Rui Jie Li . Jarvis Consulting

As someone who enjoys problem solving and technical challenges, I am looking forward to contributing to your organization! I have a year of work experience in software development, both in frontend and backend, and a bachelor's degree in software engineering from Polytechnique Montréal. I have worked with JavaScript, TypeScript, Python, Java, and C++, on projects such as a telecommunication simulator, embedded C++, a drawing app, and computer vision for robots. I am also working at a tech consultancy on the side, aside from training at Jarvis. In my free time, I'd sometimes go play badminto or basketball (not on a profesionnal level, just for fun), and sometimes I'd go running.

Skills

Proficient: Python, C++, HTML, JavaScript, TypeScript, Angular, CSS, Git

Competent: Linux/Bash, RDBMS/SQL, Agile/Scrum, Java, C, Docker

Familiar: Android, AWS, Machine learning, Linux OS, Unit testing, Kotlin, Message queue, Makefile, Boost.Python, VBA

Jarvis Projects

Project source code: https://github.com/jarviscanada/jarvis_data_eng_RuijieLi

Linux Cluster Resource Monitoring App [GitHub]: The project consists of a Linux Bash script that monitors the ressource usage of Rocky Linux virtual machines on Google Cloud. The scripts are triggered by crontab and send the information to a Postgresql database that runs in a Docker container.

RDBSM and SQL [GitHub]: This project is mainly for learning SQL, with statements like SELECT, UPDATE, DELETE, INSERT, various types of JOIN operations, etc. PostgreSQL was used to run the database, and PGAdmin was used to run the commands.

Core Java Apps [GitHub]:

- JDBC App: WIP
- Grep App: This app takes a regular expression, and recursively searches all files in a given folder before writing the results in an output file. It is written in Java 8, with Maven as a package manager. To use this app, use the command `java -jar path/to/app.jar REGEX PATH/TO/FOLDER PATH/TO/OUTPUT/FILE MAX_FILE_SIZE_WITHOUT_BUFFER` where `REGEX` is a regular expression, `PATH/TO/FOLDER` is a folder that should be searched recursively, `PATH/TO/OUTPUT/FILE` is where all the matched lines will be written, and `MAX_FILE_SIZE_WITHOUT_BUFFER` is the maximum size of a file that can be processed without an input buffer. It can also be used with `sudo docker run --rm -v path/to/data:/data -v path/to/log:/log  ruijie99/grep:latest REGEX PATH/TO/FOLDER PATH/TO/OUTPUT/FILE MAX_FILE_SIZE_WITHOUT_BUFFER`, however this is not ideal since it requires the creation of a data folder and a log folder, plus Docker isn't made to access the host file system.

Highlighted Projects

UI for a transit simulator: The project aimed at making a graphical user interface for a public transit simulator. In addition to drawing the deployment and package diagrams, I worked on the following features: displaying the events generated by the backend while a simulation is running, connecting the frontend and the backend with a message queue (ActiveMQ running in Docker), and pause/continue/stop simulation.

Cross-platform drawing app: The app consisted of a cross-platform drawing app. I made a chatting feature on Android that allows users to send messages in different channels (some groups are public, others are only for people drawing on the same picture). I also made parts of the user profile. The Android app was written in Kotlin and Firebase was used as a database.

Browser drawing app: The project consisted of making a web version of Microsoft Paint in Angular/TypeScript. I implemented the following drawing tools: straight line, polygon, bucket fill (left click to color connected pixels, right click to color all pixels on image), a part of free from selection, and aerosol. I also implemented undo/redo with the design pattern "Command", as well as unit tests with up Jasmine

Embedded programming [GitHub]: The project consisted of a robot that can complete an obstacle course (e.g. following a black line and avoiding walls) and was done as part of a class. I worked on measuring distance with sonars, CTC,

and implementing object-oriented programming to the classes representing the LEDs. There were also features such as debouncing buttons and interrupts that everyone had to implement as part of the class.

PolyStar computer vision team: The project consists of a competition where each team have to make robots (each robot have armor plates with a number on it, and LEDs indicating whether it's on the red team or the blue team). The goal is to have our robots shoot the other team's robots. I wrote a program to split 1920x1080 images into 1080x720 images based on XML data, annotated 200 pictures for the machine learning algorithm, and wrote a program to detect the color of an LED in a picture (return gray if it is off, return red or blue if it is on).

Professional Experiences

Junior Software Developer, Jarvis Consulting Group (2024-present): Training in various enterprise technologies such as SQL, Linux Bash, Maven, JDBC, and Java.

Data Analyst (contract part time), Ethos Metrics (2024-present): So far, I've completed a project on data migration from various formats (mostly Word and Excel) to Salesforce, using regex in Python and Excel to generate the import document. Currently, I am working on two projects: a document classification script with models on Hugging Face, and SEO optimization for their website (https://www.ethosmetrics.ca/)

Data Analyst, Fondation de Polytechnique (2023-2024): I implemented responsive design and improved the UI (fixing display and scrolling issues, and adding CSS transitions) of multiple web pages. Since most of the frontend was automatically generated with NetCommunity and I cannot modify the generated code, I've often had to come up with workarounds. I also made a tax credit estimation tool for the 2024 campaign and partially automated some data entry tasks.

Data management assistant, Fondation de Polytechnique (2021-2023): I was mostly working with Blackbaud products. I maintained and edited some web pages, partially automated some data entry tasks, and made a telemarketing tool that was used for a few months after Blackbaud removed theirs.

Backend development intern, Polytechnique Montreal (2020): I made a telecommunication simulator run up to 65% faster and use upt to a bit less than 50% less memory, depending on the simulation configuration. The result was achieved by adding/optimizing multiprocessing, using the **slots** attribute, and reducing function calls (especially SciPy.rvs). I also attempted further optimizations with shared memory (BaseManager) and Boost.Python (translating part of the code to C++), however they were unsuccessful due to high overhead.

Education

Polytechnique Montreal (2018-2023), Bachelor of Engineering, Software Engineering - GPA: 3.26/4.0

Miscellaneous

- I like playing basketball and badminton, just for fun
- For someone who just improvises with leftover and cheap ingredients, I'm pretty good at cooking (found out that Cheetos can melt in hot water and serve as spaghetti sauce thickener because me and my roommate didn't have flour)
- I like making memes… And maybe I am good at it!