

THIS IS NOT YOUR RESUME. WHAT IS IT EXACTLY?

The following is an exhaustive list of my duties and accomplishments from my 23-month stint at Money-Media, a Financial Times Company (New York, NY, November 2007 to October 2009). I thought it might be helpful to post this information in one document as it has been partitioned throughout my multiple resumes.

JOB TITLE

Senior PHP Developer. In my final months I assumed the managerial duties of the Director of Application Development, a position which remained unfilled up to and after I left the company.

JOB DUTIES AND ACCOMPLISHMENTS

GENERAL

- ◆ As Senior PHP Developer, developed and maintained code for front-end sites and back office applications. Worked directly with all departments in the company (C-Level, Editorial, Marketing, Sales, Customer Service, IT), analyzing business requirements and delivering subsequent systems implementations.
(The internal CMS and customer fulfillment systems are used daily by 60+ employees in the US and the UK. The front office publication sites are accessed daily by 50K+ customers - total customer login base: 150K-200K - worldwide.)
- ◆ As interim Director of Application Development, provided the CTO with periodic reports on team members' progress, and worked with him to determine their next projects. Helped determine code release schedule for the CMS, customer fulfillment system, job advertising system, and live publication sites (Ignites.com, Fundfire.com, AgendaWeek.com, BoardIQ.com, and IgnitesEurope.com).

SYSTEMS ANALYSIS/DESIGN

- ◆ Performed a full analysis of how a legacy application authorized payments, comparing it with PayPal's new requirements for payment authorization. Under a strict deadline, determined the changes necessary to bring the legacy application in line with the new requirements, and directed a junior member on how to implement those changes. The upgrades to payment authorization were coded, tested, and successfully released to production ahead of schedule.
- ◆ Designed an asynchronous solution to keep three disparate databases (with very different schemas) and a remote Lyris server in sync with each other. The "gen 1" back-end database is used each morning to create both front-end databases - "gen 2" and "gen 3". Any changes to gen 2 or gen 3 get queued and copied back into gen 1 before the next morning. Queued changes are also sent to the remote Lyris server, when applicable. Existing systems relying on the critical gen 1 database remain untouched, and changes made by customers on the front-end are re-integrated such that they appear correctly the next day in all three databases.
- ◆ Solved a months-old issue with a large client representing \$100K+ of revenue. The client (also a customer) noticed that sometimes the ads they bought did not appear in the daily emails which go out to all customers; the ads always showed in-house and for other customers. In the end I determined that an in-house naming issue caused links to the ad to break - only ads named a certain way and only on the client's particularly old version of Lotus Notes.

PROGRAMMING

- ◆ Designed and implemented the Remote Mailing Scheduler, a queuing system serving as the main interface between the editorial staff and the remote mailing system. Editors prepare their content in their CMS and then submit it (optionally time-delayed) to a queue for sending to the external Lyris email processor. All content remains in-house until transmitted to Lyris for immediate message sending, allowing submissions to be cancelled or altered until right before emails actually need to be sent. Real customer emails as well as test emails get generated through this system. Final implementation included classes for SOAP-based communication with Lyris, HTML/AJAX interface additions to the CMS, a separate batch script run from cron, detailed logging and a MySQL-based audit trail.
- ◆ Redesigned the classes which generate critical, customer-facing ad server (Lyris) HTML/JavaScript code. Used UML diagramming to plan new classes and their relations to one another, then implemented it all in a PHP PEAR package with a full unit/integration test suite using PHPUnit. Designed the classes so they could be used in four completely disparate codebases (including Zend Framework). Abstracted database access using an adapter interface so mock classes can be used (for unit testing).

(CONTINUED ON NEXT PAGE)

PROGRAMMING (CONTINUED)

- ◆ Redesigned JobBlast (a jobs listing system) so that the system can be called 100% standalone or within an iframe. Converted hardcoded functions and logic scattered across the codebase into new centralized classes, and moved key include and CSS files into one common area.
- ◆ Managed huge object-oriented code base of 16,000+ source files and hundreds of code packages covering 3 daily publications, 2 weekly publications, and 3 main web applications. Maintained code developed by several programming teams over a 10-year period.

TEAM/PROJECT MANAGEMENT

- ◆ Coordinated pre-release application testing with QA Manager. Liaised between the QA Manager and developers to facilitate successful testing of all code modules in QA and staging environments. Created test grids, release notes, and detailed installation instructions for moving web applications to staging and production environments. Used PEAR packaging and SVN to push application code between system test and QA environments.
- ◆ Worked daily with the IT Director and his Linux Administrators to solve issues such as: interfacing Active Directory/LDAP and PayPal integration with internal web applications; troubleshooting DNS, routing and reverse lookup; PEAR package installations; MySQL server performance tweaking; and troubleshooting file permission errors.
- ◆ Provided 24/7 backline support for all web applications and publications; emergency support for editorial staff (7:30-8:30am, 5:00-6:00pm); and technical support for the IT team during sites' peak access period (8:00-11:00am). Created and maintained matrix of current PEAR package levels for all web applications and sites.
- ◆ Administered site and managed users in FogBugz, an externally- hosted project management and bug tracking tool. Spearheaded centralization of all departmental documentation into the FogBugz wiki, including the creation of checklists for release procedures and the standardization of code package documentation. Met goal of keeping the queue of unhandled reported cases to less than 10 cases.
- ◆ Built PEAR code packages for shared use in system test and QA. Checked for and solved package dependency issues between multiple development teams and IT. Created and maintained matrix of PEAR package levels in department wiki.
- ◆ Enforced team standards for class design and object use. Created commenting guidelines mandating PHPDoc header docblocks for all source files, both new and edited, and encouraging liberal use of TODO notations throughout the codebase.
- ◆ Led daily, 10-minute team "scrum" sessions. Tracked the status of all open projects for my team. Trained, mentored and supervised team members. Screened developer candidates' resumes and conducted face-to-face candidate interviews.