Tu Van Nguyen Anh Quan

☎ (+84) 932 868 875 — ✉ tvnaquan@gmail.com — linkedin.com/in/quantvna
aqbtech.id.vn — github.com/aqbtech

SUMMARY

Final-year Computer Science student with solid experience in Java Spring Boot, RESTful API development, and database management. Strong understanding of OOP, SOLID principles, design patterns, and data modeling. Able to quickly comprehend and adapt to existing codebases, enabling efficient implementation of new features and improvements. Capable of working under pressure and always eager to learn and grow to deliver high-quality contributions to any team.

EDUCATION

Ho Chi Minh city University of Technology **2022 - 2026**
Bachelor of Computer Science

– **Current GPA:** 3.7 - **CPA(CGPA):** 3.5
– **Excelled in key subjects with top grades(A/A+):** Software Engineering, Data Structures and Algorithms, Computer Networks, Computer Architecture, Operating Systems.
– **Awards:** HCMUT academic encouragement scholarship semester 231.

EXPERIENCE

Dai-ichi Life Vietnam **Jun 2025 – Aug 2025**
Business Process Automation Intern

– **Scope:** Insurance domain (Claim subdomain: **GOP/ME/Reimbursement**), New Business & Data Entry. Agile/Scrum with Azure DevOps.
– **Key contributions:**
  - Wrote and maintained **Business Specification** for **Auto-assign Investigator**: routing rules (team/region/SLA), trigger conditions, exception handling, audit logging; updated spec after requirement changes.
  - Authored **Mapping Claimeform**: field mapping, validation rules, handling missing/invalid inputs to improve first-pass claim quality.
  - Configured **master data** for automation: investigator teams/regions, email templates/fonts for **generate email**.
  - Designed **test cases** and ran **ST/System Tests** for Auto-assign and Claimform (positive/negative/boundary, end-to-end across systems).
  - Modeled claim/policy flows using **BPMN** (swimlanes, gateways, events) to identify automation points and handoffs.
  - Analyzed **integration** with **Dai-ichi Connect** to enable **e-Claim submit**: payload schema, field mapping, status sync, error scenarios; proposed API versioning & idempotency, logging/audit guidelines.

– **Impact:**
  - Standardized investigator routing and reduced manual assignment steps.

  - Improved claim data consistency via Claimform mapping/validation; fewer rework loops.
  - Automated checks (waiting-period rule) and notifications (generate email) to lower manual follow-ups.
  - Established groundwork for end-to-end **e-Claim** integration; ensured traceability by linking spec–tasks–tests in **Azure DevOps**.

– **Tools/Methods:** **Azure DevOps** (boards, test plans), **BPMN**, API design/validation, ST/UAT workflows.

SKILLS

Programming languages: Java, Javascript, Python, C++.

Programming Skill: OOP, SOLID principle, Data Structures & Algorithms, Database Design.

Framework/Library: Spring boot, Reactjs, NodeJs, NextJs.

Database: SQL(MySQL), NoSQL(Firebase Realtime Database, Firebase Storage)

Version Control: Git, GitHub.

Other: Postman(API Testing), JMeter (Concurrent Testing), Swagger(API Documentation), Docker.

Languages: English, Vietnamese.

PROJECTS

(A few highlight projects; see more on my portfolio)

Project: E-commerce System **September 2024 - December 2024**

– **Description:** *An online shopping platform inspired by Shopee and Lazada, supporting product browsing, cart and order management, address book, and online payments.*

– **Source:** github.com/aqbtech/HCMUT-E_commerce

– **Technologies used:** Java Spring Boot, ReactJS, MySQL, Docker, Firebase, ZaloPay

– **Role:** Team Leader & Fullstack Developer

– **Responsibilities:**
  - Planned the project timeline, assigned tasks, and reviewed progress during weekly meetings.
  - Built REST APIs for product/user/address management and JWT authentication.
  - Designed the database schema and ORM entity using JPA, hibernate.
  - Integrated third-party services (Firebase/Cloudinary for images, ZaloPay for payments).
  - Implemented responsive UI components (product listings, shopping cart) using ReactJS and integrated front-end with Spring Boot back-end through REST APIs.
  - Containerized backend with Docker, deployed back-end to Render and front-end to Vercel.

- **Achievement:** Received a perfect score of 10/10 from the course lecturer for overall project quality. Strengthened teamwork and project management skills. Learned to optimize system performance using database-level pagination instead of in-memory paging. Gained hands-on experience with unit, integration, and performance testing to ensure system reliability.

Project: Hospital Management System **December 2024 - April, 2025**

- **Description:** Designed and developed a hospital management system from scratch as a software architect, focusing on requirement engineering, modular architecture, and SOLID principles. Applied Domain-Driven Design (DDD) and actor-action workflow to design high-cohesion, loosely coupled services with clear domain boundaries.
- **Source:** github.com/aqbtech/hospital-management-system
- **Technologies used:** Java Spring Boot, Spring cloud, ReactJS, MySQL, Docker.
- **Role:** Team Leader & Fullstack Developer
- **Responsibilities:**
  - Led a team of 7 through the full software architecture lifecycle — from requirements analysis, architectural design, and service decomposition to implementation, testing, and deployment.
  - Designed and implemented multiple backend services with Spring Boot, registered via Eureka Discovery Server, and communication through REST APIs or method call.
  - Applied SOLID principles to design maintainable components and interfaces, especially for authentication and user management modules.
  - Documented system architecture using Architecture Decision Records (ADR) and visualized key architectural views: topology view, module view, component & connector view, and deployment (allocation) view.
- **Achievement:** Acquired practical skills in applying SOLID principles and architectural design techniques to a modular system. Developed proficiency in documenting software architecture using ADRs and visualizing system views (module, component-connector, and deployment). Achieved a final project score of 9.9/10.