Systém pro správu skladu elektronických součástek

WMS – Elektronické součástky

Projektová dokumentace

Autoři: Bc. Tomáš Simandl
Bc. Tomáš Szetei
Bc. Lukáš Kvapil

Datum: 24. února 2026

Dokumentace systému pro skladovou evidenci elektronických součástek

Abstrakt

Tento dokument popisuje návrh a implementaci systému pro správu skladu (*Warehouse Management System*, WMS) určeného pro evidenci elektronických součástek. Systém umožňuje sledovat zásoby součástek (rezistory, kondenzátory, tranzistory, integrované obvody a další), spravovat skladová místa, zpracovávat příjem a výdej zboží, generovat objednávky u dodavatelů a tvořit reporty o stavu skladu.

Dokument obsahuje analýzu požadavků, datový model, architekturu systému a UML/PlantUML diagramy popisující klíčové procesy.

Klíčová slova: WMS, sklad, elektronické součástky, správa zásob, inventura, příjem, výdej, objednávky.

Obsah

Abstrakt **1**

1 Úvod **6**

1.1 Motivace 6

1.2 Cíle projektu 6

1.3 Rozsah dokumentu 7

1.4 Použité nástroje a technologie 7

2 Analýza požadavků **8**

2.1 Funkční požadavky 8

2.1.1 Správa součástek 8

2.1.2 Skladová místa 8

2.1.3 Příjem a výdej 8

2.1.4 Objednávky a dodavatelé 9

2.1.5 Inventura 9

2.1.6 Reporty 9

2.1.7 Správa uživatelů 9

2.2 Nefunkční požadavky 10

2.3 Případy užití – přehled 10

3 Architektura systému **11**

3.1 Přehled architektury 11

3.2 REST API – přehled endpointů 13

3.3 Deployment diagram 13

3.4 Bezpečnostní model 14

4 Datový model **15**

4.1 Přehled entit 15

4.2 Konceptuální model tříd 16

4.3 ER diagram 17

4.4 Popis tabulek 19

4.4.1 Tabulka `component_categories` 19

4.4.2 Tabulka `components` . . . 19
4.4.3 Tabulka `locations` . . . 20
4.4.4 Tabulka `stock` . . . 20
4.4.5 Tabulka `stock_movements` . . . 21

5 Formální specifikace dotazů 22
5.1 Definice notace . . . 22
5.2 Příklady dotazů nad modelem WMS . . . 23
5.2.1 Součástky pod minimálním stavem . . . 23
5.2.2 Celková hodnota skladu . . . 23
5.2.3 Katalog součástek dodavatele . . . 23
5.2.4 Pohyby výdeje na projekt . . . 23
5.2.5 Složité dotazy – inventurní rozdíly . . . 23

6 Popis procesů 25
6.1 Proces příjmu součástek . . . 25
6.1.1 Popis kroků příjmu . . . 26
6.2 Proces výdeje součástek . . . 27
6.3 Proces inventury . . . 28
6.3.1 Fáze inventurního procesu . . . 29
6.4 Stav objednávky – stavový diagram . . . 30

7 Závěr 31
7.1 Shrnutí . . . 31
7.2 Další rozvoj . . . 31
7.3 Diagram tříd – rozšiřitelnost . . . 32

A Slovník pojmů a zkratek 33

B Instalace a konfigurace 34

Seznam obrázků

2.1 Use Case diagram systému WMS – elektronické součástky 10

3.1 Komponentový diagram systému 12
3.2 Deployment diagram – nasazení systému pomocí Docker 14

4.1 Konceptuální model tříd 17
4.2 ER diagram datového modelu 18

6.1 Sekvenční diagram – Příjem součástek na sklad 26
6.2 Sekvenční diagram – Výdej součástek ze skladu 28
6.3 Diagram aktivit – Průběh inventury 29
6.4 Stavový diagram objednávky 30

Seznam tabulek

1.1 Přehled použitých nástrojů 7

2.1 Nefunkční požadavky systému 10

3.1 Klíčové REST API endpointy 13
3.2 Matice oprávnění dle rolí 14

4.1 Tabulka component_categories 19
4.2 Tabulka components 19
4.3 Tabulka locations 20
4.4 Tabulka stock (zásoby) 20
4.5 Tabulka stock_movements (pohyby zásob) 21

Kapitola 1

Úvod

1.1 Motivace

Správa skladu elektronických součástek představuje náročný logistický úkol. Moderní elektronika využívá tisíce různých komponent – od běžných rezistorů a kondenzátorů přes specializované integrované obvody až po konektory a mechanické díly. Bez kvalitního informačního systému dochází k:

- duplicitním nákupům téhož zboží,
- ztrátě přehledu o aktuálních zásobách,
- zbytečnému zdržování při hledání součástek v neoznačených zásobnících,
- potížím při plánování výroby a prototypování,
- nesnadné dohledatelnosti dodavatelů a nákupních cen.

Systém WMS – Elektronické součástky si klade za cíl tyto problémy eliminovat a přinést přehlednou, efektivní a rozšiřitelnou správu skladových zásob.

1.2 Cíle projektu

1. Vytvořit centrální evidenci všech elektronických součástek včetně jejich technických parametrů.
2. Zavést přesné sledování fyzických umístění v skladovém prostoru (regály, přihrádky, zásobníky).
3. Automatizovat procesy příjmu a výdeje zboží s vazbou na projekty.
4. Umožnit rychlé vyhledávání součástek dle parametrů (hodnota, pouzdro, výrobce, kategorie).

5. Generovat reporty o stavu skladu a pohybech zásob.
6. Spravovat dodavatele a objednávky.
7. Upozorňovat na součástky pod minimální hranicí zásob.

1.3 Rozsah dokumentu

Tento dokument pokrývá:

- analýzu funkčních a nefunkčních požadavků (chapter 2),
- popis navrhované architektury systému (chapter 3),
- datový model a ER diagram (chapter 4),
- popis klíčových procesů formou UML diagramů (chapter 6).

1.4 Použité nástroje a technologie

Tabulka 1.1: Přehled použitých nástrojů

Oblast	**Nástroj / Technologie**	**Popis**
Dokumentace	LaTeX, Biber	Sazba odborného textu
Diagramy	PlantUML	UML a ER diagramy
Databáze	PostgreSQL / SQLite	Relační databáze
Backend	Python / FastAPI	REST API server
Frontend	React + TypeScript	Webová aplikace
Kontejnerizace	Docker, Docker Compose	Nasazení prostředí
Správa kódu	Git, GitHub	Verzování zdrojového kódu

Kapitola 2

Analýza požadavků

2.1 Funkční požadavky

2.1.1 Správa součástek

FR-01 Systém umožní evidovat součástku s atributy: název, typ, výrobce, hodnota, tolerance, pouzdro (package), popis, datasheet URL.

FR-02 Systém podporuje kategorie součástek (rezistory, kondenzátory, tranzistory, diody, IC, konektory, ...).

FR-03 Každá součástka může mít přiřazený čárový kód (EAN/QR).

FR-04 Uživatel může vyhledávat součástky dle libovolné kombinace parametrů.

2.1.2 Skladová místa

FR-05 Systém spravuje hierarchii skladových míst: *Sklad* → *Sekce* → *Regál* → *Přihrádka*.

FR-06 Jedno skladové místo může obsahovat více druhů součástek; jedna součástka může být na více místech.

2.1.3 Příjem a výdej

FR-07 Příjem součástek ze skladu generuje pohyb typu `PŘÍJEM` a aktualizuje zásoby.

FR-08 Výdej součástek na projekt generuje pohyb typu `VÝDEJ` a sníží zásoby.

FR-09 Přeskladnění přesouvá zásoby mezi místy bez změny celkového množství.

FR-10 Systém hlídá minimální zásoby a upozorní na podkročení limitu.

2.1.4 Objednávky a dodavatelé

FR-11 Systém eviduje dodavatele s kontaktními údaji.

FR-12 Uživatel může vytvořit nákupní objednávku u dodavatele.

FR-13 Objednávka prochází stavy: `NÁVRH` → `ODESLÁNO` → `POTVRZENO` → `PŘIJATO` → `STORNOVÁNO`.

2.1.5 Inventura

FR-14 Systém umožní zahájit inventurní cyklus, během nějž uživatel ověřuje fyzické zásoby.

FR-15 Systém generuje inventurní list (PDF/CSV) se seznamem součástek a očekávaných množství.

FR-16 Po ukončení inventury systém provede korekci zásob.

2.1.6 Reporty

FR-17 Přehled aktuálních zásob s filtrací dle kategorie/umístění.

FR-18 Historie pohybů za zvolené časové období.

FR-19 Součástky pod minimálním limitem zásob.

FR-20 Ocenění skladu (celková hodnota zásob).

2.1.7 Správa uživatelů

FR-21 Systém spravuje uživatele s rolemi: `ADMIN`, `SKLADNÍK`, `ČTENÁŘ`.

FR-22 Každá akce je logována s časovým razítkem a uživatelem.

2.2 Nefunkční požadavky

Tabulka 2.1: Nefunkční požadavky systému

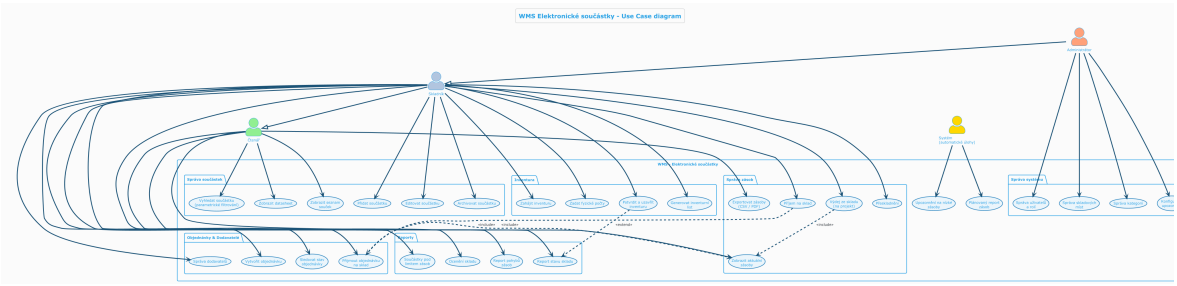
ID	Kategorie	Požadavek
NFR-01	Výkon	Vyhledávání vrátí výsledky do 500 ms pro zásoby do 100 000 položek.
NFR-02	Dostupnost	Systém musí být dostupný 99,5 % času (plánovaná údržba vyjmuta).
NFR-03	Bezpečnost	Hesla uložena jako bcrypt hash; komunikace pouze přes HTTPS.
NFR-04	Škálovatelnost	Architektura umožní horizontální škálování backendu.
NFR-05	Přenositelnost	Nasazení pomocí Docker Compose; nezávislé na OS.
NFR-06	Použitelnost	Webové rozhraní přístupné bez speciálního SW v moderním prohlížeči.
NFR-07	Lokalizace	Primární jazyk čeština; systém připraven na vícejazyčnost (i18n).
NFR-08	Auditovatelnost	Každá změna stavu skladu musí být dohledatelná v auditním logu.

2.3 Případy užití – přehled

Hlavní aktéři systému:

- **Skladník** – provádí příjem, výdej, přeskladnění a inventuru.
- **Administrátor** – spravuje uživatele, nastavení, kategorie a místa.
- **Čtenář** – prohlíží zásoby a reporty (pouze čtení).
- **Systém** – automatické upozornění na nízké zásoby, plánované reporty.

Podrobný Use Case diagram je zobrazen v Figure 2.1.



Obrázek 2.1: Use Case diagram systému WMS – elektronické součástky

Kapitola 3

Architektura systému

3.1 Přehled architektury

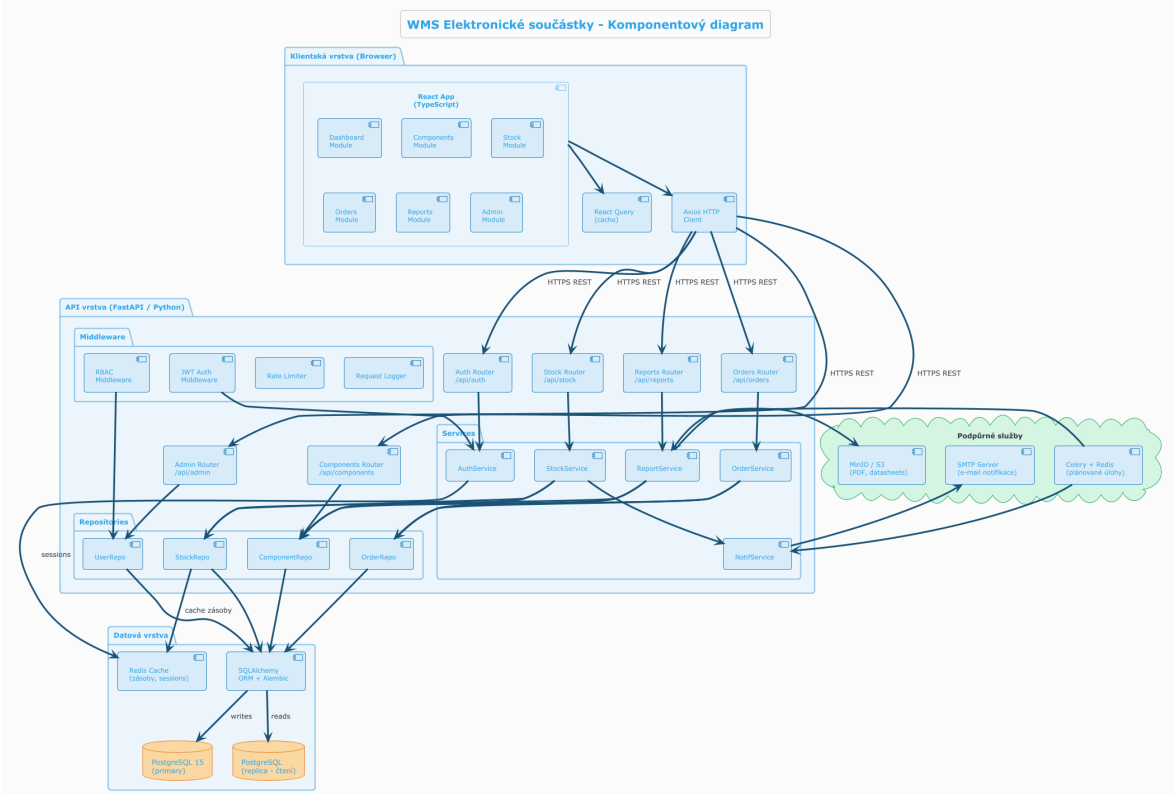
Systém je navržen jako třívrstvá webová aplikace s jasně oddělenými vrstvami:

Prezentační vrstva Webová aplikace napsaná v React + TypeScript, která komunikuje s backendem prostřednictvím REST API.

Aplikační vrstva Python backend s frameworkem FastAPI implementující business logiku, autentizaci (JWT) a validaci dat.

Datová vrstva Relační databáze správující persistenci dat; přístup přes ORM SQLAlchemy.

Komponentový diagram systému je zobrazen v Figure 3.1.



Obrázek 3.1: Komponentový diagram systému

3.2 REST API – přehled endpointů

Tabulka 3.1: Klíčové REST API endpointy

Metoda	Endpoint	Popis
GET	`/api/components`	Seznam součástek (s filtrací)
POST	`/api/components`	Nová součástka
GET	`/api/components/{id}`	Detail součástky
PUT	`/api/components/{id}`	Aktualizace součástky
DELETE	`/api/components/{id}`	Smazání součástky
GET	`/api/stock`	Aktuální zásoby
POST	`/api/stock/receive`	Příjem na sklad
POST	`/api/stock/issue`	Výdej ze skladu
POST	`/api/stock/transfer`	Přeskladnění
GET	`/api/locations`	Skladová místa
GET	`/api/suppliers`	Dodavatelé
GET	`/api/orders`	Objednávky
POST	`/api/orders`	Nová objednávka
PATCH	`/api/orders/{id}/status`	Změna stavu objednávky
GET	`/api/reports/stock`	Report stavu skladu
GET	`/api/reports/movements`	Report pohybů
GET	`/api/reports/low-stock`	Součástky pod limitem

3.3 Bezpečnostní model

Autentizace je realizována pomocí JWT tokenů (JSON Web Token):

1. Uživatel odešle přihlašovací údaje na `POST /api/auth/login`.
2. Server ověří heslo (bcrypt), vygeneruje access token (platnost 15 min) a refresh token (platnost 7 dní).
3. Klient přikládá access token v hlavičce `Authorization: Bearer {token}`.
4. Po vypršení access tokenu si klient vyžádá nový pomocí refresh tokenu.

Oprávnění jsou řešena pomocí RBAC (Role-Based Access Control):

Tabulka 3.2: Matice oprávnění dle rolí

Operace	**Čtenář**	**Skladník**	**Admin**
Prohlížení součástek	✓	✓	✓
Editace součástek	—	✓	✓
Příjem / Výdej	—	✓	✓
Inventura	—	✓	✓
Správa dodavatelů	—	✓	✓
Správa uživatelů	—	—	✓
Konfigurace systému	—	—	✓

Kapitola 4

Datový model

Datový model je zachycen platformově nezávislým konceptuálním diagramem tříd, který zachycuje doménové koncepty, generalizaci, vazební třídy a kvalifikovanou vazbu.

4.1 Přehled entit

Systém pracuje s následujícími hlavními entitami:

`Component` Elektronická součástka – základní katalogová informace.

`ComponentCategory` Kategorie součástek (rezistor, kondenzátor, ...).

`Location` Fyzické místo v skladovém prostoru.

`Stock` Zásoba – propojení součástky, místa a množství.

`StockMovement` Pohyb zásob (příjem, výdej, přeskladnění, korekce).

`Supplier` Dodavatel součástek.

`Order` Nákupní objednávka u dodavatele.

`OrderItem` Položka objednávky.

`Project` Projekt, na který jsou vydávány součástky.

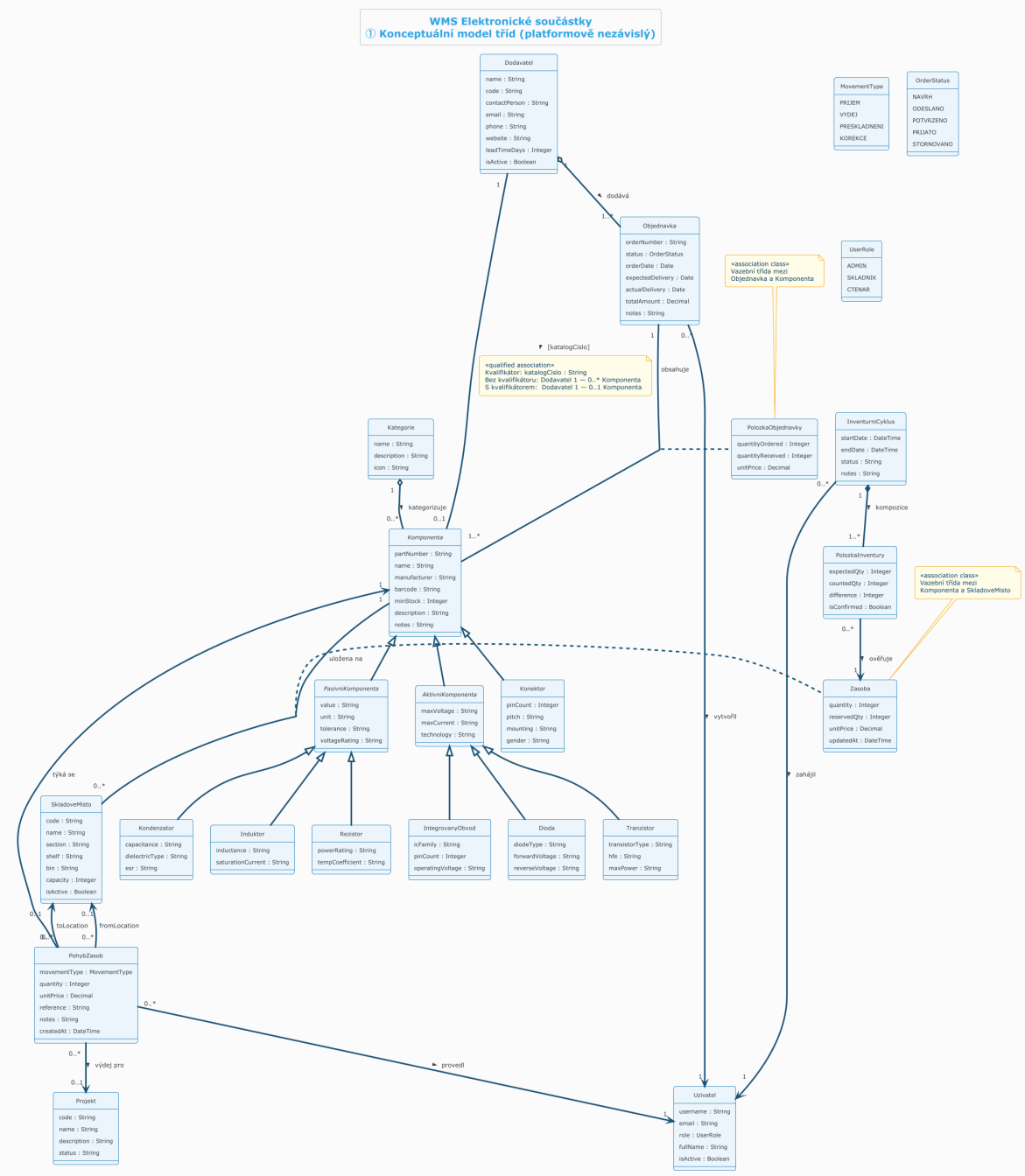
`User` Uživatel systému.

`AuditLog` Auditní záznam všech změn.

4.2 Konceptuální model tříd

Konceptuální model zachycuje:

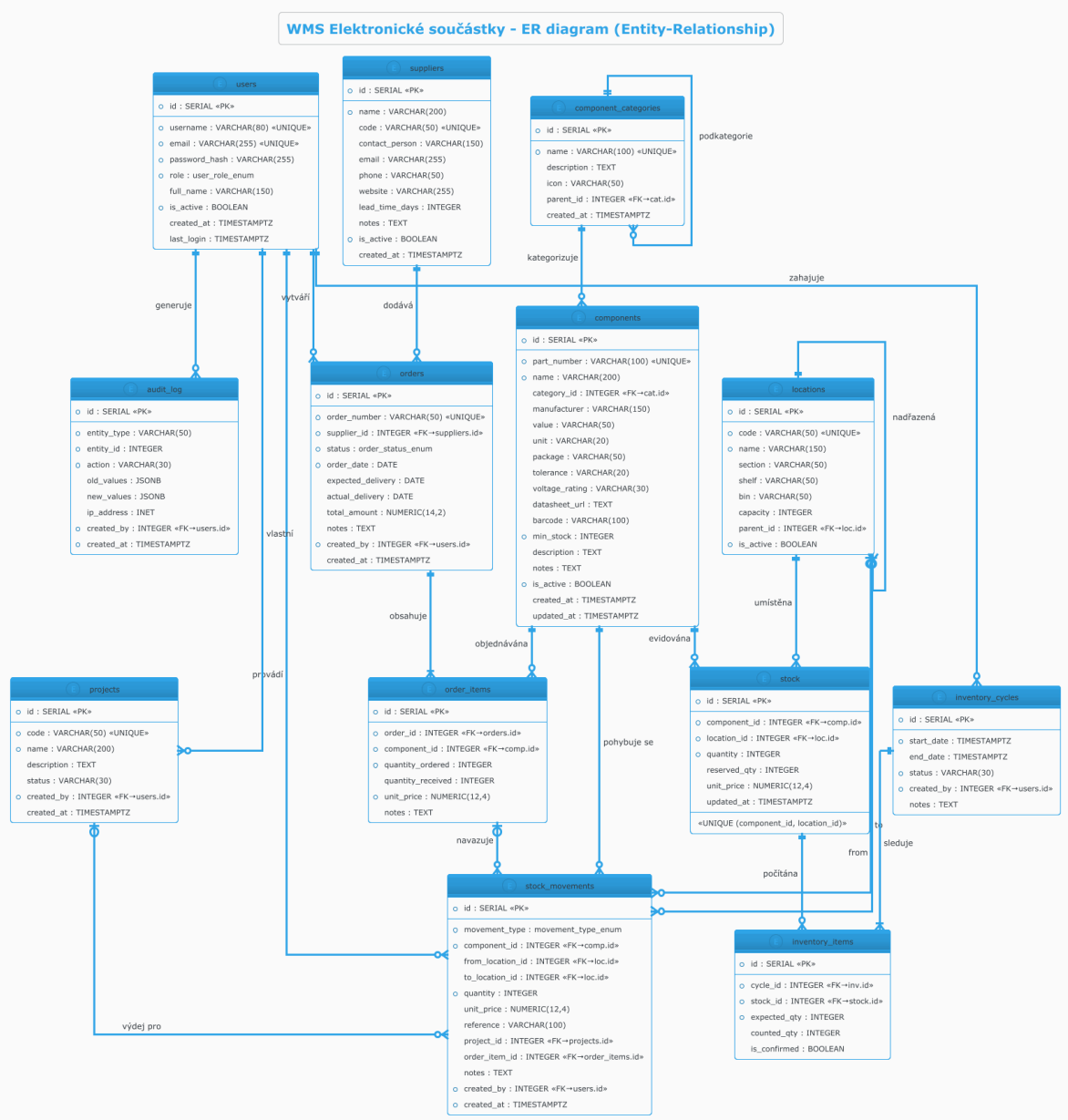
- **Generalizaci** – dvouúrovňová hierarchie: `Komponenta` → `PasivniKomponenta`, `AktivniKomponenta`, `Konektor` → `Rezistor`, `Kondenzator`, `Induktor`, ...
- **Vazební třídy** – `Zasoba` (mezi `Komponenta` a `SkladoveMisto`) a `PolozkaObjednavky` (mezi `Objednavka` a `Komponenta`).
- **Kvalifikovanou vazbu** – `Dodavatel[katalogCislo]` → `0..1 Komponenta`: pro daného dodavatele a jeho katalogové číslo existuje nejvýše jedna součástka. Bez kvalifikátoru by byla násobnost 0..∗.
- **Kompozici** – `InventurniCyklus` je složen z `PolozkaInventury` (existenční závislost).



Obrázek 4.1: Konceptuální model tříd

4.3 ER diagram

Kompletní ER diagram datového modelu je zobrazen na Figure 4.2.



Obrázek 4.2: ER diagram datového modelu

4.4 Popis tabulek

4.4.1 Tabulka `component_categories`

Tabulka 4.1: Tabulka component_categories

Sloupec	Typ	Popis
id	SERIAL PK	Primární klíč
name	VARCHAR(100) UNIQUE	Název kategorie (Rezistor, IC, ...)
description	TEXT	Popis kategorie
parent_id	INTEGER FK	Nadřazená kategorie (stromová struktura)
icon	VARCHAR(50)	Název ikony pro UI
created_at	TIMESTAMPTZ	Čas vytvoření

4.4.2 Tabulka `components`

Tabulka 4.2: Tabulka components

Sloupec	Typ	Popis
id	SERIAL PK	Primární klíč
part_number	VARCHAR(100) UNIQUE	Katalogové číslo výrobce
name	VARCHAR(200)	Název součástky
category_id	INTEGER FK	Kategorie
manufacturer	VARCHAR(150)	Výrobce
value	VARCHAR(50)	Hodnota (10k, 100nF, ...)
unit	VARCHAR(20)	Jednotka (Ω, F, H, V, A, ...)
package	VARCHAR(50)	Pouzdro (SMD 0402, DIP-8, TO-92, ...)
tolerance	VARCHAR(20)	Tolerance (±5 %, ±1 %, ...)
voltage_rating	VARCHAR(30)	Maximální napětí
datasheet_url	TEXT	URL na datasheet
barcode	VARCHAR(100)	Čárový kód EAN/QR
min_stock	INTEGER DEFAULT 0	Minimální zásoba
description	TEXT	Popis
notes	TEXT	Interní poznámky
is_active	BOOLEAN DEFAULT TRUE	Aktivní / archivována
created_at	TIMESTAMPTZ	Čas vytvoření
updated_at	TIMESTAMPTZ	Čas poslední změny

4.4.3 Tabulka `locations`

Tabulka 4.3: Tabulka locations

Sloupec	**Typ**	**Popis**
id	SERIAL PK	Primární klíč
code	VARCHAR(50) UNIQUE	Unikátní kód místa (A-01-03)
name	VARCHAR(150)	Popis místa (Regál A, přihrádka 3)
section	VARCHAR(50)	Sekce skladu
shelf	VARCHAR(50)	Regál
bin	VARCHAR(50)	Přihrádka / zásobník
parent_id	INTEGER FK	Nadřazené místo
capacity	INTEGER	Kapacita (počet zásobníků)
is_active	BOOLEAN DEFAULT TRUE	Aktivní

4.4.4 Tabulka `stock`

Tabulka 4.4: Tabulka stock (zásoby)

Sloupec	**Typ**	**Popis**
id	SERIAL PK	Primární klíč
component_id	INTEGER FK	Odkaz na součástku
location_id	INTEGER FK	Odkaz na místo
quantity	INTEGER NOT NULL	Aktuální množství
reserved_qty	INTEGER DEFAULT 0	Rezervované množství
unit_price	NUMERIC(12,4)	Pořizovací cena za kus
updated_at	TIMESTAMPTZ	Čas poslední aktualizace
UNIQUE (component_id, location_id)		

4.4.5 Tabulka `stock_movements`

Tabulka 4.5: Tabulka stock_movements (pohyby zásob)

Sloupec	**Typ**	**Popis**
id	SERIAL PK	Primární klíč
movement_type	movement_type_enum	RECEIPT / ISSUE / TRANSFER / ADJUSTMENT
component_id	INTEGER FK	Součástka
from_location_id	INTEGER FK	Zdrojové místo (NULL pro RECEIPT)
to_location_id	INTEGER FK	Cílové místo (NULL pro ISSUE)
quantity	INTEGER NOT NULL	Množství pohybu
unit_price	NUMERIC(12,4)	Cena za kus
reference	VARCHAR(100)	Číslo dokladu (faktura, projekt, ...)
project_id	INTEGER FK	Projekt (pro ISSUE)
order_item_id	INTEGER FK	Položka objednávky (pro RECEIPT)
notes	TEXT	Poznámka
created_by	INTEGER FK	Uživatel
created_at	TIMESTAMPTZ	Čas pohybu

Kapitola 5

Formální specifikace dotazů

Tato kapitola zavádí formální notaci pro popis dotazů nad datovým modelem WMS. Notace je inspirována lambda kalkulem a množinovou algebrou a je platformově nezávislá.

5.1 Definice notace

Universum. $\mathcal{U}$ označuje množinu všech uzlů (objektů) v systému. Množiny jsou reprezentovány jako charakteristické funkce; $x \in A$ je zkrácený zápis za $(A\ x)$.

Selekce (filtr). Infixový operátor // filtruje množinu:

$$A \mathbin{//} \Lambda \ \coloneqq\ \{\, x \mid (x \in A) \wedge (\Lambda\, x) \,\}$$

Transformace (map). Infixový operátor $\gg$ mapuje množinu:

$$A \gg f \ \coloneqq\ \{\, f(x) \mid x \in A \,\}$$

Flatten. Funkce flatten sloučí množinu množin:

$$\text{flatten}(M) \ =\ \bigcup M$$

Skládání do struktury. Infixový operátor $\oplus$ vytváří strukturu z hodnot:

$$a \oplus b \ \coloneqq\ \text{Struct}(a,\, b)$$

Atributy. Tečková notace je zkratkou pro funkční zápis: $s.\text{věk} \coloneqq \text{věk}(s)$.

Orientovaná hrana. $x \xrightarrow{\text{Relace}} y$ znamená $(x, y) \in \text{Relace}$.

5.2 Příklady dotazů nad modelem WMS

5.2.1 Součástky pod minimálním stavem

Množina součástek, jejichž celkový stav zásob klesl pod minimum:

$$K_{\min} \;=\; \text{Komponenty} \,//\, \Big(\lambda k \mid k.totalStock < k.minStock\Big)$$

kde $k.totalStock$ je agregovaná hodnota:

$$k.totalStock \;\coloneqq \sum_{z\,:\,k \xrightarrow{\text{ULOZENA\_NA}} z} z.quantity$$

Výstupní projekce (jméno, číslo, stav):

$$K_{\min} \gg \Big(\lambda k \mid k.name \oplus k.partNumber \oplus k.totalStock\Big)$$

5.2.2 Celková hodnota skladu

Suma hodnot všech zásob:

$$HodnotaSkladu \;=\; \sum_{z \in \text{Zásoby}} z.quantity \times z.unitPrice$$

5.2.3 Katalog součástek dodavatele

Všechny součástky nabízené konkrétním dodavatelem d_0:

$$\text{flatten}\Big(\big\{\, d_0 \,\big\} \gg \Big(\lambda d \mid \big\{\, k \mid d \xrightarrow{\text{DODAVANA\_OD}} k \big\}\Big)\Big)$$

5.2.4 Pohyby výdeje na projekt

Pohyby typu `VYDEJ` navázané na projekt p_0, provedené za dané časové období $[t_1, t_2]$:

$$\text{Pohyby} // \Big(\lambda m \mid m.movementType = \texttt{VYDEJ} \wedge m \xrightarrow{\text{VYDANA\_NA}} p_0 \wedge t_1 \le m.createdAt \le t_2\Big)$$

Výsledná projekce (součástka, množství, datum):

$$\ldots \gg \Big(\lambda m \mid m.component \oplus m.quantity \oplus m.createdAt\Big)$$

5.2.5 Složité dotazy – inventurní rozdíly

Položky inventury, kde se zjištěný počet liší od evidovaného (schválené i neschválené):

$$\text{PolozkyInventury} // \Big(\lambda p \mid p.countedQty \neq p.expectedQty\Big) \gg \Big(\lambda p \mid p.component \oplus p.expectedQty \oplus p.$$

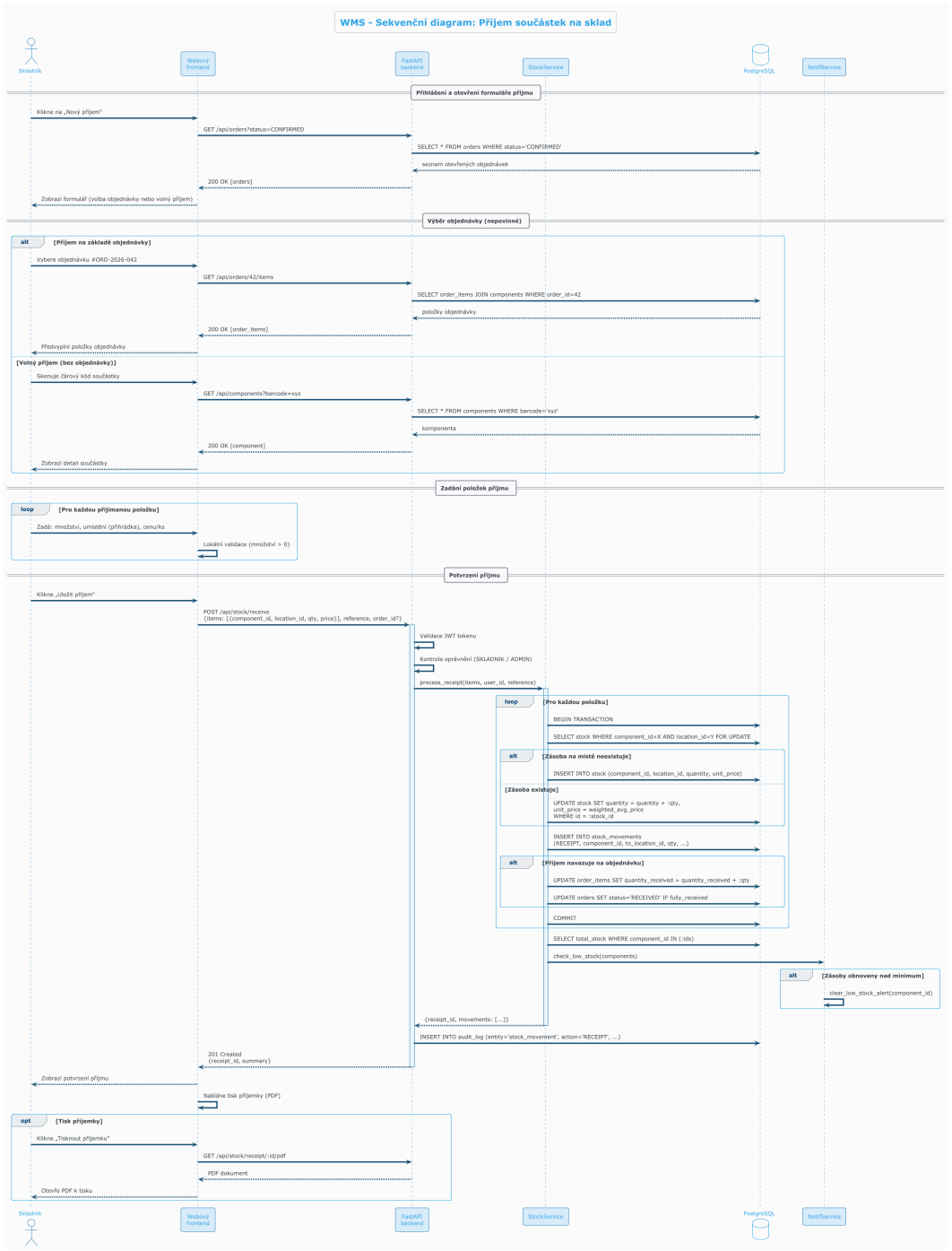
Kapitola 6

Popis procesů

6.1 Proces příjmu součástek

Příjem součástek na sklad je jedním z nejčastějších procesů. Probíhá typicky na základě dodávky od dodavatele, přičemž může, ale nemusí, navazovat na předchozí objednávku.

Sekvenční diagram příjmu je zobrazen na Figure 6.1.



Obrázek 6.1: Sekvenční diagram – Příjem součástek na sklad

6.1.1 Popis kroků příjmu

1. Skladník otevře formulář příjmu a naskenuje/zadá číslo dokladu nebo vyhledá objednávku.

2. Pro každou přijímanou položku:

   (a) Vyhledá součástku (dle part number nebo skenování čárového kódu).

   (b) Určí cílové skladové místo.

   (c) Zadá přijímané množství a jednotkovou cenu.

3. Systém ověří validitu dat (množství > 0, místo existuje, součástka aktivní).

4. Systém vytvoří záznamy v `stock_movements` a aktualizuje tabulku `stock`.

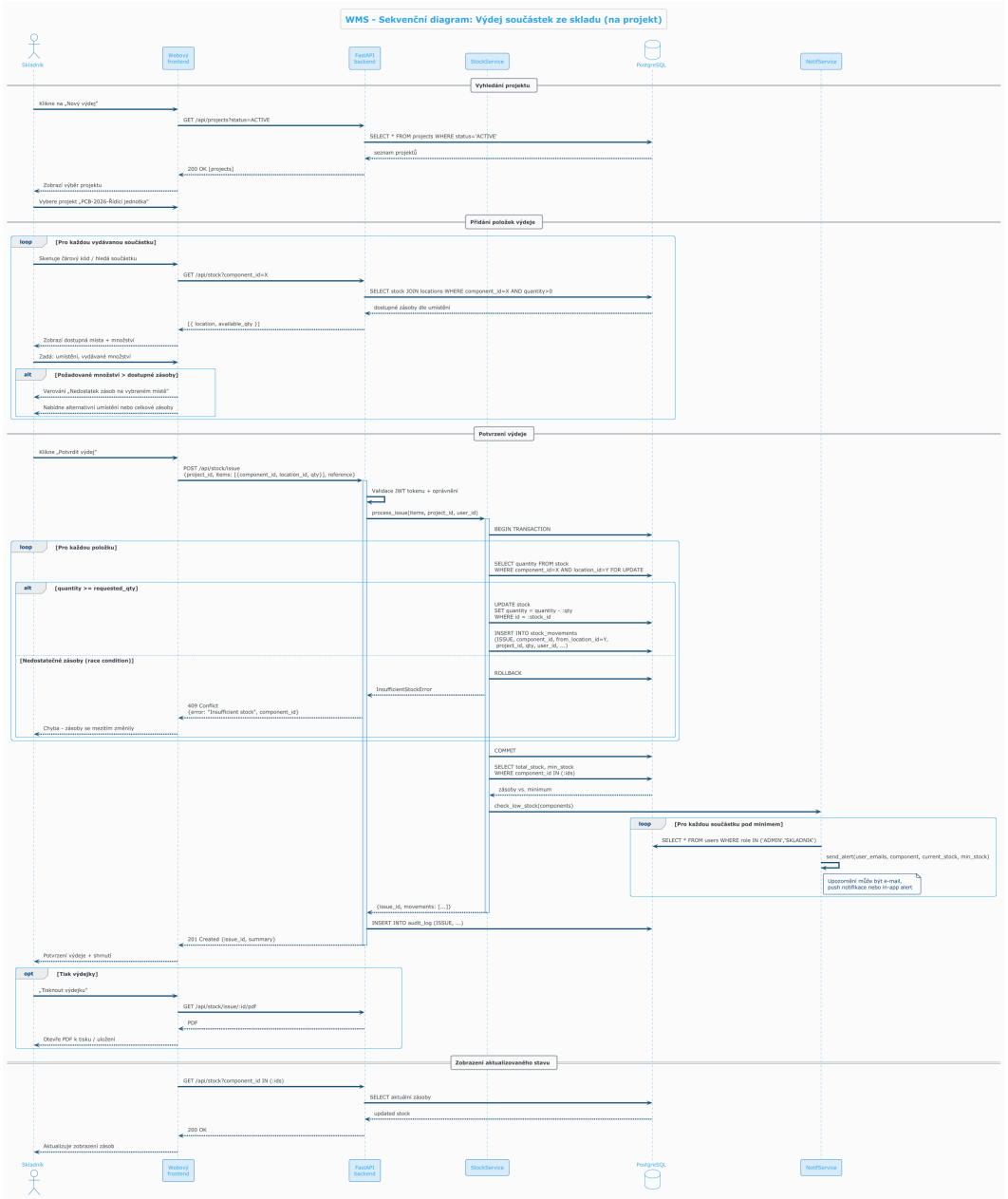
5. Pokud příjem navazuje na objednávku, systém aktualizuje stav objednávky.

6. Systém zkontroluje, zda příjem neobnovil zásoby nad minimum (zruší upozornění).

7. Vytiskne se příjemka.

6.2 Proces výdeje součástek

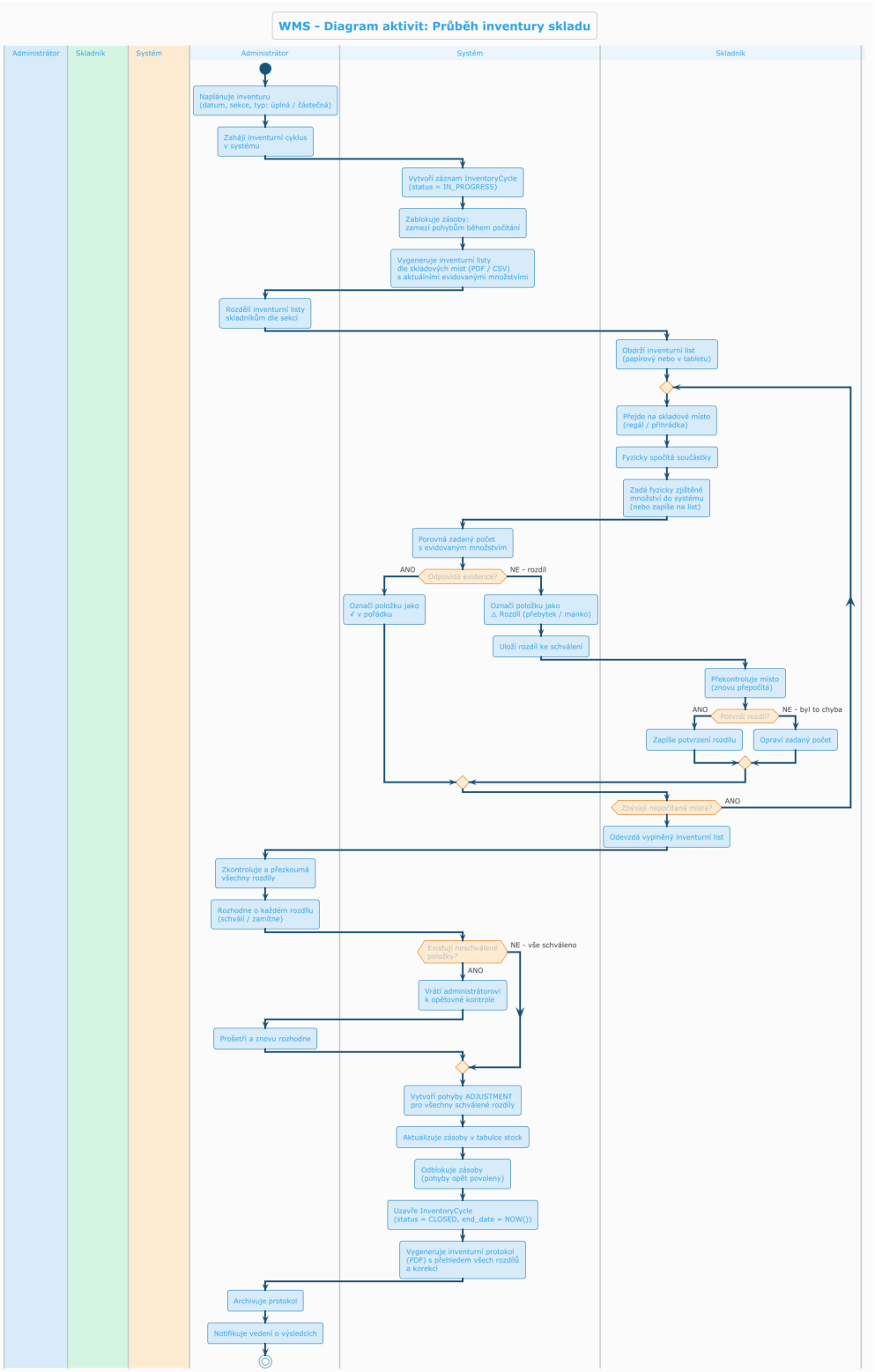
Výdej se provádí při přidělení součástek konkrétnímu projektu nebo opravě. Sekvenční diagram je na Figure 6.2.



Obrázek 6.2: Sekvenční diagram – Výdej součástek ze skladu

6.3 Proces inventury

Inventura je periodický proces ověření fyzického stavu skladu. Diagram aktivit je zobrazen na Figure 6.3.



Obrázek 6.3: Diagram aktivit – Průběh inventury

6.3.1 Fáze inventurního procesu

Příprava Administrátor zahájí inventurní cyklus. Systém uzamkne zásoby pro editaci a vygeneruje inventurní listy dle skladových míst.

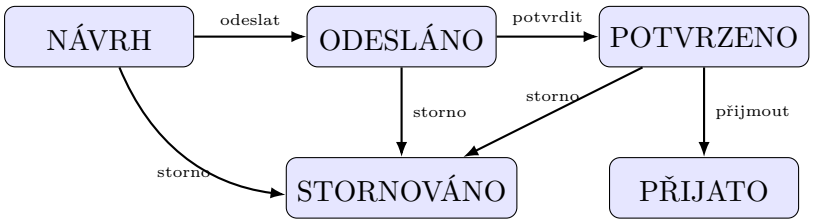
Fyzické sčítání Skladník prochází místa a zadává fyzicky zjištěné množství. Systém umožňuje zadávání postupně po sekcích.

Porovnání Systém porovná evidované a fyzicky zjištěné stavy. Zobrazí seznam rozdílů (přebytky / manka).

Schválení a korekce Administrátor přezkoumá rozdíly. Po schválení systém vytvoří pohyby typu `ADJUSTMENT` a aktualizuje zásoby.

Uzavření Inventura je uzavřena, zásoby odemčeny, vygenerován protokol.

6.4 Stav objednávky – stavový diagram



Obrázek 6.4: Stavový diagram objednávky

Kapitola 7

Závěr

7.1 Shrnutí

Tato dokumentace popsala návrh systému pro správu skladu elektronických součástek (WMS). Systém řeší reálnou potřebu přehledné a efektivní evidence zásob v prostředí, kde se pracuje s tisíci různých komponent.

Klíčové výstupy projektu:

- Analýza 22 funkčních a 8 nefunkčních požadavků.
- Navžena třívrstvá architektura (React + FastAPI + relační DB).
- Datový model s 11 hlavními entitami pokrývající kompletní lifecycle součástky v skladu.
- 7 UML/PlantUML diagramů dokumentujících use cases, konceptuální model, ER diagram, sekvenční a aktivitní procesy a komponentovou architekturu.
- REST API s více než 15 endpointy.

7.2 Další rozvoj

Plánovaná rozšíření pro budoucí verze systému:

v1.1 Mobilní aplikace pro skenování QR kódů při příjmu a výdeji.

v1.2 Integrace s online obchody (TME, Farnell, Mouser) pro automatické doplnění technických parametrů a cen.

v1.3 BOM (Bill of Materials) management – propojení WMS s návrhem DPS pro automatické ověření dostupnosti součástek.

v2.0 Multi-sklady a multi-uživatelský přístup s pokročilými právy. Notifikace přes e-mail a Teams/Slack.

7.3 Diagram tříd – rozšiřitelnost

Navržená architektura je připravena na budoucí rozšíření díky použití abstraktních bázových tříd a dependency injection v backendu, což umožní přidávat nové typy pohybů a reportů bez zásahu do existujících modulů.

Příloha A

Slovník pojmů a zkratek

WMS Warehouse Management System – Systém pro správu skladu

EAN European Article Number – čárový kód

QR Quick Response – 2D čárový kód

IC Integrated Circuit – integrovaný obvod

SMD Surface-Mount Device – povrchová montáž

THT Through-Hole Technology – průchozí montáž

REST Representational State Transfer – architektonický styl API

CRUD Create, Read, Update, Delete – základní databázové operace

ER Entity-Relationship – diagram datových entit a vztahů

UML Unified Modeling Language – unifikovaný modelovací jazyk

Příloha B

Instalace a konfigurace

Požadavky na prostředí

- Python 3.11+, Node.js 20+ nebo Java 17+
- PostgreSQL 15+
- Docker (volitelně)
- PlantUML JAR pro regeneraci diagramů

Generování diagramů z PlantUML

Listing B.1: Generovani PNG diagramu z PUML souboru

```
# Instalace PlantUML (macOS)
brew install plantuml

# Generování vsech diagramu najednou
plantuml -tpng diagrams/*.puml -o ../docs/images/

# Nebo pomoci Makefile
make diagrams
```