

Jing Cao

206-670-5221 | caoj318@gmail.com | Seattle, WA

GitHub: www.github.com/JennyJCao | LinkedIn: www.linkedin.com/in/jing-cao-neu

Available: May 2022 – Dec 2022

Education

Northeastern University

Seattle, WA

Master of Science in Computer Science

Sept 2021 - Expected May 2023

Related Courses: Programming Design Paradigm(Java), Mobile Application Development

Nanjing University

Nanjing, China

Bachelor of Management; GPA: 89.2/100

Sept 2016 - Jul 2020

Related Courses: Computer Systems, Operating System and Linux Programming, Algorithm Design and Analysis, Object-oriented Programming(C++), Information Security, etc.

Technical Skills

Languages: Java, JavaScript, Python, C/C++, Go

Framework: Spring Boot/Security, Django, Express.js, React.js, GraphQL, MyBatis

Database: MySQL, PostgreSQL, MongoDB, Redis

Tools: Git, AWS, RabbitMQ, Docker, ElasticSearch, WebSockets

Professional Experience

Xiaodu Network Technology Co., Ltd.

Xiamen, China

Software Engineer Intern - Backend

Apr 2021 - July 2021

- Developed the back end of the online shopping mall and its management system from scratch. Built the project with **Java**, **Spring Boot**, **Spring Security**, **MySQL**, **Redis**, **RabbitMQ**, **ElasticSearch**, etc.
- Designed and implemented **200+** **RESTful** APIs with **Spring Boot**, supporting functional modules like goods, users, orders, return orders, file system, and management. Contributed **15k+** **LOC**.
- Designed the mall's database with 23 tables in **MySQL**. Optimized database index structure, saving **30%** of query time. Implemented the DAO layer with **Mybatis Common Mapper**, generating most SQL statements automatically to improve development efficiency.
- Solved the **concurrency** problems in order placement with **Redis** and **JDBC Transaction**, optimizing the isolation level of the database from serial to **repeatable read**, and improve efficiency by **10+** times.
- Implemented the goods search functions with **ElasticSearch**, including word segmentation query, **multi-condition filtering**, paging, sorting, and keyword highlighting.
- Realized the flash sale feature with **Java Multithreading**, **Redis**, **RabbitMQ**, and **WeChat Pay API**, enabling asynchronous order placement, and increasing the concurrent processing capacity by **150%**.
- Integrated the file system on the **Alibaba Cloud ECS** server by applying **FastDFS** and **Docker**, to enable file storing, syncing and accessing with low latency.
- Authenticated and authorized the mall management system with **Spring Security**.
- Validated the APIs with **JUnit** and **Postman**, reached **90%+** coverage.

Selected Projects

OfferMaker - A Job-hunting Social Platform (React.js, GraphQL, Express.js, MongoDB)

- Constructed a job-hunting social platform for users to display their work experiences, resumes and discuss topics.
- Rendered the front-end pages on the server side with **React.js** and **Next.js** for **Search Engine Optimization**.
- Realized **20+** **GraphQL** APIs with **Apollo** to perform CRUD operations. Implemented back end in **Node.js** and **Express** to handle requests and launched database on **MongoDB** atlas.
- Solved user authentication and authorization with **Passport.js**.

ServiceAware - A Real-time Service Monitoring System (Go, WebSockets, PostgreSQL)

- Developed a service monitoring system with 20+ **RESTful** APIs in **Go**, which monitors remote servers and sends notifications in real time with **WebSockets**. Managed data about services, hosts, and events with **PostgreSQL**.
- Scheduled periodic monitoring events with **Cron**, and displayed responsive changes with **WebSockets** and **Pusher**. Applied **Mailhog** and **Twilio API** to notify clients by emails and text messages.

Tech Hiring Status Tracker during COVID (Python, Django, Pytest, SQLite3)

- Built the back end of an application to collect the recruitment status of companies during the COVID period.
- Implemented **RESTful** APIs with **Django** framework, integrated the email sending feature, and stored key information in the built-in **SQLite3**.
- Tested the application in **Pytest** with **40+** test cases, which consist of unit tests, integration tests, API tests, end-to-end tests, and performance tests.