

# Specifikation v1.1: Finansiell Dashboard (Råvaror + Mag 7) med Yahoo Finance

## 1) Syfte

Bygg en responsiv webbaserad finansiell dashboard med: - Topp-KPI-kort (cards) med pris + förändring och liten sparkline. - Tabbar för moduler: **Råvaror**, **Mag 7**, samt placeholders för **Inflation** och **Grafer**. - Datakälla i MVP: **Yahoo Finance** (inofficiell), med robust cache och "stale"-indikering.

## 2) Professionellt val: Teknik och arkitektur

### 2.1 Frontend (UI)

- **Next.js** (App Router)
- **Tailwind CSS**
- **TanStack Query** för datahämtning och klientcache
- Komponentbibliotek valfritt (t.ex. shadcn/ui), men UI ska matcha referensbild/PDF.

### 2.2 Backend (Data/API)

- **Python FastAPI**
- Separat "data provider"-lager (adapter) så Yahoo kan bytas ut senare utan UI-ändringar.
- Schemalagd uppdatering + server-side cache.

### 2.3 Lagring

- MVP: **SQLite**
- Produktion: **PostgreSQL**
- Valfritt: **Redis** för snabb cache.

## 3) Moduler och navigation (Tabs)

Appen har top-level navigation: 1. **Råvaror** (MVP: fullt) 2. **Mag 7** (MVP: fullt men förenklat) 3. **Inflation** (placeholder) 4. **Grafer** (placeholder eller minimal vy)

## 4) Gemensamma UI-krav

### 4.1 Header (global)

En headerrad med: - Appnamn/logotyp - Sök (filter på tabellrader i aktiv modul) - Tema-toggle (dark/light) - "Senast uppdaterad" + datakälla + stale-status

### 4.2 Färg- och formatregler

- Positiv förändring = grön, negativ = röd.
- Svenska format:
- Decimal-komma
- Tusenavgränsare (mellanslag)

- Visa alltid tecken (+/-) på procent.

### 4.3 Responsivitet

- Desktop: kort-grid + tabell som i referensen
- Mobil: kort staplas, tabell blir horisontellt scrollbar eller kolumnreducerad

## 5) Datakälla och datahantering (Yahoo Finance)

### 5.1 Primärkälla

- Yahoo Finance för priser/historik (inofficiell).
- Backend ska ha skydd mot rate limits:
- Cache (minne/Redis)
- Retry/backoff
- Max-anrop per tidsenhet (konfigurerbart)

### 5.2 Konfiguration (styr vad som visas)

Allt instrumentval ligger i en konfigfil (YAML/JSON), per instrument: - `id`, `name_sv` - `ticker` (Yahoo) - `unit_label` (USD/fat, USD/uns, USD/ton) - `price_type` (Spot eller 3 mån termin eller annan etikett) - `badge_symbol` (Au/Ag/Cu/Zn etc) - `precision` - `display_group` (cards/table) - `sort_order`

**Viktigt:** Systemet gissar inte "spot vs 3m" från Yahoo. Etiketten kommer från config.

## 6) Beräkningar (standardiserade)

Beräkningar sker i backend.

### 6.1 Grundvärden per instrument

- `last_price` (senaste tillgängliga pris)
- `day_change_pct` (förändring vs föregående handelsdag)
- `day_change_abs` (rekommenderas: absolut + procent)
- `timestamp` (Europe/Stockholm)
- `is_stale` (true om datat är äldre än threshold)

### 6.2 Perioder i tabell

- **1V:** senaste pris vs pris ~5 handelsdagar bakåt (standard)
- **1 ÅR (YTD):** senaste pris vs första handelsdagen i året
- **1 ÅR:** senaste pris vs ~252 handelsdagar bakåt

Backend ska hantera "närmsta tillgängliga handelsdag".

### 6.3 Sparklines (på kort)

- Minimal linje för senaste 24h eller senaste 30 datapunkter (beroende på Yahoo-tillgång).
- Om intraday saknas: använd senaste 30 dagliga datapunkter.

## 7) Modul: Råvaror (MVP full)

### 7.1 KPI-kort (6 st)

Varje kort visar: - Namn + enhet - Stort pris - Dagsförändring (% och ev absolut) - Badge med symbol - Sparkline - Klick kan (valfritt i MVP) öppna detaljpanel med 1V/YTD/1ÅR och större graf.

### 7.2 Tabell "RÅVAROR"

Kolumner (ordning): 1. RÅVARA (med enhet i parentes) 2. SENAST 3. +/- (abs + % om valt) 4. 1 V 5. I ÅR 6. 1 ÅR 7. PRISTYP 8. TID

UX: - Sticky header i tabellen - Sortering (default: `sort_order`, användaren kan sortera på kolumner) - Sök/filter påverkar rader

## 8) Modul: Mag 7 (MVP full, förenklat)

### 8.1 Innehåll

- Lista över "Magnificent 7" (konfigurerbar)
- Minimikrav per bolag:
- Senast
- Dagsförändring
- 1V, YTD, 1ÅR
- Standard-sort: **YTD fallande**.
- Datakälla: Yahoo Finance (tickers i config).

## 9) Placeholder-moduler

### 9.1 Inflation

- Enkel sida med rubrik + "kommer snart"

### 9.2 Grafer

- Enkel sida med rubrik + tom container
- (Valfritt) instrumentväljare + historikgraf (Fas 2)

## 10) Backend API-kontrakt (minimikrav)

Frontend pratar med backend via JSON.

### 10.1 Endpoints

- `GET /api/health` → status + datakälla + senaste uppdatering
- `GET /api/commodities/summary` → data för kort + tabell
- `GET /api/commodities/series?id=...&range=...` → sparkline/detaljgraf-data
- `GET /api/mag7/summary` → tabell/kort för Mag7
- (Valfritt) `GET /api/config` → exponerar aktiva instrument (utan hemligheter)

## 10.2 Datamodell (payload)

Varje rad/kort ska minst ha: - `id`, `name`, `unit`, `price_type` - `last`, `day_abs`, `day_pct` - `w1_pct`, `ytd_pct`, `y1_pct` - `timestamp_local`, `is_stale` - `sparkline[]` (lista med (t, v))

## 11) Uppdatering, cache och drift

### 11.1 Schemalagd uppdatering

- Standard: var 60:e sekund (konfigurerbart)
- Historik uppdateras minst 1 gång/dag, "senast" oftare.

### 11.2 Cache-strategi

- Server cache:
- "Summary payload" cacheas och återanvänds
- Historik i SQLite/Postgres
- Stale-threshold:
- Om ingen lyckad uppdatering på t.ex. 10 minuter: flagga stale.

### 11.3 Loggning

- Antal instrument ok/fail
- Senaste uppdatering
- Yahoo-fel och retry-statistik

## 12) Kvalitetskrav och felhantering

- UI får aldrig krascha om en ticker failar.
- För failad ticker: visa `N/A` och behåll övriga rader.
- Visa tydligt om data är stale.

## 13) Leverabler

- Monorepo eller två repos (frontend/backend) med:
- README (setup, start, konfig)
- Exempelkonfig för råvaror + Mag7
- Docker-compose för lokal körning (frontend + backend + db)
- En fungerande MVP som matchar referensbildens layout och PDF:ens tab/UX-krav.

## 14) MVP vs Fas 2

### MVP (ska ingå)

- Tabs, header, tema-toggle
- Råvaror: 6 kort + tabell + färglogik + uppdatering/caching
- Mag7: tabell med 1V/YTD/1ÅR och YTD-sort
- Stale-indikering

## **Fas 2 (valfritt)**

- Detaljsidor per instrument
- Valutakonvertering USD↔SEK
- Mer avancerade grafer, fler intervall, export (CSV)