

Jason Sun's Resume

jason@sunapi386.ca sunapi386.ca linkedin.com/in/sunapi386 +1 (408) 599-0428 02/02/2017

About Me

- *University of Waterloo* Computer Science, graduated in 2015. Exchange term at *EPFL (Swiss Federal Institute of Technology at Lausanne)*.
- Very driven and motivated with many hobbies. In my spare time I work on side projects, see *blog.sunapi386.ca*. Especially interested computer science, which was discovered after trying two different majors: physics, and business.
- Over ten years of experience unix: *Ubuntu*, *Arch Linux*, *OS X*, and utilities like *grep*, *find*, *fdisk*, *nmap*.
- Experienced working in startup environments, at *Velocity Residence* and *Velocity Garage*. Enjoys technology conventions; ones I had attended were: *DEFCON (#22)*, *2600 Hope Number 9*, *PennApps 2013*, *MHacks 2014*, *HackMIT 2014*, and *HackZurich 2014*.
- Won hackathon prizes at:
 - *HackZurich 2014* at *ETH Zürich*, received *Tamedia Digital Award* (all-inclusive team trip to visit startups in Berlin, Germany) with *.GIFMeIt*: An iOS app that lets a user easily capture and share GIF images.
 - *PennApps 2013* at *University of Pennsylvania*, received *Twilio's Communication Award* (\$500) with *Marmoset*: A chatbot to respond to your chosen Facebook friends without them knowing.

Work Experiences

- Software Engineer *Apple Inc.* in November 2015 – January 2017 (15 months) Cupertino, California

Designed and implemented a complete web solution (**Rails/Ember**) for managing iOS devices, hosted on **nginx** and **MySQL**. And implemented a corresponding client sided task runner delegate to work on jobs queued from the web solution. Contributed to internal iOS apps (**Swift/Objective-C**).

- Undergrad Research Assistant at *University of Waterloo* in May 2015 – September 2015 (5 months) Waterloo, Ontario

Worked with professor and a small team. Developed a data pipeline for processing chat logs, and built a search engine prototype.

- Software Engineer Intern *Shutterfly Inc.* in July 2014 – August 2014 (2 months) Redwood City, California

Develop functional load tests for distributed services. Design and implemented a distributed key value storage service, using technology like **Jersey** RESTful Web Services framework and Apache **Cassandra**.

- Software Developer at *Encircle Inc.* in May 2014 – June 2014 (2 months) Kitchener, Ontario

A startup in the *Velocity Garage*, worked in **java**, **coffee**, and **python**. Worked on the web and android application stacks.

- Software Tools Developer Intern at RIM (BlackBerry) in September 2013 – December 2013 (4 months)
Ottawa, Ontario

Built features to the GitLab open source project (**Ruby on Rails**). Developed a testing framework for testing website user interfaces, using the **Selenium** Java framework.

- Physics Teaching Assistant at *Wilfrid Laurier University* in September 2011 – April 2012 (8 months)
Waterloo, Ontario

Developed a spectrometer reading program in python, using the **pySerial** library, and automate queries over serial port - previously you had to punch numbers on the machine.

Interesting Projects

- Real-time Operating Systems (CS 452): One project for the entire semester, which is to build a real-time microkernel and write user programs to control model trains on a track. The course is famously called the Trains course.
- Artificial Intelligence (CS 486): Worked on projects involving machine learning, learning probabilistic models, Bayesian networks, search and constraint satisfaction problems.
- Advanced Algorithms class (EPFL CS 450): A graduate course in algorithms, learned theoretical techniques and their applications to solve problems. Interesting techniques such as network flow, randomization, dynamic programming.
- Computer Graphics (EPFL CS 440): Half the course was about graphics rendering techniques, and the second half about animation techniques. Implemented rendering methods into a ray-tracer (called *Nori*), and explored animation modeling and some fluid mechanics.
- Intelligent Agent class (EPFL CS 430): Developed intelligent agents to pickup and deliver parcels in a simulated environment, with intelligent behaviours: reactive, deliberative, centralized, decentralized, and auctioning.
- Concurrency class (CS 343): Multithreaded quicksort, implementation of well known concurrency control mechanisms, such as monitors. Language is in **uC++**, a concurrent dialect of C++, developed at University of Waterloo.
- Distributed Systems class (CS 454): Built a remote procedure call library in **C++** on top of TCP, both client and server side. Implemented the go-back-N reliable transmission protocol, over UDP using **Java**.
- Computer Security class (CS 458): Created exploits in **C** using techniques such as buffer overflow, and format strings. Implemented an intrusion detection program which parses output from tcpdump to detect spoofed packets, malicious hosts, and worms.
- Compilers class (CS 251): Built a MIPS compiler in C++, parsing a subset of C keywords and generating **MIPS assembly** code.
- Computer Architecture class (CS 450): Designed a pipelined CPU in **Verilog**, supporting 8 instructions for computer architecture class. This is sufficient to run machine code produced by the MIPS compiler, from the CS 251 compilers class.