APM ASSIGNMENT · APRIL 2026

AI-Driven Web Automation for SaaS User Management

A Scalable Solution for Non-API SaaS Applications

CANDIDATE
Goutham Shaji C K

ROLE
Associate Product Manager

COMPANY
CloudEagle.ai

DATE
April 2026

PROBLEM BACKGROUND

The World is SaaS-First. But Management is Still Manual.

Organizations use 100+ SaaS tools. Centralised user governance is critical — yet widely broken.

🔐 Security & Compliance

Access control, offboarding discipline, and audit readiness depend on timely, accurate user data across all tools.

💸 Cost Optimisation

Unused licenses drain budgets. Visibility into who is active — and who isn't — requires reliable, structured data.

⚙️ Operational Efficiency

Onboarding, role changes, and offboarding across dozens of tools creates massive overhead for IT teams.

The Core Gap: Many SaaS applications expose **no API** for user data access or provisioning actions — leaving organizations with manual-only workflows and zero automation path.

ROOT CAUSE ANALYSIS

Why Do SaaS Apps Lack APIs for User Management?

This is a structural gap — not an oversight. Understanding why it exists is key to solving it.

🌱 Product Maturity

Early-stage SaaS prioritises UI over developer integrations. APIs get backlogged for later.

🛡️ Security Concerns

Exposing user management endpoints increases the attack surface and misuse risk.

🪜 Legacy Architecture

Older systems weren't built API-first. Retrofitting is complex and resource-intensive.

👥 Limited Resources

Small SaaS teams can't build and maintain robust APIs. Feature dev wins.

💼 Business Strategy

APIs gated behind enterprise tiers — a revenue lever, not an oversight.

Key Insight: "Admin portals are designed for **human interaction**. APIs are designed for **machines**. This structural mismatch creates the automation gap we're solving."

PROBLEM DEPTH

What Makes Automating SaaS Portals So Hard?

Even when automation is attempted, 8 compounding challenges derail it.

🔄 Dynamic & Changing UI

React/Angular apps frequently update DOM structure, breaking static selectors instantly.

🔐 Auth Complexity

MFA, OTP via email/SMS, session expiry, and device-based restrictions all block automation flows.

🤖 CAPTCHA & Bot Detection

Systems flag automated behavior and issue challenges that block the entire workflow.

🧩 UI Variability

"Users" vs "Members" vs "Team" — every SaaS uses different labels and navigation paths.

📄 Paginated Data

User tables span multiple pages or lazy-load on scroll. Extraction must handle all patterns.

❓ No Feedback Signals

Unlike APIs, UIs give no structured success/failure response — success must be inferred from UI state.

⏳ Action Reliability

Loading states, hidden elements, and timing issues cause silent failures in automated clicks.

⚠️ Rate Limiting

Excessive automation triggers security alerts and may temporarily lock accounts.

USER DISCOVERY

Identifying User Segments

Three distinct groups are affected — but not equally.



IT Admins

Need: Manage access across all tools

Pain: Manual portal logins daily, inconsistent workflows, no central view.

HIGH FREQUENCY · HIGH SEVERITY



Security & Compliance Teams

Need: Ensure access control and audit trails

Pain: Delayed offboarding creates zombie accounts and compliance gaps.

MED FREQUENCY · VERY HIGH SEVERITY



Finance / Procurement

Need: Optimise SaaS spend

Pain: No visibility into license utilisation; all data locked behind admin portals.

MED FREQUENCY · MED SEVERITY

Selected User: "I will focus on **IT Admins** because they are the most underserved by current tooling — they face this problem **daily**, across **every SaaS tool**, and bear the full operational burden. Solving for them unblocks downstream security and cost problems too."

Reason: High frequency · High severity · Gatekeeper to all downstream issues

USER NEEDS

Reporting the Real Problems

For *IT Admins and Security Teams* responsible for managing user access across multiple SaaS tools — key problems are:

PROBLEM 1

Manual user management across SaaS tools is slow, error-prone, and does not scale.

Why it happens: No APIs → admins must log into each portal individually → every tool has a different UI and workflow → no central system to manage them all. Result: delays in onboarding/offboarding + rising operational overhead.

PROBLEM 2

Delayed or missed deprovisioning creates serious security and compliance risks.

Why it happens: Offboarding requires manual action across every tool → human dependency + no real-time visibility → high friction discourages immediate cleanup. Result: ex-employees retain access → major security vulnerability and compliance violations.

PROBLEM 3

Poor visibility into user data leads to wasted SaaS spend and inefficient license usage.

Why it happens: User data locked inside portals with no API → no structured or centralised view of active users, last login, or role utilisation. Result: paying for unused licenses with no path to optimization.

Core Insight: "The core issue is that SaaS user management is designed for **human interaction via UI** — not machine-driven automation — creating a scalability, security, and efficiency gap across the organisation."

PRIORITISATION

Choosing the Problem to Solve

Goal: Select ONE problem. Evaluate by frequency and severity.

Problem	Frequency	Severity	Optimize?
Manual user management is slow & doesn't scale	High	High	✅ Selected
Delayed deprovisioning creates security risks	Medium	Very High	Downstream
Poor visibility leads to wasted SaaS spend	Medium	Medium	Secondary

Optimize for: Frequency → High | Severity → High | Impact → Maximum

Selected Problem: "I will focus on **manual user management across SaaS tools** because it is **highly frequent and highly painful** — occurring daily, impacting every onboarding and offboarding, and forming the root cause of both the security risks and cost inefficiencies downstream."

SOLUTION DESIGN

Possible Solutions

For the problem: *Manual SaaS user management with no API access.*

Option 1

Pure RPA (UiPath / Robocorp)

How it works: Static, rule-based scripts automate browser clicks using fixed element IDs and XPaths.

❌ Breaks instantly when UI updates. High maintenance. Zero adaptability.

Option 2

Static Browser Scripts (Playwright)

How it works: Hardcoded Playwright scripts navigate portals and extract data based on known selectors.

⚠️ Faster to build but just as brittle. Every SaaS update requires re-scripting.

Option 3 — Recommended

AI Agent + Browser + Orchestration

How it works: An LLM reasons over live DOM/UI context and decides the next action. Playwright executes. A workflow engine handles retries, logging, and validation.

✅ Adapts to UI changes. Scales across apps. Auditable. Enterprise-grade.

Workflows Enabled: Scrape user data · Auto-provision new users · Auto-deprovision on exit · Handle MFA + CAPTCHA · Paginate across all data

SOLUTION EVALUATION

Evaluating the Trade-offs

Goal: Optimize for Impact → High · Effort → Medium · Risk → Low

Solution	Impact	Effort	Risk	Verdict
Pure RPA (UiPath)	Medium	Medium	High	❌ Brittle
Static Browser Scripts	High	Medium	High	❌ Fragile
AI Agent + Browser Only	High	High	Medium	⚠️ Complex
AI + Browser + Orchestration	High	Medium	Low	✅ **Best**

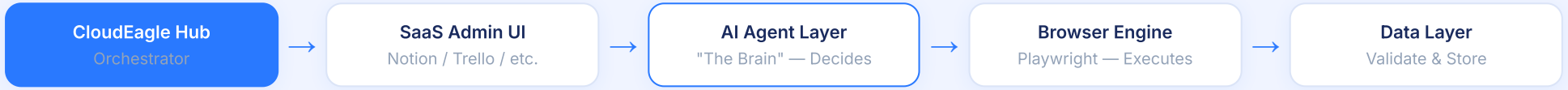
Trade-off: "This is worth doing because it balances adaptability and reliability. The upfront AI investment pays back exponentially as the system self-improves — reducing long-term maintenance to near-zero."

Best Solution: "I recommend **AI + Browser Automation + Workflow Orchestration** because it is the only approach that handles dynamic UI variability while remaining reliable, auditable, and scalable at enterprise level."

SCALABILITY & ARCHITECTURE

How the System Scales

One engine. Any SaaS. Built for change.



🔌 Modular Connector Framework

- → Standardised workflow: `Login → Navigate → Identify → Act → Verify`
- → Each new SaaS needs only ~10 lines of JSON "hints"
- → AI maps "Team" / "Members" / "Users" to the same concept
- → Parallel execution across multiple apps simultaneously
- → Event-driven: new hire → provision; exit → deprovision

🧠 AI Resilience Against UI Changes

- → Semantic detection: finds "Add User" button by meaning, not ID
- → Multi-strategy fallback: text → relative position → fallback selector
- → Continuous learning loop: failures logged → AI updates strategy
- → Computer vision as advanced layer when DOM changes drastically
- → Human-in-loop escalation for critical edge cases

Key Insight: "We scale not by standardising every SaaS UI — but by standardising the **intent** and letting AI adapt the **execution**."

PROOF OF CONCEPT

Prototype & Implementation Outline

POC validated on Trello — representative of common SaaS UI patterns.

✅ What Worked

→Browser auto-navigation to user management UI

→Successful structured data extraction from UI table

→Non-API automation proven technically feasible

⚠️ Challenges Found

→Static selectors broke when Trello updated its UI

→Email fields hidden/restricted for some user roles

→Inconsistent element structures across different pages

Key Learning: "Pure automation is insufficient — **AI intelligence** is required to handle variability and ensure reliability at scale."

```
# ── User Data Extraction ──────────────── login(credentials)
navigate_to("Members / Team / Users") while has_next_page(): users
+= extract_user_table() click_next() store(normalize(users)) # ──
Provisioning ──────────────────────── navigate_to("Add Member")
ai_fill_form(name, email, role) submit() verify_success() # ──
Deprovisioning ──────────────────────── search_user(email)
click("Remove / Deactivate") confirm() log_audit_trail(user,
timestamp) # ── Edge Cases ──────────────────────── if
element_not_found: retry_with_fallback_strategy() if
captcha_triggered: human_in_loop_escalation()
```

SUMMARY

Wrapping Up

"We focused on **IT Admins and Security Teams** who manage user access across multiple SaaS applications, facing challenges in manual, error-prone user management due to lack of APIs. We proposed an **AI-driven web automation system** combining intelligent agents, browser automation, and workflow orchestration — delivering scalable, reliable, and secure user lifecycle management across diverse SaaS platforms."

96%	98%	50%+	−70%
FASTER ONBOARDING	FASTER DEPROVISIONING	LICENSE COST SAVINGS	SECURITY RISK

CANDIDATE

Goutham Shaji C K

Associate Product Manager Candidate · CloudEagle.ai

gouthamshaji@gmail.com

linkedin.com/in/goutham-shaji-ck-879b771b9

+91 7025407167