Gennadii Nikitin

QA Engineer (Senior) · 8+ years · Manual + Automation + API

Asunción, Paraguay · Remote (UTC-3), overlapping CET and EST · relocation considered · available immediately
Contract: B2B, EOR or direct contract — negotiable; also self-employed for RU entities.
Telegram: @ni_genn · Email: g.nikitin14@gmail.com · WhatsApp: +595-993-353-602 (PY) · +7-931-288-7553 (RU)
LinkedIn: https://www.linkedin.com/in/gennadiy-nikitin-21337022b · GitHub: https://github.com/ge-a-nikitin

SUMMARY

QA engineer, 8+ years on desktop, web and mobile client-server products in high-traffic consumer promos and gamification — prize games, seasonal campaigns, loyalty mechanics — including a prize promotion with up to 3M+ unique monthly users. Owns the full SDLC on my projects: requirements and testability before development, test design, functional / regression / API / integration testing, and automation where it pays off — TypeScript on the company's in-house Playwright-based framework, k6 load testing on two projects. Leaves documentation and processes a team can pick up and reuse.

KEY ACHIEVEMENTS

- Cut regression from **2 working days for 2 testers to ~2 hours for 1 tester** by automating core desktop scenarios with Python + PyAutoGUI and image-recognition locators.
- **Took over most of the project-manager role on Pushkin 225 Years without the title** while the team was short-staffed; the promo **launched on schedule** with me as its only QA — launch readiness, daily statuses, backlog priority, requirements with the stakeholder product, research into new integrations (API and Kafka), mentoring new colleagues.
- An AI chatbot answering L1 inquiries from prepared documentation **closed 30% of requests without reaching the team** — tracked per channel and per project on a dashboard.
- Co-designed the department's automation template on the company's in-house framework eva.io and contributed code to it; **took it into five further projects of my own, and colleagues in adjacent teams used it on about ten more**.
- Moved performance testing from production-only, backend-owned runs to a **dedicated environment owned by QA**: benchmark, stability and maxperf (peak-load) scenarios in k6 on two projects, metrics in Grafana and Cosmos; the department-approved methodology with template and quickstart is now reused on new projects.
- Automated suites gated every release where we wrote e2e tests: a red pipeline held it until we knew why.
- Visual regression (screenshot tests) on **daily CI runs**, reports delivered to the corporate messenger.

CORE SKILLS

Testing (quality assurance): functional, regression, integration, system, UI, API, cross-browser, performance/load, visual regression (screenshots).

Test design & analysis: boundary value analysis, equivalence partitioning, state transition diagrams, requirements analysis, 3 Amigos sessions (PM, developer and QA together), testability review.

Test automation: TypeScript on eva.io (Selenium at first, then Playwright; most work in the hybrid period), Playwright (eva.io modules at work; the public e2e suite below), Python (PyAutoGUI, image recognition, Tillypad), parameterized tests, visual regression, Allure TestOps reporting, reusable module and page structure.

API & data: Postman, Insomnia, Swagger/OpenAPI, Charles, Fiddler, OWASP, SQL (SELECT, JOIN, GROUP BY, subqueries, INSERT/UPDATE), MSSQL replication verification.

Performance: k6 (benchmark, stability, maxperf), load profiles built from production logs, Grafana (plus the internal metrics service, Cosmos).

Platforms & infrastructure: Docker, BrowserStack, real iOS/Android devices, Android Studio, Chrome and Firefox DevTools.

AI in testing: LLM-assisted checklists and test scenarios, prompt iteration, Nestor AI (the company's own Cline-like agent) for test code in the IDE; shared the practice with the team.

QA process: test strategy, test plans and test cases, shift-left (requirements and testability review before development), release criteria and quality gates, defect lifecycle and triage, test documentation, QA metrics; Agile — Scrumban (Scrum/Kanban).

Tools & CI/CD: Git, GitLab CI with daily scheduled runs, Jira, Confluence, Allure TestOps, Redmine, Kafka topic integrations.

EXPERIENCE

Current status: left T-Bank for personal reasons.

T-Bank (ex Tinkoff) — Quality Engineer — April 2023 – September 2026

Web and webview applications inside Android/iOS apps: promo mechanics that grew into products — a monopoly-style prize game that reached up to 3M+ unique monthly users, longer-lived services, multiple third-party integrations. Sole QA for the first two years, then one of two equal QA engineers after the teams were rebuilt around larger products; owned the test strategy for my own projects. Work ran in Scrumban, the team's name for a Scrum/Kanban hybrid chosen for project-based work. Release cycle: 3–7 months of development on staging, one release day, then follow-ups daily to weekly by severity. On-call for support and alerts: QA triages and routes to PM, developer or team lead; incidents reviewed with backend and the lead.

- **Automation across the department.** Co-designed the department's test template on eva.io (TypeScript; Selenium first, Playwright modules after the migration): authorization via the internal test environment QA Utils (ready-made test users), test execution and Allure TestOps reporting, repository structure, deployment guide. Reviewed code and contributed to the template repo; adopted across own and adjacent teams.
- **Game Center (unified entry point for special projects).** Built the testing process from scratch — test plans, Allure reports, FAQ and knowledge base; ran system, UI, API and integration testing; added screenshot-based visual regression on the Playwright module in eva.io with CI/CD runs (red about once a month, when seasonal assets or the project list changed). Handed over with documentation and processes in place.
- **Test pyramid and TestOps.** With the second QA over the last year and a half: unit and integration runs from the backend and frontend repos, e2e and screenshot suites, all reporting into TestOps — statistics, test-case management, and unit, integration and e2e layers kept deliberately separate. A contract sweep and field validation across the API layer caught a broken API contract and a field left unvalidated.

- **Shift-left as the default, not an initiative.** In requirements before development started: co-wrote functional requirements in 3 Amigos sessions, worked to the department RFC on involving testing earlier, kept a running page of open questions during design, and called out weak spots while changing plans was still cheap.
- **Support load.** Set up AI chatbot auto-responses in the corporate messenger from prepared documentation; a per-channel, per-project dashboard showed **30% of inquiries resolved there, never reaching the team**; kept ongoing support and defect tracking.
- **Performance testing.** With SRE, set up a dedicated performance environment; benchmark, stability and maxperf scenarios in k6; metrics in Grafana and Cosmos. The runs exposed database saturation — slow queries and lock contention under load — and it was fixed before launch. Wrote the department-approved methodology plus a template and quickstart the QA team reused on new projects.
- **Pushkin 225 Years: sole QA and most of the PM role.** With the team short-staffed, took over the PM's operational work — daily statuses and priority, launch readiness and dates with the lead PM, the launch content, requirements from the stakeholder product — and mentored the two new joiners (a backend developer on probation, an intern PM) through onboarding and 1:1s.
- **Cross-team relief.** Twice covered a QA gap in a neighbouring team for 1–2 months: release testing plus the strategy context, releases kept moving with no resident QA, handed back with documentation.

Tillypad — Quality Engineer — May 2018 – April 2023

Grew from Test Specialist through Test Engineer to Quality Engineer in a team of 3 testers and 6 developers. Client-server system for restaurant automation: desktop apps for waitstaff, bartenders, cooks and managers, with mobile (Android/iOS) and web versions added later; MSSQL with replication between the restaurant database and head office / cloud storage.

- **Full-cycle manual testing and support.** Created and maintained test plans and test cases; executed unit, integration, system and regression testing; API checks in Postman and Swagger, traffic analysis in Charles and Fiddler; tracked defects in Redmine; elaborated requirements to catch weak spots early in the cycle. Troubleshot user issues with the support team and onboarded new team members on the product and processes.
- **UI automation.** Automated core desktop scenarios in Python + PyAutoGUI, using image recognition of buttons and UI elements for coordinate detection and baseline comparison. Regression went from 2 working days (2 testers) to ~2 hours (1 tester).
- **MSSQL replication testing.** Helped shape the approach; wrote the SQL scripts that insert, update and delete source data, then verified the target database held what should replicate and nothing else; ran at the start of every release cycle: **around ten defects over three years, all high severity**, on logic the product could not work without.

SELECTED PROJECTS

- **hecho-demo** — https://github.com/ge-a-nikitin/hecho-demo — HECHO, a client-server order board built to demonstrate QA practice end to end: React 19 + TypeScript SPA, NestJS API with Swagger/OpenAPI, PostgreSQL + Prisma, SeaweedFS, Docker Compose. Unit and integration API tests on Vitest, bilingual product and QA documentation, GitHub Issues bug-report template, CI deployment.
- **hecho-e2e** — https://github.com/ge-a-nikitin/hecho-e2e — Playwright end-to-end suite driving a real stack (web, API, PostgreSQL, SeaweedFS): customer-order cases across chromium and Pixel 7 projects, tags, Qase TestOps catalogue sync, SUT pinned by commit, reports.
- **TShop** — TBank's merchandise store inside the mobile app; order-fulfilment integrations, sole QA.
- **Birthday 4.0** — congratulatory service with prize giveaways that grew from a one-off promo into a product; testing from scratch, automation on the framework, then two years as the knowledge holder and entry point for its testing questions — support requests, investigations, triage, release fixes.
- **New Year 2024** — annual gamified promotion; Kafka topic integrations and deep-link testing across app versions. Pulled the app versions users actually ran from the logging system into charts on the department wiki, and those numbers decided which versions we tested.
- **Pushkin 225 Years** — quests, prizes and loyalty for the poet's 225th anniversary; built from scratch, development April – July 2024, live July – August 2024. Sole QA across frontend and backend: shift-left requirements, Kafka topic integrations for the new task types, API scripts for working with the project, AI-generated test data and helpers. Automation was a conscious no — timeline, staffing and functional depth left no room for a suite that would pay for itself.
- **Stickermania with Vkusno i Tochka** — co-promotion with the restaurant chain; parameterised automated tests, an authorization module added to the shared template, AI-assisted checklists. An R&D study on parameterised layout automation showed a high false-positive rate, so we dropped it and saved the resources; the parameterisation itself went into functional autotests, cutting duplication.
- **UserAgent Redirect** — browser user-agent research for a fallback page; findings in an ADR plus a usage guide, automated tests over large value lists manual checks could not cover, knowledge-sharing session for the department's teams.
- **Prizopoly 2.0** — prize promotion reaching up to 3M+ unique monthly users; k6 performance testing covering every critical endpoint available at the time, three planned scenarios distilled into one universal one, load-testing ownership moved from backend to QA, results presented to both teams.

EDUCATION

- **St. Petersburg College of Technology** (sptcol.ru) — Programming, secondary vocational education — 2023 – 2027 (expected).
- **Emperor Alexander I St. Petersburg State Transport University** (PGUPS) — Faculty of Mechanical Engineering, Railway Rolling Stock and Wagon Facilities — 2007 – 2011, three years completed (incomplete higher education).

Training. Internal employer courses before each new scope: end-to-end automation on eva.io and load testing with k6. No external certifications — SQL, test theory, programming and English come from college, university and self-study.

LANGUAGES

Russian — native (C2). English — B2, working proficiency in reading, writing and speaking: documentation, tickets, reports and interviews.