# 1. Úvod

### 1.1 Účel dokumentu

Tento dokument specifikuje softwarové požadavky na skladový informační systém **Stockly**, jehož hlavním cílem je usnadnit a zefektivnit řízení skladových zásob ve firmách. Dokument slouží jako podklad pro vývoj, testování, nasazení i následnou správu systému. Poskytuje jasnou strukturu očekávané funkcionality, uživatelských scénářů a technologických i procesních specifikací, které mají být dodrženy během realizace projektu.

### 1.2 Cíl systému

Cílem systému Stockly je nabídnout jednoduché, intuitivní a zároveň robustní řešení pro firmy, které chtějí mít přehled nad svými zásobami napříč více sklady, minimalizovat chybovost při naskladnění a vyskladnění, a automatizovat rutinní činnosti skladníků i administrátorů.

Stockly je navržen jako **víceskladový systém**, který:

* **Podporuje přesnou evidenci zásob** na více fyzických místech
* **Umožňuje převody mezi sklady** s plnou sledovatelností pohybů
* **Automatizuje porovnávání skutečného naskladnění s očekávanými dodávkami**
* **Podporuje import a export dat**, např. z Excelu, což zjednodušuje integraci s dalšími systémy
* **Umožňuje řízení přístupu pomocí rolí**, čímž zajišťuje bezpečnost a přehlednost dat

### 1.3 Důraz kladený při návrhu systému

Při návrhu systému Stockly jsou kladeny důrazy na následující oblasti:

* **Uživatelská přívětivost**  
  Systém musí být jednoduchý na používání pro běžné zaměstnance (např. skladníky) bez nutnosti dlouhého školení. Rozhraní má být čisté, srozumitelné a funkčně oddělené podle typických úloh.
* **Automatizace a úspora práce**  
  Cílem není pouze digitalizovat stávající procesy, ale je aktivně zjednodušit a částečně automatizovat. Příkladem je automatická kontrola naskladnění podle očekávaných dodávek nebo chytrý import produktů z tabulkových souborů.
* **Modularita a rozšiřitelnost**  
  Systém je budován s ohledem na možnou budoucí rozšiřitelnost o další funkce jako jsou např. sledování šarží, expirací, nebo napojení na účetní systémy.
* **Bezpečnost a přístupová práva**  
  Díky víceúrovňové autentizaci a přiřazení práv podle rolí se zajišťuje, že každý uživatel vidí jen data, která má mít k dispozici, a může provádět jen ty operace, ke kterým má oprávnění.
* **Víceskladová architektura**  
  Od počátku je systém navržen tak, aby umožňoval správu více skladových lokalit, včetně převodů mezi nimi, samostatných zásob a uživatelů s přístupem k jednotlivým skladům.
* **Důraz na jednoduchou integraci**  
  Export/import dat, dobře strukturované API a přehledné datové toky umožní v budoucnu snadno propojit systém s ERP, e-shopy nebo dalšími nástroji.

### 1.4 Cílová skupina systému

Tento skladový systém je navržen s důrazem na jednoduchost, přehlednost a efektivitu. **Primárně je určen pro malé a střední firmy**, které potřebují udržovat pořádek ve svých zásobách, aniž by musely investovat do robustních a složitých podnikových řešení.  
Cílem je nabídnout systém, který:

* nezatěžuje uživatele zbytečnými kroky,
* umožňuje rychlé naskladnění a vyskladnění zboží,
* podporuje více skladů a oprávnění uživatelů,
* zvládne importy produktů z Excelu,
* zůstává cenově dostupný a snadno nasaditelný.

# 2. Použité technologie

Systém **Stockly** je vyvíjen jako webová aplikace s důrazem na jednoduchost, přehlednost a možnost budoucího rozšíření. Architektura systému je rozdělena na backendovou a frontendovou část, přičemž pro databázové služby a autentizaci je využívána platforma **Supabase**.

### 2.1 Backend

Backend je tvořen pomocí frameworku **NestJS** s využitím **Prisma ORM** pro komunikaci s databází. NestJS umožňuje přehlednou a modulární architekturu, která je vhodná pro větší aplikace, jako je systém pro správu skladových zásob.

**Použité technologie:**

* **NestJS** – framework pro tvorbu škálovatelných serverových aplikací v TypeScriptu
* **TypeScript** – typový nadstavbový jazyk nad JavaScriptem
* **Prisma ORM** – nástroj pro databázový přístup, migrace a validaci dat
* **Supabase** – poskytovatel databáze (PostgreSQL) a dalších backendových služeb, včetně autentizace
* **JWT (JSON Web Tokens)** – autentizace a správa přístupových práv uživatelů
* **Bcrypt** – hashování hesel (v případě vlastního auth řešení mimo Supabase)
* **class-validator** – validace vstupních dat

### 2.2 Frontend

Frontendová část je vyvíjena v **Reactu**, s důrazem na jednoduché a čisté uživatelské rozhraní bez složitých frameworků nebo knihoven. Stylování komponent je řešeno pomocí klasického **CSS**, bez použití utility frameworků jako Tailwind nebo Bootstrap.

**Použité technologie:**

* **React** – JavaScriptová knihovna pro tvorbu uživatelského rozhraní (SPA)
* **JavaScript / TypeScript** – dle preferencí vývojáře
* **CSS** – klasické kaskádové styly pro stylování komponent
* **Axios** – pro komunikaci s backendem přes REST API

### 2.3 Databáze

Databázová vrstva je plně spravována pomocí **Supabase**, která poskytuje PostgreSQL databázi jako službu. Přístup k databázi zajišťuje Prisma, která umožňuje definovat modely, pracovat s migracemi a generovat typově bezpečný kód.

**Použité technologie:**

* **Supabase PostgreSQL** – hlavní relační databáze systému
* **Prisma ORM** – nástroj pro datový přístup, generování modelů a migrací

### 2.4 Nasazení a vývoj

Projekt je vyvíjen s důrazem na jednoduché nasazení a správu.

**Technologie a nástroje:**

* **Git + GitHub** – správa verzí a zdrojového kódu
* **Supabase hosting** – pro databázi
* **Postman** – testování API endpointů

# 3. Funkční požadavky

Níže jsou uvedeny požadavky, které musí systém Stockly splňovat z hlediska funkčnosti. Jsou rozděleny dle jednotlivých modulů systému.

**3.1 Správa uživatelů**

* FR-01: Systém umožní registraci a přihlášení uživatelů.
* FR-02: Uživatel po registraci získá roli administrátora a může přidávat další uživatele.
* FR-03: Administrátor může nastavovat role ostatním uživatelům (např. skladník).
* FR-04: Uživatel má přístup pouze k funkcím odpovídajícím jeho roli.

**3.2 Správa skladů**

* FR-05: Systém umožňuje vytvořit více skladů.
* FR-06: Uživatel může přiřadit produkty ke konkrétnímu skladu.
* FR-07: Skladník vidí pouze sklady, ke kterým má přístup.

**3.3 Naskladnění a vyskladnění**

* FR-08: Uživatel může vytvořit záznam o naskladnění produktu (StockMovement).
* FR-09: Uživatel může vytvořit záznam o vyskladnění produktu.
* FR-10: Uživatel může naskladnit více produktů najednou.
* FR-11: Pohyb zboží je zaznamenáván v historii skladu.

**3.4 Očekávané dodávky (přednaskladnění)**

* FR-12: Uživatel může založit záznam o očekávané dodávce (např. dle dodcého listu).
* FR-13: Uživatel může potvrdit dodávku a tím ji převést na reálné naskladnění.
* FR-14: Systém porovná očekávané produkty s reálným naskladněním a upozorní na nesrovnalosti.

**3.5 Práce s daty**

* FR-15: Uživatel může importovat produkty k naskladnění z Excel souboru.
* FR-16: Při importu systém upozorní na produkty, které chybí v databázi.
* FR-17: Chybějící produkty lze doplnit manuálně ze seznamu po importu.
* FR-18: Systém umožňuje export dat o pohybech a skladových stavech.

# 4. Nefunkční požadavky

Tato část specifikuje požadavky, které nesouvisí přímo s funkcionalitou systému, ale mají vliv na jeho kvalitu, použitelnost a provoz.

**4.1 Výkon a odezva**

* NFR-01: Systém by měl reagovat na uživatelské akce do 1 sekundy ve většině případů.
* NFR-02: Import souboru do 500 řádků by měl být zpracován do 3 sekund.

**4.2 Dostupnost a spolehlivost**

* NFR-03: Systém musí být dostupný alespoň 99 % času během pracovních hodin.
* NFR-04: Data nesmí být ztracena při výpadku díky zálohám Supabase.

**4.3 Bezpečnost**

* NFR-05: Přístup k systému je chráněn autentizací přes Supabase.
* NFR-06: Citlivá data (např. hesla) musí být uložena bezpečně (např. hashování, HTTPS).
* NFR-07: Každý uživatel může přistupovat jen k datům, která odpovídají jeho oprávněním.

**4.4 Uživatelská přívětivost**

* NFR-08: Uživatelské rozhraní musí být intuitivní a srozumitelné i pro méně zkušené uživatele.
* NFR-09: Formuláře musí obsahovat validaci a uživatelsky srozumitelné chybové hlášky.

**4.5 Údržba a rozšiřitelnost**

* NFR-10: Backendový kód je psán modulárně (NestJS), aby umožnil snadné rozšiřování.
* NFR-11: Použité technologie (Supabase, Prisma, React) umožňují přenesení na vyšší hosting bez zásahu do jádra systému.

# 5. Zabezpečení a oprávnění

Systém využívá **JWT (JSON Web Token)** pro autentizaci a autorizaci uživatelů. Každý uživatel získá po přihlášení bezpečný token, který je následně využíván pro ověřování přístupu k chráněným částem systému.

**Role v systému:**

* **Admin**
  + má plný přístup ke všem funkcím systému
  + spravuje uživatele, sklady, produkty i zásoby
  + může upravovat role ostatních
* **Skladník**
  + má přístup pouze k operacím spojeným s pohybem zásob a očekávanými dodávkami
  + nemůže měnit nastavení systému, produkty ani spravovat jiné uživatele

**Princip přístupů:**

* Každé API má definováno, jaká role má oprávnění volat daný endpoint
* Oprávnění jsou validována na backendu při každém požadavku
* Rozhraní (frontend) zobrazuje pouze funkce, ke kterým má uživatel přístup

Tímto způsobem je zajištěno, že uživatelé nevidí ani nemohou ovlivnit data, která jim nepřísluší.

**Tabulka oprávnění rolí**

| **Funkce** | **Admin** | **Skladník** |
| --- | --- | --- |
| Zobrazit uživatele | ✔ | ✖ |
| Přidat / upravit uživatele | ✔ | ✖ |
| Správa skladů | ✔ | ✖ |
| Správa produktů | ✔ | ✖ |
| Import produktů z Excelu | ✔ | ✖ |
| Naskladnit zboží | ✔ | ✔ |
| Vyskladnit zboží | ✔ | ✔ |
| Zobrazit zásoby | ✔ | ✔ |
| Zobrazit / potvrdit dodávku | ✔ | ✔ |
| Filtrovat pohyby zásob | ✔ | ✖ |

Obsah obrázku snímek obrazovky, černobílá, černá, kruh

Obsah vygenerovaný umělou inteligencí může být nesprávný.

# 6. Popis uživatelského rozhraní

Uživatelské rozhraní systému **Stockly** je navrženo s důrazem na přehlednost a jednoduchost. Všechny funkce jsou rozděleny do logických celků podle rolí a jejich oprávnění. Rozhraní je responzivní, přístupné z desktopových i mobilních zařízení, a staví na známých interakčních vzorech.

## 6.1 Hlavní části rozhraní

### 6.1.1 Přihlášení a registrace

* Uživatel zadává e-mail a heslo.
* Pokud ještě není registrován, může být účet založen (první uživatel se stává administrátorem).
* Po přihlášení je uživatel přesměrován na dashboard odpovídající své roli.

### 6.1.2 Dashboard

* Shrnutí základních informací (např. počet skladů, poslední pohyby).
* Pro administrátora přehled uživatelů a správa přístupů.
* Pro skladníka přímý vstup do naskladnění/vyskladnění.

### 6.1.3 Správa skladů

* Seznam všech skladů, možnost vytvořit nový.
* Přístup ke konkrétnímu skladu umožňuje zobrazit jeho aktuální stav a historii pohybů.
* Možnost filtrovat podle data, typu pohybu nebo produktu.

### 6.1.4 Naskladnění / vyskladnění

* Formulář pro vytvoření nového pohybu s výběrem:
  + typu pohybu (naskladnění/vyskladnění),
  + skladu,
  + produktů (možnost přidat více najednou),
  + poznámky.
* Možnost importu položek z Excelu.

### 6.1.5 Očekávané dodávky

* Sekce pro správu přijatých objednávek nebo faktur.
* Zobrazení očekávaných položek, možnost jejich potvrzení jako reálné naskladnění.
* Funkce porovnání s reálně dodanými položkami.

### 6.1.6 Správa produktů

* Seznam všech produktů v systému.
* Vyhledávání, filtrování a úprava produktů.
* Možnost manuálního přidání chybějícího produktu (např. při importu z Excelu).

### 6.1.7 Uživatelské role a oprávnění

* Administrátor může:
  + přidávat uživatele,
  + přiřazovat role (admin, skladník),
  + spravovat přístup ke konkrétním skladům.
* Běžný uživatel (např. skladník) vidí jen data, ke kterým má přístup.

### 6.2 Design a styl

* Použit je jednoduchý, konzistentní design bez použití frameworků jako Bootstrap nebo Tailwind.
* Barevnost je neutrální, s jasně oddělenými akcemi (např. potvrzení vs. zrušení).
* Formuláře obsahují validaci, chybové hlášky jsou přehledné a srozumitelné.

### 6.3 Navigace

* Navigační menu je umístěno vlevo (nebo nahoře na mobilu).
* Dostupné položky menu závisí na roli uživatele.
* Každá akce je potvrzena vizuální zpětnou vazbou (např. notifikace po uložení).

# 7. Datový model

Databáze systému je navržena relačně a běží na platformě **Supabase (PostgreSQL)**. Pro práci s daty na backendu se používá **Prisma ORM**. Datový model podporuje více skladů, různé uživatelské role a záznam všech pohybů zásob.

Níže je stručný přehled hlavních entit a jejich vztahů:

* **User**  
  Uchovává informace o uživatelích systému, jejich e-mailu, roli a přístupových právech.
* **Storage**  
  Reprezentuje jednotlivé sklady. Každý sklad má vlastní správce a evidenci pohybů zásob.
* **Product**  
  Obsahuje informace o produktech (název, kód, jednotky). Produkty mohou být použity ve více skladech a pohybech.
* **StockMovement**  
  Zaznamenává naskladnění a vyskladnění. Obsahuje typ pohybu, sklad a uživatele, který pohyb provedl.
* **StockTransaction**  
  Vztah mezi pohybem a konkrétními produkty – obsahuje množství a produktové položky.
* **ExpectedDelivery**  
  Uchovává informace o plánovaných naskladněních (např. z objednávek nebo faktur). Slouží k následnému potvrzení a porovnání s reálným pohybem.

**Vztahy mezi entitami**

* Jeden **uživatel** může mít přístup k více **skladům**.
* Každý **sklad** má vlastní **produkty** a **pohyby zásob**.
* Jeden **pohyb zásob (StockMovement)** může obsahovat více položek (**StockTransaction**).
* **Očekávané dodávky** lze převádět do reálných pohybů.

Obsah obrázku snímek obrazovky, Obdélník, design

Obsah vygenerovaný umělou inteligencí může být nesprávný.

# 8. Přehled hlavních API funkcí

Systém poskytuje REST API pro komunikaci mezi frontendem a backendem. API je navrženo modulárně podle jednotlivých oblastí systému. Uživatelé musí být ověřeni pomocí JWT tokenu. Níže je uveden základní přehled funkcionalit:

**Autentizace a správa uživatelů**

* POST /auth/login – přihlášení uživatele
* POST /auth/register – registrace prvního admina (nebo ručně přes Supabase)
* GET /users – seznam uživatelů
* POST /users – vytvoření uživatele s danou rolí a přístupem ke skladu

**Sklady**

* GET /warehouses – seznam skladů dostupných uživateli
* POST /warehouses – vytvoření nového skladu
* PUT /warehouses/:id – úprava skladu
* DELETE /warehouses/:id – odstranění skladu (soft-delete)

**Produkty**

* GET /products/by-company – seznam produktů
* POST /products – přidání nového produktu
* PUT /products/:id – úprava produktu
* DELETE /products/:id – smazání produktu

**Pohyby zásob**

* GET /movements/company – výpis všech pohybů
* POST /movements – vytvoření nového naskladnění/vyskladnění
* GET /movements/:id – detail pohybu
* DELETE /movements/:id – zrušení pohybu

**Očekávané dodávky**

* GET /expected-delivery – výpis očekávaných dodávek
* POST /expected-delivery – vytvoření nové dodávky (např. na základě faktury)

### Seznam zkratek a pojmů

| **Zkratka / pojem** | **Význam** |
| --- | --- |
| **SRS** | *Software Requirements Specification* – specifikace požadavků na systém |
| **API** | *Application Programming Interface* – rozhraní pro komunikaci mezi systémy |
| **JWT** | *JSON Web Token* – bezpečný způsob autentizace uživatelů |
| **CRUD** | *Create, Read, Update, Delete* – základní operace nad daty |
| **BPMN** | *Business Process Model and Notation* – standard pro modelování procesů |
| **UI** | *User Interface* – uživatelské rozhraní |
| **UX** | *User Experience* – uživatelský zážitek |
| **CSV / XLSX** | Formáty tabulkových dat, často používané pro import a export |

# 9. Závěr

Navržený skladový systém je vytvořen s důrazem na jednoduchost, přehlednost a maximální automatizaci běžných skladových operací. Umožňuje efektivní správu více skladů, snadnou evidenci produktů, práci s očekávanými dodávkami a kontrolu pohybu zásob. Díky intuitivnímu rozhraní a jasnému rozdělení rolí je systém snadno použitelný jak pro administrátory, tak pro běžné skladníky.

**Přínosy systému**

* **Podpora více skladů** – snadná organizace a filtrování zásob dle lokace.
* **Přehledný uživatelský interface** – jednoduché ovládání i pro méně technicky zdatné uživatele.
* **Import produktů z Excelu** – šetří čas při zakládání a úpravách produktů.
* **Modul pro očekávané dodávky** – umožňuje plánování a potvrzování naskladnění.
* **Zabezpečení přístupu dle rolí** – ochrana citlivých funkcí pomocí rolí (admin/skladník).

**Možnosti budoucího rozšíření**

* **Notifikace** – upozornění na očekávané dodávky, nízký stav zásob, nebo neaktivitu uživatelů.
* **Historie změn** – záznam o tom, kdo a kdy provedl jakou úpravu (tzv. audit log).
* **Mobilní verze / offline režim** – pro skladníky, kteří nejsou vždy u počítače.
* **Napojení na externí systémy** – propojení s účetnictvím nebo e-shopem (např. API integrace).
* **Pokročilé statistiky a grafy** – pro lepší rozhodování na základě dat.
* **Více úrovní oprávnění** – možnost jemnější správy přístupů podle konkrétních funkcí.