

Omri Levia

omrilevia.com

linkedin.com/in/omrilevia/

Email : omrilevia@gmail.com

Mobile : +1-408-781-3509

Education

- **San Jose State University** San Jose, CA
Master of Science in Software Engineering; GPA: 3.9 *Aug. 2019 – Dec. 2021*
- **University of California, San Diego** San Diego, CA
Bachelor of Science in Computational Physics *Oct. 2014 – June. 2018*

Experience

- **Vocera Communications** San Jose, CA
Software Engineer Intern *June 2020 - August 2020*
 - **Repaired Admin Console:** Repaired and modernized early 2000s admin console web application.
 - **Identified Problematic Areas:** Identified problem areas involving loading of modal dialog windows. Implemented modernized version with HTML5 and jQuery.
 - **Put Changes into Production:** Upon internship completion, admin console application operation was restored and changes were put into production.
 - **Collaboration and Team Building:** Collaborated via pair programming with mentor, and participated in team building activities with other interns. Recognized by management for extraordinary performance.
- **Alphabet, Waymo** Mountain View, CA
Software Quality Operations Associate *August 2018 - July 2019*
 - **Nominal Agent Simulation:** Constructed simulations of road agents using CarCraft to support self driving operations.
 - **Most Creative Project:** Won award for most creative project in the Software Quality Innovation Expo.
 - **Facilitated New Employee Training:** Encouraged collaboration, people development, exercised mentor-ship via new employment training.

Projects

- **Home Automation and IoT Management Platform**
github.com/HoMi-Iot
 - **Problem Statement:** Exists strong desire for devices connected to the internet, remotely controllable, and easy to automate. Management platforms have difficulty keeping up with new technologies, and devices from differing vendors usually fail to interop. Enter the Home Machine Interface (HoMi).
 - **The Home Machine Interface:** Aims to tackle market issues with a universal, scalable plugin development framework. Implements a modular plugin based design around a system core. The flexibility of HoMi allows for users to customize the platform to their liking.
 - **Design Motivations:** Uses self enforcing service protocols, with inter-plugin communication built in. Provides runtime execution of scripts, as well as a rule engine.
 - **How it Works:** The system core provides plugin management, loading and communication. Plugins are loaded dynamically, and are designed to extend the system's capabilities based on needs of the user. Scripts and rules can be run to make use of provided plugin services. Rules are triggered based on conditions and evaluate scripts.
- **Project Leadership and Agile:** Led project meetings consistently to keep team on track. Utilized agile methodologies for rapid development. Actively shaped management style based upon peer feedback.
- **Python Bittorrent Client**
github.com/Bittorrent-Client
 - **Complete Bittorrent Specification:** Implements the complete Bittorrent specification.
 - **Uses Optimization Algorithms:** The client makes use of optimization algorithms like rarest block first: the block or data pieces that are rarest will be requested from peers first, and end game: avoid slowdown of download toward end of download by issuing request for last blocks to all peers, then send cancel when a block arrives.

- **UML Parser**
github.com/umlparger (private, please request access to view)
 - **Parses Java Code:** Parses abstract syntax tree using Javaparser.
 - **Stores UML metadata:** During traversal of compilation units, metadata of java classes, relationships, and other information useful for making uml diagrams is stored in JSON objects.
 - **Generates UML class diagram images:** Makes API calls to YUML using a generated YUML string constructed from the metadata.
- **Ethereum Livestream Application**
github.com/livestream-app
 - **Starts Livepeer node:** Uses a Livepeer node for video encoding. The Livepeer protocol is built upon Ethereum.
 - **Publishes to HLS video player:** Uses an HLS player implemented in react to publish the stream data.

Programming Skills

Languages: Java, Python, Javascript, C++, SQL.

Technologies: AWS, Google Compute Engine, React, Relational Databases, noSQL databases, wireshark.

Programming/Knowledge: Design patterns, object-oriented programming, data structures, operating systems, virtualization, algorithms.