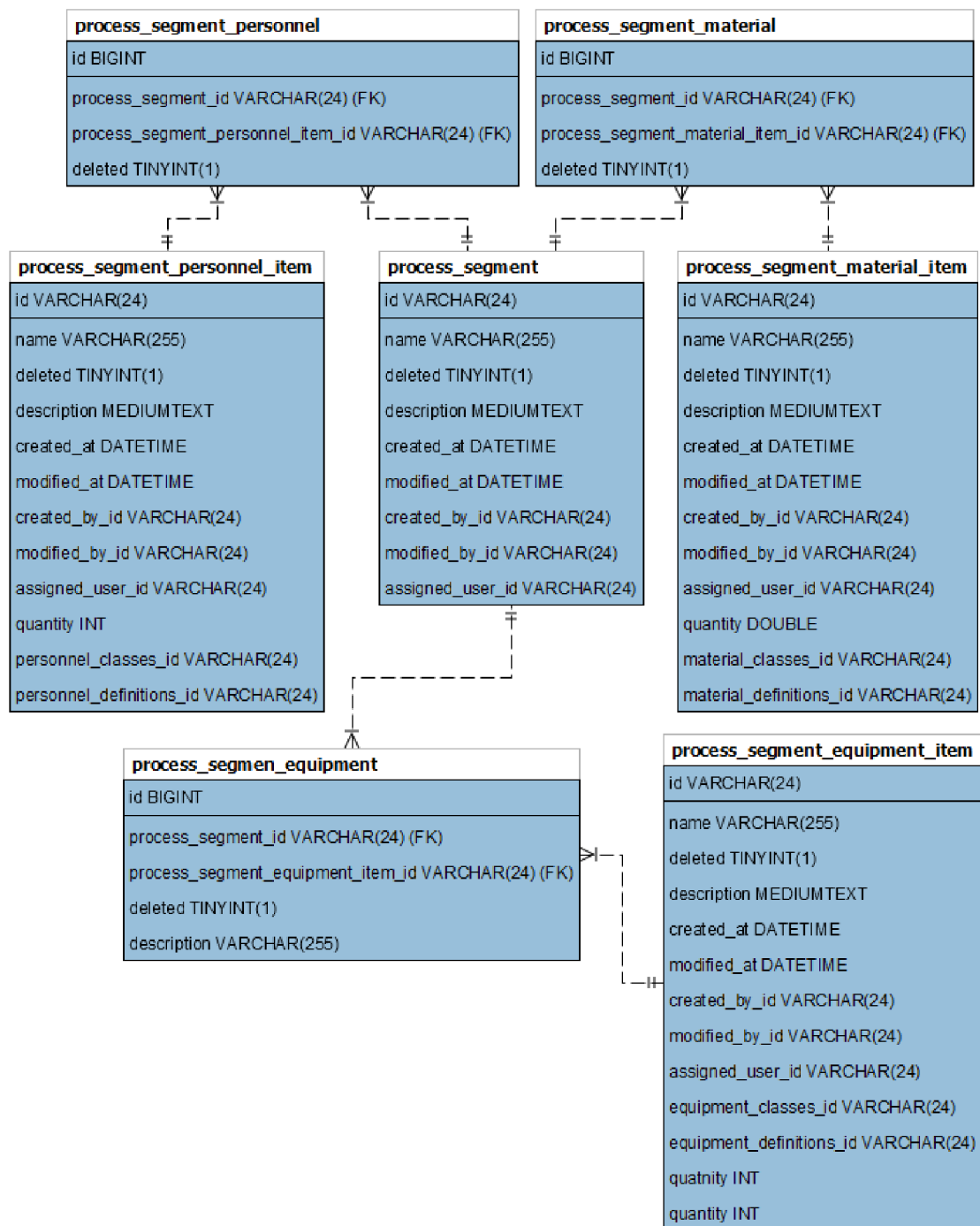
V systému lze definovat procesní segmenty, což jsou kombinace vybavení a personálu potřebné k vykonání konkrétního procesu. Při tvorbě segmentů si u připojených položek lze vybrat, zda bude stanovena konkrétní definice či pouze třída.

Procesní segmenty lze demonstrovat na příkladu segmentu Mixování. K tomuto záznamu je připojeno vybavení (definice) Shaker process cell.



Obrázek 37: Příklad segmentu procesu

Databázový model se skládá z tabulky pro definici procesního segmentu. K segmentu je přes vazební tabulky navázáno vybavení a personál.



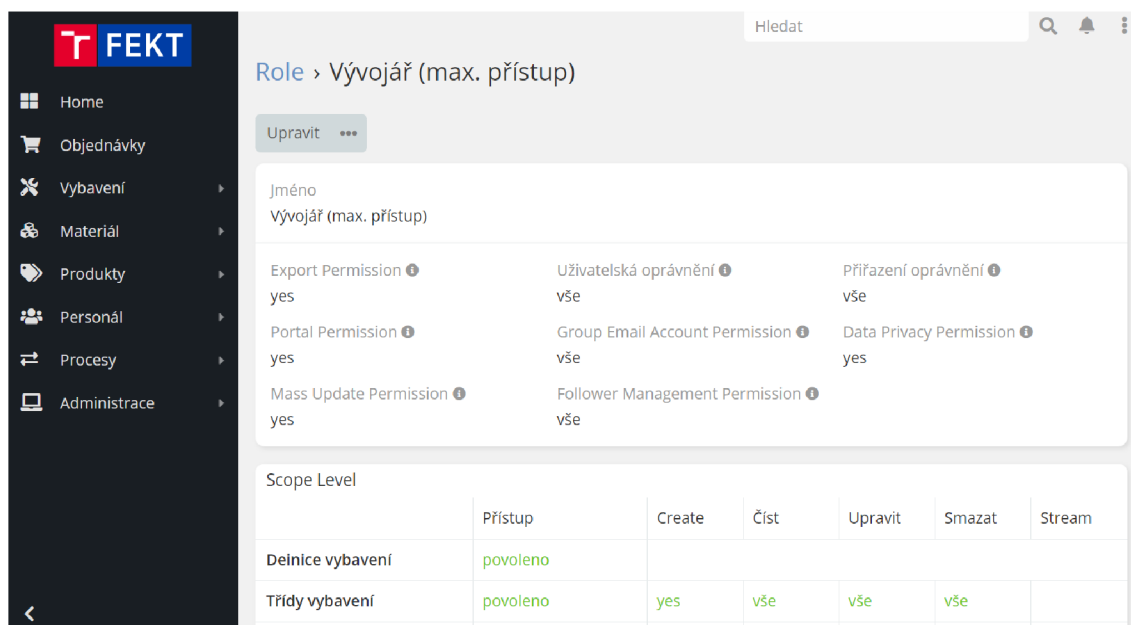
Obrázek 38: Databázový diagram procesů

5.2.7 Správa uživatelů, jejich rolí a práv

V rámci aplikace EspoCRM je možné spravovat uživatele, týmy a role s právy. [23] Při vytváření uživatele je potřeba vyplnit základní údaje, tzn. jméno, příjmení, e-mailová adresa, uživatelské jméno a heslo. Dále lze nastavit další volitelné informace. Po vytvoření uživatele je nutné mu přiřadit roli a týmy, ke kterým má přístup.

Při vytváření týmu je nutné určit název týmu a popis, a následně přidat uživatele, kteří mají být do týmu zařazeni. Tým lze následně přiřadit k modulům a datům, ke kterým mají uživatelé v týmu přístup.

V systému lze editovat oprávnění a přístup k datům pro jednotlivé uživatele či týmy prostřednictvím rolí. Pro každý modul lze definovat oprávnění pro čtení, zápis, editaci a mazání dat. Role lze následně přiřadit k uživatelům a týmům.



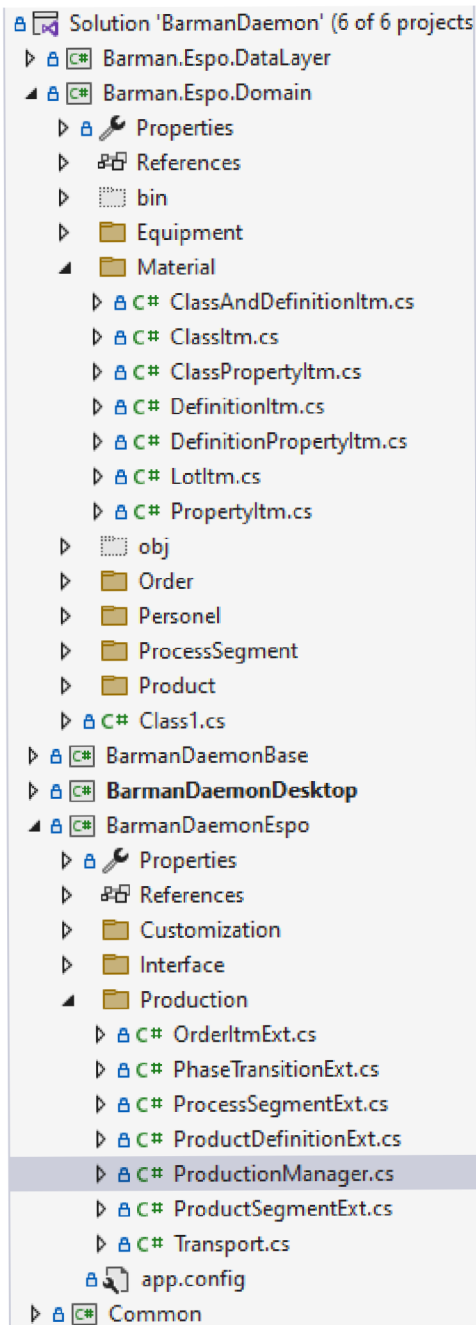
Obrázek 39: Role a práva v aplikaci EspoCRM

5.3 Implementace exekučního modulu

Pro implementaci exekučního modulu bylo potřeba vytvořit třídy reprezentující objektové modely (tabulky v databázi). Díky návrhu nového MES systému dle standardu ISA-95 třídy téměř odpovídaly třídám již existujícího exekučního modulu. Cílem tedy bylo urovnat nedostatky a vyřešit drobné odlišnosti tříd. V novém systému byly přidány sloupce v databázových tabulkách, které obsahoval starý systém a v novém chyběly, protože některé z těchto sloupců exekuční modul využívá.

Hlavním problémem implementace exekučního modulu byl rozdíl datového typu identifikátorů databázových tabulek. Zatímco ve předchozím řešení byly id typu int, tedy inkrementace hodnot od 1, v novém systému jsou id typu string. Např. tedy definice materiálu Mléko má v předchozím řešení id 56 a v novém řešení id 63bddbef0a4c3eb77. Metody tříd exekučního modulu byly přizpůsobené pro id typu int (číslo), zatímco nový MES systém, založený na systému EspoCRM, vyžadoval id typu string (řetězec). Bylo proto nutné najít řešení vyhovující oběma stranám, čehož bylo dosaženo přidáním sloupců externí ID (external_id), kde se pro jednotlivé záznamy generuje unikátní číslo

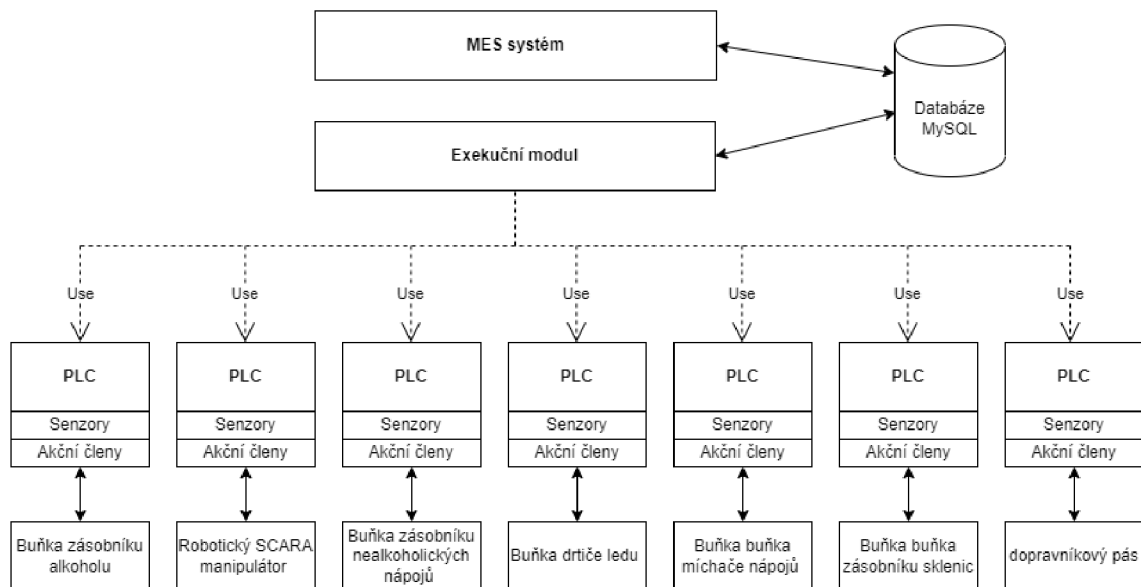
(pomocí inkrementace od 1). Nově implementovaný exekuční modul tedy používá výhradně identifikátor `external_id`, se kterým už zvládne pracovat.



Obrázek 40: Repositář exekučního modulu

5.4 Hierarchie celého systému

Po implementaci nového MES řešení systém funguje tak, že MES má vlastní MySQL databázi, do které ukládá data, která zároveň zobrazuje. Následně databázi využívá exekuční modul se svým GUI pro vyčítání objednávek a všech objektů a parametrů potřebných ke spuštění a vykonání výroby. Exekuční modul následně komunikuje pomocí S7 protokolu s příslušnými PLC výrobních buněk, které zároveň posílají data do PLC např. ze senzorů.



Obrázek 41: Hierarchie celého systému

5.5 Možnosti další práce

Systém je pro potřeby demonstrátoru funkční a využitelný, nicméně existují možnosti jeho rozšíření o další moduly a funkce, které by mohly přinést výhody pro uživatele a zlepšit celkovou efektivitu systému.

5.5.1 Skladový modul

Dalším možným krokem ve vývoji tohoto systému je tvorba skladového modulu. Pomocí něj by uživatelé mohli spravovat skladové zásoby materiálů a výrobků, které mají k dispozici.

5.5.2 Modul pro KPI

Jedním z dalších kroků je vytvoření modulu KPI (klíčové ukazatele výkonnosti). Modul by uživateli umožnil sledovat a měřit účinnost výrobních procesů analýzou dat. Pro

implementaci modulu a zobrazení KPI metrik se nabízí využít open-source grafických knihoven.

5.5.3 Využití umělé inteligence

Pro aplikaci také dává smysl prozkoumat možnosti využití umělé inteligence, tzn. zjistit, jaká data a v jaké podobě by mohla být užitečná a které algoritmy použít. Příkladem využití může být předpověď objednávek, doporučení objednávek materiálu apod.

5.5.4 Vytvoření či integrace nadřazeného ERP systému

Dalším možným směrem práce je vytvoření či integrace nadřazeného ERP systému, který by zastřešoval celkový chod podniku a zároveň měl k dispozici určité údaje z výrobního systému. Nabízí se po vzoru této práce využít systém EspoCRM a přetvořit jej na ERP systém.

6. ZÁVĚR

V této bakalářské práci jsem se zaměřil na problematiku výrobního procesu a jeho řízení pomocí MES systému. Cílem práce bylo nahradit stávající řešení MES systému pro demonstrátor výrobního podniku novým řešením, které bylo potřeba navrhnout a implementovat.

V první části práce byly stanoveny požadavky na nový systém a byla provedena rešerše dostupných open-source řešení MES (či jiných) systémů a na základě potřeb demonstrátoru byl jeden ze systémů zvolen pro další práci. V druhé části bylo navrženo a implementováno nové řešení s využitím open-source systému EspoCRM. Byly provedeny úpravy a konfigurace systému dle požadavků našeho demonstrátoru výrobního podniku, došlo k vytvoření několika modulů pro řízení výroby a správu zdrojů. V poslední části práce jsme provedli integraci do existujícího systému demonstrátoru pomocí implementace již existujícího exekučního modulu.

Požadavky na nový systém byly splněny, zároveň byly identifikovány oblasti, ve kterých by bylo možné systém dále rozšířit a vylepšit, například přidáním modulů pro skladové zásoby, KPI apod.

LITERATURA

- [1] MES centrum, „MES center,“ [Online]. Available: <http://www.mescenter.org/cz/>. [Přístup získán 2022].
- [2] Crosser, „ISA-95 Automation Pyramid,“ [Online]. Available: <https://www.crosser.io/media/r4gnymym/crosser-edge-analytics-isa-95-illustration.svg>. [Přístup získán 2021].
- [3] „Krypton Solid,“ 2023. [Online]. Available: <https://kryptonsolid.com/cs/>.
- [4] Epicor Software Corporation, „Epicor,“ [Online]. Available: <https://www.epicor.com/en/>.
- [5] Vorne Industries Inc., „OEE: Overall Equipment Effectiveness,“ [Online]. Available: <https://www.oee.com/>.
- [6] Václav Kaczmarczyk, „An Industry 4.0 Testbed (Self-Acting Barman): Principles and Design,“ 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318309108>.
- [7] Odoo SA, „Odoo,“ 2022. [Online]. Available: <https://www.odoo.com/>.
- [8] Educius z.s., „Třívrstvá architektura (Three-tier architecture),“ [Online]. Available: <https://managementmania.com/cs/trivrstva-architektura-three-tier-architecture>.
- [9] D. Čápka, „IT Network,“ [Online]. Available: <https://www.itnetwork.cz/images/5/patterns/mvc.png>.
- [10] T. Segal, „Product Lifecycle Management (PLM): Definition, Benefits, History,“ 17 3 2023. [Online]. Available: <https://www.investopedia.com/terms/p/product-life-cycle-management.asp>.
- [11] Educius z.s., „Řízení kvality (Quality Management),“ [Online]. Available: <https://managementmania.com/cs/rizeni-kvality>.
- [12] J. Trout, „MTBF: A Complete Overview,“ Noria Corporation, [Online]. Available: <https://www.reliableplant.com/mtbf-31702>.
- [13] J. Truelsen, „WKhtmlTOPdf,“ [Online]. Available: <https://wkhtmltopdf.org/>.
- [14] PostgreSQL Global Development Group, „PostgreSQL,“ [Online]. Available: <https://www.postgresql.org/>.
- [15] Qcadoo Limited Sp. z o.o., „Qcadoo,“ 2022. [Online]. Available: <https://www.qcadoo.com/en/>.
- [16] D. Čápka, „ITnetwork,“ [Online]. Available: <https://www.itnetwork.cz/>.

- [17] VMware, Inc., „Spring Framework,“ [Online]. Available: <https://spring.io/projects/spring-framework>.
- [18] The Apache Software Foundation, „Apache Tiles,“ [Online]. Available: <https://tiles.apache.org/framework/index.html>.
- [19] OpenJS Foundation, „jQuery,“ [Online]. Available: <https://jquery.com/>.
- [20] The Apache Software Foundation, „Apache Maven,“ [Online]. Available: <https://maven.apache.org/>.
- [21] Dolibarr foundation, „Dolibarr ERP/CRM,“ [Online]. Available: <https://www.dolibarr.org/>.
- [22] Frappe, „ERP Next,“ [Online]. Available: <https://erpnext.com/>.
- [23] C. Visitsaktavorn, „IMES smart industry,“ github.com, [Online]. Available: <https://github.com/jukbot/smart-industry>.
- [24] EspoCRM, „EspoCRM,“ 2022. [Online]. Available: <https://www.espocrm.com/>.

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

API	Application programming interface
CRM	Customer relationship management
CSS	Cascading Style Sheets
DAO	Data Access Object
DTO	Data Transfer Objects
ERP	Enterprise resource planning
GUI	Grafické uživatelské rozhraní
HMI	Human-Machine Interface
HTML	HyperText Markup Language
MES	Manufacturing Execution System
MTBF	Mean time between failures
MRP	Material requirements planning
OEE	Overall equipment effectiveness
ORM	Object–relational mapping
OS	Operační systém
PLM	Product Lifecycle Management
REST	Representational state transfer
SaaS	Software as a service
SCADA	Supervisory control and data acquisition
SSH	Secure Shell