

# GIANNI MAIZE

(949) 439-4293, [gmaize@alumni.stanford.edu](mailto:gmaize@alumni.stanford.edu)  
github.com/gmaize, [linkedin.com/in/gmaize](https://www.linkedin.com/in/gmaize)

---

## EDUCATION

---

### Stanford University

*March 2014*

*B.A. Economics & Minor Computer Science*

---

## EXPERIENCE

---

### NetApp

*Nov. 2014 - March 2016*

*Software Engineer (WAFL Performance Engineering Group)*

*\*\*\*SPOT Award Recipient - Sep 28th, 2015\*\*\**

- Designed, developed, and supported performance analysis and reporting tools and infra.
- Ensured tools' maximal benefit across various performance groups
- Worked with performance labs' infrastructure and automation to enable tool workflow
- Designed and implemented ONTAP performance exceptions surveillance system:
  - Leveraged company-wide performance data-stores
  - Allowed developers to easily define variety of performance exception conditions
  - Enabled QA to detect a wider class of comprehensive performance issues

*php | jQuery | javascript | python | angularjs | sqlite3 | html5 | css*

### NetApp

*April 2014 – Nov. 2014*

*Web Apps Developer (WAFL Performance Engineering Group)*

- Owned full-stack of intracompany data visualization and analysis web-tool
- Drove tools' continuous innovation, modularity, performance, and architectural design
- Added new tool features for cross-team functionality, responded to bug reports, and answered questions regarding tool usage and design
- Trained co-developer on tool architecture and best practices
- Redesigned codebase to achieve 80% loading speed improvements, decreases in long-term storage requirements of 150MB per performance test (avg. 10+ tests per day), and reduction in run-time memory overhead by 400x

*php | jQuery | javascript | C | sqlite3 | html5 | css*

---

## PAST PROJECTS

**ExQUIZit** — Quiz website designed and implemented in a team of 4, featuring user profiles, quiz-creation, quiz-taking, quiz-grading, website administration, and more.

*java | html5 | MySQL | javascript*

**Heap Allocator** — Implementation of malloc function in a team of 2. Required collaborative design, testing, and performance optimizations.