

Two thirds of our customers came once. So I built a loyalty product that knows nothing about them.

Canna Cash is a cash-back program running on our POS partner API, live at the counter. It exists because the software that does this properly costs more per month than a first-year single store can justify, so the constraint was set before the design started. Five decisions shaped it, and each one had a more conventional alternative I turned down for a stated reason. This is that reasoning, including what each choice cost and what would make me choose differently.

Built and operated at Pacific Wellness, Sherman Oaks, California — live at cash.pacificwellness.uk.

Ticket with the card \$68.58 Average, against \$36.66 without it.	Margin cost 3 pts 50% against 53% undiscounted. A vendor promotion costs nine.
---	--

The problem, stated honestly

Five months of transactions say the store is growing on returning customers while new-customer intake sits flat. New customers per day have moved between 8.4 and 13.3 since April and gone nowhere in particular. Everything that looks like growth is repeat traffic stacking on a fixed intake.

Which would be fine, except repeat behaviour is getting worse, not better. Thirty-day repeat rate by monthly cohort, censoring-corrected: 41.3%, then 29.7%, 28.0%, 25.7%,

23.0%. And **66% of 1,740 customers came exactly once.**

The obvious lever was already in place and does nothing. Our first-time-purchase discount has been used by 276 customers, and their repeat rate is **34.1% against 33.5%** for everyone else. A first-time deal buys a visit. It does not buy a customer, and we had been paying for it as though it did.

So the brief was a returning-visit mechanic that a bartender can explain in one sentence during a Friday rush, that costs less margin than the vendor promotions already running, and that does not create a database I would rather not own.

Five decisions

Spend

Build it for the cost of printing, rather than buy the category

There is a mature, genuinely good software category for exactly this — compliant loyalty and customer data platforms built for cannabis retail, with consent management, segmentation, SMS and attribution already solved. I didn't buy one.

Why not. Cash, and the honest arithmetic behind it. Third-party pricing for that category is commonly cited in the high hundreds to low thousands per store per month. Our gross profit — revenue after product cost, and before rent, payroll, security, utilities and the POS itself — runs about \$15,100 a month, and in a store this young the fixed costs behind it account for all of it. A loyalty subscription isn't carved out of profit; it's carved out of a pool already spoken for, it lands at the same size in a bad month as a good one, and it would be doing so before it had proven it retains anybody. Every dollar not committed there is a dollar in inventory, which is the only thing on the floor a customer can actually buy.

What it cost. No SMS automation, no attribution, no segmentation, no win-back flows — and I own the maintenance of everything I built instead. Those are real capabilities and I'm not pretending otherwise.

What would change it. A second store, or a month where the line item can be justified against gross profit rather than hoped at. At that point buying beats building and I'd buy — as a customer who knows precisely what the thing costs to build, which tends to make for a shorter sales cycle and a more specific set of questions.

Cash back with an expiry, not points

Spend over a threshold, get a fixed dollar amount off your next visit within a window. The Kohl's Cash structure rather than the airline structure.

Why not points. Points accrue silently and forever, which makes them a liability that grows without anybody deciding to grow it, and they defer the return visit indefinitely instead of scheduling it. Our POS proved the point before I could argue it: its native loyalty module was holding **88,179 points matching nothing** — no campaign, no redemption path, accruing indefinitely against customers who had no idea they had them. I disabled it.

What it cost. Expiring value annoys a subset of customers, and an expiry is a harder sell at the counter than "you're earning points." Accepted, because a liability I can size beats a liability I can't.

No customer database

The program holds no names, no purchase histories, no identifiers. A phone number, when one is taken at all, is used once to send a single message and destroyed after it sends.

Why. This is a risk decision before it is anything else. A single-store cannabis retailer operating in a federally illegal industry, holding a named list of who bought what and when, has created a subpoena target, a breach liability and a CCPA obligation — and has done it to run a punch card. The asset is worth less to me than it is worth to a plaintiff, a regulator or an attacker.

How I tested it. I ran the design past a constructed panel of three opposing positions — a privacy absolutist, a CRM advocate arguing hard for full capture, and a neutral operator — specifically to attack my own preference rather than confirm it. The CRM case is genuinely strong: segmentation, win-back, attribution, lifetime value. What came out of it was a number rather than a verdict. The strongest argument I could build for full capture conceded that a paper card reaches roughly 85% **of the value** of a full CRM at this scale. That is an estimate produced by an exercise, not a measurement, and I'd treat it as such — but it converted the question from a belief into a price.

What would change it. Multiple stores, consent collected properly at the counter, and a tier structure customers actively want to be identified for. At that point the missing 15% is worth building the infrastructure to hold, and the liability is spread across a company that can carry it. At one store it isn't, and I'd make the opposite call without much argument.

Medium

A physical card, not an app

Five hundred printed cards. A budtender writes the date in a white box on the back. That is the entire enrollment flow.

Why not an app. Install friction at the counter is the highest-attrition step in any retail loyalty program, and it lands precisely when the customer wants to leave. A card is handed over in two seconds and requires no account, no download, no identity. It also fails safe: if our system is down, the card still works, because the card *is* the system.

What it cost. No push notifications, no automated win-back, no per-customer reporting. I can see what the program does in aggregate and nothing about who is doing it.

Threshold

A basket minimum set one cent low on purpose

The minimum is \$28.80 pre-tax rather than \$28.81, and the discount is \$3.60 rather than \$5.00.

Why. Our best-selling all-in-one vape lands at exactly \$28.81 pre-tax and would otherwise sit precisely on the threshold, and the platform does not document whether its comparison is strict or inclusive. Rounding down costs nothing and removes the question. On the discount side, \$3.60 pre-tax reaches the register as \$5.00 off and holds at every basket size; \$5.00 pre-tax would have handed the customer \$6.94. The full derivation is in the [pricing migration case study](#).

What it cost. Nothing, which is the point. Undocumented platform semantics are a design input, not an excuse.

What it's doing

Tickets carrying the Canna Cash promotion average **\$68.58** against \$36.66 without it, at a **50% margin** versus 53% undiscounted.

The margin comparison is the one that matters, and it is clean: three points of margin, against nine points for the vendor-funded brand promotions running alongside it — promotions that, measured at store level, [give up about \\$8,000 a year](#) without producing traffic.

The basket comparison is not clean, and I'd rather say so than let someone find it. A \$28.80 minimum *selects* for larger carts by construction, so some of the gap between \$68.58 and \$36.66 is mechanical rather than caused. The honest claim is narrower than the headline number: the program costs three points of margin instead of nine, and the customers using it are worth more per visit than the customers who aren't. Whether it made them that way is a question this data can't answer, and answering it properly would need a holdout the store is too small to run.

The part I didn't plan for is that the name took. Customers ask for Canna Cash by name at the counter and staff use the term with each other without being told to. That's anecdotal and I can't put a number on it, but a program with a name in the store's own vocabulary is getting distribution that no amount of signage buys — and it's the clearest argument I have that the design work on the card was worth doing rather than shipping something functional and plain.

How it's built

The web front end at cash.pacificwellness.uk is the referral and redemption surface. Behind it, the POS partner API — two header credentials, one issued by the vendor and one self-serve, with promotions, rewards, products and store endpoints. I wrote a read-only puller with the HTTP verb hardcoded to GET so that an early version could not write to a live store under any circumstance, handling pagination and the platform's habit of formatting dates as calendar strings on some endpoints and epoch milliseconds on others.

Full write-back sync is blocked on a partner key the vendor issues on request, which takes weeks. That is a dependency I should have identified and requested on day one rather than the day I needed it, and it is the main thing I'd do differently.

The card itself went through two design directions built and rejected before the one now in customers' hands: a dark currency-styled treatment that I liked and that my graphic designer told me was unreadable under retail lighting, and the lighter direction that replaced it. She was right. The designer overruled me on legibility and the decision took about thirty seconds, which is the correct amount of time to spend defending an aesthetic preference against someone whose eyes are better than yours.

What I'd change in the product

Most of the above is work I did because the platform doesn't do it. The gaps are general ones:

- Expiring cash-back should be a first-class promotion type. Every POS I've used models points natively and makes the Kohl's Cash structure something you

assemble out of a basket minimum, a fixed discount and a manual date — which is exactly the assembly I did by hand.

- A points balance is a liability, and it should be reported as one. 88,179 points matching nothing sat in our system with no aging report, no redemption-rate view and no way to size the exposure. A retailer should be able to see what their loyalty program owes.
- There should be a zero-PII loyalty mode. Not every retailer can safely hold a customer list, and in federally illegal industries that isn't a preference. A program that identifies a card rather than a person would serve a large share of this market, and no vendor sells it.
- Single-use contact handling should be a supported pattern — take a number, send one message, destroy it, with the destruction logged rather than promised. Retention tooling assumes permanent capture as the only model, and permanent capture is a liability the vendor doesn't carry.

None of this is an argument against customer data or against the platforms that sell it. At scale, with consent and a real value exchange behind it, the CRM wins and I'd buy rather than build. The argument is that the platform should support both shapes — because the retailer who can't safely hold the data, or can't yet carry the line item, is the one who most needs a program that works without either.

Kyle Williams

Managing Partner, Pacific Wellness — Sherman Oaks, California. Retail operations, purchasing, financials, POS administration and analytics. Fifteen years in cannabis, from delivery in 2011 through a decade of wholesale brokering.

[linkedin.com/in/kylewilliams021](https://www.linkedin.com/in/kylewilliams021)