Ryan Zabcik

ryanzabcik@gmail.com
281.769.2793

http://zabcik.com
http://github.com/rzabcik

Graduate of the University of Texas at Austin, I am looking for full-time employment opportunities in the Austin area. I am always eager to learn new technologies and explore the sub-fields of CS. I want to help apply my skills towards noble causes.

Education - The University of Texas at Austin

- Major: B.S. Computer Science, graduated May 2016
- Notable Courses: Data Structures, Honors Computer Architecture, Honors Operating Systems, Autonomous Intelligent Robotics, Programming Languages, Compilers, Security and Privacy
- Other studies: abstract mathematics, classical physics, rhetorical writing

Employment

Microsoft (summer of 2015)

- Worked on the OneNote service architecture, specifically the notifications API (all C#)
- Learned the techniques and consequences of large-scale deployment
- First experience with large team dynamics in a corporate environment

Offers.com (summer of 2014)

- Built a virtual server manager using Ruby, Docker, and AWS as a developer tool
- Gained experience in API and CLI design, cloud networking, and deployment

NCI Building Systems (summer of 2013)

- Built a web application interfacing with Google Maps
- Worked with consultants on a document management system to help their engineering business adapt to digital needs

Projects

- Textbook Trader, under development in Android
- Designed and implemented an instruction set architecture for an FPGA, and an 8-bit noise generator, using Verilog
- Implemented protocols for distributed and concurrent systems, using Java
- Autonomous Twitter robot personality for Freshman Research Initiative using ROS
- Weather balloon data collector built using Android and Arduino

Skills

- Linux system management and bash scripting for personal and corporate machines
- Java for most school assignments, independent projects, and Android
- Ruby, Ruby on Rails, C# for server-side web development
- HTML/CSS/Javascript for client-side web development
- C/C++ for operating systems and hardware programming
- Functional programming with Haskell