David Buchanan - http://joesvolcano.net

Cell: 801 557-5568
Home: 801 766-8345
Email: dj8@joesvolcano.net
Home Address: 421 W 2375 N Lehi, Utah 84043

Job Title: Senior Application Engineer
Desired salary: $80,000/year salary w/ benefits

Education

Masters Degree in CS and Internet Engineering (Equivalent Experience, 12 years)
Institution: Self Taught ;)
Years: 1997 - 2009

I am 100% self-taught by rubbing shoulders with the great minds at Inetz Media Group, Freeservers, Primedia, About.com, NetZero (United Online), Prosper and Eli Kirk. My 12 years of building enterprise-level internet appliances have lead me to be an indispensable member whatever team I am a part of, and in many cases to become the leader of that team.

Education Focus: Effective System Design, Object oriented and MVC design patterns, Elegant Database Design, and Web 2.0/AJAX to make a fluid user experience

Experience

Eli Kirk: 2007 - Present, http://www.elikirk.com

Worked as architect and primary engineer for two large projects with timelines longer than one year. One of these was a collaboration network linking many of the largest technology companies in the world including: Microsoft, Oracle, HP, EMC, Red Hat, and Symantec. Wrote and worked with these clients to refine business requirements, milestones and deliverables. Was lead engineer in teams of 3 to 5 people and code reviewed most new functionality. Designed and built enterprise-level database logic and constraints.

Skills Applied: Perl, PHP, PostgreSQL, MySQL, Enterprise-level security and scalability design and Web 2.0 AJAX UI implementation

ODYC Hosting (Self-Employed): 2004 - 2007, http://www.aodyc.com

Designed and built a billing system engineered for enterprise-level application including advanced revenue recognition and multi-currency. An addition to it's use for the hosting system, the billing system was intended for eventual packaging for other industries as a standalone product.

Skills Applied: Perl, PHP, PostgreSQL, Apache/ModPerl, XML/XSD, Linux, Accounting/Finance Engineering

NetZero (Web Services): 2001 - 2004, http://megawebservices.com

Wrote accounting/billing interface for 60,000+ paid web hosting accounts on various recurrence cycle billing. Created recapture system for billing declines which increased overall capture by 10%. Wrote the account upgrade/settings code used by all 4 million hosted web sites. Migrated 50,000 paid accounts from an older billing system to a new one. Integrated with several 3rd party API's for real-time credit card processing (AuthorizeNet, CyberSource, and ProPay), registered domain administration (Register.com), billing systems (Extent, EclipseNet), and auction software (CommerceFlow).

Skills Applied: Perl, Oracle, Apache/ModPerl, XML/XSD, Linux, Engineering for System Scalability, Accounting/Finance Engineering

Inetz Media Group: 1998 - 2001, http://inetz.com

Engineered many client's shopping carts, intranets, extranets, and site utilities. Wrote an HTML-based programming language used to speed development and allow HTML coders to build shopping carts with Database Integration. Key developer on most major projects.

Skills Applied: Perl. PHP, MySQL, Apache/ModPerl, Linux/BSD, JavaScript/DHTML, Versatility in Handling Many Projects/Clients

References

Tom Gay, CTO/Co-Founder Inetz Media Group

Phone: 801 415-2512
Email: tom@inetz.com

Doran Barton, President/Co-Founder Iodynamics LLC

Phone: 801 520-9875
Email: fozz@iodynamics.com

Code Samples Available Upon Request

Project Highlights

TSANet Inc. Marketing and Member websites, Eli Kirk: 2008 - 2009

TSANet ("Technical Support Alliance Network") is a collaboration network formed by many of the largest technology companies including: Microsoft, Oracle, HP, EMC, Red Hat, and Symantec. It facilitates easier resolution of multi-vendor technical support cases. They had been using a legacy system for several years, that was inflexible and didn't allow them to change and scale their network. We were contracted to re-build the system from the ground up, make it more flexible and efficient, provide a clean new look and use the latest Web 2.0 interface technologies. One of the biggest challenges this project presented was a high level of data complexity and access permissions. For example, one participant can only see another participant's information if their member subscription and per-user settings allow it. The new system allows TSANet to run their system on less hardware, have faster performance, and be able to easily adapt it to serve new markets and industries.

ODYC Hosting System, Self-Employed: 2004 - 2006

After a previous hosting business, LiquidOrange.com failed in the end of 2003, my partner and I set out to rebuild the platform from scratch to make it more scalable and robust. We were tried to use some of the old code from LiquidOrange.com, but in the end started completely fresh. I laid out a completely new hosting solution which included a reseller model, and a billing solution, site builder, e-commerce application, and hosting administration system. The billing solution was designed to allow strategic partners, or ourselves to easily monetize the hosting accounts and run all types of promotional offers online or offline. The site builder included a powerful and flexible site template system, an image manager, and advanced web-entities to run photo albums or other path based processes. The site builder also included a multi-lingual engine to allow whole-site language switching. The e-commerce application included advanced product sub-categorization and inventory management, international standard tax and shipping algorithms and a modular checkout system. Several times through the project we outsources graphics work to independent designers who I was responsible to manage. My partner Steve provided funding and business knowledge, but I was the only developer that created these systems, working self-motivated for 2 years on this project.

IDBS / Synoma Programming Environment, Inetz Media Group: 1999 - 2000

This was a programming language for web systems that greatly increased the speed of development for Inetz projects. I created the language for myself and my own development efficiency and used it for several months before the owners became aware of it. Originally, I named it IDBS for "Inetz DataBase System". The language is HTML embedded with external object definition files which greatly streamline SQL queries and web object pagination. A database layer allows developers to have full access in a simplistic database-agnostic manner. The database access and query methods were simple enough to empower HTML coders to develop feature-rich dynamic shopping carts, intranets, and web portals. Inetz's low-end HTML people became mid-level application developers, and soon the majority of new Inetz projects were coded in IDBS. In Jan 2000 we began looking into building IDBS into it's own standalone marketable product. The new product "Synoma" was to compete in the same space as Cold Fusion. Investors were secured and we selected programming teams in Syria where I would also move during the development. The project was to be fast-tracked and aimed for an IPO mid-2001. In May 2000 however, after much research we decided that we were too late in the game to compete, and abandoned these business plans. Inetz continued to use IDBS for the rest of the time that I worked there.

Billing and Recapture System, NetZero (Web Services): 2002 - 2004

At the end of 2001 NetZero (Web Services) realized that their current billing system, Extent RBS, was not going to scale and decided to purchase Sandlot EclipseNet and migrate their customers. I had been the primary engineer over billing and was tasked with the project. During March and April I re-wrote the billing libraries into a billing-system-universal API so that we would be able to have all other code be unaffected by the changeover. By switching to EclipseNet we would have to switch from "Anniversary-Day" billing to "First-of-Month" billing which meant that all our 50000+ customers needed prorated billing for July. Also, missing in EclipseNet was a system to auto-run billing each month, so I had to create a billing daemon and check system to see each of our customers were billed properly.

Once the billing system was transitioned, we began to look at retention and recapture. Out of the gates we only tried our customers credit cards once per month around the 6th of the month. We thought up many ideas for retrying the customers credit cards, and eventually I put together the following system: each decline is attempted up to 9 tries over 3 months (when the transaction fees outweigh the amount captured); communications are sent to people telling them their credit cards are expiring, then notifying them of declines on retry rounds; each decline round increased the disabled-level of the site until after the third try when all site activity was suspended; customers can login online to self-clear payments; card processing rounds were placed on dates or greatest opportunity when charges are most-likely to clear. Before the recapture system 75% of our customers cleared and 25% declined. After the new recapture system was implemented and tuned, we cleared %89, leaving only 11% declined.