

# Jason Chau

SOFTWARE ENGINEER · MATH AND COMPUTER SCIENCE MAJOR

☎ (626) 757-1902 | ✉ [jasondchau@gmail.com](mailto:jasondchau@gmail.com) | 🏠 [jdc9870.github.io/jasonChau/](https://jdc9870.github.io/jasonChau/) | 💻 [github.com/jdc9870](https://github.com/jdc9870) | 🔗 [www.linkedin.com/in/jason-d-chau/](https://www.linkedin.com/in/jason-d-chau/)

## Education

**University of California, San Diego**

*La Jolla, California*

B.S. IN MATHEMATICS AND COMPUTER SCIENCE

*Sept. 2016 - Mar. 2019*

## Skills

<b>Programming Languages</b>	Java · JavaScript · C++ · Python · C · Bash · MATLAB · ARM Assembly
<b>Technologies/Tools</b>	Git · Redux · Firebase · Numpy · JUnit · HTML5 · CSS · Bootstrap · SQLite · Node · Express
<b>Frameworks</b>	React Native · React · Flask
<b>Operating Systems</b>	MacOS · Linux (CentOS) · UNIX · Microsoft Windows
<b>Methodologies</b>	Agile

## Projects

### GymPal - JavaScript

*Apr. 2019 - PRESENT*

- Developed an iOS app that helps users find potential workout/fitness partners by matching them together
- Implemented with React Native for cross-platform capabilities and Redux was used for state management and cleaner code structure
- Leveraged react-navigation to implement smooth animations when switching between screens resulting in a better user experience
- Leveraged Firebase Authentication to sign users in securely and Firestore NoSQL cloud database to store user information

### Gravitate - Java, Python

*Sept. 2018 - Dec. 2018*

- Developed an Android app written in Java that groups users who want to carpool from San Diego to LAX to save money from using other rideshare services such as Uber and Lyft.
- Leveraged Firestore NoSQL cloud database to store user information such as flight times and profile details
- Leveraged Traxo API to retrieve real time flight stat information to implement an algorithm to group users with similar flight schedules
- Login security was implemented using Firebase authentication
- Used Python's Flask framework for handling API endpoints

### Mapping Networks - C++

*June 2018 - June 2018*

- Developed a program to read a network, translated it into a graph with nodes and found the shortest and most efficient path by creating a minimum spanning tree using Dijkstra's, Prim's, and Kruskal's algorithm.

### File Compression Program - C++

*May 2018 - June 2018*

- Implemented Huffman's encoding algorithm for lossless data compression
- Able to encode and decode bits to convert text files into binary and vice versa

### Auto Correction & Spell Checker Program - C++

*April 2018 - May 2018*

- Implemented a string lookup table using C++ STL implementations of a BST and HashTable.
- Created a ternary search tree data structure for storing strings
- Implemented auto-correction to return a list of strings that are most likely the intended word given a certain prefix.
- Implemented spellchecker to return the correct spelling of a query that is mistyped.

## Relevant Coursework

**Basic Data Structure & Object Oriented Design:** Designed and implemented code for basic data structures and ADTs in Java and C.

**Advance Data Structures:** Implemented algorithms that leverage advanced data structures in C++. Learned how to design and debug medium-scale C++ programs and use the C++ STL effectively.

**Computer Organization & Systems Programming:** Learned high-level concepts related to the structure and organization of computer systems.

**Design & Analysis of Algorithms:** Covered general algorithm techniques, such as divide-and-conquer, greedy algorithms, and dynamic programming.

**Theory of Computation:** Applied classical techniques including pumping lemma, determinization, diagonalization, and reduction to analyze the complexity of languages and problems.

**Software Engineering:** Learned about software practices used in industry, such as design patterns and version control, and applied them to an application with a small team of students.

**Introduction to Machine Learning:** Learned the basics of the statistical classification methods, such as linear classifiers, support vector machines, and decision trees, and kernel estimations, and how to apply these algorithms to large data sets.

**Software Tools & Techniques Lab:** Learned how to write Shell Scripts to automate tasks and build better software with understanding