**DANIEL NORMAN HUFFMAN**

|  |  |
| --- | --- |
| **Candidate Summary Sheet** | |
| **Required Skills** | **Candidate’s Experience Years** |
| Experience designing and implementing multi-tiered enterprise applications. | 10+ years |
| Experience with SQL preferably with PL/SQL and Oracle | 10+ years |
| Agile experience | 10+ years |
| Programming in Java | 7+ years |
| Web development experience with some of the following technologies; (MVC framework, HTML, JavaScript, CSS, XML, JSP, Servlets, Web Services) | 2+ years |
| J2EE | 2+ years |
| Experience in web-related and client-server/web service applications development | 2+ years |

**SUMMARY:**

A highly motivated Software Developer with 10+ years of IT background, highlighting his solid Java/J2EE development experience for large scale companies such as Capital One, Northrop Grumman, WorldCom, Raytheon, etc.   Performing in analysis, design, development, testing, integration, deployment and implementation of multi-tier distributed applications; developing new Java code from requirements or without requirements and troubleshooting and integrating multiple technologies, including, but not limited to, hardware, PL/SQL, Oracle SQL embedded procedures, C, C++, Java, UNIX shell scripts in order to function seamlessly are his core competencies. Adept in using NetBeans IDE for debugging Java and JavaScript and J2EE for Java JDBC code. Strong front-end UI development skills using scripting languages like JSP, HTML, JavaScript, and CSS.

Daniel is a USA Citizen and lives in Gaithersburg, MD. He is available for interviews given 1 day notice and can start immediately if offered the position.

**WORK EXPERIENCE:**

**Employment Gap March 2019 – Present**

* Looking for opportunities.

**Capital One August 2018 - February 2019**

**Software Engineer**

* Resurrected mothballed DLL, debugged for new platform, and created new code base in GIT. This financial calculating DLL was in use at the time, but had not been worked on or compiled for years. The DLL processed information from several databases, providing predictions for "balance attrition model" originally written by "quantitative analysis people." Using C++, and MS Studio, compiled, debugged, and created production DLL for use in Citrix environment. Added multi-threading and a thread pool design pattern object. (Added other useful design pattern like singleton, of which this code was lacking.
* Provided documentation, in the code, describing them and their use.)
* Migrated DLL for use in new platform of Linux in AWS.
* Created Python interface into DLL for other python developers in AWS environment.

**Employment Gap March 2018 – July 2018**

* Looking for opportunities.

**USTech – Raytheon June 2017 - February 2018**

**Software Engineer**

* Using Java, C++, worked on the SOTC team, supporting the JPSS-1 weather satellite that was launched in November of 2017.
* Wrote and debugged code for telemetry and command and control interface.
* Fixed code, and wrote new code, in Java, C++, Python, shell script languages. Often interpreting, and referring to older revisions, in C++ and Fortran, to fix bugs in the newer version in Java, and Python.
* Using ClearQuest bug reports, fixed bugs in the code baselines. ClearCase was used as version control. Debugging telemetry included TCP/IP and proprietary protocols, using sockets in C++ code.

**Employment Gap May 2016 – May 2017**

* Personal time off.

**NTT Data - Raytheon contracted to National Weather Service February 2013 - April 2016**

**Software Engineer**

* Worked on the National Weather Service AWIPS 2 platform.
* Debugged code in Java using Eclipse and plugins.
* Translated C/C++ code to Java code from AWIPS one to AWIPS 2 attempting to allow the user experience be identical except for the enhanced graphics abilities of AWIPS 2.
* Developed new Java code from requirements or without requirements by emulating AWIPS one.
* Debugged and wrote new Java code from code that was written by a different group of developers.
* Handled, debugged, and coded, in Java, resolutions to discrepancy reports created from trouble tickets from Weather Forecast Offices. When the front line tier one, and tier two could not fix the problem the Trouble Ticket morphed into a Discrepancy Report and handled by our group to debug. On AWIPS 2 platform: Cave UI, D2D, Radar Server subscription and heartbeat, Edex dx1, dx2, dx3, dx4, HDF5 files. Sockets using TCP/IP as well as pipes, the database, and nfs files facilitated communications between physical machines listed above.
* Technologies include: Git revision control, PostgreSQL database, XML Java, including framework Spring, Hibernate framework, C++, Eclipse.

**Employment Gap August 2012 – January 2013**

* Looking for opportunities.

**ACS October 2010 - July 2012**

**Software Engineer**

* Migrated, improved, and added functionality to a database vehicle toll, traffic and financial system.
* Created new development to interface with existing system containing limited documentation.
* Interpreted classical sequential C code to create modular and reusable C++ code for new development. The new development contains comprehension towards the ideas of retroactively replacing the 15+ year old C code with the new modular code, and the idea that the new code may be active and require to be maintainable 15+ years hence. Some aspects to achieve 15+ year maintenance on the new development are to use an extremely limited set of system calls and resources, and any system calls or resources are abstracted. Also an extremely limited set of C header files are utilized. The purpose being that as these header files change in the future, since they are not used by this code, no maintenance will be required. E. g. the few required templates, e.g. a linked-list, have been developed so that the C++ standard template library is not required and therefore with new releases of that massive library, this code will require no maintenance. Interfacing into the database is achieved by ProC and SQL. An endeavor and enhancement I proposed was to increase database efficiency by the use of stored procedures, but their use was not agreed upon in this development cycle. The creation of abstracted database objects will allow for an efficient change to stored procedures when approved.
* Tasked to troubleshoot and integrate multiple technologies, including, but not limited to, hardware, PL/SQL, Oracle SQL embedded procedures, C, C++, Java, UNIX shell scripts in order to function seamlessly. Applications tasked to account for, and reconciliation of, vehicles traveling through toll lanes at highway speed using video license plate recognition technology. The application, which cannot error, do to ridged accounting rules, operates for one of the busiest toll systems in the world; namely the MTA in New York City. Data transfer from toll plazas to the main application was facilitated by using C/C++ UNIX sockets using the TCP/IP protocol.

**Northrop Grumman February 2009 - May 2010**

**Software Engineer**

* Migrated improved, and added functionality to embedded software written in C++.
* Wrote documentation about the past development environments for migration to future projects, processes, instructions for use. Updated written documentation including the API for the libraries.
* The migration was from an older PowerPC to the newer PowerPC, MPC7410, on a proprietary DoD processing board. The processing board and the software drivers control low level monitoring and mechanics of nuclear reactor power plants. Being that it is extremely important for the software and drivers to perform correctly, code migration includes inspection of the assembly code and assembly code level proof of proper functioning. Unlike the boards that are being replaced. The newer PowerPC chips pipeline, so examination of the assembly code must insure that I/O functions in the correct order and out of order I/O does not occur. The migration involved resurrecting code and a development environment that was obsolete because the customer deliverables included the old code libraries, the new code libraries, and proof that the new code works identically to the old code. The obsolete development environment was Microsoft Visual C++ 6.0 with Nucleus EDE plug in, and the Diab 4.3g compiler. The new development environment is Wind River Work Bench 2.5, using the Wind River 5.5.1.0 compiler.
* Created new and perform existing unit tests on code using Parasoft C++ Test tool. Create documented test results on modified and existing code.
* Wrote procedure documents for technical audience on update procedures to military grade touch screen apparatus.
* Followed and improved documentation on test procedures for a turbine generator control and fail-safe system. Insufficient documentation required me to draw on electronic as well as software expertise in order to clarify, improve, and proceed with the testing of the embedded system wind turbine apparatus. For example knowledge of a circuit board, its grounds and voltages, was required to force faults testing if the code recognized and logged the fault.
* Designed, implemented, and tested a post processing utility that took machine language output, used in a DSP, from the Code Composer compiler and convert the raw hex into a usable format for use by Corelis ScanExpress 1.05 flash memory burner connected to a high speed USB JTAG controller. The utility also created and included a military required code checksum, and flash memory offsets. Since this implementation of the Code Composer 3.1 executed in Windows, I used the Microsoft Studio 2005 IDE to implement this command line driven post processing utility. The utility was configurable, on the command line, for use in the post processing of many different DSP applications, for the Rod Drive Power Supply system used in the nuclear power plant on Navy Aircraft Carrier ships. Implemented the utility by including it in the Code Composer project files of the many DSP applications requiring its execution during the post processing phase. Tested results by flashing the memories of the DSPs, connecting them to the system, and executing various tests. Used Code Composer IDE and emulator to modify, write, and debug code on both the simulator and loaded onto the DSP from the Emulator. Developed, debugged, and tested Java Front-end to nuclear power plant on Navy Aircraft Carrier ships. The Java front-end is a human-computer interface providing nuclear control rod operation and status and provides controls for operation augmentation.

**Kastle September 2008 - November 2008**

**Software Engineer**

* Designed, developed and deployed embedded software for a dual redundant Linux single board computers. The project involved using their proprietary current library of code to create an embedded driver that maintains a connection from a building's fire safety panels to Kastle's internal database and response centers. This project used C/C++, the gnu compiler, and a proprietary development environment. Short project assignment.

**Raytheon April 2007- July 2008**

**Software Engineer**

* Troubleshot, debugged and performed maintenance on the NASA EMD, Earth Observatory System, the deployed and operational system code tree.
* Designed and written new code for development future deployed system code tree.
* Troubleshot and debugged non-conformance reports for the Order Manager Server subsystem, the most complex subsystem of the total system. Includes manipulating all tiers of Order Manager: the database, the C++ middleware server, caching, and the customer GUI. Bugs include segmentation faults, memory leaks, stack overflow, etc.
* Worked closely with team members on reengineer to simplify and reduce lines of code, integrate new C++ threaded frameworks; reviewed team member’s code.
* Debugged, maintained, and created new Sybase stored procedures.
* Corrected, maintained, and created new C++ code for system functioning.
* Fixed, and maintained Perl and CGI code. Tools used in this environment include: g++ Gnu compiler, vi, gdb, Gnu Debugger, Purify, Valgrind, Clear Case version management, RougeWave code library.
* Other duties include work on the Datapool subsystem, when problems spanned both subsystems.
* Used NetBeans IDE; debugging Java and JavaScript.
* Fixed Java jdbc code, using J2EE.
* Reviewed and verified other developer’s Java code.
* Updated stored procedures to work with both C++ odbc and Java jdbc.

**PREVIOUS PROFFESIONAL EXPERIENCE**

**TCOM January 2007- March 2007**

**Software Engineer**

**ACS March 2006 - November 2006**

**Lead Software Developer**

**Secured Services August 2005 - March 2006**

**Software Developer**

**Employment Gap January 2004 – July 2005**

* Personal time off.

**Telenix September 2002 - December 2003**

**Software Engineer**

**NAPA May 2002 - June 2002**

**Software Engineer Consultant**

**WorldCom May 2001 - December 2001**

**Software Engineer Consultant**

**Decision Consultants September 2000 - December 2000**

**Software Engineer**

**EDS April 2000 - August 2000**

**Software Engineer Consultant**

**Information Mechanics Inc. July 1999 - March 2000**

**Software Engineer**

**IBM Corp. February 1998 - April 1999**

**Software Engineer Consultant**

**US West April 1997 - February 1998**

**Software Engineer Consultant**

**Atlantic Aerospace Electronics Corporation August 1995 - April 1997**

**Software Engineer**

**University of Maryland at College Park January 1993 - May 1995**

**Software Engineer**

**University of Maryland at College Park August 1992 - January 1993**

**Network Technician**

**Employment Gap November 1990 – July 1992**

* Personal time off.

**Distribution Plus July 1989 - October 1990**

**Electronic/Computer/Network Technician**

**Employment Gap August 1988 – July 1989**

* Personal time off.

**Montgomery College Physic Lab August 1987 - July 1988**

**Lab Technician**

**EDUCATION:**

* Bachelor of Science in Computer Science. University of Maryland at College Park, 1995
* Associates of Arts in Electronic Technology, Electronic Technology, Montgomery College, Rockville, Maryland, 1991
* Graduate course, C programming, MIS department, University of Denver, 1991
* Graduate courses, Computer Security, High-Speed Networking, Johns Hopkins University, Maryland, 1997