**SRS (Software Requirements Specification) pro projekt**

**1. Úvod**

Tento dokument popisuje technické požadavky a specifikace pro webovou aplikaci pro správu skladového systému. Aplikace umožňuje správu produktů, kategorií, skladů, dodavatelů a pozic, naskladňování a vyskladňování produktů, a historii pohybů zásob. Systém bude obsahovat administrativní rozhraní pro správu a uživatelské rozhraní pro zaměstnance.

**2. Technický popis řešení**

Aplikace je postavena na moderním webovém stacku s cílem zajistit vysokou dostupnost a bezpečnost. Využívá technologii serveru, databázi pro uložení dat a front-end framework pro interaktivní uživatelské rozhraní.

**2.1. Použité technologie**

* **Backend**:
  + **Node.js** s frameworkem **Express.js** pro zajištění rychlého zpracování požadavků a flexibilní struktury.
  + **JWT (JSON Web Token)** pro autentifikaci a správu uživatelských relací.
  + **MySQL** nebo **MongoDB** pro správu databáze a uložení dat o uživatelských účtech, produktech, skladech a historických pohybech.
* **Frontend**:
  + **HTML5**, **CSS3**, a **JavaScript** pro tvorbu uživatelského rozhraní.
  + **React** nebo **Vue.js** pro interaktivní komponenty, jako je autocomplete vyhledávání produktů a dynamické formuláře pro naskladnění a vyskladnění.
* **Autentifikace a autorizace**:
  + **JWT** pro bezpečné přihlášení a autentifikaci uživatelů.
  + Role-based access control (RBAC) pro správu práv přístupu (admin, zaměstnanec).
* **Další nástroje**:
  + **Docker** pro snadné nasazení aplikace v izolovaných kontejnerech.
  + **Git** pro verzování kódu a spolupráci při vývoji.

**2.1. Definice, Akronymy a Zkratky**

**RESTful API**

* **RESTful API** (Representational State Transfer) je architektonický styl pro návrh síťových aplikací, který využívá HTTP metody (GET, POST, PUT, DELETE) pro práci s daty. Tento styl je zaměřen na zjednodušení a škálovatelnost. API tohoto systému je navrženo jako RESTful, což znamená, že všechny operace nad daty (jako získání, přidání, úprava a smazání) jsou prováděny pomocí standardních HTTP metod.

**API**

* **API** (Application Programming Interface) je rozhraní, které umožňuje vzájemnou komunikaci mezi softwarem a různými systémy.

**JWT**

* **JWT** (JSON Web Token) je standard pro bezpečnou výměnu informací mezi klientem a serverem prostřednictvím tokenu, který obsahuje zakódované informace o uživatelských datech.

**RBAC**

* **RBAC** (Role-Based Access Control) je přístupový model, který zajišťuje, že uživatelé mají přístup pouze k těm funkcím, které odpovídají jejich rolím v systému.

**React**

* **React** je knihovna pro tvorbu uživatelských rozhraní, která umožňuje vývoj komponentových aplikací. Je populární pro svou jednoduchost, výkon a schopnost spravovat stav aplikace pomocí virtuálního DOM.

**Node.js**

* **Node.js** je open-source JavaScriptová platforma pro vývoj serverových aplikací. Umožňuje spuštění JavaScriptu na serveru a je oblíbená pro svou asynchronní povahu a škálovatelnost.

**Express**

* **Express** je minimalisticý framework pro Node.js, který usnadňuje tvorbu webových aplikací a API. Poskytuje různé nástroje pro správu směrování, middleware a šablonování.

**PostgreSQL**

* **PostgreSQL** je open-source relační databázový systém, který je známý svou robustností, výkonem a podporou komplexních datových typů a operací. Používá se pro ukládání dat a poskytuje silnou integritu a spolehlivost.

**Supabase**

* **Supabase** je open-source alternativa k Firebase, která poskytuje backend jako službu. Nabízí funkcionalitu pro správu databáze (PostgreSQL), autentifikaci a real-time funkce.

**JWT (JSON Web Token)**

* **JWT** je standard pro bezpečnou výměnu informací mezi klientem a serverem prostřednictvím tokenu, který obsahuje zakódované informace o uživatelských datech.

**Prisma**

* **Prisma** je ORM (Object-Relational Mapping) nástroj pro práci s databázemi v aplikacích napsaných v JavaScriptu nebo TypeScriptu. Umožňuje jednoduchý a efektivní přístup k databázi pomocí generování typově bezpečného kódu pro databázové operace. Prisma podporuje různé databázové systémy, včetně PostgreSQL, MySQL a SQLite.

**3. Struktura kódu**

Struktura kódu bude organizována do několika klíčových modulů, které usnadní rozšiřitelnost, údržbu a testování aplikace. Hlavní složky aplikace:

* **/models**: Obsahuje definice databázových modelů pro produkty, kategorie, sklady, pozice, uživatele, atd.
* **/controllers**: Logika pro zpracování požadavků, validace dat a interakce s databází.
* **/routes**: Definuje směrování HTTP požadavků na příslušné kontrolery.
* **/frontend**: Soubory pro frontendové zobrazení a interaktivní uživatelské rozhraní (pokud se používá server-side rendering).
* **/uploads**: Statické soubory jako obrázky.
* **/middleware**: Middleware pro autentifikaci, logování a další pomocné funkce.
* **/services**: Obsahuje logiku pro externí služby nebo zpracování složitějších obchodních pravidel.

**4. Funkcionality systému**

**4.1. Uživatelské rozhraní**

**1. Registrace**

* Uživatel vyplní své údaje a při registraci se zároveň vytvoří firma. Po registraci obdrží aktivační odkaz na e-mail, na který klikne a následně bude přesměrován na stránku pro přihlášení.

**2. Přidávání zaměstnanců**

* Admin má možnost přidávat zaměstnance do systému. Po zadání e-mailu zaměstnance, companyId a storageId je zaměstnanci odeslán odkaz na registraci. Tento odkaz obsahuje již předvyplněné údaje o firmě a skladu. Zaměstnanec si pouze nastaví své heslo, aktivuje účet prostřednictvím e-mailu a může se následně přihlásit.

**3. Přihlášení**

* Uživatel zadá své přihlašovací údaje (e-mail a heslo). Po úspěšném přihlášení je vygenerován a uložen JWT token do cookies pro autentifikaci uživatele.

**4. Správa produktů, skladů a kategorií**

* Admin má plný přístup k přidávání, úpravám a mazání produktů, skladů, dodavatelů a kategorií. Zaměstnanci mají omezený přístup podle svého skladu. Administrátor může přidávat produkty, vyplňovat základní údaje a přidávat obrázky.

**5. Naskladnění a vyskladnění**

* Uživatel vybere produkt pomocí autocomplete inputu, poté zadá cenu a množství. Admin má možnost vybrat sklad, do kterého produkt naskladňuje, zatímco zaměstnanec má sklad automaticky přiřazen podle svého ID.

**6. Historie pohybů**

* Admin může vidět všechny pohyby produktů včetně typu akce (naskladnění/vyskladnění), data, uživatele, množství a skladu. Data lze filtrovat a hledat podle různých parametrů.

**4.2. Backend**

* **Autentifikace a autorizace**: Použití JWT tokenů pro bezpečné přihlášení a validaci uživatelských požadavků.
* **CRUD operace**: Admin může provádět operace pro vytváření, čtení, aktualizaci a mazání produktů, kategorií, skladů atd.
* **Skladové operace**: Systém bude mít logiku pro správu skladových pohybů (naskladnění/vyskladnění) a sledování dostupnosti produktů.
* **Historie pohybů**: Všechny akce budou zaznamenány a admin je bude moci prohlížet.

**4.3. Role uživatelů**

V aplikaci jsou definovány dvě hlavní role, které mají různý přístup k funkcím systému: **admin** a **zaměstnanec**.

**4.3.1. Admin**

Admin má plný přístup ke všem funkcím aplikace a může provádět jakékoliv operace na všech úrovních. Hlavní práva admina zahrnují:

* **Správa uživatelů**: Admin může vytvářet, editovat a mazat uživatele. Také může přiřadit uživatele k různým rolím.
* **Správa produktů**: Admin má možnost přidávat, editovat a mazat produkty ve všech skladech firmy. Má přístup k detailním informacím o produktech a může upravovat jejich ceny, množství a další atributy.
* **Správa skladů**: Admin může přidávat, editovat a mazat sklady. Může také přiřazovat produkty k různým skladům a definovat jejich množství na jednotlivých skladech.
* **Správa kategorií, pozic, dodavatelů**: Admin může spravovat kategorie produktů, pozice v rámci skladu a dodavatele.
* **Naskladnění a vyskladnění**: Admin může naskladnit a vyskladnit produkty do libovolného skladu. Má možnost vybírat sklad při naskladnění.
* **Historie pohybů**: Admin má přístup ke kompletní historii pohybů produktů (naskladnění, vyskladnění), včetně detailních informací o každé akci.
* **Přístup k reportům**: Admin může generovat reporty o stavu zásob a historických pohybech.

**4.3.2. Zaměstnanec**

Zaměstnanec má omezený přístup v porovnání s administrátorem. Hlavní práva zaměstnance zahrnují:

* **Zobrazení produktů**: Zaměstnanec může vidět produkty pouze v rámci svého přiřazeného skladu. Může si prohlížet produkty, jejich množství, ale nemůže je upravovat nebo mazat.
* **Naskladnění a vyskladnění**: Zaměstnanec může naskladnit nebo vyskladnit produkty, ale pouze v rámci svého skladu, který je automaticky přiřazen na základě jeho uživatelského ID. Nemá možnost vybírat jiný sklad.
* **Zobrazení historie pohybů**: Zaměstnanec má omezený přístup k historii pohybů, obvykle pouze pro svůj vlastní sklad a akce, které prováděl.
* **Přístup k vlastnímu účtu**: Zaměstnanec může aktualizovat pouze své vlastní údaje a nastavení, ale nemá právo měnit údaje ostatních uživatelů.

**5. Bezpečnost**

Aplikace bude zajištěna následujícími bezpečnostními opatřeními:

* **Šifrování hesel**: Hesla uživatelů budou šifrována pomocí algoritmu **bcrypt**.
* **JWT autentifikace**: Všechny citlivé akce budou chráněny pomocí JWT tokenů, které budou uloženy v cookies.
* **Role-based access control**: Přístup k určitým částem aplikace bude omezen podle uživatelských rolí (admin, zaměstnanec).

**6. Nefunkční požadavky**

**6.1. Výkon**

* **Rychlost odpovědí**: Systém by měl zpracovávat požadavky uživatelů do 2 sekund pro všechny operace.
* **Škálovatelnost**: Systém by měl být navržen tak, aby byl schopen zvládnout až 1000 simultánních uživatelů bez výrazného poklesu výkonu.

**6.2. Bezpečnost**

* **Autentifikace**: Veškeré přihlašování uživatelů musí být zabezpečeno pomocí **JWT** tokenů.
* **Šifrování**: Všechna citlivá data (např. hesla, osobní údaje) musí být šifrována pomocí standardních šifrovacích algoritmů (např. AES-256).
* **Role-based Access Control (RBAC)**: Systém bude implementovat přístup na základě uživatelských rolí, aby bylo možné spravovat přístup k citlivým funkcím a datům.

**6.3. Spolehlivost**

* **Dostupnost systému**: Systém by měl být dostupný 99,9 % času během pracovních hodin (od 8:00 do 18:00) a 99 % mimo pracovní dobu.
* **Zálohování**: Veškerá data musí být pravidelně zálohována každý den.

**6.4. Údržba**

* **Snadná údržba**: Kód musí být dobře strukturován a dokumentován, aby byl snadno udržovatelný a rozšiřitelný.
* **Testování**: Kód musí být pokryt jednotkovými a integračními testy s minimálním pokrytím 80 %.

**6.5. Použitelnost**

* **Responzivní design**: Aplikace musí být přístupná na různých zařízeních (mobilní telefony, tablety, desktopy) a správně zobrazená na širokém spektru velikostí obrazovek.
* **Uživatelský zážitek (UX)**: Aplikace by měla mít intuitivní a snadno použitelné rozhraní. Všechny funkce by měly být snadno přístupné a logicky uspořádány.

**6.6. Kompatibilita**

* **Prohlížeče**: Aplikace musí být kompatibilní s moderními webovými prohlížeči (Chrome, Firefox, Safari, Edge).
* **Operační systémy**: Aplikace musí být plně funkční na hlavních operačních systémech (Windows, macOS, Linux).

**6.7. Zabezpečení API**

* **Omezený přístup k API**: Veškeré RESTful API musí být chráněno před neautorizovaným přístupem pomocí autentifikace JWT.
* **Rychlá detekce a reakce na bezpečnostní incidenty**: Systém musí být schopen detekovat pokusy o útoky, jako je SQL injection nebo Cross-Site Scripting (XSS), a musí automaticky omezit přístup v případě podezření na útok.

**7. Odkazy na dokumentaci**

* **React**: https://reactjs.org/docs/getting-started.html
* **Node.js**: https://nodejs.org/en/docs/
* **Express**: https://expressjs.com/en/starter/installing.html
* **PostgreSQL**: <https://www.postgresql.org/docs/>
* **Supabase**: https://supabase.com/docs
* **JWT**: <https://jwt.io/>
* **RBAC**: <https://en.wikipedia.org/wiki/Role-based_access_control>
* **Prisma**: https://www.prisma.io/docs/

**8. Závěr**

Tento technický popis a specifikace definuje základní rámec pro vývoj webové aplikace pro správu skladového systému. Použité technologie, struktura kódu a bezpečnostní opatření zajistí spolehlivý a bezpečný systém pro správu produktů, skladových pohybů a historie.