

# Souhrn cílů

## 1) Vize & hodnota

- **Hlavní cíl:** zrychlit a zpřesnit interní přepravu půl-palet (colli) mezi úložišti v hale tak, aby operátor zadal převoz v co nejméně krocích a manipulant měl jasnou frontu práce. Vše musí být stabilní, protože prostor pro chybu není možný. Je třeba zajistit robustní a stabilní BE přes POSTGRESQL.
  - **Pro koho:** Operátor (objednává přepravu), Admin (spravuje data a pravidla), Manipulant (vyřizuje přepravu).
  - **Hodnota:**
    - méně omylů a dotazů (jednoznačné „ODKUD → KAM“),
    - standardizace a auditovatelnost procesu,
    - škálovatelnost (snadné přidání nových úložišť/tras),
    - zkrácení času zadání (slot→přívoz→odeslat).
- 

## 2) Rozsah verze v1 (co dodáváme)

- **Operátor UI (top-down A/B):**
  - horní mřížka max. **12×12 slotů** – aktivní = zelené tlačítko s písmenem (label), neaktivní = šedá „díra“,
  - **Stav A:** vyberu ODKUD (klik na zelený slot),
  - **Stav B:** zobrazí se povolené KAM (oranžová tlačítka z tras), klik vybere cíl,
  - formulář: ODKUD (read-only), KAM (read-only), **Urgence** (STANDARD/URGENT, default dle pravidla), **Poznámka** (volitelná), **Odeslat**,
  - **po odeslání:** ODKUD zůstává, KAM+Poznámka se vynulují (rychlé zadávání více převozů ze stejného místa),
  - spodní **vizuál v1:** pouze vrstvy pozadí (žlutá čára, šedý pruh, bílá linka, modrý chodník) – **bez** dlaždic/slotů.
- **Admin moduly:**

- **Úložiště:** CRUD (code, name), (volitelně) lokální info mřížky w×h.
  - **Trasy:** správa `from` → `to`, abecedně, stavy (existuje/není), hromadné akce (vybrat vše / zrušit vše / invertovat), **Undo toast** po uložení.
  - **Urgence – pravidla:** `scope=route`, `from`, `to`, `defaultUrgency`, `enabled` – určují default pro Operátora po zvolení KAM.
  - **Layouty – editor max. 12×12:** klikáním aktivuji buňky; pro aktivní vyberu **storageCode** (musí existovat) + **label** (volitelný, fallback = 1. písmeno kódu). **Žádné duplicity storageCode v mřížce.**
  - **Uživatelé:** CRUD, aktivace/deaktivace, reset hesla, **home úložiště** (přiřazení uživateli).
- **Manipulant – pracovní fronta (tablet, v1):**
    - **Moje mise:** zobrazuje aktivní úkol (in\_progress) s tlačítkem **DOKONČIT**.
    - **Fronta úkolů:** seznam `status='new'` seřazený s **URGENT** návrhu; akce **PŘEVZÍT** (→ in\_progress) a **SMAZAT** (→ canceled, viz RBAC).
    - **RBAC:** Manipulant smí **PŘEVZÍT/DOKONČIT**; **SMAZAT** (zrušit) jen úkol ve stavu **new**.
    - **Realtime (SSE):** push události *order.created*, *order.status\_changed* a *metrics.updated* pro <u>okamžitou</u> aktualizaci fronty a „Moje mise“.
- 

### 3) Mimo rozsah v1 (vědomě neřešíme)

- Geolokace/manuální navádění manipulantů, live obsazenost, capacity >1.
  - Zakázané zóny/uličky v layoutu (v1 nepoužíváme)
- 

### 4) Základní pravidla a rozhodnutí

- **Půl-paleta = 1 slot** (capacity vždy 1).
  - **Jasná terminologie:** „úložiště“ (ne „sklad“), „přívoz“ pro cíl trasy.
  - Odvozové úložiště: např. A01-A, A01-B, E15-C, B02-A,...
  - Přívozové úložiště: např. A01, E15, B02,...
  - **Priorita rychlosti pro Operátora** (co nejméně kliků, konzistentní rozložení).
  - **Validace proti chybám:**
    - aktivní buňka v layoutu musí mít existující `storageCode`,
    - žádné duplicity `storageCode` v 12×12,
    - KAM musí být mezi povolenými trasami pro dané ODKUD.
      - spravuje Admin u Home\_storage pro daného Operátora
- 

## 5) Data & úložiště dat

- **BE v1**
  - `layout_v1_grid = { layers, grid }` – zdroj pravdy pro Operátora (sloty) i vizuál (vrstvy). Tato data jsou **\*\*perzistentně uložena na Backendu v PostgreSQL a spravována Adminem prostřednictvím dedikovaného API\*\*** [1-5].
  - `layout_v1_layers` – kompatibilitní kopie jen přepínačů vrstev (volitelné pro zpětnou kompatibilitu, pokud je potřeba) [1].
  - `storage_layouts_v1` – (volitelně) info w×h pro úložiště [1].
  - `layout_v1_layers` – kompatibilitní kopie jen přepínačů vrstev,
  - `storage_layouts_v1` – (volitelně) info w×h pro úložiště.
  - `/login` (token),
  - `/storages` (CRUD),

- `/routes` (GET by from, create/delete),
  - `/priority-rules` (CRUD route-scoped pravidel),
  - `/users` (CRUD, reset hesla, `homeStorageCode/homeStorageId`).
- 

## 6) UX zásady

- **Top-down:** nejdřív ODKUD, pak KAM, pak urgency/poznámka → Odeslat.
  - **Srozumitelné stavy:**
    - zelené = aktivní slot, šedé = neaktivní díra,
    - oranžové = povolené cíle,
    - disabled slot = neznámý kód (tooltip s radou).
  - **Rychlé opakování:** po odeslání zůstávám ve „Stav B“ se stejným ODKUD.
  - **Žádné přetěžování** – žádná legenda, žádné sloty ve spodním vizuálu; jen vrstvy.
- 

## 7) Branding a UI

- **Název:** „Manipulátor“ (slovní hříčka, odkaz na VZV i „ovládání“).
  - **Sloganové varianty:** „Ovládá palety, ne lidi.“ / „Poctivá logistika bez psychologických triků.“
  - **Logo:** nápis „MANIPULÁTOR“ + piktogram **půl-palety shora** se dvěma šipkami. Finální barva šipek: **modrá** (původně se testovala oranžová).
  - **Vizuál v1 (operátor – spodní pásy):** světle šedé pozadí; pořadí shora dolů: **žlutá čára** → **pruh VZV (tmavší šedá)** → **tenká bílá linka** → **modrý chodník**.
- 

## 8) Operátor – cílové chování

- **Layout max .12×12** definuje přesnou vizuální mapu: které sloty se zobrazí nahoře, s jakým písmenem. Zobrazují se pouze sloty definované Adminem (max. 12x12)
  - **Klik na slot (A)** → načti `/routes?from=...`, přepni do (B).
  - **Klik na cíl (B)** → vyplň `to`, předvyplň **Urgenci** podle pravidel (jinak STANDARD), fokus do „Poznámky“.
  - **Odeslat** → validace (`from`, `to` ∈ povolené), ulož do `orders_v1`, zobraz toast, zůstaň v (B).
  - **Robustnost**: když layout chybí → infobox „Nastav v Admin → Layouty“; když nejsou trasy → upozorni a porad' Admin → Trasy.
- 

## 9) Admin – cílové chování

- **Úložiště**: rychlý CRUD (bez kapacit); přehledné kódy/názvy.
  - **Trasy**: UI „z pohledu FROM“, přehled TO abecedně, stavy (existuje/není), hromadné volby a undo.
  - **Urgence**: pravidla pro `from-to` (default v Operátorovi).
  - **Layouty – 12×12**: klikáním zap/vyp; pro zapnuté – vybrat `storageCode` + volitelný `label` (fallback písmeno); validace existence a neduplicit.
  - **Uživatelé**: CRUD, aktivace/deaktivace, reset hesla, **home úložiště** uživatele (pro budoucí auto-předvýběr slotu v Operátorovi).
- 

## 10) Technické zásady

- **FE**: Vite + Vue + TS, Tailwind" nebo přesněji "FE: Vite + Vue 3 (Composition API) + TS, Tailwind
- **BE**: Node.js + Express
- **Auth**: token v `localStorage.auth_token`, do `Authorization: Bearer ...`

- **Výkon:** načítat layout z BE (při prvním načtení aplikace nebo po přihlášení Operátora) [10]; trasy až po výběru ODKUD [10, 11].
  - **Realtime:** v1 používá **SSE (Server-Sent Events)** endpoint /events; kanály: order.created, order.status\_changed, metrics.updated. (WebSocket si necháme pro v2 rozšíření – presence/navádění.)
- 

## 11) Rizika a mitigace

- **Nekonzistence layout vs. storages:** v UI disabled slot s tooltipem; Admin a Operátor mají jasnou instrukci co opravit.
  - **Chybějící trasy:** stav B zobrazí vysvětlení a odkaz „Admin → Trasy“.
  - **BE změny schémat:** FE pracuje s chybami milostivě (hlášky, UI běží dál).
  - **Přetížení operátora informacemi:** v1 drží minimalistické UI (bez legend, bez dolních dlaždic).
- 

## 12) Úspěch v1 (měřitelné)

- **≥ 90 % přeprav** zadáno do 10 s (od prvního kliknutí na slot po „Odeslat“).
  - **0 duplicit tras** (UI neumožní založit existující).
  - **< 1 % chyb** způsobených špatným výběrem (díky validacím a přehlednosti).
  - **Adopce:** Operátoři sami preferují nový systém před ad-hoc domluvou.
- 

## 13) Postup nasazení

1. Admin vytvoří **Úložiště** (A01, ...).

2. Admin v **Layouty** aktivuje buňky a přiřadí **storageCode** + (label).
3. Admin v **Trasy** nadefinuje hlavní **from-to**.
4. (Volitelně) Admin v **Urgence** nastaví defaulty.
5. Admin v **Uživatelé** založí uživatele a nastaví **home úložiště**.
6. Operátor začne používat **/operator**.