Yisong Zou

919-937-6663 | yisong.zou97@gmail.com| Github: https://www.linkedin.com/in/yisong-eason-zou-152858122/

EDUCATION

Duke University: *M.S in ECE - Software Engineering Track* **Aug 2019 - May 2021**

Key Courses: Data Structure and Algorithm, Engineering Robust Server Software, Parallel Computing, Computer Security, Cloud Analysis

Xi'an Jiaotong University (C9 League): *B.E. in Microelectronic Science and Engineering.* **Sep 2015 - Jul 2019**

Key Courses: OOP Java programming, Data structure, C programming, Linear Algebra, Probability and Statistics, Computer Architecture

TECHNICAL SKILLS

Programming Language: C++/C, Java, Golang, Python, Shell, HTML, JavaScript, CSS, MATLAB, Verilog

Tools & Frameworks: RPC, Protocol Buffers, Django, PostgreSQL, GoMonkey, SqlMock, Docker, MySQL, JSON, Gradle, Git, Emacs

AWARDS

- "Siyuan" Scholarship and "Excellent Student" of Xi'an Jiaotong University **Sep 2016 - Jun 2018**
- Awarded **Provincial Prize** in 2017 National College Students Mathematical Modeling Competition **Sep 2017**

WORK EXPERIENCE

Tencent, *Backend Software Engineer Intern,* **(Golang)** **May 2020 - July 2020**

- Worked as a **backend developer** for Tencent's **Tianyi agile product development** platform at **PCG group**.
- Used **Go programming language** together with **Protocol Buffers.** The whole structure follows **MVC pattern.**
- **API layer**, based on **RPC,** provides API for frontend. **BIZ layer** provides all **business logic** functions, including data conversion, judgment and assembly. **DAO layer** used **xorm** ORM library to access database.
- I committed to several features. Such as **project space**, **Wework group establishing reusing and notification**, **team information editing, URL backend reshape** as well as some smaller ones and **finally launched them.**
- The **agile development** platform is used during the development process of **Chinese largest** social app **Tencent QQ** and the video streaming website **Tencent Video**. It provides rich configurable functions such as **iteration planning & tracking, defect tracking management, test planning & cases, CI/CD & deployment, etc.**

Graduate Teaching Assistant, ***(ECE651 software engineering)*** *Duke University* **Jan 2021 – May 2021**

- Teaching assistant for graduate level course ECE651 software engineering at **Duke Pratt School of Engineering**.

SELECTED PROJECTS

HTTP Caching Proxy (C++), *Duke University* **Jan 2020**

- Established a proxy server on **Linux VM** to handle **GET, POST and CONNECT** requests from browser.
- Implemented **multithreading** for concurrent requests and achieved synchronization by **RAII strategy.**
- Integrated an **LRU cache** for efficiency and accomplished response expire-checking and re-validation.
- Achieved robustness to external failures by providing **exception guarantee** for request and response.

Customized Command Shell (C++), *Duke University* **Nov 2019 - Dec 2019**

- Constructed a **multi-process UNIX** command shell that supports functionality of **Linux Shell.** Achieved the majority part of operations by **parsing user input** in parent process and **invoking system calls** in child process.
- Designed **built in commands** to provide access to **environment variables** in parent process. **Handled exceptions.**

Thread-Safe Malloc (C), *Raleigh-Durham, North Carolina Area* **Feb 2020**

- Implemented **thread-safe malloc** library in C based on **best-fit allocation** policy.
- Allocated new memory space in heap by calling the **sbrk function** to move program break and managed sperate memory by inserting meta data as header connected in a **linkedlist.**
- **Eliminated race condition** by using lock-based synchronization in pthread library.

Risc Online Board Game (Java Full-Stack, Gradle, Docker), *Duke University* **Feb 2020 - May 2020**

- Created online game with **Java**. Applied **agile development method** and **CI/CD** to update new features.
- The game includes **3 iterations** using **Git** for feature update and uses **factory and state design patterns.**
- Each player can choose a **game room** to join **through web socket**. And there is flexible built-in map for **2-5 players.** Each player will have **different resources** to make actions and use units to **attack others and conquer.**

Mini Ups and Amazon (Python Full-Stack, Docker), *Duke University* **April 2020 - May 2020**

- Constructed web app with separate tiers **(client, server and database)**, provide shopping and delivering service.
- Set up **client tier**. For **backend** used **Django,** for **frontend** used **HTML, CSS and JavaScript.** The client can receive requests from users and interact with **server using protocol buffers** and **cloud Postgres database**. Users can **register account** and manage. The Amazon and Ups server provides **warehouse and truck management.**