

Dokumentace projektu

Informační systém na správu vězňů a výdej jídla

Tým Naboo

Členové:

Erika Blažeková, Jaromír Bobek, Radim Kopřiva

1. Úvod

Cílem tohoto projektu je digitalizace a automatizace procesů spojených se stravováním ve vězeňském zařízení. Projekt reaguje na potřebu nahradit zastaralé, manuální a papírové procesy řešením postaveným na architektuře mikroslužeb. Systém propojuje správu vězňů, management kuchyně a skladové hospodářství do jednoho koherentního celku, který zajišťuje efektivitu, bezpečnost a transparentnost.

2. Funkční design

Tato část popisuje business logiku aplikace, definuje klíčové problémy, které systém řeší, a specifikuje chování jednotlivých modulů z pohledu uživatele.

2.1 Popis řešeného businessu

Provozování stravování ve vězeňském prostředí není jen o vaření – je to operace, kde se prolíná ekonomika s tvrdými bezpečnostními nároky. Vězeňská kantýna je uzavřený ekosystém, kde jakékoli selhání neznamena jen nespokojeného strávnicka, ale reálné riziko finančních ztrát, a v krajním případě i bezpečnostních incidentů.

Klíčové problémy:

- *Neefektivní správa surovin*

Evidence surovin závisí na papírových seznamech nebo izolovaných excelových tabulkách, které nekomunikují s realitou v kuchyni. Protože data nejsou sdílena v reálném čase, kuchaři nemají přehled. Chybí automatické varování, že dochází klíčové suroviny (např. mouka, maso). To vede k nákupům na poslední chvíli, které mohou být cenově nevýhodné.

- *Hazardování se zdravím a diskriminace*

Informace o zdraví vězňů (alergie) a jejich náboženských omezeních (např. zákaz vepřového) jsou vedeny v jiném systému než samotný výdej jídla.

Cíl:

Systém, který v reálném čase propojuje evidenci vězňů (strávníků) s plánováním výroby (kuchyně) a dostupností surovin (sklad). Výdej stravy je podmíněn identifikací vězně, která okamžitě ověří nárok na stravu, dietní omezení a dostatek kreditů.

2.2 Big picture modulů

Systém je rozdělen na následující mikroslužby.

Správa vězňů (Prisoner Service)

Tahle část systému jednoduše spravuje kompletní informace o vězních.

- **Informace:** Vězeň (ID, jméno, cela, rizikový profil, kredity, náboženství, datum nástupu a propuštění).
- **Funkce:**
 - Evidence příjmů a propuštění vězňů
 - Správa profilů
 - Evidence dietních požadavků a náboženských restrikcí. Tyto atributy jsou poskytovány Kitchen Service pro validaci jídelníčku

Kuchyně (Kitchen Service)

Centrální prvek celého procesu stravování a zajišťuje, aby vše od plánování až po samotnou přípravu jídel probíhalo konzistentně a bez chyb.

- **Funkce:**
 - Tvorba jídelních lístků
 - Recepty
 - Kontrola alergenů: Systém automaticky upozorní, pokud recept obsahuje alergen, který má nahlášený strážník přihlášený k odběru
 - Generování požadavků na výdej ze skladu na základě počtu objednaných porcí

Sklad (Inventory Service)

Zajišťuje logistiku a řízení pro plynulý chod kuchyně bez výpadků nebo nadbytečných přebytků.

- **Funkce:**
 - Evidence zásob (Suroviny, měrné jednotky, expirace)
 - Příjem zboží od dodavatelů (naskladnění)
 - Výdej surovin pro potřeby kuchyně (vyskladnění na základě požadavku z Kitchen Service).
 - Inventury a reportování stavu zásob

Podpůrné služby (Support Services)

- **Auth Service:** Přihlašování personálu (dozorci, kuchaři, skladníci). Využívá JWT (JSON Web Tokens) pro zabezpečení komunikace mezi frontendem a API.
- **Payment Service:** Spravuje účty vězňů. Umožňuje nákup nadstandardních položek v kantýně (pokud je povoleno) a strhávání kreditů za stravu. Integrováno s platební bránou (Stripe) pro vklady z venku věznice (např. od rodinných příslušníků).

2.3 Návrh Use Casů

3. Technický Design

Tato část popisuje, jak je systém implementován, jaké technologie využívá a jak je navržena infrastruktura pro běh mikroslužeb.

3.1 Popis Architektury

Projekt využívá architekturu **Microservices**. Jednotlivé služby jsou volně svázané (loosely coupled), nasaditelné nezávisle na sobě a komunikují spolu primárně synchronně přes REST API (HTTP). Celý systém je zabalen do kontejnerů (Containerized).

Hlavní komponenty architektury:

- **Frontend:**

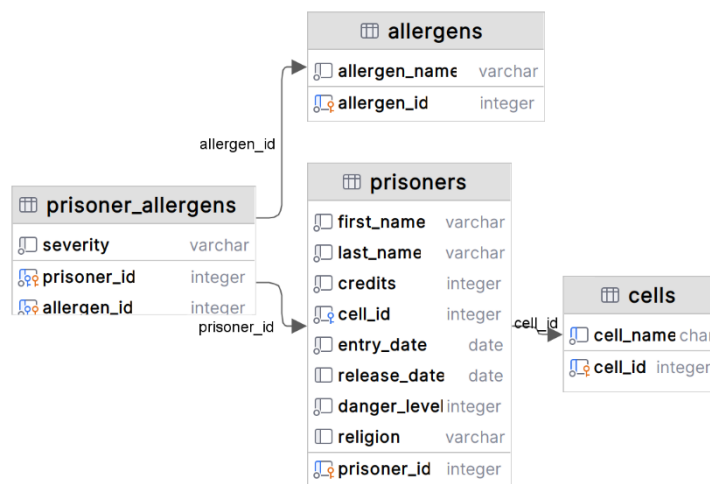
- *Technologie:* Angular + Ionic
- Single Page Application (SPA), která je plně responzivní. Díky Ionic frameworku je připravena pro nasazení na tablety, které mohou používat dozorcí přímo v terénu (mobilita).
- Frontend nekomunikuje přímo s mikroslužbami, ale zasílá požadavky na API Gateway.
- Slouží jako klientské rozhraní pro zaměstnance věznice (dozorce, kuchaře, skladníky)
- Zajišťuje prezentační logiku a ověřování uživatele
- **API Gateway / Reverse Proxy:**
 - *Technologie:* Nginx
 - Reverse Proxy pro všechny požadavky z frontendu
 - Směřuje požadavky na příslušné mikroservisy (např. /prisoners -> Prisoner Service, /items -> Inventory Service)
- **Backend:**
 - *Technologie:* Node.js + Express.
 - Každá služba má vlastní zodpovědnost a vlastní databázi. To zajišťuje, že pád jedné služby (např. Inventáře) neshodí celý systém (např. stále lze vydávat jídlo)
- **Data Layer:**
 - *Technologie:* PostgreSQL
 - Každá mikroslužba má svou vlastní izolovanou databázovou instanci definovanou v docker-compose

3.2 Datový model

Datový model je rozdělen do logických domén odpovídajících jednotlivým službám.

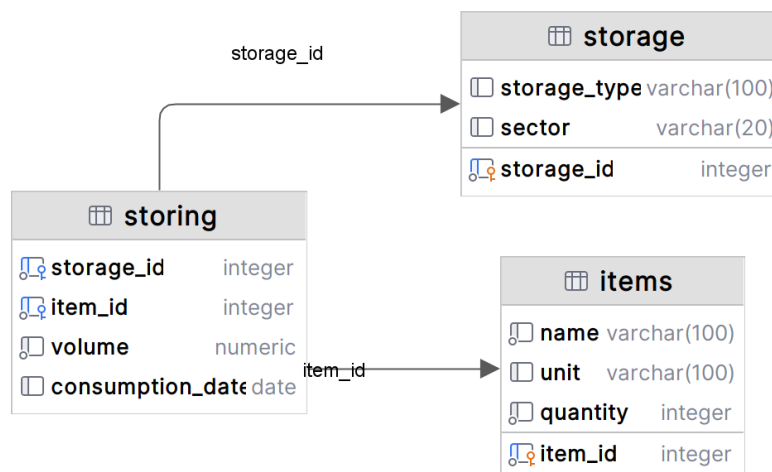
Prisoner Service

- *Prisoners*: prisoner_id, first_name, last_name, cell_id, danger_level, credits
- *Cells*: cell_id, cell_name.
- *Allergens*: allergen_id, allergen_name
- *Prisoner_allergens*: propojuje vězně s alergeny a definuje závažnost alergie



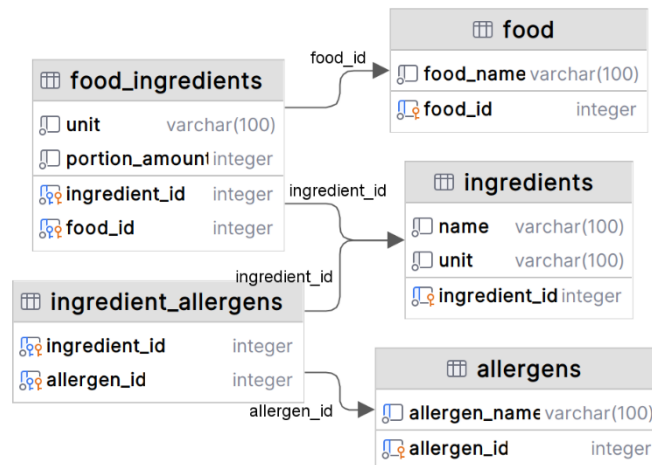
Inventory Service

- *Items*: item_id, name, unit, quantity
- *Storage*: storage_id, storage_type, sector
- *Storing*: volume, item_id, storage_id, consumption_date

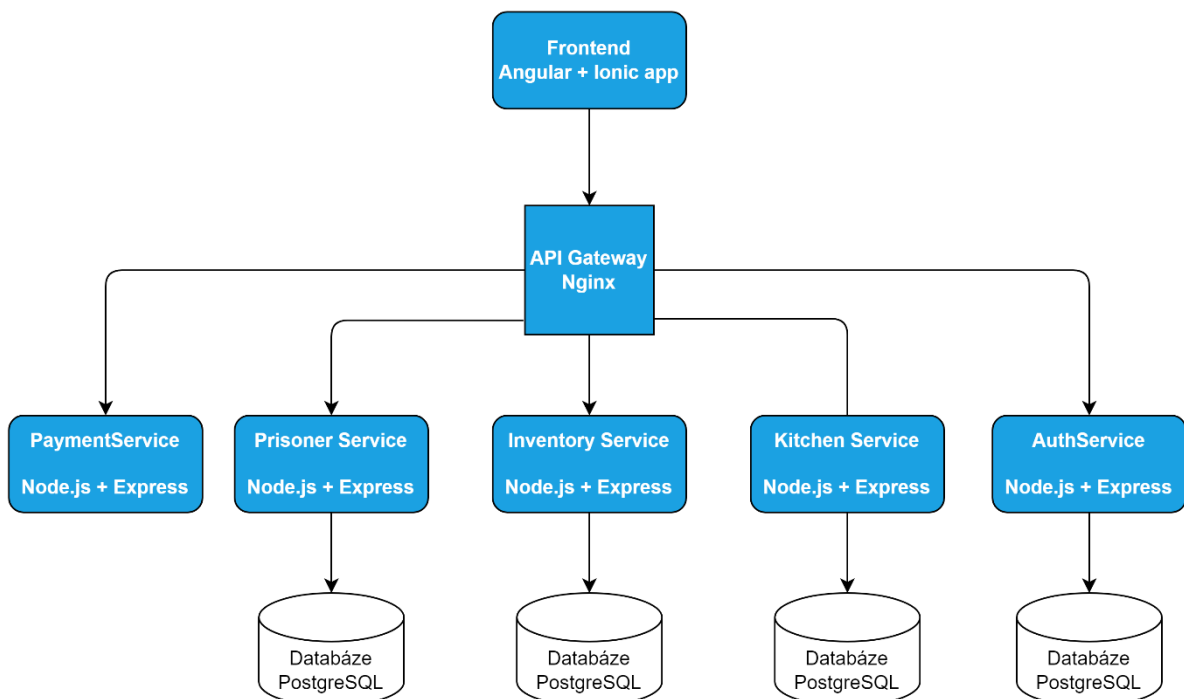


Kitchen Service

- *Food*: food_id, food_name
- *Ingredients*: ingredient_id, name, unit
- *Food_ingredients* (recepty): portion_amount, unit, ingredient_id, food_id
- *Ingredient_allergens* (alergeny v surovinách): ingredient_id, allergen_id
- *Allergens* : allergen_name, allergen_id



3.3 Architektura



3.4 Návrh infrastruktury

Infrastruktura je navržena s důrazem na kontejnerizaci a snadnou orchestraci pomocí Dockeru. Celý systém je definován v *docker-compose.yml* souboru , který zajišťuje spuštění a propojení všech služeb.

Spuštění: *docker compose up*

Networks:

- všechny služby kromě API Gatewaye a frontendu jsou dostupné pouze v rámci interní prison-net sítě a tedy nedostupné jiným způsobem než skrz Nginx.

Inicializační data:

- k automatickému vytvoření a naplnění databáze při prvním startu jsou použity SQL inicializační skripty (např. *inventory_data.sql*)
- */docker-entrypoint-initdb.d/init.sql*

4. Ukázka aplikace

Inventář a editace položky

Inventory

18	Těstoviny	4	kg	EDIT
19	Olej slunečnicový	2	l	EDIT
20	Kečup	1	kg	EDIT
21	Kuřecí maso	5	kg	EDIT
22	Pepř	20	g	EDIT

Name
Kuřecí maso

Quantity
5

Unit
kg

SAVE CHANGES

CANCEL

Dashboard Dispensing Customers Inventory Settings

Seznam vězňů/stravníků

Customers

Search & Filter

Search (ID / Name)

9

Cell

C ▾

ADD NEW RECORD

Customer List

ID	First name	Last name	Credits	Cell	Entry date	Release date	Danger level	Religion	Actions		
9	Tomas	Král	60	C	12/9/25		2	Christian	EDIT	DELETE	CHARGE
									ALLERGENS		
39	Ben	Walker	95	C	12/9/25		2	Christian	EDIT	DELETE	CHARGE
									ALLERGENS		

Dashboard

Dispensing

Customers

Inventory

Settings

Platební brána – dobíjení kreditu

Částka (Kč)

100

Platební údaje

VISA

4242 4242 4242 4242

04 / 28

102

☐ Ušetřete s [link](#)

ZAPLATIT A DOBÍT

Zamítnutí jídla z důvodu alergie

Dispensing

Meal Dispensing

Enter consumer number

10

CHECK

NOT ISSUED ✖

Consumer is allergic to this meal

[Dashboard](#)[Dispensing](#)[Customers](#)[Inventory](#)[Settings](#)

Nastavení tmavého režimu

Settings

Appearance

Follow system theme
Use system light / dark mode

Dark mode
Manual theme switch

Current user

Username kitchen

Current role kitchen

You are not allowed to change roles.

LOGOUT

[Dashboard](#)[Dispensing](#)[Customers](#)[Inventory](#)[Settings](#)