Kyle Sherman

Email: kylewsherman@gmail.com
612-208-6855

INTRODUCTION

I am a degreed computer professional with fourteen years of experience in software engineering with an emphasis on high performance, low latency development, as well as database architecture and administration. In recent years my focus has been on Java, Ruby, Perl, and C++ development. Most recently I've started developing Android applications. I am seeking a full-time position in computer systems programming.

WORK EXPERIENCE

Android Developer *March 2011 – April 2011*
MotherRobot
Minneapolis, MN
http://motherrobot.com/

Assist in the ongoing development of a consumer Android application for a local retail operation. Develop classes, modules, and other functionality as needed, using best practices from the Android development community.

- Created and updated multiple linear and relative layouts to accommodate complex designs, including standard Android views such as gallery, grid, language localized text, image, button, etc.
- Created dynamic layouts that change based on user input. For example: A "Delete" button that disappears when there are no more items to delete, and a "Buy" button that changes to an "Add" button once an item is in the cart.
- Wrote asynchronous tasks to handle web connectivity to the master data repository. Created web connection methods that handle marshaling Java objects to and from JSON as well as making the web calls to query and update the data repository.
- Modified existing artwork so it looked good at the required dimensions.
- Wrote JUnit tests as needed to test new methods as well as added functionality for existing methods.
- Worked in a team environment with multiple developers sharing a Subversion code repository.

Android Developer *January 2011 – March 2011*
Nullware
Minneapolis, MN
http://nullware.com/fortunequote/

Design and write an Android application called FortuneQuote. FortuneQuote is a version of the old *nix fortune program for Android. The fortune database contains 44 topics and over 15,000 entries. The goal was to turn what is a traditional command line application into a full featured Android application.

- Designed applicaiton overveiw and process flow. Main components consist of the main screen with the current fortune displayed (pressing the screen displays the next random fortune) and a menu of options (about, preferences, send, copy, share, and exit).
- Used the built in SQLite Datbase to store and index the fortunes. This allows for speedy lookups and easy randomization. The database is initialized from flat files in a background task when the program is first run.
- Added widget support so users can put fortunes on their home page.
- Added notification support so users can have periodic fortunes appear.

- Preference support allows changing the selection of topics to pull from, the font style and size, widget refresh rate, and notification settings.
- Published FortuneQuote to Google's Android Market.

Software Engineer *April 2006 – January 2011*
Dow Jones
Minneapolis, MN
http://www.dowjones.com/

Senior Software Engineer for ongoing Market Data project to redesign existing data request infrastructure. Some data requests covered were: stock quotes and trades, ticker symbology, real-time and delayed data, and data aggregated over time, for U.S. and foreign markets. GigaSpaces (a Java based cloud platform) was used for data storage and retrieval, with MySQL for persistent storage. Client facing services were written in Java and C++ using ZeroC Ice and SOAP interfaces respectively.

- Lead designer and developer for client services that allow for requesting data from the GiaSpaces repository. These services were written using the ZeroC Ice framework and support all of the various data types (quotes, symbology, time-series, etc) in the repository. Ice was used to allow for easy and fast (no serialization needed) cross platform consumption of our services.
- Lead designer and developer for an entitlement system that allows access to market data. The system is Java based and is integrated into all of our services. It can filter data based on type, source, query request, query result, including/excluding specific fields (by row or by column), and via custom plug-ins (for more complicated business logic). A web based administration panel handles updates, which are applied (via notifications) to running services. All changes are logged for auditing.
- Developed an Ice based time-series service that returns aggregated data between requested timestamps. Aggregated block sizes can be minute, hour, day, week, month, or year. Holidays, weekends, exchange timezones, and market hours are all accounted for.
- Updated a minimal Rails based entitlement administration program to a more full-featured and user friendly web application. It allows for full modification to the entitlement settings after user authentication. It uses modern web techniques, such as CSS and Javascript, to achieve more of an application feel. It also has import and export (to JSON or XML) functions as well as the ability to merge entitlements from one system to another. This merge feature is used to deploy changes from the Development environment to QA, Test, and Production.
- Assisted in developing and maintaining C++ SOAP services for quote and symbology requests. The data request and entitlement portions are handled with calls to our Java library, then C++ is used for speedy marshaling of the data from our internal format to SOAP XML.
- Wrote a Ruby FIX (Financial Information Exchange) file generator that builds a FIX44 file from Wombat sources that QuickFIX can consume.

Database Engineer *August 2003 – April 2006*
MotivAction
Minneapolis, MN
http://www.motivaction.com/

Senior database architect/developer/administrator, responsible for the administration of production, pre-production, testing, development, and reporting database servers running SQL Server 2000 on Windows 2000.

- Lead database developer for a six month project to rebuild the core production web environment (consisting of SQL DB, C#, and ASP.Net). Designed, developed, and implemented the following modules for this system: Data Dictionary, Clients, Users, Groups, Promotions, Activities, Actions, Transactions, Shopping, Email, Lottery, and Reporting. These modules encapsulated the following attributes:
  - Hierarchical recursive tables (automatically tracking the hierarchy and level of parent/child relationships).
  - Vertically partitioned tables (divided by date ranges for large volume tables).
  - A central ID generator for all primary keys to allow for the easy migration of data from development, to QA, to staging, and to production environments.
  - Unlimited custom properties for all major objects (allowing internal and external clients to easily add new indexed properties of most any data type to the system via an administrator interface).
  - Double entry transaction system (to maintain the integrity when moving funds).
  - Replication of commonly requested meta data to MSDB servers running on the web servers to alleviate the load on the production server.
  - The ability to replicate all data to read-only servers for load sharing if needed (including a system that tells the web pages to pull from PROD if their data is out of date for a given user).
  - A read-only mode that can be turned on for a user, client, stored procedure, or everything, that will prevent any data writes from occurring while allowing all reads (which is very useful when doing maintenance or when migrating clients/users from one system to another).
  - A verbose logging mode that can be turned on for a user, client, stored procedure, or everything, that will cause all database writes to be logged at the stored procedure level. A verbose level can be chosen to limit the logging based on the severity level (ERROR/WARNING/INFO/DEBUG).
  - An XML template based email system for sending custom emails.
  - Flexible shopping system with multiple catalog/item sources (both internal and external) which are combined into custom virtual catalogs for clients. This enables a user to shop with multiple fulfillment centers at the same time using one cohesive interface. Full Text Indexing was used for searching item descriptions. Most of our fulfillment is handled by Amazon via this automated system.
- Designed and developed a generic survey/test system that asks questions, controls allowable answers, and records responses, scores, number of attempts, pass/fail status, etc.
- Designed and built an email system that generates custom newsletters for users based on their properties and group affiliation. This system currently sends out 150,000 custom emails in a 4 hour period.
- Wrote a system that copies the production databases to a reporting server every night from the daily backups. It sets up permissions for internal users to access the databases via their windows logins. It uses detach and attach methods so downtime is less than a minute.

Database Engineer *May 1999 – August 2003*
CBS MarketWatch (now Dow Jones)
Minneapolis, MN
http://www.marketwatch.com/

Oracle Developer/Systems Administrator *December 1997 – May 1999*
The Boeing Corporation
Long Beach, CA
http://www.boeing.com/

COMPUTER EXPERIENCE

OPERATING SYSTEM ENVIRONMENTS

Linux (RedHat, CentOS, Fedora, Ubuntu, Debian, Gentoo)
Windows XP/Vista/7
Windows NT/2000/2003

PROGRAMMING LANGUAGES

Java
Ruby/JRuby
Perl
Python
C/C++
Common Lisp/Emacs Lisp/Clojure
HTML/CSS/Javascript
XML/XSD/XSLT/DTD
ASP
SQL
Unix shell (sh, BASH)

EDUCATION

Bachelor's Degree in Computer Science *Graduated in June of 1996*
California State Polytechnic University, Pomona
Pomona, CA
http://www.csupomona.edu/

Online Version
http://kylesherman.com/resume.html

Contact vCard