

## **Jun Zhang**

### **Education:**

#### **University of Iowa- Bachelor's Degree, Computer Science**

Minor: Mathematics

2022 Fall Dean's List, International Distinction in Education Award (2021 – 2024)

### **Skills:**

JavaScript, Java, Junit Test, C, python,

MySQL, Linux English at professional working level

Chinese at native level.

### **Personal Projects:**

#### **RailTrack: A database system for train station operations and maintenance.**

January 2024 – May 2024

##### **Description:**

The database system can handle the daily operations of a train network efficiently and effectively. The system can store, and process various types of information related to the train network, such as the details of the trains, stations, tracks, routes, passengers, and tickets. The system can also provide assorted services to the passengers, such as helping them to find the best train for their destination by the optimized scheduling and price, book and cancel tickets online or at the station, and check the seat availability and occupancy of the trains, and passenger could provide feedback and support to the station. The system can also assist the staff in managing the train schedules, seat assignments and maintenance issues, as well as reporting any problems or incidents that may occur. It can also generate useful reports and statistics on the train station's performance and revenue, such as the number of passengers, the ticket sales, the train delays, and customer satisfaction. The database is drawn in Entity-Relation diagram, written in Schema, normalized to 3NF and BCNF, has referential integrity, and finally implemented by MySQL.

#### **WARP Sensor Network Refactoring and**

##### **Analysis November 2023 – December 2024**

Collaborated within a team to refactor the Swift codebase into Java, prioritizing modularity, and maintainability to streamline the development process.

Applied object-oriented design patterns (AGILE) to the Java rewrite, enhancing code readability and efficiency, and contributing to a more scalable and adaptable system.

Developed and integrated a latency evaluation module within the WARP system to measure the end-to-end latency of network flows, emphasizing precision and performance evaluation. Methods are implemented corresponding to interfaces by a planned sequence diagram.

Designed and implemented a GUI in Eclipse configuration along with a UML diagram (by yatta.de) for real-time visualization of latency metric, providing a user-friendly interface for monitoring system performance.

Conducted thorough testing and debugging procedures to ensure the reliability and accuracy of the latency evaluation feature, demonstrating a commitment to delivering a robust and error-free system.

### **Network Intelligence Suite and Research**

#### **Methodology Implementation**

August 2023 – December 2023

Developed Python scripts to emulate key feature of ping and traceroute utilities, employing raw sockets and ICMP for echo requests and replies. Implemented traceroute functionality by manipulating Time-To-Live (TTL) values to interpret ICMP timeout responses.

Designed and deployed a robust multi-threaded TCP server in Python, accommodating concurrent client requests. Integrated real-time network status, bandwidth analysis, and latency measurements based on client demands. Ensured stable server operation through comprehensive error handling and communication protocols.

Created a methodology for class research using Selenium WebDriver for automated web crawling and data collection. Developed scripts to navigate, extract, and process data from web pages, showcasing proficiency in web automation and data handling. Analyzed and interpreted collected data to provide effective insights in response to the research question.

### **Predicting Stock Market Trends Using Sentiment Analysis on Google**

#### **News: An empirical study**

December 2023

This project focuses on the utilization of sentiment analysis applied to financial news articles from Google News, aiming to predict stock market trends. The volatile nature of the stock market, heavily influenced by news and human sentiment, provides a rich dataset for analysis using NLP techniques.

Methodology:

Utilize a web crawler to gather news articles from Google News.

Apply NLP techniques to analyzed article sentiment, generating positive, negative, and compound sentiment scores.

Use the sentiment scores as features to fit machine learning models, with stock data from Yahoo! Finance as labels.

Evaluate model performance using accuracy, precision, recall, and F1 score metrics to determine the most suitable algorithm for trend prediction.

### **Data Structure Utilization for Query Optimization**

April 2023 – May 2023

Leveraged Sets and Maps data structures to enhance query performance on a U.S Department of Transportation dataset. Developed efficient Java algorithms for tasks such as identifying distinct destinations, calculating flight frequencies, and analyzing intra-state flight percentages.

Demonstrated proficiency in advanced Java collections, algorithm thinking, and Junit for debugging. Successfully manipulated large datasets, optimized query performance using

appropriate data structures, and ensured accuracy through comprehensive testing, showcasing a strong combination of technical skills in data manipulation and algorithm design.

### **Geo-Twitter Search Application**

November 2021 – December 2021

Developed a Python-based application integrating Google Maps API and Twitter API for a unique geo-social media experience. Utilized Tkinter module for GUI, enabling users to input search keywords. Implemented advanced API integration, fetching, and displaying relevant tweets with user information through the Twitter API, while simultaneously using the Google Maps API to pinpoint and visualize tweet locations on a map.

Showed strong skills in API integration, GUI development using Tkinter, and an innovative approach to merging social media and geographic data using JSON. The application enhances user experience by providing a comprehensive view of tweet context and origins, offering valuable insights into geographic distribution of social media trends.

### **Sorting Network Tool Development in Haskell**

March 2024

Developed a comprehensive tool in Haskell for simulating and analyzing sorting networks as part of a programming language course.

Established foundational operations for sorting networks, allowing for reading, writing, and reversing comparator sequences.

Features:

- Implemented functionality to read and visualize comparator networks from files, displaying them in a user-friendly format (IO).
- Added capability to reverse the order of operations in a sorting network to explore different computational paths.
- Ensured robustness through comprehensive error handling despite minimalistic UI requirements.
- Programmed the tool to simulate the operation of a sorting network on user-provided input sequences to validate sorting policy.
- Engineered a method to transform a linear sequence of comparators into a parallelized format, optimizing sorting time by executing independent operations simultaneously.
- Developed a feature to rigorously test if a given network qualifies as a sorting network using the zero-one principle, ensuring reliability across varied inputs.

### **Light projects in Operating System**

February 2024 – April 2024

- Developed a Shell-like command line interface tool in C that executes user commands and displays outputs.
- Implemented a process monitor to track and display current user process details such as PID and context switches.
- Enhanced the process monitor with a GUI using Python, improving user interaction and accessibility.
- Implemented multiple scheduling algorithms including FIFO, SJF, and SRTF to determine optimal process scheduling based on different criteria.

- Conducted comparative analysis on turnaround and response times for various algorithms to identify the most efficient scheduling practices.
- Extended the scheduling to a Multi-Level Queue system, accommodating processes of differing priorities and requirements. Implemented RR algorithm.
- Developed and tested multiple page replacement strategies like FIFO, LRU, Second Chance, and Enhanced Second Chance to report on their efficiency via metrics like page fault rates and disk writes.
- Analyzed performance impacts of increasing physical memory frames on different algorithms, providing insights into optimal memory management practices.