

Alexander Lee

Graduate June 2020

aljilee@ucsc.edu

1-510-565-2808

Fremont

linkedin.com/in/alexanderjlee98

OBJECTIVE

Software engineering position.

EDUCATION

Major: Technology Information Management

University of California Santa Cruz

09/2016 – 06/2020

Minor: Computer Science

University of California Santa Cruz

09/2016 – 06/2020

CERTIFICATES

IBM Applied Data Science Specialization (09/2019)

Certificate ID: P67RW94NU2N4

IBM Data Science Professional Certificate (07/2020)

Certificate ID: R7J3GRCEQZJS

AWS Certified Cloud Practitioner (07/2020 – 07/2023)

Validation Number: JTY1C3NCF1B41G33

TECHNICAL SKILLS

Languages: Python, Javascript, Java, SQL, C, Ethereum Solidity, OpenFlow, UML

Data Science: Pandas, Matplotlib, Numpy, Seaborn, Folium, BeautifulSoup

Machine Learning: Regression, Decision Tree, Support Vector Machine, K-Means, KNN

Neural Networks: Keras

Blockchain : Ethereum

Virtualization: Virtualbox, Docker

Databases: Postgres, JDBC

Web Development: HTML, CSS, Angular

Cloud: AWS Cloud

WORK EXPERIENCE

Blockchain Developer

Ubicast

07/2018 – 09/2018

Tokyo, Japan

Achievements/Tasks

- Derived use cases and technical specification for the crowd funding platform.
- Created an ethereum crowdfunding smart-contract that allows the public to pull resources together to acquire and hold assets with ethereum.
- Created an ICO with the crowdfunding smart-contract that complies with the ERC20 token standard.

WORK EXPERIENCE

Assistant Instructor

Martial Arts America

07/2014 – 07/2015

Fremont, California

Achievements/Tasks

- Assisted lead instructors in teaching, drills, and activities during class and constructed weekly lesson plans.

Volunteer

Sunday Friends

06/2014 – Present

San Jose, California

Achievements/Tasks

- Taught underprivileged children and adults basic computer skills such as functionality of the computer and navigating the operating system.

TECHNICAL PROJECTS

Distributed Systems: Distributed Key Value Storage (11/2019)

- Implemented a fault tolerant, eventually consistent key-value storage system using Docker and Flask.
- Key-value storage system stored key-value pairs across different nodes and successfully resharded nodes upon addition or deletion of a node.
- Used gossip protocol to communicate updates to values with each other.

Machine Learning: The Best Classifier (07/2019)

- Used scikit-learn to build different machine learning classifiers such as KNN, Decision Trees, Support Vector Machines, and Logistic Regression to build a model to predict which users are likely to pay back their loans.
- Created classifiers by reading, cleaning, and normalizing the data set, splitting the dataset into training and testing sets, building the model, and evaluating the accuracy of the model with the jaccard index or a confusion matrix.

Networking: Openflow Network Topology (11/2018)

- Created a network topology using Openflow, consisting of hosts, switches, server, and a firewall that regulates traffic between hosts.
- Firewall controlled the flow of traffic between hosts by implementing a set of rules in a pox controller, allowing and blocking traffic from certain hosts and communication protocols.

Data Science: Pandas, Numpy, Folium (09/2019)

- Created a tool for displaying median housing prices in Vancouver by scraping housing prices and converting the dataset into a CSV, cleaning the data by only keeping the necessary information, performing statistical analyses to identify data trends, and creating a visual representation of the data in the form of a choropleth map for insight.

Information Systems: Automated Hiring System with UML (03/2019)

- Derived and analyzed use cases for a hypothetical automated hiring information system.
- Created UML activity diagrams, sequence diagrams, and use cases to identify and depict the workflow and behavior of the system.

Data Compression: Huffman Trees, Stacks, and Priority Queues (05/2017)

- Created a data compressor and decompressor using Huffman Trees, Stacks, and Priority Queues.
- Encoder encoded a source file and produces a compressed file by constructing a huffman tree and dumping its contents onto a file.
- Decoder took the compressed file and produced a duplicate of the original file by reconstructing and stepping through the huffman tree.

RELEVANT COURSES TAKEN:

Algorithms and Abstract Data Types

Computer Systems and Assembly Language

Distributed Systems

Database Systems I and II

Information Systems

Introduction to Computer Networks

Introduction to Computer Graphics

Introduction to Deep Learning & Neural Networks with Keras

Introduction to Econometrics

Machine Learning with Python

Management of Technology I & II

Python for Data Science and AI

Systems Analysis and Design