

CYRILL HOCH · SELECTED PROJECTS · 2026

Project Portfolio

Selected work in MS Power Platform, process digitisation, full-stack development, data applications and practical AI integration.

BUSINESS PROJECTS · 01 Business Apps · 02 PlainPPM · 03 Client Applications
INDEPENDENT PROJECTS · 04 Project Overview · 05 MedSave

MS Power Platform & Business Apps

Client applications for data management, project governance and process automation.

Brokers · Qualibroker AG

MS POWER PLATFORM · APP OWNER

Architecture. Internal business application for one of Switzerland's largest insurance brokers. MS Power Apps and Power Fx provide the interface and logic; MS Power BI and MS Power Automate support reporting and automation. The app integrates with MS SharePoint, MS Outlook and MS 365 and uses a REST API. Communication, templates and client records stay within the same working context.

My responsibility. App ownership for daily use by approximately 250 people, covering operations, technical administration, support, access rights, testing, data migrations and functional development throughout the application lifecycle.

- MS Power Apps · Power Fx · MS SharePoint · REST API

PlainPPM · DACS AG

MS POWER PLATFORM · UI REFACTORING

Architecture. MS Power Apps application for ideas, projects and the agile backlog, connected to MS SharePoint and MS Teams. Views and filters expose status, ownership, dates, effort and budgets. The structure links strategic portfolio work to operational planning and individual work items.

My contribution. Technical and visual refactoring from rigid pixel positioning to a flexible layout system. Semantic colour names enabled themes, while a consistent naming convention replaced generic object names. I rebuilt the Kanban module with flexible axes, automatic spacing, draggable sidebars and user-adjustable dimensions.

- MS Power Apps · Power Fx · MS SharePoint · MS Teams

Customer Repair Cases · ADENSYS

MS POWER APPS · END-TO-END IMPLEMENTATION

Data model. Two MS Lists store repair cases (REK) and items submitted for repair (REP), with multiple items linked to each case. Two connected galleries display the cases and filter related products according to the selected REK record.

Application logic. Search, company filters, forms, a multilingual auto-mailer and conditional product checks support the workflow. I implemented the data structure, forms and logic. The completed app has been used daily by approximately ten people for four years, without requiring corrective maintenance or fault resolution.

- MS Power Apps · Power Fx · MS Lists

Travel Reimbursement · IKEA

MS POWER APPS · END-TO-END IMPLEMENTATION

Architecture. Two independently developed MS Power Apps serve approximately 3,500 employees through a shared MS Lists data source. Employees submit ticket details and receipts; HR reviews the records and submission history.

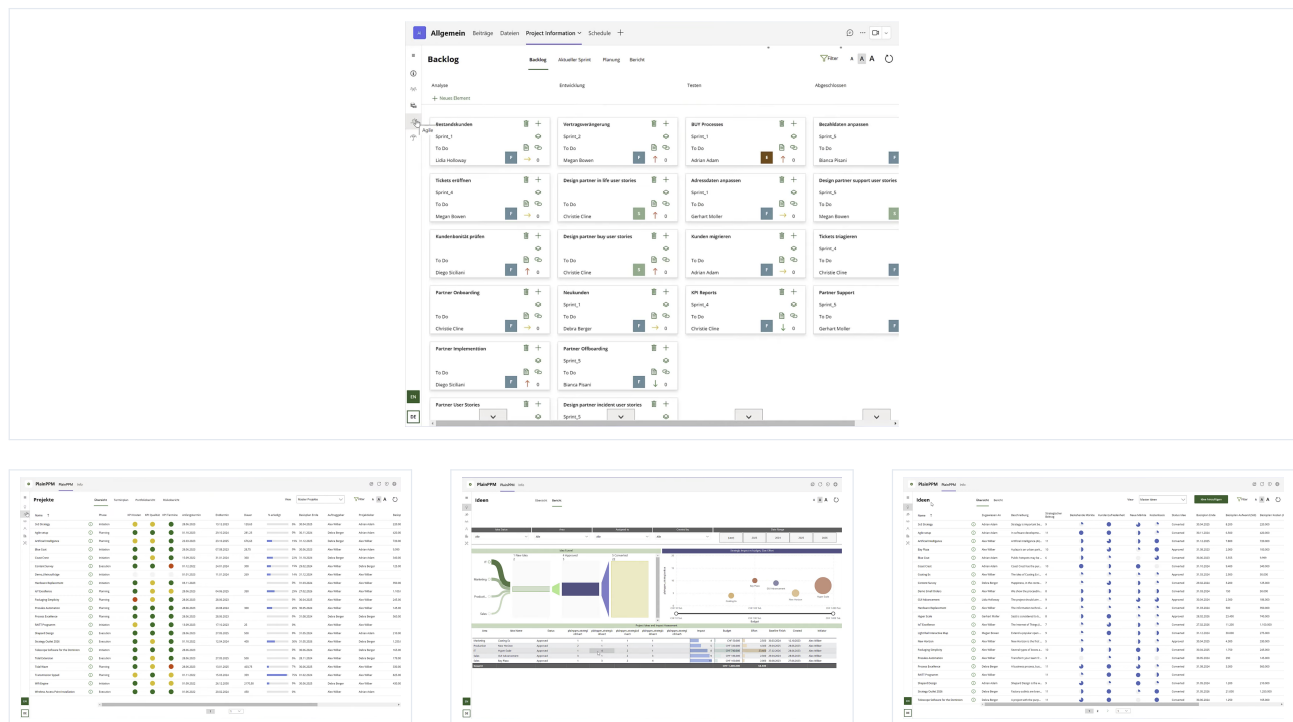
Workflow logic. Review actions trigger emails, while receipts are named and stored in employee folders. Month-end automation totals reimbursements, transfers them to Finance and opens the next monthly list. It reduced one employee's monthly processing effort from several days to approximately 20 minutes.

- MS Power Apps · Power Fx · MS Lists

02 / MS POWER PLATFORM · CLIENT APPLICATION

PlainPPM · DACS AG

Contributing to a business application for idea, project portfolio and backlog management.



PLATFORM & INTEGRATION

Architecture. PlainPPM is built on MS 365 and the MS Power Platform. MS Power Apps provides the user interface, MS Power Automate supports workflows, and MS Power BI delivers reports and dashboards.

Integration. The client solution connects to MS SharePoint and MS Teams, bringing project information into team collaboration. Its modules support initiatives from initial idea to project closure. The platform accommodates traditional, agile and hybrid project methodologies, alongside resource planning, risk management and time tracking.

MY CONTRIBUTION

UI refactoring. I converted the rigid pixel-based interface into a dynamic layout system. Flexible axes, automatic spacing and draggable sidebars prevent downstream elements from requiring manual repositioning after layout changes.

Design system. Semantic colour names replaced hard-coded hex values and enabled theming. Names based on container, type and function replaced automatically assigned generic names. I completely redesigned the Kanban module, with dimensions controlled centrally and, in part, by users.

FEATURES

Portfolio & Reporting. An idea funnel and an impact/budget chart support comparison and prioritisation of initiatives. Project overviews show KPIs for cost, quality and delivery dates. Date filters define reporting periods; progress indicators show the current completion status.

Planning & Control. The agile backlog organises work items by workflow stage and sprint. Additional modules support resource planning from project and line-management perspectives, recording risks and countermeasures, and time tracking to evaluate effort and project progress.



VIDEO & PRODUCT DEMO

PlainPPM · DACS AG

Scan the QR code or open: youtube.com/watch?v=bDDfwsT7kYQ

03 / MS POWER APPS · PREVIEW

Client App Preview

Selected application screens from two independently developed client solutions.

Customer Repair Cases · ADENSYS

Linked cases and items submitted for repair, record creation, editing, an auto-mailer and product checks.

CASE OVERVIEW

EDITING

RECORD CREATION

PRODUCT CHECKS

Travel Reimbursement · IKEA

Employee public-transport ticket entry, receipt photos and submission history, with HR review and validation.

EMPLOYEE SUBMISSION

HR REVIEW & HISTORY

04 / INDEPENDENT PROJECTS

Project Overview

QR, satellite monitoring, learning, communication, healthcare coordination and data analysis.

LIT · Pass the Flame

DEPLOYABLE MVP · NEXT.JS · REACT · POSTGRESQL

QR platform for artists and bands: physical lighters become travelling music objects.

Permanent QR codes open Spotify music, a map and a public journey timeline.

Scans record approximate locations; visitors refine stops. Responsive views and sharing support each journey; a protected admin area manages lighters. Up to five Spotify songs or a playlist, social profiles, artwork-derived colours and public rankings complete the artist page. Mobile QR scanning simplifies check-ins.

• Next.js · React · PostgreSQL · Prisma · NextAuth · OAuth · QR

Doomsday Trainer

NATIVE IOS APP · SWIFTUI · MVVM

Training app for quickly determining the weekday of any date mentally, with guided practice.

A central ObservableObject view model generates date questions and manages tabs, answer state and four unlockable guided steps.

The Doomsday algorithm implements leap-year, century-anchor and month-anchor logic for the years 1500-3000.

Response times, accuracy, streaks and per-weekday statistics are aggregated throughout each session. The session summary captures both speed and accuracy.

• Swift · SwiftUI · MVVM · XCTest

Bobi · Lightweight Shared Chat

PRIVATE BETA · 5 TESTERS · NEXT.JS · OPENAI API

Lightweight shared chat application for small teams, with a straightforward operating model.

A shared passcode controls access to chat, while a protected server route handles model requests and keeps the API key out of the browser.

Minimal application state simplifies deployment and operation. Controlled error responses protect internal details and credentials. The deliberately reduced interface keeps onboarding and administration simple for testers and small teams.

• Next.js · React · TypeScript · OpenAI API · Vercel

CareRoute

INTERACTIVE CONCEPT · NEXT.JS · OFFLINE

Interactive product concept for coordinating time-critical maternal referrals in underserved regions.

The interface shows reported facility readiness, guides a three-step referral and visualises acknowledgement and transport status.

A Today dashboard, facility availability, a three-step referral, Ride View, transport assignment, and call, messaging and fallback views demonstrate the planned workflow. Ride View links the receiving facility, vehicle and timeline while keeping hand-off actions available in context. Role-specific views make each hand-off clear.

• Next.js · React · TypeScript · PWA · Service Worker

FlareScope · Oil Refineries

FUNCTIONAL MVP · PYTHON · FASTAPI · NEXT.JS

OSINT monitoring of thermal satellite signals at oil refinery locations.

FIRMS/VIIRS anomalies are matched against verified coordinates, a time window and a search radius.

Matching logic, debug endpoints and a dashboard make results inspectable. The dashboard maps thermal anomalies to the corresponding refinery locations. Time windows and search radii define the scope; facility coordinates and satellite matches can be reviewed together in the dashboard.

• Python · FastAPI · Next.js · FIRMS/VIIRS · OSINT

Photography Wizard

NATIVE IOS APP · SWIFTUI · AVKIT

Local video-learning app with a structured content source and synchronised transcripts that follow the active video passage.

LessonStore loads and indexes bundled lesson assets and decodes the JSON transcripts; AVPlayer handles MP4 playback.

An AVPlayer time observer synchronises playback with transcript cues every 100 ms; UIScrollViewReader automatically follows the active passage.

The macOS/iOS app offers a reusable set of playback and navigation controls.

• SwiftUI · AVKit · Combine · Codable · MVVM

Telegram Memories

LOCAL-FIRST PROTOTYPE · REACT · PRIVACY

Local memory explorer for personal Telegram archives, with chat, gallery, calendar and insights.

HTML or ZIP exports are normalised into IndexedDB with messages, replies, reactions and media references; virtualised timelines keep long histories responsive.

Parsing and analysis run locally in the browser. The security design requires encryption and an app lock for sensitive archives. Filterable views for chats, media, calendar and insights support navigation through large archives.

• React · TypeScript · IndexedDB · fflate · TanStack Virtual

MedSave

FUNCTIONAL MVP · NEXT.JS · DATA APPLICATION

Medication price comparison using live pharmacy queries and reference data from the Swiss Compendium.

Provider-specific parsers normalise offers; Pharmacode and EAN matching groups variants and removes duplicates.

The search API validates requests, calculates final prices including shipping, and runs reference-backed quality checks. Automated daily audits continuously monitor the quality and availability of connected provider data. Source status, matching attributes and quality gates keep every result traceable.

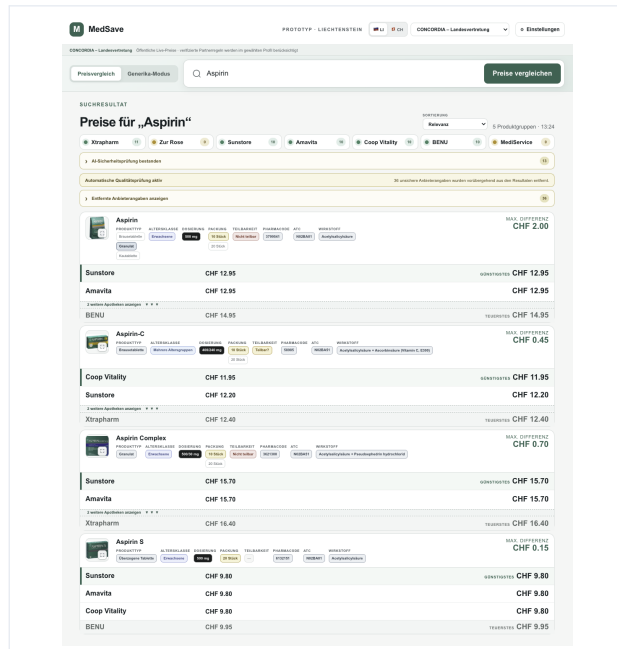
• Next.js · React · TypeScript · Cheerio · JSON Schema

05 / INDEPENDENT PROJECT

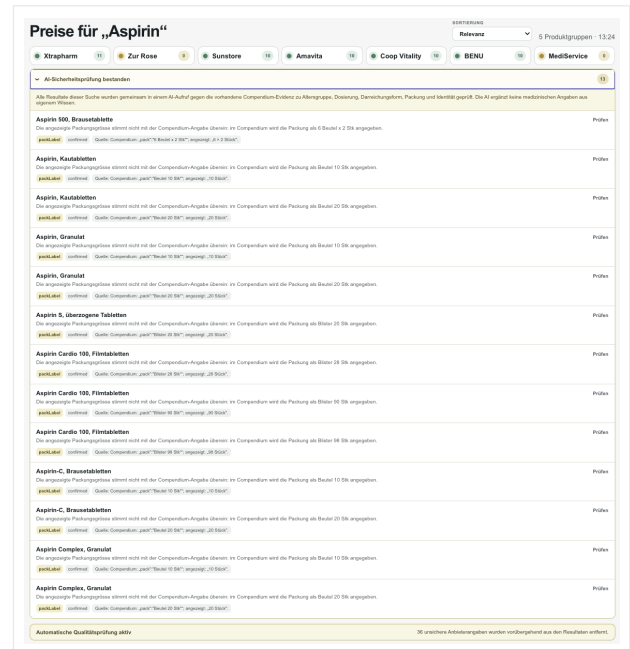
MedSave

Medication price comparison with product matching, live data queries and evidence-based checks.

PRICE COMPARISON



AUTOMATED QUALITY CHECKS



PROVIDERS & MATCHING

Provider layer. Provider-specific Cheerio parsers query pharmacies and the Swiss Compendium in parallel, converting each result into a unified Offer data type.

Identity. Exact Pharmacode and EAN matches link provider offers, remove duplicates and enrich records with the official name, active ingredient, ATC code, dosage and pack details. The enriched records feed product grouping and price comparison across pharmacy providers.

QUALITY & SAFETY

Quality gate. Results are checked against the search term and Compendium index before grouping. Fixture-based parser tests, health checks and daily audits validate provider data quality.

Safety review. AI checks product information against supplied Compendium references and returns schema-validated results. Field-level checks determine which information is displayed; requests use store:false.

API & PRICING

Search API. GET /api/search validates query length, search mode, insurance profile and prescription status. The API queries provider prices on demand. Cache-control directives prevent response storage.

Calculation. A discount engine and verified shipping rules calculate a comparable final price. Variants are grouped by form, strength, pack and target group, then sorted by price and potential savings.

- Next.js · React · TypeScript · Cheerio · JSON Schema · Node Test Runner