

PayNow Requests



Privacy-Preserving Payment Requests for Singapore

Sivasubramanian Ramanathan

*Product Owner | Fintech & Innovation
Ex-BIS Innovation Hub Singapore*



Seeking Opportunities in Singapore

I am looking for roles in **Product Management, Fintech, Payments, RegTech, and Digital Assets.**

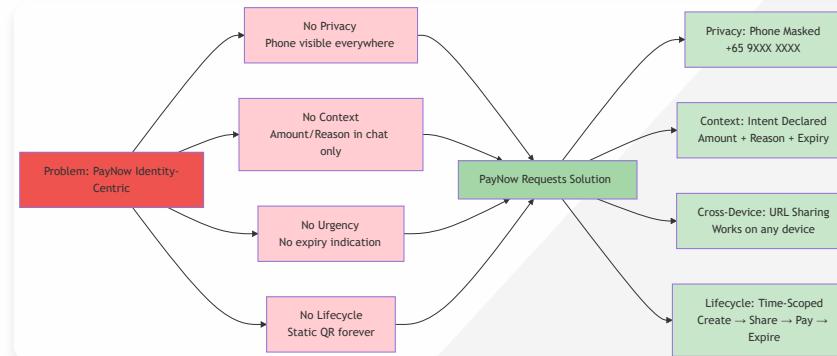
"I am not just a Product person. I **build**."

I specialize in taking complex financial systems and structuring them into reliable products that bridge **policy** and **code**.

I care deeply about building products that create real impact.

The Problem Solved

What Problems Does PayNow Requests Address?



Problems → Solutions

Problem	Solution
No Privacy	Phone masked (+65 9XXX XXXX) in shared links
No Context	Intent declared: Amount + Reason + Expiry
No Cross-Device	URL-encoded sharing works on any device
No Lifecycle	Time-scoped: Create → Share → Pay → Expire

The Paradigm Shift

From Identity-Centric → Intent-Centric

Current (Identity-Centric):

"Here is my phone number (+65 91234567), pay me"

New (Intent-Centric):

"Alice requests SGD 50.00 for Friday dinner at Nando's, valid until 8 PM"

“This treats the **payment request as a first-class object** — with lifecycle, state, and metadata — rather than just a QR code utility.

”

Push vs Pull Payment Models

Why India's UPI Discontinued P2P Collect Requests

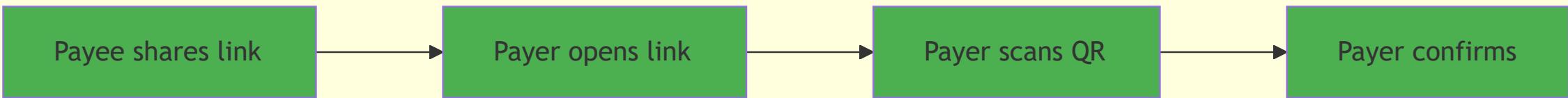
IMPORTANT: NPCI permanently discontinued UPI "collect request" for P2P transactions from **October 1, 2025** due to rising financial fraud.

- P2P Collect: **Discontinued** (85% fraud increase year-over-year)
- Payer-Initiated Only: All personal transfers must now be **push** payments

Push vs Pull: Visual Comparison



Push Payment - SAFE



Key Difference

Aspect	Pull (UPI Collect - BANNED)	Push (PayNow Requests - SAFE)
Initiation	Payee sends notification	Payer opens link voluntarily
Control	Payee controls flow	Payer has full control
Verification	Single-tap approval	Scan QR in bank app + verify
Fraud Rate	0.05% - 0.12%	0.01% - 0.03%

Competitive Analysis 📱

What About DBS PayLah! Request Feature?

Feature	DBS PayLah!	PayNow Requests
Cross-Bank	✗ DBS/POSB only	✓ All Singapore banks
App Required	✗ Must have PayLah!	✓ Works via SMS link
SMS Fallback	✓ Downloads app prompt	✓ Opens in any browser
Privacy	✗ Phone visible	✓ Phone masked
Intent Display	Amount only	✓ Amount + Reason + Expiry
Open	✗ Closed ecosystem	✓ Web-based, open

PayLah! Innovation: When a recipient doesn't have PayLah!, it sends an **SMS asking them to download and pay** — but this is still a **closed ecosystem** requiring the DBS app.

PayNow Requests Advantage: Works with **any banking app** — DBS, OCBC, UOB, Standard Chartered — and only requires a **browser** to view the request.

How PayNow Requests Works

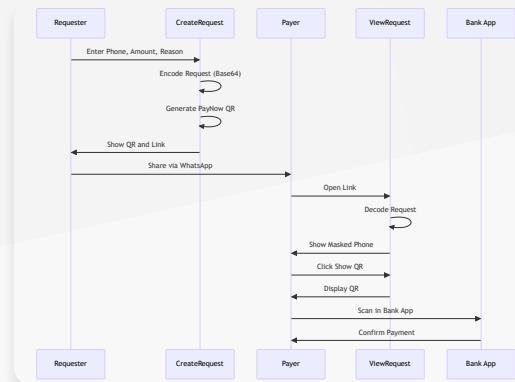


For Requesters

1. Fill in: Phone, Amount, Reason, Expiry
2. Generate QR + Shareable Link
3. Share via WhatsApp/SMS/Email

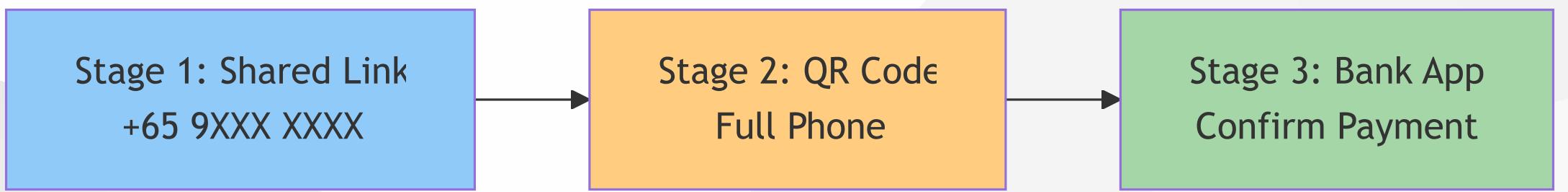
For Payers

1. Click shared link (opens in browser)
2. View: "X requests SGD Y for Z"
3. Click "Show QR to Pay"
4. Scan in **any** banking app
5. Verify and confirm



Privacy: Progressive Disclosure

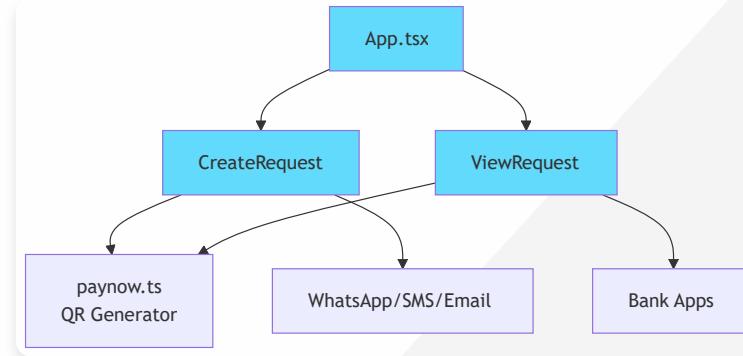
Three-Stage Privacy Model



Stage	Phone Visibility	Purpose
Shared Link	+65 9XXX XXXX	Prevents phone harvesting in group chats
QR Code	+65 91234567	Required by PayNow for payment execution
Bank App	+65 91234567	Standard PayNow confirmation flow

Technical Architecture 🛡

Client-Side Only, Serverless Design



Tech Stack

- **Frontend:** React 19, TypeScript, Vite
- **Styling:** Tailwind CSS
- **Routing:** React Router (hash mode for GitHub Pages)
- **QR Generation:** EMVCo-compliant PayNow QR
- **Deployment:** Static hosting (GitHub Pages, Netlify, Vercel)

Fraud Risk Assessment

PayNow Requests vs Discontinued UPI Collect

Aspect	UPI Collect (DISCONTINUED)	PayNow Requests
Payment Model	Pull (payee-initiated)	Push (payer-initiated)
Initiation	Push notification to app	User clicks link voluntarily
Verification	None (direct approval)	Must scan QR in bank app
Phone Privacy	Full phone visible	Masked in link
Fraud Risk	HIGH (85% YoY increase)	LOW-MEDIUM

Why PayNow Requests is Safer

- Push Model:** Payer initiates via scan (same as discontinued UPI Collect)
- Manual Banking App:** Human-in-the-loop required
- Clear Intent Display:** Amount, reason, expiry shown
- Phone Masking:** Reduces social engineering risk
- No Artificial Urgency:** Expiry is UX-level only

Real-World Use Cases



Problem → Solution Examples

Group Dinner

- *Before:* "How much do I owe? What's the reference?"
- *After:* Click link → See SGD 45 for "Friday dinner at Nando's" → Scan & Pay

Freelance Invoice

- *Before:* "Pay invoice #2025-001 to +65 91234567"
- *After:* Click link → See SGD 750 for "Logo design - Invoice #2025-001" → Scan & Pay

Event Payment

- *Before:* Organizer's phone shared publicly to 50+ people
- *After:* Link shared, phone masked until payment

Marketplace

- *Before:* Seller's phone visible to strangers on Carousell
- *After:* Link with masked phone, revealed only on payment

Engineering Decisions (ADRs)

Key Architecture Decisions

ADR-001: URL-Based Data Encoding (No Backend)

- **Decision:** Base64 encode request data in URL fragment
- **Benefit:** Zero infrastructure, cross-device sharing, privacy (no server logs)

ADR-002: Hash-Based Routing

- **Decision:** Use React Router with hash mode
- **Benefit:** GitHub Pages compatible, no server config needed

ADR-003: Progressive Phone Disclosure

- **Decision:** Mask phone in link, reveal only in QR/bank app
- **Benefit:** Privacy preservation while maintaining PayNow compatibility

“**Philosophy:** Work within existing constraints rather than fighting them”

What This Does NOT Do

Honest Limitation Communication

Aspect	Capability
Execute Payments	✗ All payments occur in banking apps
Deep-Link Bank Apps	✗ Requires manual scan
Enforce Expiry	✗ UX-level only, expired QRs still scannable
Cryptographic Anonymity	✗ Base64 is encoding, not encryption
Fraud Prevention	✗ Users must verify through separate channels

What This DOES Do

Aspect	Capability
Reduce Ambiguity	✓ Clear intent: amount, reason, expiry
Protect Privacy	✓ Phone masked in shared links
Enable Cross-Device	✓ URL works on any device
Work with All Banks	✓ DBS, OCBC, UOB, Standard Chartered
Deploy Anywhere	✓ GitHub Pages, Netlify, Vercel

Market Opportunity

Singapore PayNow Ecosystem (2024 Data)

Metric	Value
PayNow Users	3.5 million (60% of population)
Annual Transactions	100+ million
Annual Volume	SGD 15 billion
Growth Trend	+40% YoY

Market Gap Analysis

Competitor	Payment Requests	Privacy	Cross-Platform
DBS PayLah!	Yes (closed ecosystem)	✗	✗
OCBC Pay Anyone	Partial	✗	✗
GrabPay	Partial	✗	✗
This Project	✓	✓	✓

Development Roadmap



Implementation Status

Phase 1: Core Functionality ✓ Complete

- [x] Request creation form (phone, amount, reason, expiry)
- [x] PayNow EMVCo QR generation
- [x] Base64 URL encoding/decoding
- [x] Phone masking display
- [x] Share buttons (WhatsApp, SMS, Email, Copy)

Phase 2: UX Polish ✓ Complete

- [x] Responsive mobile-first design
- [x] Expiry validation and warnings
- [x] How-to-pay instructions
- [x] Privacy notices

Phase 3: Future Enhancements 📝 Planned

- [] LocalStorage request history
- [] Dark mode support

Live Demo 

Try It Now

Create a Payment Request:

1. Go to: paynow-demo.example.com
2. Enter: Phone, Amount, Reason, Expiry
3. Generate QR and shareable link
4. Share via WhatsApp/SMS/Email

View a Sample Request:

1. Click any shared request link
2. See: Masked phone, amount, reason, expiry
3. Click "Show QR to Pay"
4. Scan with your banking app

"All client-side. No backend. No data collection.

”

About the Builder



Sivasubramanian Ramanathan

Product Owner | Fintech, RegTech & Digital Innovation

I specialize in taking messy, real-world complexity and structuring it into reliable products that bridge **policy** and **code**.

Experience:

- BIS Innovation Hub Singapore — Cross-border payments research
- Stablecoin Clearing & Settlement Engine — Full-stack Web3 development
- PayNow Requests — Privacy-preserving payment infrastructure

Philosophy: Build products that work within real-world constraints, don't overstate capabilities, and create genuine user value.

Let's Connect

I am ready to bring this level of engineering rigor and product thinking to your team.

-  **Portfolio:** sivasub.com
-  **LinkedIn:** linkedin.com/in/sivasub987
-  **Code:** github.com/siva-sub/PayNow-Requests
-  **Docs:** [Full Documentation](#)

Live Demo: paynow-demo.example.com

Research: See [UPI_RTP_FRAUD_RESEARCH.md](#) for detailed fraud analysis