Willis Wang

Software Developer

wwang4292@berkeley.edu williswang.dev github.com/willislwang (408) 966-2909

Education

B.A. Computer Science UC Berkeley

3.74 GPA August 2018 - May 2022

Courses taken: Data Structures, Algorithms, Machine Structures, Operating Systems, Networking, Artificial Intelligence, Discrete Math, Probability Theory

Experience

Undergraduate Researcher Video and Imaging Processing Lab

August 2020 - Present Berkeley, CA

- Collaborated to develop a multi-modal indoor proximity detection framework with Professor Zakhor
- Design and implement magnetometer trace batching, feature extraction, and proximity classifier
- Deploy mobile app for public COVID-19 contact tracing

Software Engineer Intern 8th Wall

May 2020 - August 2020

- Work to improve existing infrastructure and testing methodology crucial for company growth
- Build automated database migration and CDN update pipelines currently used for all 8th Wall databases and web assets
- Design and build policy-driven customer repository backup and restoration framework for over 10,000 users
- Extend on production APIs to support bulk repo parsing for backup functionality
- Used AWS services: Lambda, S3, API Gateway, DynamoDB, Cloudwatch, Cloudfront, SNS, Elastic Beanstalk

CS182/282A Reader UC Berkeley

Janurary 2020 - May 2020 Berkeley, CA

- Reader for upper division Neural Network course CS182/282A
- Assist in assignment release and rubric creation, grading both homeworks and exams

Projects

Delta Lake Caching

October 2020 - Present Scala

- Research project for graduate level class CS 262A: Advanced Topics in Computer Systems
- Work with Databricks engineers to design and implement a read/write caching layer for Delta Lake using RocksDB

PintOS

Feburary 2020 - May 2020 C

- A simple operating system framework for the x86 architecture
- Wrote a complete BSD Fast File system, supporting dynamic sector allocation and file growth
- Implemented kernel syscalls, processes, synchronization, scheduling, filesystems, and buffer cache

TSP Variation Approximator

December 2019 Python

- Write an optimal solution finder for a version of the TSP problem
- Uses a combination of approximation algorithms such as Chistofides, k-means clustering, and local search to find a valid and cost-efficient solution

Languages

Programming Languages: C/C++, Java, Python, SQL, Javascript, Scala

Frameworks: Node, Express