

TAHER ABOUZEID

Profile

Software Engineer experienced in a large investment-banking environment, with globally scaled systems. Highly adept at analyzing problems to find their root cause and suitable solutions. Excellent communications skills. Successful in redesigning ineffective services in order to boost performance, improve accuracy, manageability and reliability of services. Experienced in creating several scalable and efficient processes for high volume trading systems.

Motivated to venture into the development arena in a highly creative and active environment, in order to have exposure to additional aspects of the computing world and gain enriching experience in the design, development and testing of projects.

Experience

Dev Lead/Architect – Northwestern Mutual (NM) New York, NY, 03/2018–Now
Northwestern Mutual is an American financial services mutual organization which provides consultation on wealth and asset income protection, education planning, retirement planning, investment advisory services, trust and private client services, estate planning and business planning. Its products include life insurance, disability income, and long-term care insurance; annuities; investments; and investment advisory products and services.

A member of the Solution Engineering team, driving the initiative to modernize legacy platforms across the enterprise in order to support current and future needs of the company. Leading a small team to evaluate, architect, design and implement modernized versions of existing platforms, as well as building new ones.

- **Data Engine** – Designed and implemented a platform that reads vendor data in multiple formats and distributed across multiple files then aggregates them and standardizes the format to JSON data. (Java, Apache Nifi, Vert.x, Spring Boot, Cassandra, Kafka, MySql, AWS)
- **ETL-Testsuite** – Designed and implemented a new testing suite to support the migration from the legacy third party ETL (Extract, Transform, Load) system, to a newly developed ETL system. The suite made sure that there were no differences in the data between the new and the old systems when any task is run. (Java)
- **API** – Lead a team of 3 people, to design and implement a platform to allow the client experience team, as well as other groups in NM to communicate with third party external vendors, in order to perform transactions as well as query for reports and records. (Java, Spring Boot, REST, Cassandra, Apache Ignite, nginx, Docker, MySql, DB2, ELK stack, Vert.x)
- **Client Update** – Designed and implemented a new feature to update client information on 3rd party vendor systems, in an effort to centralize client data in the enterprise. (Java, REST, Swagger, MySql)
- **Prod Data Scrubber** – Implemented a new AWS Lambda function to handle s3 events regarding new files with PII being uploaded to S3, which need to be scrubbed and made available to the data science team for analytics. (AWS Lambda, Java, Jackson)
- **IPG** – Designed, implemented and Lead a team to modernize migrate an outdated legacy platform running on Websphere, which was going to be retired soon. (AWS Lambda, Java, React, NGINX, Kubernetes, Docker)

Senior Java Developer – Learnvest New York, NY, 06/2015–03/2018
Learnvest is widely recognized as a disruptive player in the personal finance industry, LearnVest is an award-winning subscription-based program designed to help people make progress on their money. The company combines a dedicated planner with innovative technology to help make financial planning affordable, accessible and even delightful.

Responsible for developing and maintaining a java library that generates the financial plans that planners use to help their client create a budget for their finances and achieve their financial goals. I am responsible for the general design and architecture of the library. I make sure that the code is scalable, reusable, simplified and flexible towards growth and future development. I also interact with other teams in order to provide them with the means to retrieve and extract the data generated by our library, which involves multi team cooperation to accomplish the deliverable. Ultimately this work allows for more work to be done with less time and effort.

- **Ongoing Refactoring of the Financial Library** – Developed, Improved and Maintained the design and implementation of the financial library that the core product of the company runs on. (Java)
- **Data Validation Framework** – Designed, Implemented and Integrated a new validation framework to replace the rigid and non-scalable legacy implementation. (Java)

- **Test Suite** – Built a new test suite framework to allow generic testing for all inputs without having to specify expected output for each individual test case. (Java)
- **Migration and Improvement of CI/CD** - Responsible for multiple migrations of the CI/CD systems.
- **Development Tools** – Built various tools for developers and QA members to be able to perform their tasks much more easily and efficiently (Java)
- **AWS Lambda Testing** – Built a new tool/framework to allow testing thousands of test cases at high speeds. (Java)

Senior Java Developer – Sailthru New York, NY,

01/2013-06/2015

Sailthru is a Smart Data company with an international client base. Based in New York City, the company's Smart Data solution makes automated decisions based on user activity across clients' websites, emails and mobile applications.

Responsible for the development and support of the backend systems responsible for generating dynamic smart lists of emails to send to for various clients with whatever email templates they wish. I'm also responsible for refactoring and improving the services responsible for sending out emails as well as maintaining statistical data to be processed and viewed by our clients.

- **Solving ongoing issues** - Improved memory usage for company mailing services, which in turn improves daily operations and ongoing service continuity. (Java)
- **Mailer Refactoring** - Refactored existing java code for the mailing services, to make the implementation more flexible to change and development of new features to support the growing needs of the company. This change also resulted in a drastic decrease in the number of support alerts that required manual intervention, which in turn gave the development team a lot of time to focus on new initiatives. (Java)
- **Converting Legacy PHP Code** - Converted existing php code to Java services to take advantage of threading techniques as well as moving away from crons to run tasks and expand the available options to managing and processing company data. (Java, PHP)
- **Refactoring PHP Codebase** - Redesigned the php codebase to establish a proper data layer and decoupling the various components of the code to make them more generic and flexible towards change during the evolution and growth of the product in order to accommodate the growing needs of our clients. (PHP)
- **Design and Implementation of New Features** – API and UI modifications to allow clients to request the generation of mailing of email/sms/push notification campaigns with varying and user specific content. UI changes include a complete redesign of the interface and re-evaluation of the campaign workflow. The API changes involved an intimate understanding of our clients needs and coming up with a suitable solution to meet those needs while making the best use of the company's resources. (PHP, JavaScript)
- **Development Standards** – Am a member of the group responsible for devising and implementing the coding standards and best practices for coding in java. The group was responsible for determining the best way to redo the java code to make it more modular and generic in order to make the components reusable and more efficient in the future endeavors of the company. The work involved setting company standards for coding in java as well as implementing the workflow changes to enforce them.
- **Support Tools** - Developed new support tools to help the support team do their jobs more effectively and in a timely manner (Java, PHP)

Senior Java Developer – Citigroup New York, NY,
(Contractor until August 2011)

02/2010-08/2012

Citigroup Inc. has the world's largest financial services network, spanning 140 countries with approximately 16,000 offices worldwide. The company employs approximately 260,000 staff around the world, and holds over 200 million customer accounts in more than 140 countries. It is a primary dealer in US Treasury securities.

Responsible for the development and support of the credit electronic trading systems, which deal with post trade messaging, market communication, electronic trading, pricing and negotiation.

- **Creation of a New Framework** – Created a new framework that employs the concept of workflow with reusable components, resulting in a framework that allowed several groups in multiple regions of the world to construct services simply by creating a new configuration file. Among the services created were pricing sources, post trade feeds, data bridges and price calculation services. This framework constitutes 90% of the credit e-Trading environment that our project has built. (Java, JavaScript, FIX, EMS, JDBC, KDB, ION MarketView, Spring)
- **Refactoring of Legacy Code** – Refactored existing services to accommodate new requirements and to make it more reliable and less error prone. As a result modifications became a lot more easy and required less time and code modification. (Java, JMS, Spring)
- **Migration of Existing Services** – Migrated several post trade feeds from legacy code to the new framework. The result is greater flexibility to accommodate change and almost no production issues due to the framework. (Java, JMS, Spring, JDBC, Sybase)
- **Design of the eTrading Environment** – Responsible for the environmental infrastructure and allocation of information within the eTrading platform, to make sure that the information is normalized and available to all services that seek to use it to perform various functions and process data in a variety of ways.
- **Design and Implementation of all Data Processing Services** – responsible for designing and implementing all of the services that constitute the platform and process all of the data contained within it, using the new framework

mentioned above, save for the central notification system that the environment uses. (Java, Spring, XPath, XStream, bash, JDBC, ION, Sybase)

- **Design of the Central Notification System** – One of two individuals who designed the central hub of the eTrading platform by creating the set of interfaces necessary to implement the hub as well as deciding on the various features and policies to be implemented by it.
- **Design of Client API for the Hub** – Designed the client API that allows services to communicate with the central hub. This allows the user to synchronously query for information as well as subscribe to future updates to data depending on the query submitted, the implementation alleviates the burden of API specific code for the persistence technology being used by the hub so that if a new persistence technology is used, none of the clients of the hub will have to make any modifications, which in turn makes it ideal for environments where the latest technology and changing resources is a constant. The user is also able to request that the data be throttled in its delivery, to allow the client an extra measure of control.
- **Implementation of New Operating Standards** – Implemented new standards of operation by making full use of source code maintenance and versioning to streamline deployment and debugging production issues. Also instilled the use of Task Management tools to manage the assignment of tasks as well as provide stakeholders with immediate status reports rather than consuming time keeping superiors informed manually.
- **Consolidation of Fragmented Code** – Various versions of source code were consolidated into a single codebase to reduce the number of variables in play when analyzing production problems and coming up with solutions, which would work for all users rather than just a single branch of the code. This way the amount of time invested in development will be significantly reduced since you don't need to account for multiple forms for a single solution.
- **Training and Support** – Pushed for the use of a support team to allow developers to focus their time on delivering to the business rather than support. Provided tutorials to various stakeholders, perspective users as well as the support team so that they would gain a better understanding of the environment in order to support it.

Java Developer – Bank of America New York, NY,
(Contractor)

08/2008-10/2009

Bank of America is the largest financial services company in the world, largest bank by assets, second largest commercial bank by deposits, and (previously) third largest by market capitalization in the United States. Also, Bank of America is the number one underwriter of global high yield debt, the third largest underwriter of global equity and the ninth largest adviser on global mergers and acquisitions.

Responsible for development and support of a globally scaled trading system, dealing with 2 distinct trading groups, corporate bonds and rates bonds, which encompasses government and agency bonds.

- **ICAP Market Data Service** – Created a service that connects and receives market data from ICAP by employing the FASTFix protocol. (FASTFix, Java, Spring)
- **Datalink Trade Bridge** – Designed and implemented a service that connects to an MQ queue and receives various trades from various regions. The service was designed to make it capable of connecting to any number of queues and handle the data received in any number of ways, all through configuration. It is an essential component in bridging trade information from Merrill systems to Bank of America Systems.
- **Fanny Mae Exchange** – Assisted in the creation of a service that connects to Fanny Mae and receives a Post Trade Feed by employing the FIX Protocol. (FIX, Java, Spring)
- **Ion Publishers** – Designed and implemented a service that subscribes to various instruments quotes from the Pioneer environment and maps then publishes them to various chains on the ION Platform. The code for the service was designed flexibly enough so that there could be as many instances as necessary running, each easily modifiable through configuration to have as many listeners and publishers as the user may desire, making the need for code changes an absolute minimum. (ION, Java, Spring)
- **Bloomberg Agency Price Feed** – Designed and implemented a service that took prices that were bridged over from Merrill Lynch Legacy systems and mapped them to the correct fields in the Bloomberg messages so that they would be sent across properly and displayed correctly in the desired Bloomberg pages. (Java, Spring)
- **Trade Straight Through Processing** – Designed and implemented services that receive confirmation of trades performed from various markets such as Tullet Liberty, MarketAxess and Bloomberg. (Java, Spring)
- **Treasury Dealer Negotiation Engine Manager** – Designed and implemented a service that holds and updates the various parameters involved in calculating prices to quote out for various instruments and various markets. It acts as the bridge between the Traders trading application and the various platforms that are required in order to update various pieces of information necessary for the traders to setup automatic quoting for an instrument. (Java, ION)

Platform, Spring)

- **Price Volume Tool** – Created a tool that plots either a price/volume graph for each instrument or compares the prices for multiple instruments. (Java)
- **Start Execution Processor** – Modified and upgraded an existing service to read trades from other groups and forwards them to the service responsible for processing the trade on Bloomberg, but only after processing the information and mapping the various fields of the trade to the required Bloomberg fields, as well as the necessary logic in order to set the various indicators of the Bloomberg trade for it to indicate the trades status. (Java, Spring)
- **Pioneer Adapter** – Created an adapter to allow other users from other groups to be able to use our API to subscribe to and process notifications from our environment. (Java)
- **Pioneer MQ Wrapper** – Created a wrapper class for IBM's Websphere MQ in order to be able to interact with other systems that employ MQ Queues in order to transfer information. (Java, Websphere MQ, Spring)
- **MarketAxess Autospotting** – Modified an existing service to support handling requests for autospot prices for 'On The Run' treasury bills. (Java, XML, Spring)
- **Ion ARTS (Anvil Repo Trading System) Support** – Supported a 3rd party system from ION, which is used by the traders in order to perform repo trades. Responsibilities of the support team include creating accounts, modifying static data, diagnosing existing problems and passing them along to the vendor in order to fix or perform further diagnosis, which can only be done by the vendor. Developed scripts that generate various reports and perform certain tasks are also part of the duties of the support team.
- Constant modification and upgrading of various services in order to accommodate the requests of the various trades.

Java Developer - UBS investment bank Stamford, CT,
(Contractor for the first 10 months)

10/2006-05/2008

UBS is a diversified global financial services company, with its main headquarters in Switzerland. The world's largest manager of private wealth assets and second-largest bank in Europe, by both market capitalization and profitability.

Responsible for development and support of a globally scaled trading system, dealing with 2 distinct trading groups, corporate bonds and rates bonds, which encompasses government and agency bonds. One of only a few people to work on almost all parts of the system.

- **Market Infrastructure** - Incorporated various ECN Markets into the application, which enabled traders to post offerings, accept orders as well as respond to bid and offer requests from various brokerage systems, using ION MarketView gateways as well as in house multi-threaded gateways that handle market communication and STP. (Java, JDBC)
- **Trace** - Redesigned and re-implemented the multithreaded Trade Reporting and Compliance Engine. Contributed to the redesign and re-implementation of a Swing based GUI Tool, which was used to interact with the service. The results of these efforts were improved performance, reliability, scalability and manageability. (Java, JDBC, Swing)
- **Two Sided Match** - Changed and safeguarded the access of the various threads to the cache in order to insure data integrity, which resulted in improved matching accuracy and efficiency. (Java)
- **Tradeweb STP Changes** - As part of incorporating a new market with non-standard STP requirements, a resulting fundamental change to the trade metadata handling was required, which involved modifications to stored procs and several server applications as well as handling of ION MarketView gateway information. This resulted in improved data allocation in the database and improved the ability to retrieve information for a trade. In this case worked as a business analyst by approaching the end users and discerning how they wanted the information presented to them and in what format along with the specific content desired. (Java, JDBC, Stored Procs)
- Served as the 'Go To' guy for all Matching Engine issues (Two Sided Match) as well as trace related issues while also taking on the role of a mediator and communication conduit between NASD and all of the stakeholders.

Our group was the second and final level of production support globally.

Intern - Microsoft Cairo, Egypt,

7/2004-8/2004

Member of the Visual Studios Team, responsible for checking and reporting bugs in the latest developments for Right to Left programming in Visual Studio 2005 for the purpose of professional usage. Presented remedial solutions or program additions to avoid any recurrent errors.

Education

Cornell University, Ithaca, NY, USA — Masters of Engineering – Computer Science - 2006

University of Warwick, Coventry, West Midlands, UK — Bachelor of Science - Computer Science - 2005

Skills & Activities

Languages: Java, SQL, PHP, Bash

Tools and IDE's: Eclipse, Microsoft Visual Studio, DBArtisan, IntelliJ IDEA

Databases: Sybase, Oracle

OS: Unix, Linux, Windows, Mac

Fluent in Arabic and English. Conversational level of French, Beginner in Japanese.

- Webmaster for the Warwick Anime Society. – 2004/2005
- Volunteer work teaching Sudanese refugees how to use a computer. - 2001/2002

Projects

PROJECT	PROJECT DESCRIPTION
Media Manager (2008)	A Program that organizes and archives various media files and formats, using the Nexus API.
Nexus (2008)	A generic API that allows a programmer to design and code truly flexible and powerful processes, by providing an internal database running within the process allowing the user to easily and flexibly tie various components using SQL queries to subscribe for various notifications and cached data.
Masters Project (2006)	Biologically Inspired Evolutionary Machines. The investigation performed in this project was aimed at discovering whether there is a correlation between the speed with which a virtual robot could learn to walk, and the number of legs that it had. A set of neural nets that had a genetic algorithm applied to them to produce multiple generations; were used to control the robots and investigate this issue.
Routing Project (2006)	Software that acts like a router, which receives packages with different headers, takes them apart, places a new header on them and forwards it to the appropriate port to be handled.
Cornell Treasures GPS Tour (2006)	Client/Server software where the client is used on the phone to connect to the server in order to view content for the 'Cornell Treasures', based on the users location. Given the location of the user, the phone would display various point on the map that a visitor can go to in order to see a particular display, along with a small picture and blurb about that display.
Warwick Graduation Project (2005)	The use of optimized graphical performance of the Xbox to perform rendering jobs on behalf of other rendering software that runs on pc. A group of Xboxes would have Linux and a certain rendering program on them, which was used by the software created for this project to render various portions of an image and then send them back to the main user, which is using the Xboxes as a rendering farm
Supermarket Project (2003)	A Client/Server system created to be used on multiple cashiers in a supermarket.
Robot Explorer (2002)	An algorithm to allow a robot to explore a maze and find the direct route to the goal.