brandon KOROUS

p.o. box 7233, visalia, ca 93290 | 559.623.3290 | me@brandonkorous.com

professional summary

Forward driven talented technologist with over 17 years of extensive experience, specializing in leading high-performing and multi-disciplinary teams from requirement gathering through successful implementation. Accustomed to managing multiple full stack projects, teams, and priorities in fast-paced environments. With experience in all avenues of technology and implementation, and a strong desire for success, I bring a unique level of expertise to any business.

skills

- Full stack development lifecycle expert
- Project management – Scrum
- Advanced critical thinking
- Operational analysis
- Superior time management
- Business process improvement
- Requirement gathering
- Risk mitigation and management
- Expert in multiple programming languages including: C#, HTML, JavaScript, JAVA, SQL, and more
- Advanced knowledge of: n-Tier, MVC, OOP/OOD, WinForms, Threading, API, Source Control, Data Warehousing, Cloud Technologies
- Multiple location networking principles
- Business needs analysis
- Enterprise application integrations

professional experience

Developer III (03/2010 to 04/2010)
Manager of Technology (04/2010 to 01/2012)
Director of Information Technology (01/2012 to Current)
Principal Developer (06/2012 to current)

Employer Driven Solutions – *Formerly BEN-E-LECT* – Visalia, CA

- The planning, implementation, organization, and oversight of the migration from a POTS telephone service to a Shoretel VOIP system. This also included the migration from a T1 service to a Cable with tandem Point-to-Point for secure remote connections for a Mid-Market Insurance firm with many locations and external agents and agencies.
- Infrastructure migration that involved the existing multiple-role server system converting into a single-role virtualized solution running in Hyper-V. The servers have been migrated from Windows Server 2003 through 2016 over my tenure. These migrations included all aspects of a Microsoft based domain foundation, including: Active Directory (including the connection of internal AD to Azure AD); Virtual Private Networking (VPN); Distributed File System (DFS); Group Policy; Application Server; SQL Server 2004 through 2016; Exchange 2007 through 2016.

- To meet the growing requirements of our expanding organization, we shifted from a single domain forest to a multi-domain forest for our Exchange, DFS, and AD components to support the co-existence of multiple organizations. This organizational method succeeded in ensuring separation of organization, while maintaining an invisible feel to the end user.
- With over 100 different vendors and partners, there was the need to establish required data transmissions to and from these vendors and partners. Using Visual Studios .Net Windows Service and multiple Class projects in C#, I replaced the legacy Batch File/Windows Scheduled Task method of transmission to a more modern, effective, and trackable automated service that continuously runs, maintaining compliance and vendor requirements. This is a data driven automation service written in C# with class libraries that house the code for the .Net 4.0 Entity Framework (SQL Server connection to our 3 different database servers housing our nearly 20 different SQL Server databases) data layers and subsequent logic operations for their related data libraries. While the Automator project itself is built with its own singular SQL Database connect for its data driven instructions, all other logic and data libraries are loaded and called through reflection driven off the data instructions. This was done to reduce the maintenance overhead of changes to this core system. The tasks involved in this service consist of: Extract/Export, Import, NCPDP, X12, SSH, FTP, Backup, data based DevExpress XtraReport generation, Email/SMTP, conversion, and more. These projects migrated from being source controlled with Source Gear Vault initially, to an internal Team Foundation Server, and finally to Cloud Based Visual Studio Team Services.
- Ensure that all systems and protocols comply with the regulations of multiple organizations including: Federal and California based DOI, DOJ, DOL; HIPAA, PCI, NAIC, FIO; as well as any partner and/or vendor requirements based on scope of relationship.
- Cross application data access was achieved with Entity Framework (edmx) technology, that I developed a C# wrapper library to provide required control and implementations directly in the data layer. The result of these libraries can be very similarly compared to the Code First version of Entity Framework. When Entity Framework 6 came out with .Net 4.5, I began replacing my custom LINQ to Objects layer with the newly improved Code First.
- The need of a centralized method of access to our elaborate data systems for our staff was urgent. Resulting in the creation of: Primary Functionality Base (PFB). Which is a C# WinForms MDI application that was originally designed to provide this singular access point while replacing a piecemeal Visual Basic application for internal staff to use for daily processes. The solution for this includes 41 Dynamic Link Libraries (built with mostly C#, however there are components that are written in C++). This solution shares its data and process libraries with the Automator when applicable, but adds a GUI layer that is inherited from a few DevExpress components. Adobe and Microsoft Office are also embedded and controlled through C#/C++ api calls in their required areas to support document creation and maintenance from with the application itself, relieving the technical requirements of the end user. This solution was migrated through both our internal Team Foundation and the Visual Studio Team Services for its source control.
- Utilizing C#, CSS, DevExpress and SQL Server built an ASP.NET Web From site that would easily plug in to place with our existing Drupal website, giving our clients the ability to upload, download, or enter information while not changing the content of the main site. This was a full stack application, but only fitting niche content areas of an existing application. This application was strong in SQL queries that were called through ADO.NET connections.
- As a pet project, based on business needs, I built a large-scale C# ASP.NET MVC Razor web portal. This portal had the need to allow employees and their families, employer groups (our direct clients), insurance brokers, health care providers, and our partners access to the relevant health care information that we have processed. This portal was built with the data libraries that are used throughout the businesses other

applications providing access to all SQL Server databases and their data. A messaging component was also built into the system, with canned and open responses allowed based on permissions. This portal had to provide the utmost level of security, which I created a wrapper for ASP.NET Identity (which was migrated to Identity 2) that would allow action based authorization tied to the user's role and/or the user themselves; this was also a data driven wrapper to alleviate the need to republish the application if a roles access was adjusted. There is a strong infusion of JavaScript and JQuery throughout to provide an in-depth search and graphic representation. Extensive helpers were created to improve upon the reusability of views and actions, many of these being used in conjunction with AJAX calls to compartmentalize data loads for optimization purposes. Because of the volume of data that we present, optimization was key. Most of the increases in speed were found in a custom method of using reflection to limit the query calls to SQL based on View Models and mapping through C# helper methods built into the support library of the application and referencing this in each data layer. Also, to keep the overhead down, limited 3rd party frameworks were used. This was built with Bootstrap 3.3, with JQuery, x-Editable, Dropzone, and font-awesome. Overall design was accomplished with my interpretation of Google Material Design documentation coupled with Bootstrap LESS and adjusted through a centralized LESS file.

- Full stack design, development, and implementation of a RESTful API with supporting documentation front end for internal application data access consolidation. This solution utilized the C# version of the ASP.NET Web API in combination with the EF Code First data connection layers to streamline the extensive database connections and datasets that all our systems utilize. Providing responses in the expected: JSON and XML; as well as also creating custom export formats ranging from PDF to X12 5010 to Excel. This API portal has 256 SSL encryption to ensure data protection during transit and is hosted on one of our internal virtual web servers running Windows Server 2016 Internet Information Services (IIS) 10.0.
- Extract, Transform, and Load (ETL) is key to our operations. I had to migrate Visual Basic WinForm application into C# that managed the ETL processes as they continued to grow. This system, while composed of C#, required more SQL queries and Stored Procedures than C#. These queries and procedures would take from a flat file non-normalized system, and convert the data into a relational database, while also joining in existing data as well as data from other sources. The work of this application was performed in SQL queries and SQL Bulk Copy, while the C# GUI and Service handled the operation of the SQL.
- To meet the demands of our clients' needs to provide a speedy resolution to our wrapping insurance product, we developed a C# WinForms application that ties into our Entity Framework libraries that JQuery, JavaScript, C#, and HTML to access and retrieve data from 3rd party websites and stores the retrieved data in one of our SQL Server databases.
- Built in the DevExpress XtraReport library to allow our organization to update our reporting structure to a more modern design. These reports are housed in a .Net Library project written in C# with the relevant data layers referenced. For the more complex reports, SQL queries and stored procedures are used directly from the code.
- Developed a C# class library that could take a feed of data and convert that data into an X12 5010 HIPAA Transaction based on fluid schematics that are stored in a SQL Database and accessed through relevant Entity Framework data access libraries.
- Because of the scope of code required to create an in house centralized software system was enormous, I created a C# application that allowed me to point it to a SQL Database and using elaborate combinations of inline queries, LINQ, and stored-procedure results would generate the relevant GUI interfaces for both WebForm and WinForm integrations within a C# class library for each need. This allowed our team to more rapidly include these prebuilt libraries, and apply design practices throughout.
- Drive staff in multiple subsets of the IT department to grow their own personal experiences abilities and reach outside of the accustomed box.

- Due to a shift in business model, the need arose to create duplicated systems (with expansion potential for more duplication) that would allow multiple corporations to run simultaneously while not creating double the work for our internal staff. There had to be a clean and concise separation of work, and seamlessly ensure that our users did not cross-mingle data.

Technologies: .Net up-to 4.6.1, C#, Entity Framework, Code First, Web API 2.0, Windows Server up-to 2016, SQL Server up-to 2016, Linq, Team Foundation Server, SMTP, Javascript/JQuery, Visual Studio, WinForm, Bootstrap, JSON, XML, X12, TSQL, HTML, CSS, LESS, SCSS, CSV, Razor, NCPDP, FTP, SFTP, FTPS, SSH, ODBC, ADO, Compression, Automation, Threading, Vault Source Gear, Tortoise SVN.

Partner - Principle Developer (03/2009 to 03/2010)

GMO Developing – Visalia, CA

- Led design and development of custom Web, Desktop, and Mobile Applications, to meet and exceed client needs and requests. The tools used in this role were based on client requirements. They ranged across all operating systems, many languages (C#, PHP, JQuery, C++, Objective-C), running multiple environments (Visual Studios, Eclipse, X-Code, Adobe), accessing database in multiple environments (SQL Server, MySQL, PostgreSQL, SQLlite), building WinForms, WebForms, Windows Service, iOS, and Android applications to meet the needs as the client had specified. Many times these were full-stack implementations, with plenty of code replacement or insertion opportunities as well.
- Led client based planning meetings to develop strategies for implementations. Many times these meetings would start with a clients statement: "I think I want…", which I would then lead them in the direction to determine what they actually did want.
- Drove client decisions through wire-framing, story boarding, and market research.
- This position also required my direct involvement in our whole business process (Leads, Sales, Management, Finance, Development, Testing, etc...).

Technologies: .Net up-to 4.0, C#, Entity Framework, Linq, PHP, Drupal, Obj-C, X-Code, Visual Studio, Eclipse, SQL Server up-to 2008, MySQL, C++, SVN, Adobe ActionScript, JavaScript, JQuery, CSS, HTML, WinForm.

Application Developer (01/2006 to 02/2009)

Oce Business Services/CaseData – Bountiful, UT

- Modified existing C# based WinForm and Window Service applications to correct errors, upgrade interfaces and improve performance.
- Develop and integrate a universal document processing component for existing web and desktop applications with a combination of C# and C++ creating an Active Template Library (ATL) tool that can be embedded in any application type. This tool need to read any document type (doc, txt, xls, pdf, Lotus, etc…) and read that document content while simultaneously presenting its best interpretation of its native look. Allowing the user to save any remarks or pinpoint key elements for review.
- Rebuild the primary data migration solution for effective optimization. This was a C# desktop application that would take up to 7 days to process, duplicate, move, and import large (Terabytes) of information between our development environments and production environments based on the specification of the production development team. It was also my task to remove the need of the development staff to provide 24 hour supervision of the process by reducing errors. To optimize and remove the staffed oversite

requirement of the solution: I implemented intelligent error correction methodologies; moved the solution into a dynamic mutli-threaded environment; and separated some of the processes into separate service applications for co-processing. After the revisions, the process that took up to 7 days to complete was now complete in 15 minutes, effectively reducing the time to .001 percent of it's original.

- Utilized multiple languages to create interfacing between multiple systems.
- Consulted directly with the executive staff to create standards for new client implementations.
- Direct software design and development of new OCR and complete document review components for both our client facing web as well as our internal quality control staff.
- Create interfacing components that will directly connect to both our internal systems and third party systems including: Lotus Notes, Exchange, Office, WordPerfect, and other data storage methods.

Technologies: .Net up-to 3.5, MVC, ATL, Visual Basic, C#, C++, ADO, ODBC, Visual Studio, OCR, TSQL, SSIS, SQL Server up-to 2008, Threading, WinForm, WebForm, ActiveX, Tortoise SVN

Web Application Developer - Contract (03/2006 to 08/2006)

RealSource – Salt Lake City, UT

- Developed web application with database interaction using C# and MSSQL to allow users to input new information and allow for modifications to existing data for an online property management suite.

Technologies: .Net up-to 3.0, C#, Visual Basic, ADO, ODBC, Visual Studio, SQL Server 2000, TSQL.

Student Technical Site Management (01/2003 to 01/2004)

Utah State University Distance Education – Monticello, UT

- Ensured proper distance education site functionality to meet the student and faculty needs.
- Designed, developed, and maintained a student resource scheduling application.
- Ensured network's optimal performance and security.
- Created effective communication channels between satellite, cable, internet, and telephone systems.

Technologies: .Net, C#, ADO, ODBC, Flat File, Visual Studio

Lead Web Developer 01/2001 to 01/2003

Nothing Developing – Southern Utah

- Led a small team of entry level developers and interns in the development of HTML based websites for local clients.
- Fostered personal growth for the team in both best design and development practices.
- Tested and reviewed code and approved for production.
- Worked directly with clients to design and develop websites to further grow their companies in the rapidly expanding web market.
- Identified and eliminated website bugs to improve user experience.

Technologies: HTML, JavaScript, C++, C#

education

Utah State University: Management Information Systems & Business Administration

I completed extensive course work in: Application Development, Computer Systems Management, Computer Programming, Website Development, Network Administration, Office Systems, Business Management, Hardware Maintenance/Upgrade/Troubleshooting.