I bridge the gap between product design and engineering by building flexible architecture that empowers teams to iterate quickly and make better decisions.

This document summarizes lessons from over a decade of hands-on work.

Design Engineer

Valentine Zubkov

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

Design Engineer

The Freak Company

Bali, Tbilisi

2024–Present

At The Freak Company, I design, build, publish, and operate a set of digital products across native Apple apps, open-source developer tooling, and the web. I own the complete path, from product discovery and design to release, subscriptions, and continuous improvement.

Projects

Freak Apps Bundle

Five native Apple apps for health, finance, memory, and learning: designed, shipped, and maintained as one product family.

FreakUI

A handwritten, MIT-licensed SwiftUI framework, paired with an MCP-assisted generation harness for Apple development.

Tinypad

A bookmark management tool and web launchpad, spanning UX, frontend architecture, authentication, data, billing, analytics, and lifecycle operations.

VoiceKey

A voice keyboard with a companion application.

Trysy

Virtual AI wardrobe studio and unified

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

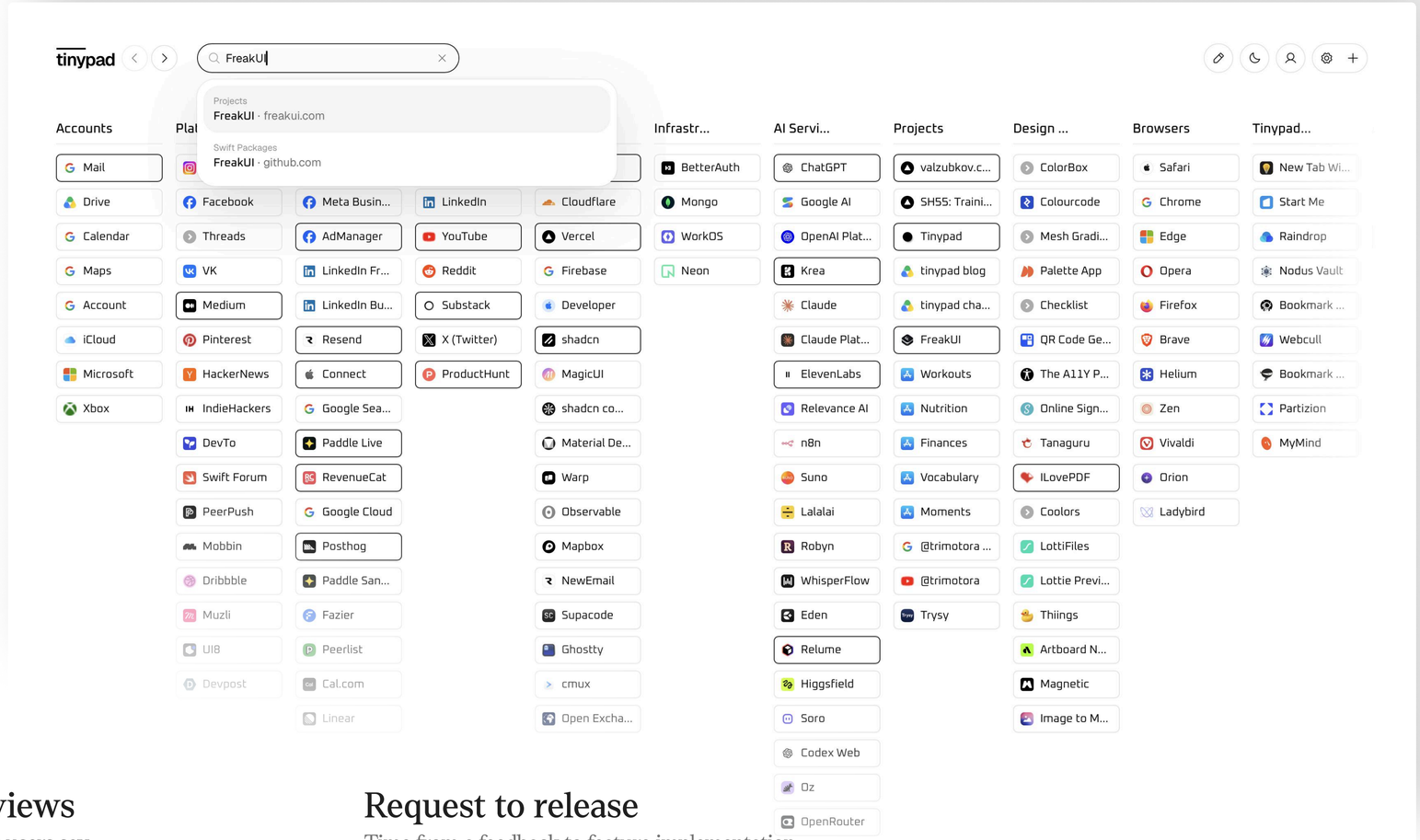
The Freak Company Tbilisi — 2026

Tiny product, massive response

Design Engineering

Tinypad combines years of design-engineering experience with strong product opinions. I keep the scope focused, own the complete system, and shorten the feedback loop.

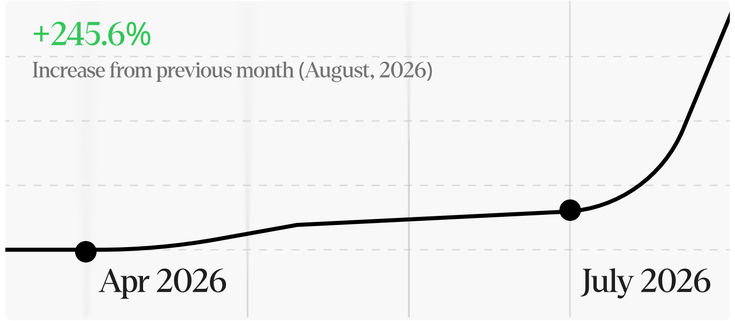
That approach turned a compact idea into a product with ***rapid growth and sustained engagement.***



Product metrics I focus on

User growth

Month to month dynamic



Reviews

What users say

@mktpavlenko

Product builder

Bookmarks didn't die because browsers killed them. They died because we stopped revisiting the same sites. everything became one-off. Tinypad brings back the habit, not just the UI.

@maria r.

Freelance writer

The grouping works really well for me, because

Request to release

Time from a feedback to feature implementation

12 Aug

User signal

"Please add Sign in with Apple" , "I can't edit bookmarks on iPad"

14 Aug

Engineering

Feature designed and implemented by a single person, with no delays

15 Aug

Release

Feature tested and released, user notified

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

The Freak Company Bali, Tbilisi — 2024–2026

Health and wealth: building apps in evergreen niches

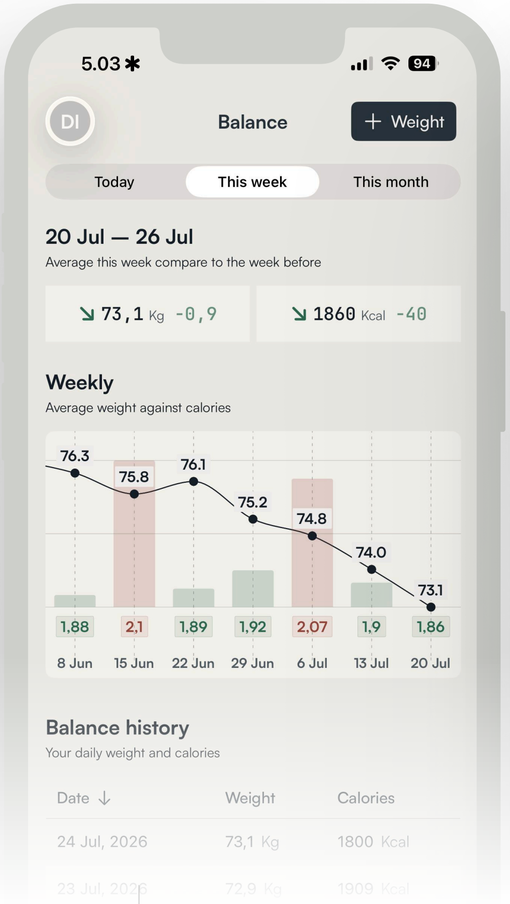
Design Engineering

Nutrition, training, and personal finance are not one-off problems, but personal systems that ***we aim to constantly improve.***

Each product began with a narrow purpose and evolved through continued use: managing nutrition with precision, planning progressive overload, and understanding budgeting across currencies.

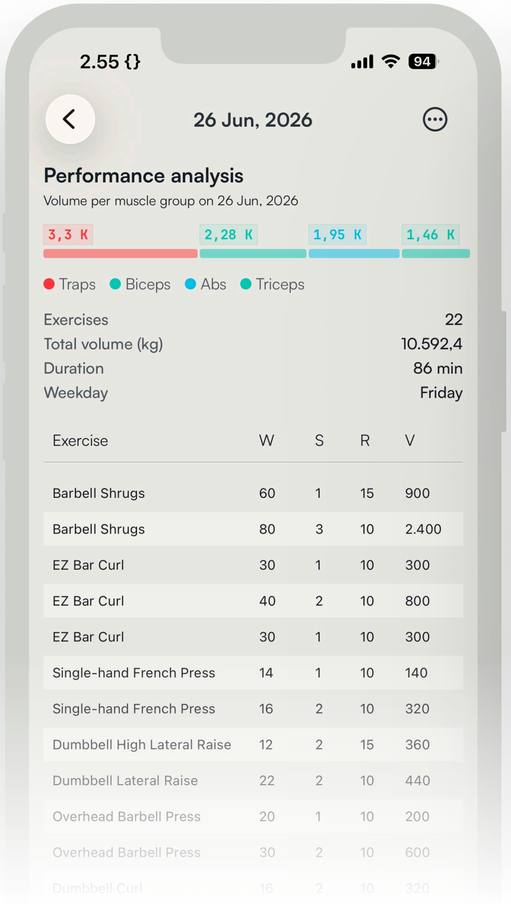
Nutrition

Precision nutrition manager



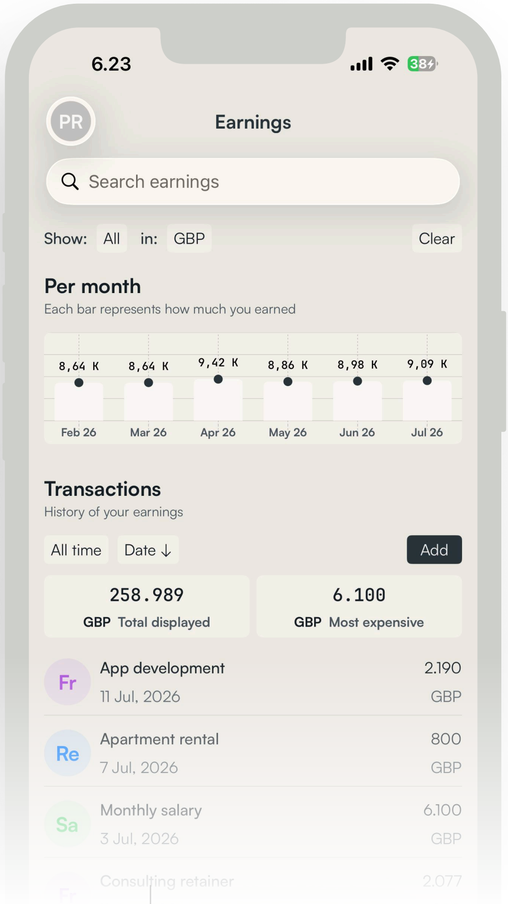
Workouts

Progressive overload tracker

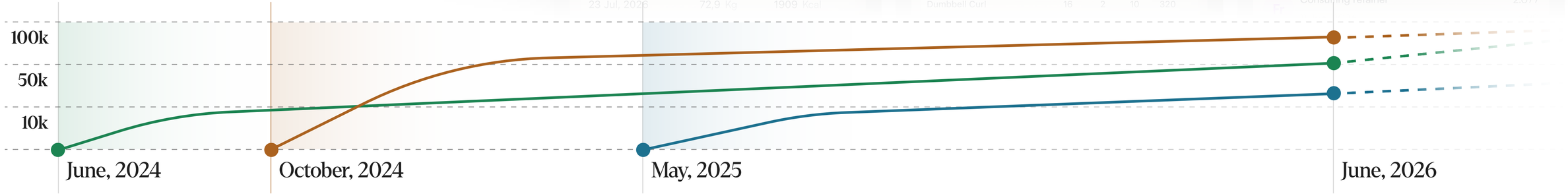


Finances

Multi-currency budget tracker



Estimated monthly logged actions over time



All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

The Freak Company Bali, Tbilisi — 2025–2026

One framework to rule them all and ship blazingly fast

Design Engineering

FreakUI is a highly opinionated library of components and extensions built on SwiftUI. It covers formatters, accessibility, haptics, custom theming, and other ***improvements to the development experience.***



Workouts

Live, 3.0.0
Subscriptions
iPhone, iPad, macOS



Nutrition

Live, 3.0.0
Subscriptions
iPhone, iPad, macOS



Finances

Live, 3.0.1
Subscriptions
iPhone, iPad, macOS



Vocabulary

Beta, 2.1.0
Free
iPhone

Moments

Beta, 1.0.0
Free
iPhone



FreakUI

Public beta, 0.5.0
MIT-licensed core · Paid generation
iPhone, iPad, macOS

Foundation

Swift 6
Concurrency, @Observable...

SwiftUI Primitives
Components, Values...

UIKit Bridges
Navigation Bars,
Liquid Glass...

Accessibility
Labels, Alternatives...

Formatters
Numbers, Dates...

Haptics
Physical Feedback

Toasts
Successes, Warnings

Theme

Scale Tokens
Radius,
Paddings...

Color Variables
Colors,
Shapes...

Typography Variables
System Fonts,
Custom Fonts...

UI Dimensions
ExtraLarge,
Large,
Medium,
Small...

Surface Styles
Elevated,
Outlined...

Components

```
FViewLayout(
  role: .parent,
  title: "Workouts",
```

```
FSection(
  heading: .titleAndS...
  title: “Weekly vol...
```

```
FContainer(
  appearance: .elevated,
  size: .medium,
```

```
FDashboardCell(
  value: .double(volu...
```

```
FButton(
  style: .primary,
  title: “Log weight”,
```

```
FSelector(
  label: “Muscle gr...
  options: muscleG...
```

```
FMenu(
  label: “Period”,
  size: .medium,
```

```
FPicker(
  label: “Balance”
```

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

2026

Clean architecture as a foundation for rapid growth

After years of watching startups collapse under over-engineering, fake Agile, misaligned values, and UI band-aids, I arrived at a simple principle: ***great architecture exists to make change fast and cheap.***

A clear mental model

Business logic, user flows, state, and UI components should be separated from day one. This separation is not about following principles of programming or satisfying ideas of abstraction. It's about clarity for the team, so people can make the right decisions faster.

Lightweight design system

A component library should support change, not become a product of its own. It can be separated from the main codebase, but must stay close to where product decisions are made. The system should be driven by user experience, not graphic design, and contain only what is actually used.

Extra layer of safety

Early-stage products change constantly, and architecture should expect rewrites, pivots, and reversals. Flexibility today beats theoretical scalability tomorrow, and adding a thin layer between modules makes change cheaper and safer. Complexity should grow with demand.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

The Freak Company Bali, Tbilisi — 2026

Framework as a service: Free open source, paid harness

Design Engineering

*FreakUI builds an accessible **one-prompt generation** on top of the **hand-written open source framework**.*

Developers can adopt the complete MIT-licensed package for free, while Codex and Claude Code plugins can provide an assisted development with a paid FreakUI subscription.

FreakUI distribution graph

FreakUI Core

Dependency-free Swift source for themes, components, charts, navigation, and interaction across iOS and macOS.

Swift Package

Versioned FreakUI Core package, installed directly in Xcode through Swift Package Manager.

For AI-assisted development • Paid subscription

MCP freakui.com/mcp

- OAuth authentication
- Subscription access verification
- Versioned generation contracts
- Private recipes and templates
- Structured application blueprints

Website freakui.com

- Documentation
- Version history
- Authentication & accounts
- Subscriptions & billing

Codex Plugin

- Collects and confirms the application brief
- Sends the request to the MCP
- Receives the versioned blueprint
- Creates and validates the project locally

Claude Code Plugin

- Runs the same supported workflow
- Exchanges requests and blueprints with the MCP
- Creates and validates the project locally

For direct development control • Free and open source

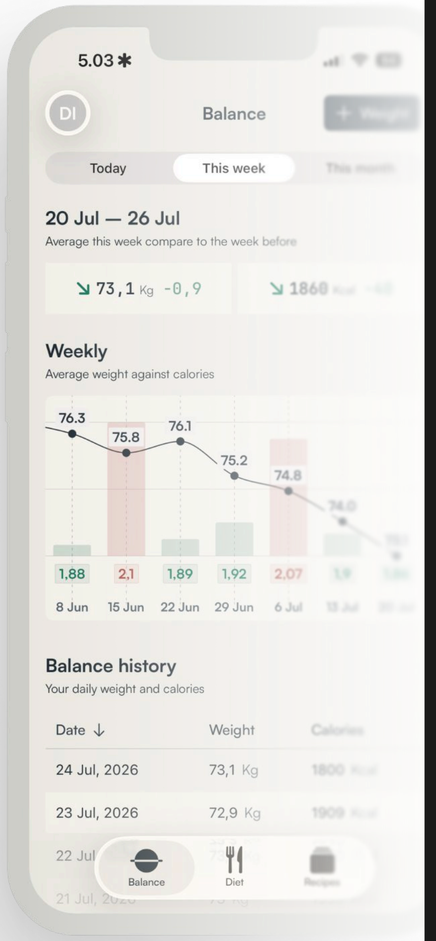
GitHub Repository

Complete MIT-licensed source, tagged releases and version history, migration notes and documentation, public issues and contributions, direct Swift Package Manager installation

Application

iPhone, iPad, macOS

Customer-owned, single-target SwiftUI foundation for iPhone, iPad, and macOS, built against an exact FreakUI release.



All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

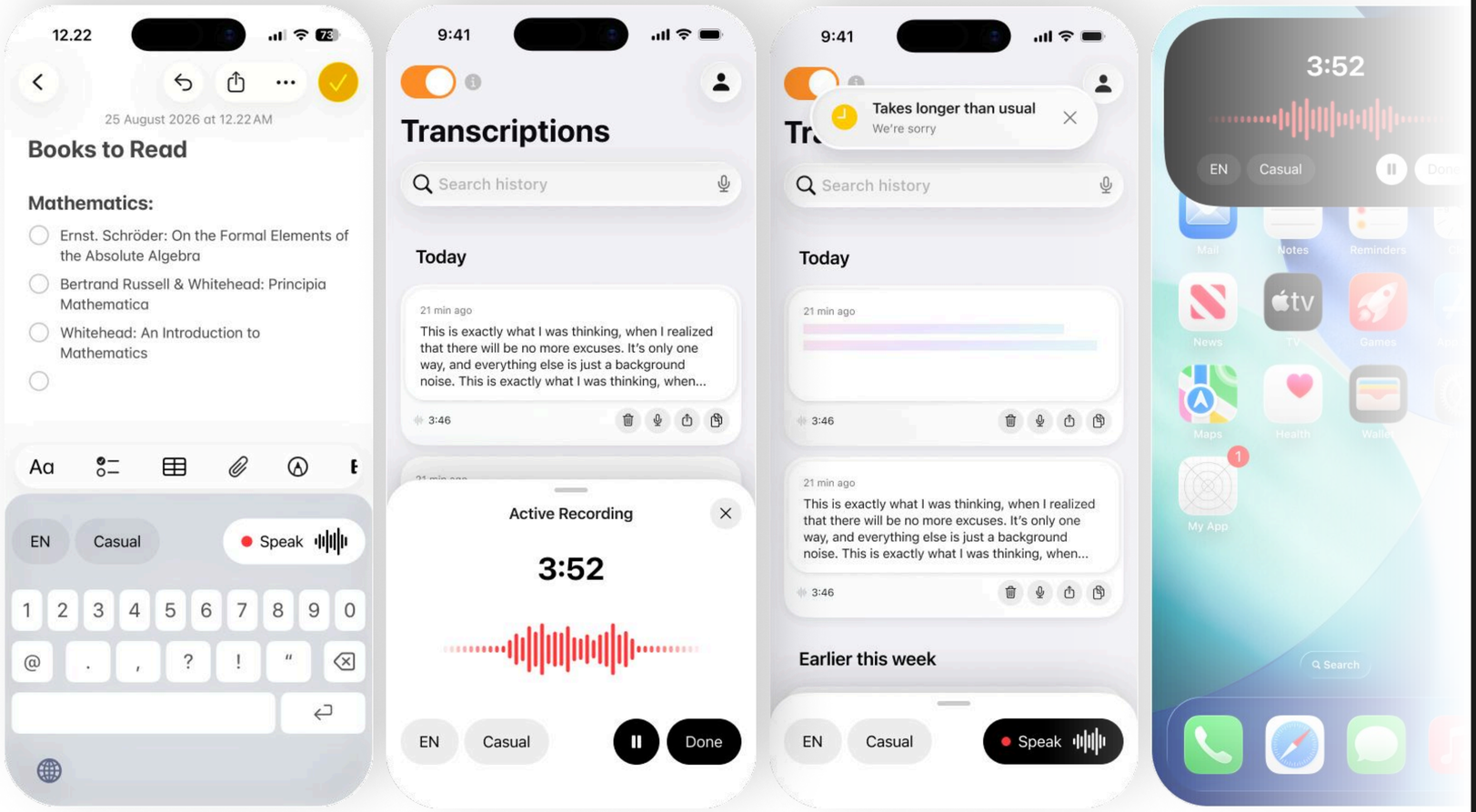
The Freak Company Tbilisi — 2026

Voice recording across keyboard, app, and system

UX/UI Design

VoiceKey extends voice input across a custom keyboard, companion application, transcription history, and Live Activities.

I designed the complete experience, including the states and transitions that keep the recording flow understandable across every surface.



One experience across three surfaces

Keyboard in any app

Start recording from any text field while preserving the current context.

Application

Manage active sessions, transcription history, and delayed processing.

System

Monitor and control the recording through a persistent Live Activity.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

Product & UX/UI Designer

BlockTrust

Hong Kong, Sydney 2023

BlockTrust shipped multiple blockchain-powered campaigns on a shared product system I designed. That system evolved into BrandPay, where I defined the core product journeys and designed the connected brand, customer, venue, wallet, and redemption experiences.

Projects

BrandPay

Most people don't understand blockchain, don't enjoy it, and don't want to interact with it.

Chainz

Ideas like digital twins or provenance tracking sound convincing to brands.

Colt SZ

Ideas like digital twins or provenance tracking sound convincing to brands.

No Limits

Ideas like digital twins or provenance tracking sound convincing to brands. But end users just want to buy,.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

BlockTrust Hong Kong, Sydney — 2023

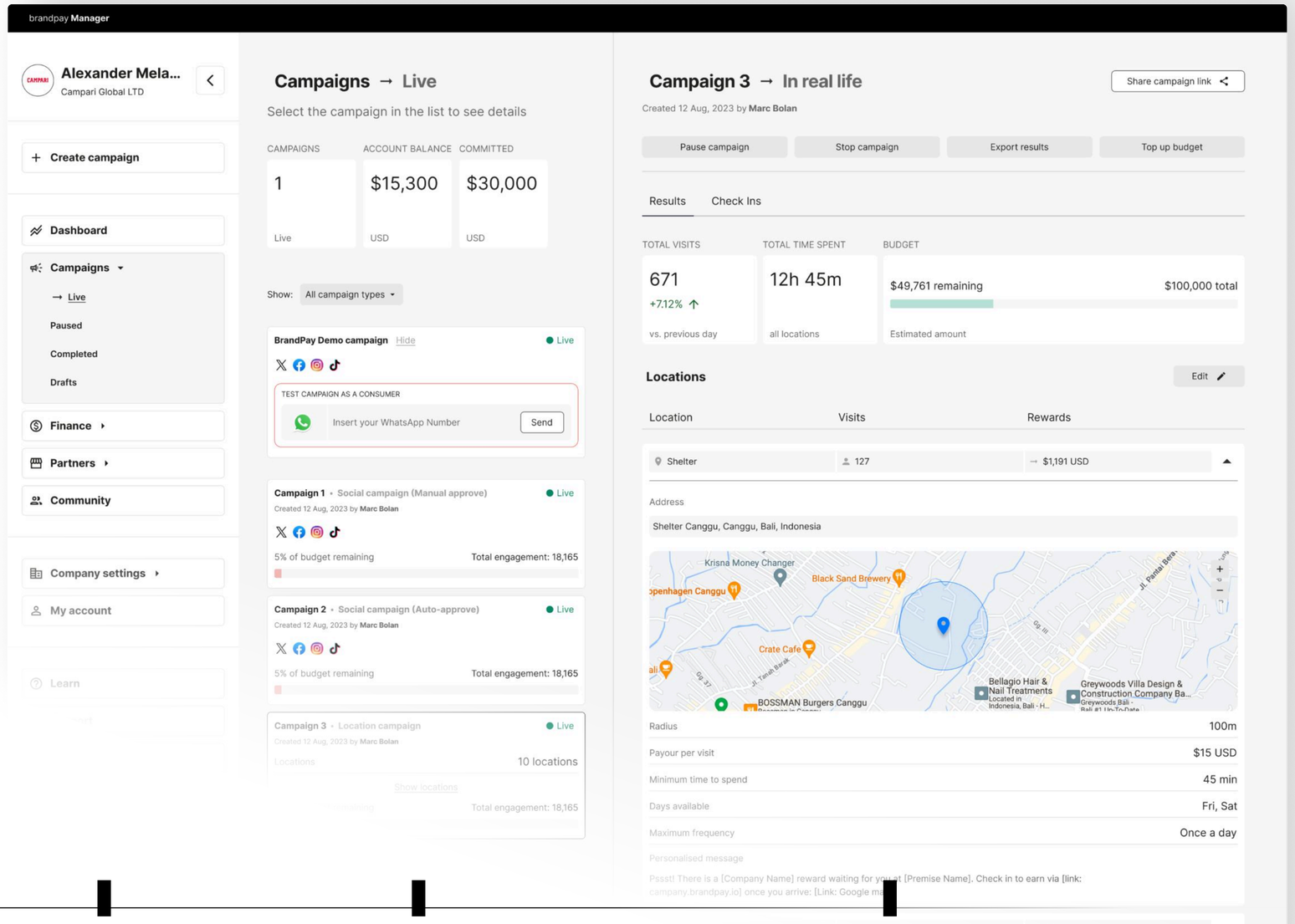
Turning social media campaigns into real-life engagements

Product & UX/UI Design

At BlockTrust, we turned our blockchain expertise into BrandPay, a social-currency product that let brands create customer wallets and reward participation. My role was to ***design every interaction along the way.***

Activity snapshot: Jun–Aug 2023

12	21	1.5M +
Brands participated	Campaigns launched	Engagements reported



Decision path: from product discovery to live campaigns

01
Define the opportunity
Mapped the client need, customer behavior and model that shaped the MVP.

02
Prototype the journey
Tested key flows with clients and focus groups, using early feedback to simplify the experience and sharpen the scope.

03
Shape the delivery
Converted the validated concept into a practical plan for design and development.

04
Design every interaction
Designed the connected brand, customer, venue, and wallet experiences, working directly with the engineering team.

05
Launch and improve
Refined the product around client and customer feedback.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

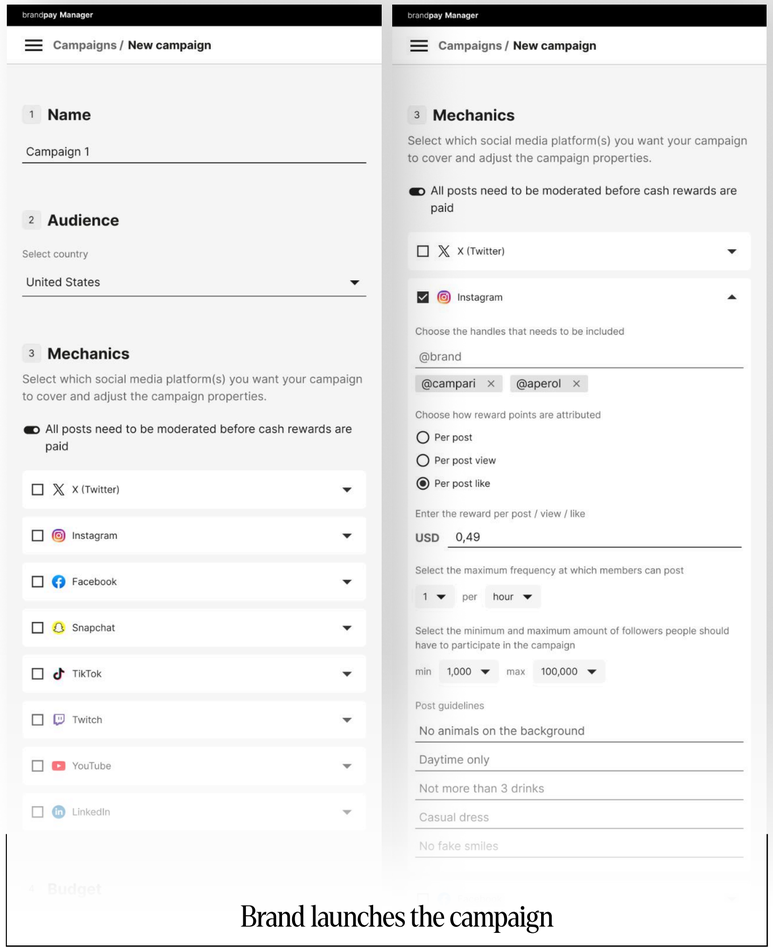
BlockTrust Hong Kong, Sydney — 2023

From customers to followers, from followers to influencers

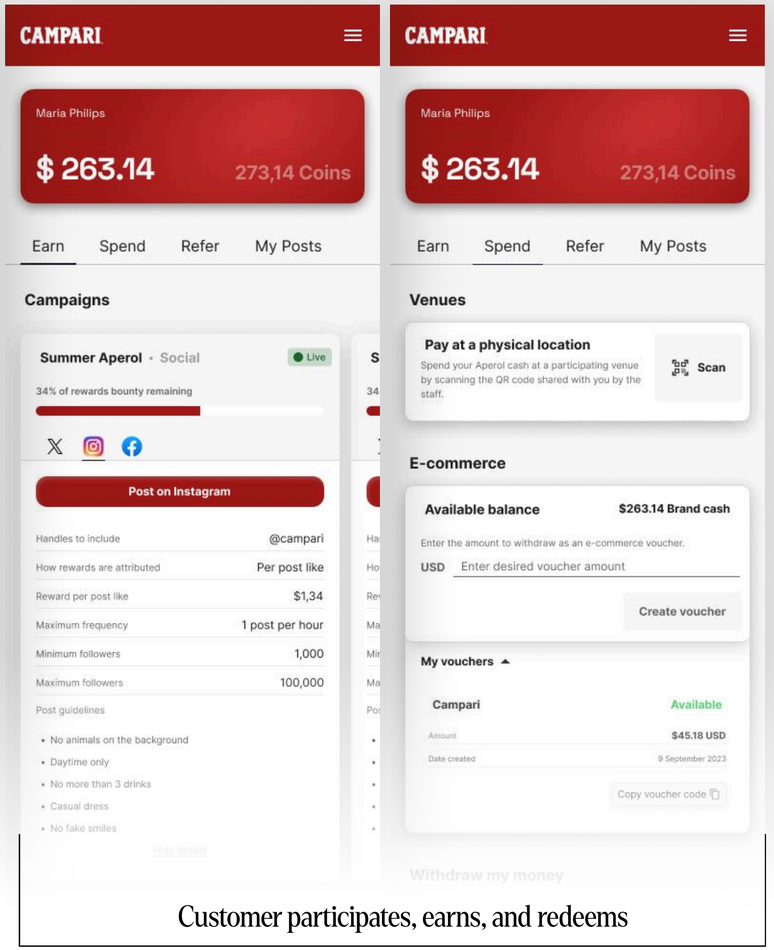
Product & UX/UI Design

Brands used BrandPay to encourage different types of social engagement. Each campaign appeared inside a ***brand-specific web experience.***

Customers discovered the campaign, followed the brand, participated, and received reward in their wallet, powered by BrandPay. Rewards could be spent at sponsored venues or converted into e-commerce vouchers, ***turning existing customers into followers, and followers into a network of influencers.***



Brand launches the campaign



Customer participates, earns, and redeems

Engagement path: from existing customer to organic influence

Active campaign phase

Customers

Already value the brand, buy its products, or attend its events, but their relationship remains mostly private and transactional. BrandPay gives them a clear reason to make that affinity visible and participate in a campaign.

Followers

Join the brand's audience to discover campaigns, understand the rules, and participate in return for clearly defined rewards. Following creates an ongoing connection beyond a single purchase or event.

Influencers

Create branded content, earn from its performance, and extend the campaign through their own networks.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

2020 – 2024

My honest thoughts on Web3 and Blockchain

*Blockchain is a fascinating technology. It has proven its value and earned real enthusiasm within a geeky audience. But **mass adoption remains extremely slow**. Here's what I learned working across Web3 projects, from failed pitches to profitable platforms and enterprise clients.*

People pursue value, not technology

Most people don't understand blockchain, don't enjoy it, and don't want to interact with it. Wallets, registrations and tokens demand too much attention and feel unnatural. Users instinctively try to skip them whenever possible.

Business is interested, consumers opt out

Ideas like digital twins or provenance tracking sound convincing to brands. But end users just want to buy, own, and resell products with minimal effort. When blockchain adds extra steps, they silently ignore it, and adoption quietly dies.

Reputation beats trends

In fantasy sports, gambling, and trading platforms, where trust and brand loyalty matter a lot, users don't care whether assets are "on-chain." They care whether the platform pays out, works reliably, and feels fair.

Blockchain that actually works

Adoption happens when blockchain removes real friction: international transfers without bank fees, transparent event ticket resale, or ownership that truly needs public verification. In these cases, blockchain works best as infrastructure, not as a feature to promote.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

Senior UX/UI Designer

Stimul

Moscow (Remote) 2021–2023

At Stimul, I designed a connected suite of internal products for a large distributed organization. The system linked executive campaign control, regional execution, engagement analytics, operational planning, and field work.

Systems

Executive Control

Central objectives translated into campaigns and assignments across hundreds of regional teams.

Engagement Analytics

Large-scale audience data made navigable through profiles, tasks, reports, posts, and notifications.

Operational Ecosystem

Four services, fourteen modules, and one design system supporting almost every operational workflow.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

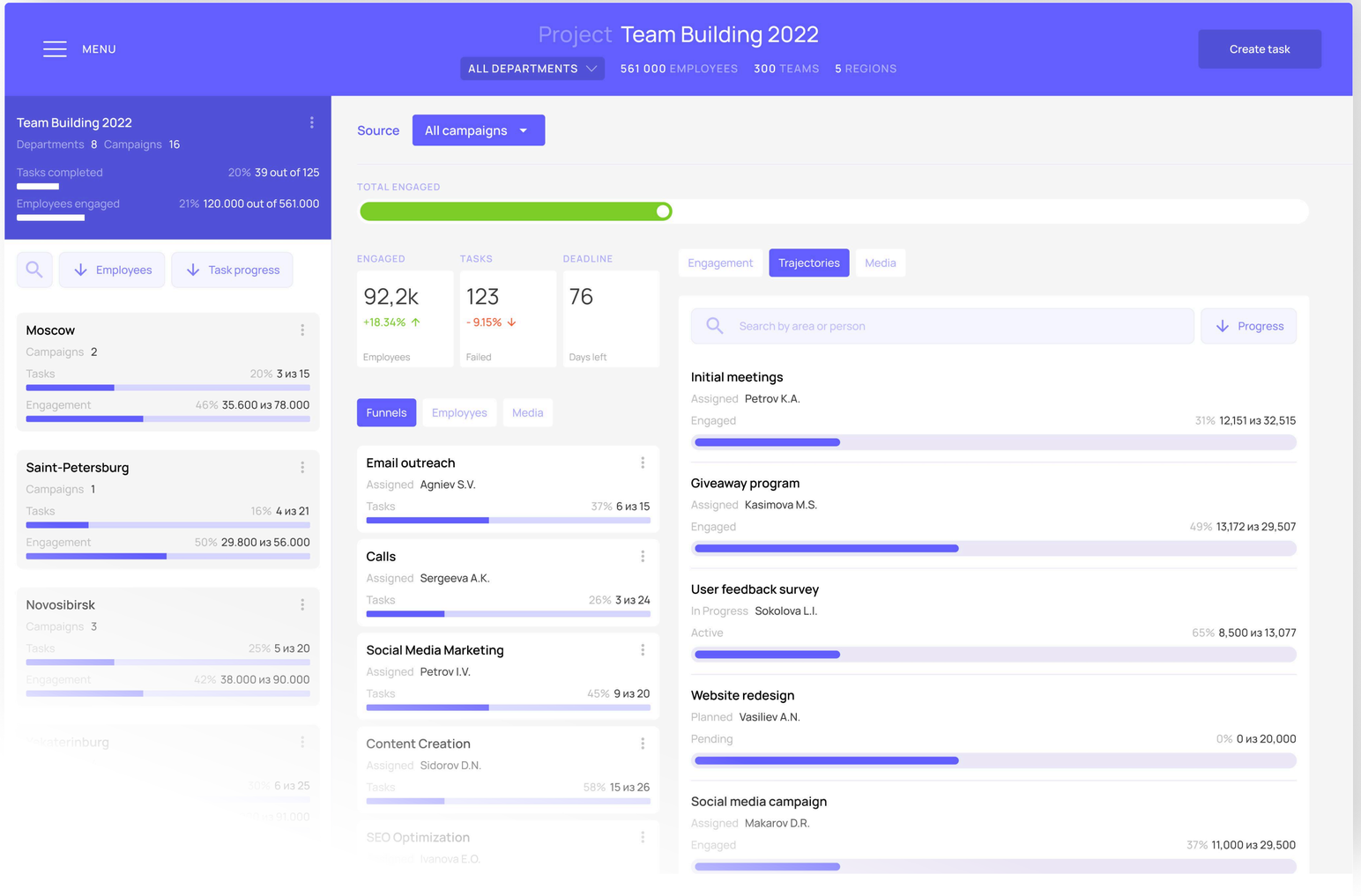
Stimul Moscow (Remote) — 2022

Delivering objectives to hundreds of teams

UX/UI Design

Senior leadership needed one place to see whether the organization was moving toward the same objective, and where regional execution was falling behind.

To connect central strategy with local execution,
I designed the control center.



Activity snapshot: June–July 2022

5	300	15
Regions	Teams	Concurrent campaigns

Control path: from central strategy to local execution

Leadership

Owns the grand strategy: defines the objective, scope, and deadline; monitors progress across the organization.

Departments

Convert strategy into campaigns, owners, deadlines, and locally relevant work.

Teams

Most of the work happens here. Teams coordinate daily execution across regions and offices, surface local constraints, and return progress through the same control loop.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

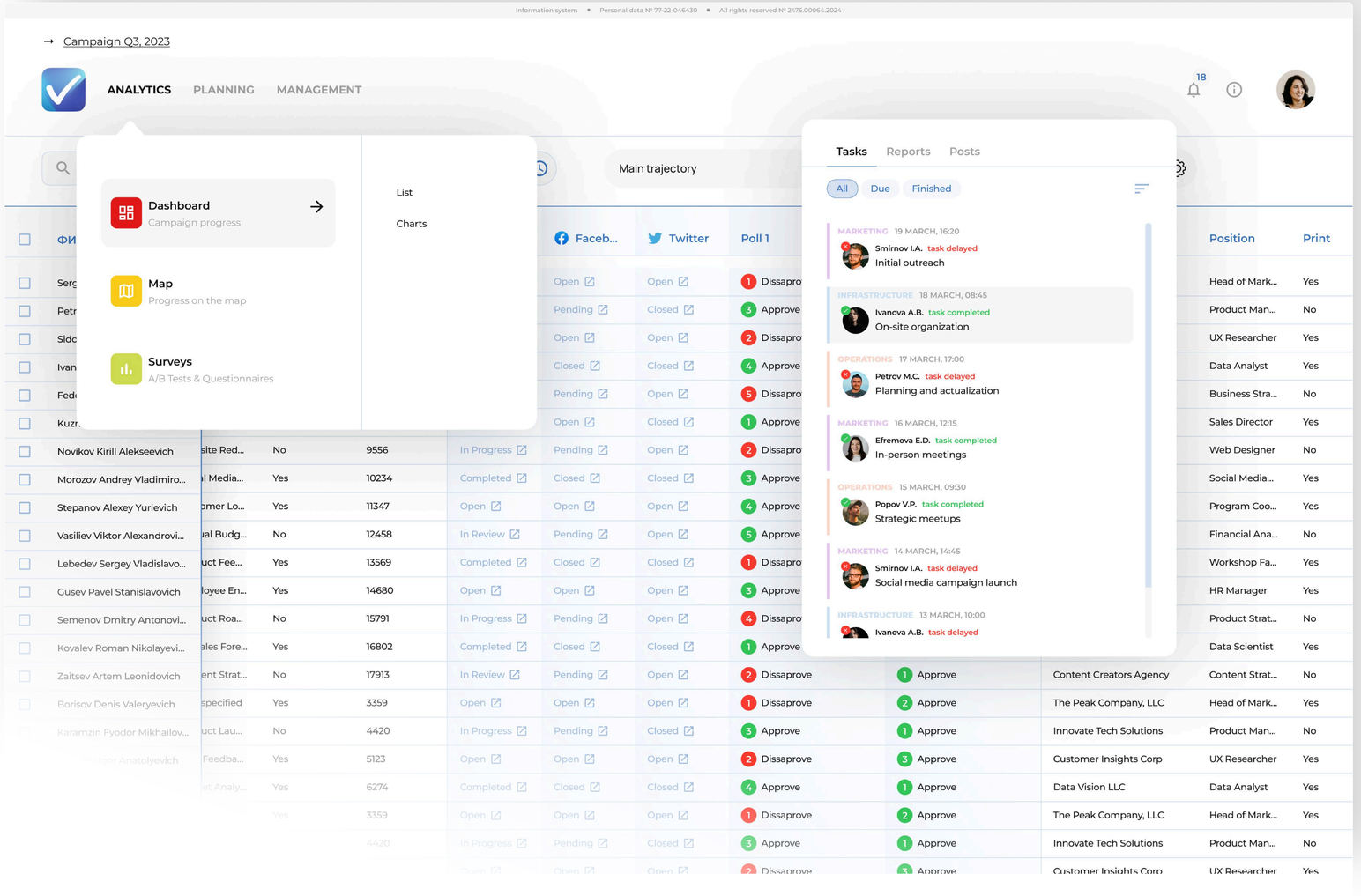
Stimul Moscow (Remote) — 2023

Enormously large dataset, extremely fast decisions

UX/UI Design

The dataset combined people, regions, roles, campaigns, and years of activity at a scale no one could read row by row.

*My job was to **make a data actionable** through tasks, reports, posts, and notifications. It needed to be dense where comparison mattered, spacious during navigation, and responsive.*



Decision path: from raw data to coordinated action

01
Unify sources
Bring profiles, regions, activity, and history into one model.

02
Structure data
Organize records around the questions people need answered.

03
Surface the signal
Reveal patterns without removing their context.

04
Turn insight into work
Create tasks, reports, posts, and notifications.

05
Close the loop
Return completed work to the same operational picture.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

2018 – 2025

Most teams misunderstand Agile

*Sprint planning, time estimates, and Jira tickets are often mistaken for Agile methodology, while **real user feedback barely reaches the development team**. I've seen this slow teams down more than any so-called "waterfall" ever did.*

Scrum is not Agile

Breaking work into sprints, estimating tasks, and tracking tickets is basic project management. It can be useful, but it's never enough. Agile is not a management framework, it's the ability to adapt product direction based on feedback from real users, taken directly by people who build.

Loop should be short

User feedback is direct and unfiltered, and usually points straight at the problem. But once it's translated through management layers, it loses precision. By the time it becomes a ticket, the original meaning is often blurred. Real Agile means collecting feedback as fast and as often as possible, and feeding it directly into the development pipeline.

Ownership over process

Real Agile shifts responsibility from process to people who are close to the system: designers and engineers should own decisions, trade-offs, and outcomes. When teams are trusted to react directly to feedback, progress accelerates and products improve faster.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

Stimul Moscow (Remote) — 2023

Platform that can do everything, design system that supports

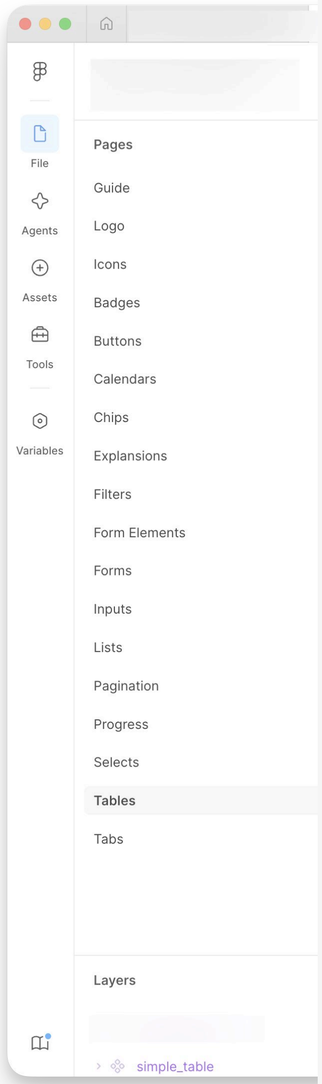
UX/UI Design

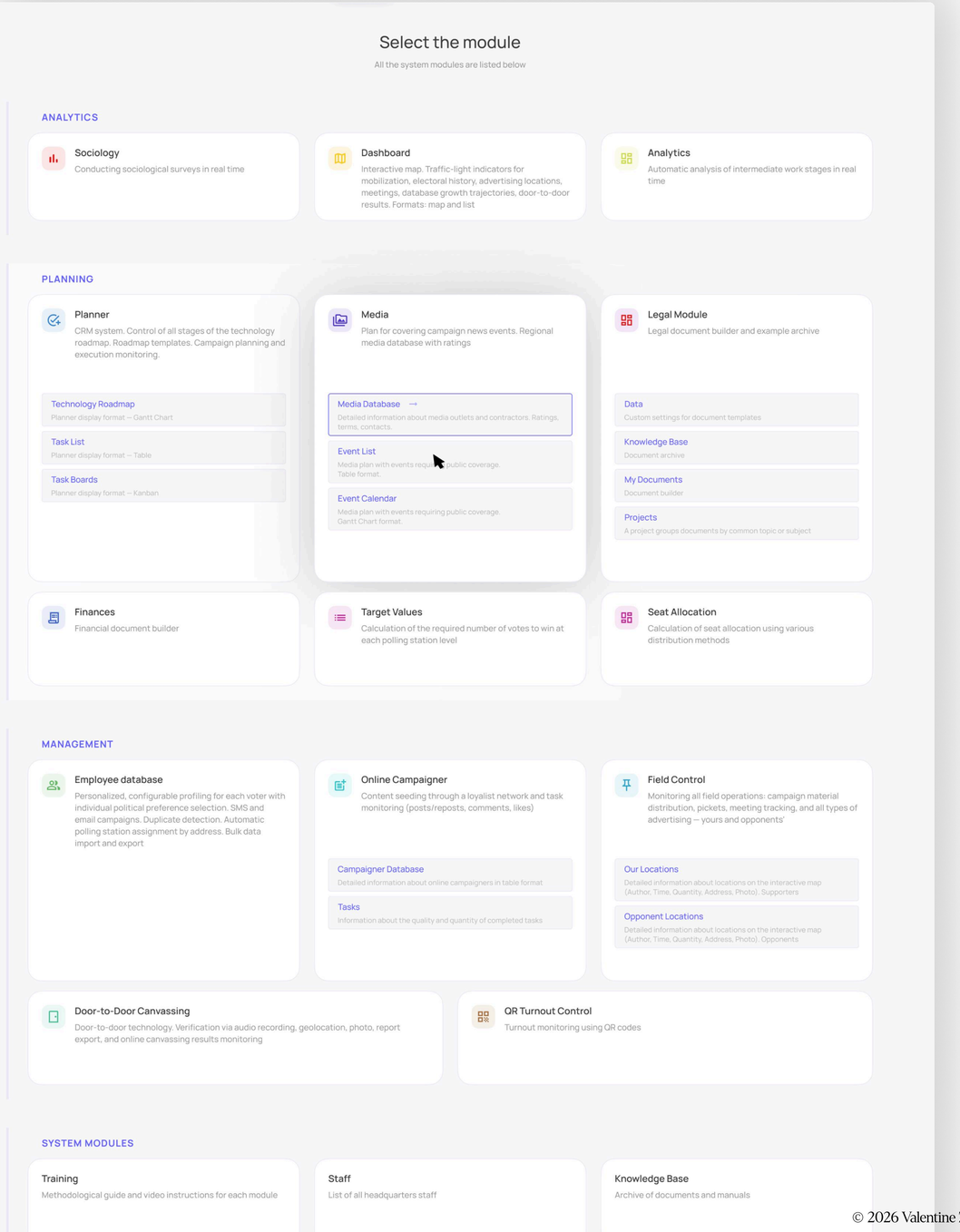
Analytics, planning, media, legal, finance, staffing, field operations, and campaign management lived inside one product ecosystem.

The design system expanded beside it: tables, filters, forms, states, navigation patterns, and components for nearly every workflow.

One system supported the organization, and another lay beneath it.

14	100+	50+
Connected modules	UI Components	User flows





All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

2018 – 2025

Never over-engineer a startup

Early-stage teams often try to build "the right architecture" from day one, before value and demand are proven. This approach works at the growth stage, but early on it's usually fatal. I learned this the hard way: ***speed of validation matters more than architectural elegance.***

Enterprise mindset kills startups

Tech-driven founders and experienced CTOs often come with an enterprise approach. It pushes teams toward complex, scalable solutions too early. When an MVP launches, after years of engineering, it often doesn't fit the market.

Microservices amplify complexity

Starting with microservices compartmentalizes both the product and the team's mental model into rigid boxes. Middleware complexity grows faster than the developers' ability to maintain the system. Precious time is spent polishing abstractions that exist only to satisfy technology for its own sake.

Change becomes expensive

Once real feedback arrives, even small changes take weeks. Every adjustment ripples through services and contracts and requires rebuilding essential parts of the architecture on the fly. At this exact point, many startups run out of time and budget.

Always start with a monolith

A simple, flexible monolith is faster to ship, easier to change, and cheaper to maintain early on. Extracting services makes sense only after demand and retention are proven, the business model is sustainable, and growth becomes the priority.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

UX/UI Designer

Freelance

Global (Remote)

2017–2022

From 2017 to 2022, I worked remotely with early-stage founders across real estate, interior design, fintech, marketplaces, and social products. I turned ambitious product concepts into clear information architecture, interactive prototypes, responsive interfaces, and implementation-ready design systems.

Projects

Qwincy

Commercial real estate platform for property discovery, private workspaces, and deal collaboration.

Pad Stylist

Interior design platform connecting inspiration, designer discovery, room planning, and shared projects.

LinkFace

Dating application built around profile discovery, matching, and private conversation.

NFT Marketplace (NDA)

Marketplace concept for collecting and trading tokenized digital assets.

Hades Money

Digital-asset product covering cryptocurrency discovery, portfolio management, transfers, and exchange.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

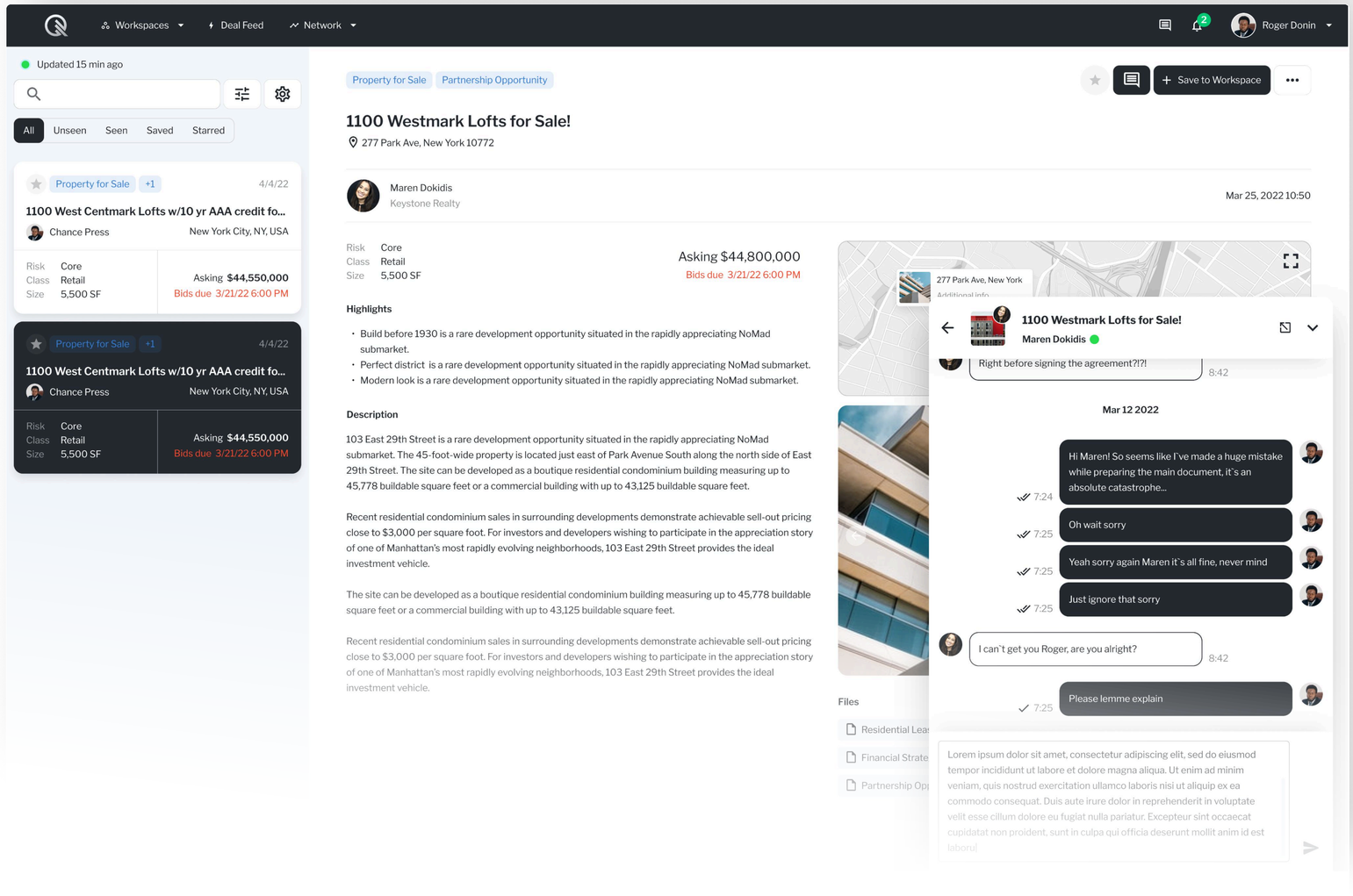
Qwincy Delaware, USA (Remote) — 2022

The complete commercial real estate dealspace

UX/UI Design

*Qwincy was conceived as a **private workspace for commercial real estate dealmakers.***

I designed the responsive product experience across opportunity discovery, property evaluation, team workspaces, document access, and private deal conversations.



Core deal journey

01
Find an opportunity
Search and filter the commercial property feed.

02
Evaluate the property
Review pricing, location, media, and deal details.

03
Save to a workspace
Organize opportunities around a team or mandate.

04
Collaborate privately
Discuss decisions and exchange documents.

05
Advance the deal
Track progress and coordinate the next step.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

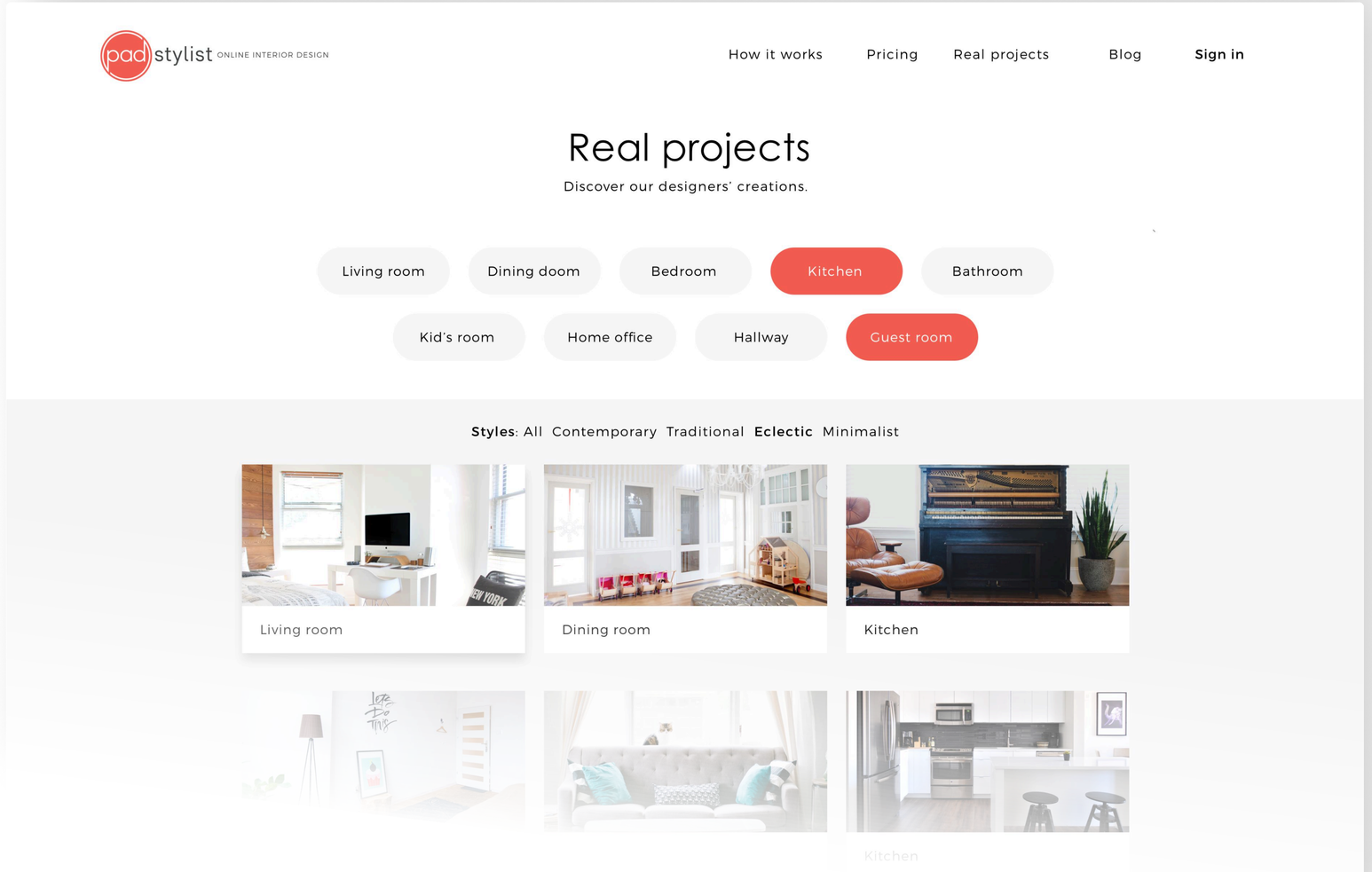
Pad Stylist London, UK (Remote) — 2017

The complete interior design workspace

UX/UI Design

*Pad Stylist was conceived as an **end-to-end interior design platform** connecting homeowners with designers, inspiration, portfolios, room planning, and project collaboration.*

I designed its public and signed-in experience across style discovery, designer profiles, project galleries, room briefs, and shared workspaces.



The complete interior design journey

Inspiration

Homeowners browse real projects by room and style. Filters turn broad taste into a clearer direction before the project begins

Designers

Clients explore profiles, portfolios, and preferred styles. The platform helps them understand who is suitable for their room and brief.

Projects

Selected ideas and designers move into shared workspaces. Clients and professionals coordinate briefs, decisions, communication, and progress.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

2014 – 2021

“Let’s SaaS this part of life” is not a business model

*Most of the ~10 early-stage startups I worked with **collapsed under their own complexity** before users ever understood the value. Across web and mobile, B2B and B2C, I’ve seen this pattern repeat.*

Not everything should become software

Startups often try to wrap complex, field-specific human interactions into a single platform. In practice, complexity exceeds the maintainability threshold before public release. After launch, users don’t understand what the product is actually for, get annoyed, and abandon it.

Keep it simple, really

If an MVP requires 10 complex flows just to start communicating value, the idea is already dead. No amount of UX polish, clean code, or project management will save a product that can’t explain itself in one interaction.

A better investment

Solving one painful problem is cheaper, clearer, and far more likely to earn real usage than trying to “cover the whole process.” Successful products ship one thing, validate demand, and only then grow the business on top of a proven solution and an existing audience.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

2014 – 2024

Great graphics don’t fix broken experience

*What I repeatedly saw across startups and mature teams alike: strong visuals creating a **false sense of progress**, while core product and UX problems remained unsolved.*

Good products don’t need onboarding

If users can’t find what they’re looking for, this is a UX problem often caused by a deeper product issue. Changing components, colors, spacing, typography, or adding animations won’t solve it. Tooltips and “knowledge bases” won’t help either.

Stunning visuals trick teams

It’s human nature to get excited when what we’re building looks visually impressive. Confident branding, stellar UI, and slick motion design can deceive teams into feeling like they’re creating value, long before real validation happens.

Polishing UI creates UX debt

When teams overspend time and budget refining visuals instead of simplifying flows, reducing steps, and clarifying customer value, they create massive UX debt that will have to be paid later.

Clarity over beauty

Minimal, clean, and even slightly generic UI is often enough early on. The team first needs to prove that users understand the product, get value from it, and return to it. Only then does investing in visual identity actually compound results.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

Co-Founder, Face Control

RAF25

Saint-Petersburg 2018–2021

In 2018, I co-founded RAF25, a nightclub that quickly became a leading hard-techno venue in Russia. Located in a Soviet bomb shelter from the Cold War era, RAF25 formed a post-apocalyptic labyrinth of concrete and bombproof metal and hosted more than 200 events with artists from all over the world. The club was closed because of the pandemic and political pressure.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

ДОГМА

September 2020 26



RAF25

- KUBRAKOV
- A/DV
- GUTKEIN [LIVE]
- ROMA PTASHENKO

K – HALL

- RTE DISSIDENT
- HQCONTROLLER
- GETMIOUD
- ZEYGMAN

На афише:
Клаудия Андухар «Yanomami»

Saint Petersburg
Svetlanovskiy pr. 3/4

Догма
26.09.2020

RAF25

Tickets – raf25.com

16.03

RAF25

- HÉCTOR OAKS
- KYK
- INWARD FLIP
- HELLPER

Engine Room

- ROMA PTASHENKO
- CURRENT CALL
- DANYA

Right Round



All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

2011 – 2020

Real–world trust is harder than software growth

These lessons come from running a nightclub and event agency, promoting festivals, selling 50k tickets, and being shut down by the global pandemic. ***I paid for them myself.***

Community can't be faked

When people don't trust your authenticity, no promotion strategy will save the brand from being abandoned and laughed at.

Consistency beats stars

If people show up every weekend regardless of the headliner on the stage — you've built something real.

Offline always wins

Convincing 100 people to show up physically is harder than getting 1000 clicks. But when they come, they bring 10 times more growth.

Some risks can't be mitigated

During the pandemic, independent event agencies and venues received no government support. It all ended overnight – regardless of execution quality.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov

Never for money, always for love.

All thoughts shared here reflect my personal experience in software development. They are not intended to devalue other approaches or experiences.

© 2026 Valentine Zubkov