

Observer Protocols

Senior iOS Developer - Panera Bread

Singleton, TX - Email me on Indeed: [indeed.com/r/Observer-Protocols/4daef43fc42f0be2](https://www.indeed.com/r/Observer-Protocols/4daef43fc42f0be2)

Willing to relocate: Anywhere

WORK EXPERIENCE

Senior iOS Developer

Panera Bread - Wilmington, DE - March 2017 to Present

Tasked with developing enterprise iOS applications for data analysis and to improve employee efficiency.

- ❖ Designed and developed the Sales Monitor application as the sole iOS developer.
- ❖ Created a custom UI/UX design for the app as the desktop program has a very poor user interface and was inflexible in the data that it displayed.
- ❖ Built the native iOS application from the ground-up in Objective-C using Xcode.
- ❖ Utilized Firebase integration for Apple Push Notifications.
- ❖ Involved in all aspects of mobile application development from design to deployment.
- ❖ Managed myself and my workload with Jira Kanban board, tracking issues and managing tasks.
- ❖ Employed an Agile/SCRUM methodology, outlining sprints and setting priorities for daily scrums.
- ❖ Worked closely with off-site development teams including backend services development.
- ❖ Worked closely with the network manager to push the Sales Monitor application to approximately 90 company devices and various employee personal devices using the AirWatch platform.

Technologies Used: Xcode 8.2.1, Xcode 8.3, Objective-C, Storyboard, CocoaPods, AFNetworking, MKDropDownMenu, FSCalendar, SWRevealViewController, DR-Charts, Apple Push Notifications, Firebase, Postman, Instruments, PromiseKit, SQL Server 2016, AirWatch

Senior iOS Developer

Bosch - Chicago, IL - July 2016 to January 2017

Worked on 3 different projects during my stay at Bosch. Thingbook, Goto, and Dremei 3D

- ❖ Kept Jira regularly updated to document the tasks and work that was done during each sprint.
- ❖ Worked closely with UI/UX design manager to provide feedback for designs
- ❖ UI developed using Storyboards with Autolayout constraints
- ❖ Implemented CoreData to store data persistently when working on the Thingbook app
- ❖ Migrated legacy code from Swift 1 and Objective C to Swift 3 when working on the Dremei 3D app
- ❖ Cocoapods utilized for management of 3rd party frameworks and dependencies
- ❖ Updated Goto app to work with recent API updates from Apple and worked on releasing said updates to the iOS app store
- ❖ Debugged various issues using a combination of Breakpoints, LLDB statements, NSLog statements, Crash Logs, and Instruments

Senior iOS Developer

Allure Medical Spa - Shelby, MI - April 2015 to June 2016

Main emphasis on this particular project was user interface; I worked heavily with core graphics, core animation, UIViewAnimateWithDuration, and UIDynamics to spruce up and make the app more enjoyable to use

- ❖ Application is compatible with devices running iOS8+
- ❖ App was originally XIB based but was migrated to a multiple storyboard setup for smarter easier setup

- ❖ Lots of work with size classes to handle all possible screen sizes; priorities were managed to ensure no conflicts arose between constraints
- ❖ UIImagePickerController within AVFoundation was used to capture/scan credit cards, or barcodes/QRcodes; POC'd Scandit and RedLaser
- ❖ Worked heavily on the shopping cart portion of the application; CoreData was used to ensure the shopping cart information was never lost regardless of user logging out, loss of internet connectivity, or any other miscellaneous occurrence
- ❖ All multithreaded executions were managed by Grand Central Dispatch calls
- ❖ Distributed team required lots of communication via email, Lync, skype, and such; accurate and precise communication was primordial due to time zone differences as well
- ❖ JIRA was used for tracking and assigning of tasks
- ❖ Lightly worked on apple pay and passbook integration; did utilize the Local authentication framework for biometric fingerprint scan usage
- ❖ Continuous integration handled with Jenkins containing rules to push out builds every morning

iOS Developer

Küdzoo Inc - Reading, PA - February 2014 to March 2015

Heavy emphasis was placed on MVVM and decoupling in this project; code reuse was at a maximum and everything was created programmatically

- ❖ App was developed to be compatible with devices running iOS7+
- ❖ In charge of creating constraints programmatically using NSLayoutConstraint and by way of the visual format language at times
- ❖ NSOperations were used for all complex multithreading needs including but not limited to getting points information, popular items to shop, motivational reminders, and the like. Both NSBlock and NSInvocation operations came in to play as well as assignment of priorities for streamlining and optimizing network utilization
- ❖ Got to work with CorePlot for creating line graphs, bar charts, pie charts, and similar
- ❖ Lightly involved with StoreKit and the in app portion of the application
- ❖ Evolving team of 4-6 developers maintained and kept adding features to the app as necessary
- ❖ NSCache utilized to mitigate loss of internet connection
- ❖ Worked with several 3rd parties like AFNetworking and MBProgressHUD which were managed with cocoapods

iOS Developer

Snapp Development - New York, NY - September 2013 to January 2014

Main role focused on utilization of mathematical libraries for performing calculations; extensive work done to create a framework capable of easily scaling and being simple to maintain as well as use

- ❖ Application developed to be compatible with devices running iOS5+
- ❖ Grand Central Dispatch utilized to periodically get/post conversion requests ; XML data would be parsed with NSXML parser and model objects created for tracking trends and to make predictions
- ❖ Created POC's using push notifications in regards to pre computing conversions for a user based on the location and perceived destination / route
- ❖ SQLite was leveraged for storing the most recently retrieved earthquake data, and to ensure functionality when lack of Wi-Fi and cellular connectivity existed; reachability classes were used to determine changing states of connectivity
- ❖ Participated in daily standup meetings as part of the agile scrum process, and worked very collaboratively with other developers. We pivoted at times rather sharply within sprints but the team was always very good at taking changing requirements well

- ❖ Several categories were created for simplification of commonly used code. For example NSNumber category for signifying metric to imperial system conversions, to go between different variations of measurements, and such were made.
- ❖ Performance tuning of the application was done with instruments to ensure no leaks, zombies, or other adverse conditions existed in the app
- ❖ SVN used for housing source code and keeping version history

iOS Developer

Hex Inc - Holmdel, NJ - June 2011 to August 2013

Designed to work with iOS 4 and later

- ❖ App UI done programmatically for both iPhone and iPad. Rotation handling was done by registering for UIDeviceDidChangeOrientation notifications and firing a selector that would set appropriate frames for objects
- ❖ Back end was JSON and response was parsed using SBJSON ; NSURLConnection and ASIHTTP utilized
- ❖ UIKit framework was used to design and implement user friendly UI
- ❖ Memory management system used was Manual Retain Release ; worked with retains, releases, autoreleases, and autoreleases pools heavily
- ❖ Worked with design team to build UI based off wireframes from Illustrator files
- ❖ Fixed bugs when notified by local QA team and overseas team in India ; unit tests created using Sen Testing Kit, OCunit and OCMock objects
- ❖ Environment for development was Agile Scrum with a high emphasize on test driven development
- ❖ Conducted memory management tests using Static Analyzer, Instruments, and NSZombie

Software Engineer

Tribune Broadcasting Company - Chicago, IL - January 2010 to May 2011

Designed GUI for HEX systems

- ❖ Implemented Core Location and MapKit frameworks for web services and displaying information on a map
- ❖ UITableViews with custom UITableView cells used to portray all calculated information to users in the most efficient and elegant manner possible ; a wide array of tableview datasource and delegate methods were used
- ❖ AVFoundation and MediaPlayer frameworks used for playing of all video material
- ❖ Worked on beta versions of iOS and XCode for creation of POC's on Apple Watch and working with Swift
- ❖ Utilized a continuous integration process and kept source code up-to-date and well-organized using Bitbucket.