Hojin Nam

E-mail | Github | LinkedIn | Website | ☎1-650-888-0407

EXPERIENCE

GSCMS Inc, Seoul, South Korea **Software Engineer** **Feb 2017 - Dec 2018**

- Create **task automation(dynamic-link library, C\#, .NET)** on Windows server with an open-source(selenium). The assembly reduced the time for the laborious task in the accounting dept, which manually fetched bank statements of various online banking services, **from a few hours to a second, and released it to the production.**
- Build the **test automation environment** in various Operating Systems (OS) on **VMware** for operational acceptance and regression testing in an exploratory approach. The corner, critical, and custom cases by analyzing error logs and defects from the beta users are considered. It **cut off the beta version error rate by 99%.**
- **Create new UIs, modify legacy functions in the company's Windows app made of C#, On Windows Server with Oracle SQL,** create the documentation for the main program according to various clients' needs.
- Build a **responsive web page** with AngularJS to support several **dashboards** in order to monitor server health and enable controlling the on-Premises service on the local server remotely for on-call team members.

SKILL

PROFICIENT: **Java, C#, Python, Git, AWS(S3, Elastic Beanstalk), Spring**
FAMILIAR: JavaScript, HTML5, C/C++**,** Oracle SQL, React, Node.js, .NET framework, Docker, VMware

PROJECT

- **Microservices on Spring,** Implemented Login service and Post service which is for CRUD of the post and communication in services via Eureka server. **[Java, Spring framework, JPA, h2, HATEOAS, Netflix Eureka]**
- **Prouhet–Tarry–Escott problem on Message Passing Interface(MPI),** Implemented MPI parallel program parallelizing the operation between 16 threads on a computer cluster and compared the performance of O(**2^n**) computation between on single laptop and cluster computers.**[C, Message Passing Interface library]**
- **Compute Unified Device Architecture(CUDA) programming with shared memory,** Optimized the use of memory between threads on **parallel computing** to maximize the bandwidth. It avoids redundant transfers by loading and storing data in a coalesced pattern to shared memory from global memory. **[C++, CUDA Toolkit]**
- **Web App with React.js, Node.js,** Implemented CRUD table, role-based access control, session cookies on the front-end with Reactjs, encrypted password, REST architecture, test functions with Promise to manage asynchronous executions on the back-end with Nodejs. **[JavaScript, React, Node, NeDB, RESTful]**
- **Strategy Pattern implementation,** Contains the abstract class which is HAS-A relation and the interface which is extended to concrete implementation classes. The objects for the client are assigned at runtime and it is tested based on equivalence classes in JUnit jupiter. **[Java, JUnit]**
- **Bounded-buffer problem**, Used the POSIX Threads(pthreads), semaphore to control access to shared resources. There are three functions such that producer which writes an element in buffer A, consumer which removes an element in buffer B, middleman which moves an element from A to B. **[C, semaphore, Ubuntu]**

EDUCATION

Cal State University East-bay, Hayward, CA	**M.S. in Computer Science**	Jan 2019 - May 2021
Samsung Multicampus, Seoul, South Korea	[NCS] Android Developer Course	May 2016 - Nov 2016
Sejong University, Seoul, South Korea	B.S in Electronic Engineering	Mar 2009 - Aug 2015

COURSE

Adv. Algorithms, Adv. Computer Networks, Adv. Theory of Computation, Adv. Software Engineering(Microservices), Web Systems, Cybersecurity, Parallel Computing(MPI, CUDA), Machine Learning