Contact

mark.rubin@gmail.com

www.linkedin.com/in/markrubin3 (LinkedIn)

Top Skills

HTML

Agile Methodologies

Git

Patents

Personalizing an Online Service Based on Data Collected for a User of a Computing Device

Mark Rubin

Android engineer

San Carlos, California, United States

Summary

• Experienced Android developer with expertise in Kotlin and Java. My most recent full-time programming employment was for Square/ Block.

• I have over 25 years experience as a developer for a variety of other platforms in a variety of other languages, including Kotlin, Java/ Android, single-page Javascript/HTML clients, Node.js services, Flex/Flash/ActionScript 3.0 clients, and Win32/C++/COM clients. I have had some brief stints with Objective-C/iPhone and C#/.NET development as well.

Experience

Square

Senior Software Engineer

May 2024 - January 2025 (9 months)

San Francisco Bay Area

I return to the Appointments team, now called the Health and Beauty team, to work on its Point of Sale application running on Square's custom devices and standard third-party devices.

Big Nerd Ranch

Instructor

December 2022 - May 2024 (1 year 6 months)

Teaching bootcamps for Android, Kotlin, Compose, and Go.

Johns Hopkins Center For Talented Youth (CTY)

Online Computer Science Instructor`

July 2022 - April 2023 (10 months)

Instructor for the Introduction to Java and Advanced Java courses for academically talented youth taking CTY courses online. The courses are described at https://cty.jhu.edu/programs/online/courses/introduction-to-java-ija and https://cty.jhu.edu/programs/online/courses/advanced-java-programming-aja.

Square
Senior Software Engineer
September 2017 - July 2022 (4 years 11 months)
San Francisco, California

In 2017, I joined the Terminal Android team at Square. Based on my success, 6 months after joining, I was asked to become the sole Android engineer for the Business Banking team, developing support for the Square Card. I drove product and technical conversations, established priorities and timelines, and implemented all aspects of the application flows. I drove conversations with the server team to make our server APIs friendly for the mobile client. I even earned a patent (SQ-1069-US1) along the way.

After the Square Card's release, 6 months later, I was asked to take a position on the SDX team, the team responsible for developing the version of Square's Point of Sale running on custom hardware running a custom version of Android. My primary achievement was the development of the Sign In App, an application that extracted the Point of Sale authentication flow into its own application and redirected the seller to one of now several possible Points of Sale they could choose on their customer hardware.

Square's custom hardware does not have Google Play Services, and so does not get push messaging for free. I next worked with the Android Platform Team to integrate a custom push implementation into our Points of Sale running on our custom hardware, having a strong hand in the API development and integration across the platform and the application.

After about a year, a Square reorg led me to joining the Appointments team. I designed and implemented the Appointments Point of Sale for Register, as well as serving as the key figure in coordinating all teams across Product, UX, Marketing, QA, hardware acquisition, Android Platform, Analytics, and Engineering Management.

I then successfully lobbied for a new architectural team at Square, focused specifically on simplifying development of applications on Square's custom hardware and custom build of Android, making it more uniform with development for third-party Android development. I became one of two founding members.

Yahoo!

Principal Software Engineer

July 2013 - August 2017 (4 years 2 months)

Sunnyvale and San Francisco

In 2015, I began work as an Android and Java developer, spending the year working on Yahoo's new Messenger client for Android. The application was built from scratch, and despite no previous Android experience, I taught myself enough to successfully be a primary decision maker (and implementer) for its initial architecture and dependencies. That fundamental architecture remains and has proven quite successful and flexible. Some of my notable contributions to the project include its heavy use of Dagger2, Retrolambda, Glide, OkHttp, and EventBus, and its implementations for @mentions, notifications, compose, photo and GIF picking, debug drawer, accessibility, and many more internal utilities and mechanisms large and small. I recently revised the app to make use of Android's data binding facilities, shimming it with a homegrown data binding solution built before Android's was available.

For a few months in 2016, I also learned Objective-C and the basics of iOS programming and contributed some significant bug fixes and small features to the iOS version of Messenger while the iOS team was shorthanded.

Xobni

Software Engineer

November 2009 - July 2013 (3 years 9 months)

I worked on Xobni’s generic out of box/on-boarding application. It was a flexible, rich web client backed by Node.js. It was used by several web and native mobile clients to walk users through our login, signup, and third-party OAuth flows.

I was one of the lead developers for a Chrome/Firefox/Safari extension that integrates Xobni functionality with GMail. This was a single-page HTML5/ Javascript application that used Xobni's cloud API to fetch rich contact data to display as the user interacted with email messages in GMail.

As a pet project, I built a CardDAV server for Xobni contacts in Node.js (on top of jsDAV). This allowed the iPhone and OSX Address Book, for example, to sync locally a user's Xobni contacts without a special Xobni client.

As a pet project, I built an LDAP adapter for Xobni contacts in Node.js (on top of ldapjs). This allowed the iPhone and other LDAP clients to perform LDAP directory searches against a user's Xobni contacts.

C#/.NET programming for the world's leading Outlook plugin, focusing particularly on the .NET/COM bridge, Win32 API hooks (e.g. subclassing Windows messages), and COM Addin support. I was also one of two engineers to refactor the plugin to support x64.

Gaia Online

Software Engineer

March 2008 - November 2009 (1 year 9 months)

Senior Flex 3/ActionScript 3.0 developer for zOMG!

I added many game features, but was particularly focused on optimization and efficiency issues (e.g. finding and fixing reasons for low frame rates and excessive garbage collection).

I helped spearhead the project to retroactively add Flex to the game, and architected a new UI framework based on Flex modules. I also implemented several new and improved UI elements.

I wrote the MMO's wrapper around AOL’s ActionScript AIM API (wimas3), implementing chat over AIM for the game.

Gaia had no build process for its front-end applications when I joined, so I created a continuous integration system, back-porting several projects to use it, including zOMG!

Yahoo!

Senior Software Engineer

June 2006 - March 2008 (1 year 10 months)

Senior developer for a Flex 3/ActionScript 3.0 implementation of Y! Messenger for the Web

I was one of the key architects and implementers of Yahoo!’s web messenger product, built using ActionScript 3.0 and the Flex 3 framework. I also designed and built a continuous build system for the project. This product was spun off into JetBlue’s instant messenger solution on its planes in flight.

The product was also spun off into myM, a web messenger product that integrated messages, wall posts, vitality information, etc. from Facebook, MySpace, and AIM. myM was released in beta internally and to some test groups with much acclaim, but Yahoo! shelved the product in early 2007 before releasing it to the general public.

Intuit

Staff Software Engineer

February 2005 - June 2006 (1 year 5 months)

Served as technical lead for a team of five regular (and sometimes as many as ten) developers refactoring the QuickBooks code base to remove the limits on the number of elements in a QuickBooks list (these are feature-level lists, such as the Customer List, Vendor List, Item List, etc.). Essentially, this involved changing a fundamental data type used in a very large codebase. Because the list subsystem and this data type permeated the codebase, the project had tremendous impact on the product and required changes to almost every major piece of functionality in QuickBooks (approximately 200,000 lines of code were changed across more than 20 modules); it also posed particular problems for serialization code and external (e.g. API) clients. The challenges in the project included rooting out what portions of the codebase (often old and undocumented) would be affected, figuring out how to change them, and helping to manage and coordinate a very large set of changes.

Critical to the success of the project was my development of an instrumentation class to cause compilation errors in many places where we missed changing one of these fundamental data types.

FactSet Research Systems

3 years 4 months

Lead Software Engineer

January 2004 - January 2005 (1 year 1 month)

Lead architect and programmer for many solutions that integrated FactSet’s data with Microsoft Office (Active Publishing Workstation (APW) and the FactSet-Excel Link), using C++, COM, ActiveX, Win32, DDE, Excel's binary file format, and VBA.

APW was FactSet’s publishing solution, allowing users to populate and update Office documents with FactSet financial data; the FactSet-Excel Link allowed clients to integrate FactSet data into their spreadsheets in a myriad of ways, including the exporting server-based reports to Excel (using XML and HTML), providing custom worksheet functions to retrieve FactSet data, real-time updates in Excel cells (using DDE), and merging requested FactSet data into Excel files on disk by manipulating the Excel file format (BIFF).

Helped spearhead FactSet’s transition from multibyte to Unicode in its PC applications. Wrote the code to enable all the Office integration products and features to support Unicode.

Senior Software Engineer, Lead Excel developer

October 2001 - January 2004 (2 years 4 months)

Rearchitected FactSet's Excel spreadsheet updating functionality (called Data Downloading), replacing a difficult to maintain and to enhance algorithm involving file format manipulation with an algorithm based on OLE automation of Excel and HTML/XML/CSS. This enabled several important enhancements that were outstanding requests for many years. The process involved touching a great deal of low-level code and modifying FactSet's proprietary protocols for transmitting data.

Rearchitected FactSet's mechanism for exporting online reports to Excel, replacing a DDE-based mechanism with an XML/HTML/clipboard based one. Based on my architecture and leadership, FactSet is now able to export reports to PowerPoint presentations and to Word documents.

Implemented linking of Excel cells populated with FactSet data to online documents from which the data was pulled, architecting a solution that uses a custom URL protocol and COM.

FactSet Software Systems

Software Engineer

October 1998 - October 2001 (3 years 1 month)

Taught myself C++, COM, MFC, and the Windows API after not having programmed in C for 10 years.

Various enhancements to FactSet's home-grown thin client, using C++, MFC, and the Windows API.

Rearchitected FactSet's custom worksheet function for retrieving user-requested data. Converted the code from an old Excel C API to using Excel OLE automation, fixed many bugs, and implemented many enhancements and stabilizations.

Johns Hopkins University, Center for Talented Youth

Instructor, Teaching Assistant

January 1985 - January 1997 (12 years 1 month)

Instructor and co-author of a text for Introduction to Theoretical Computer Science at Johns Hopkins Center for Talented Youth (CTY) summer program. Johns Hopkins’ CTY program allows academically gifted junior high school and high school students to study college and graduate level material over the summer. In a single three week session, they learned in my course Pascal or Scheme; Propositional Logic; finite state automata theory; the Chomsky hierarchy; and Turing Machines and undecidability. Since no course like this exists elsewhere, a co-author and I devised a 200+ page textbook to serve as the course text.

Education

University of California, Los Angeles

Ph. D., Philosophy · (1991 - 1998)

Princeton University

A. B, Philosophy · (1986 - 1990)