

SHELDON A D'COSTA

Lead Android Developer

Phone: (718) 554-3058

PROFILE SUMMARY

- 9+ of expertise in developing Android apps, with a proven track record of delivering high-quality solutions; capable of developing Android apps for diverse devices including tablets, phones, smart TVs, and Android Smart Watches.
- Successfully published multiple apps on the Play Store, demonstrating a strong portfolio of commercially available applications.
- 3+ years of experience in Flutter and Dart development and 8+ years in Mobile Development
- Proficient in Internet of Things (IoT) technologies, leveraging IoT knowledge and experience to develop innovative and interconnected mobile solutions.
- Extensive proficiency in Android Studio for Android app development, utilizing Java and Kotlin along with native and third-party libraries.
- Demonstrated experience in transitioning code from Java to Kotlin and implementing new features in Kotlin, enhancing codebase maintainability and performance.
- Skilled in working with RESTful Web Services, employing frameworks such as Retrofit, Volley, and OKHTTP for seamless data communication.
- Proficiency in web technologies including JavaScript, HTML, AJAX, JSON, XML, and SOAP, facilitating integration with web-based content.
- Utilization of JetPack components to streamline Android app development and maintenance, ensuring scalability and efficiency.
- Implemented ADA-compliant features in multiple applications, ensuring accessibility for users with disabilities.
- Experienced in Agile Scrum software development methodologies, actively participating in daily Scrums, Sprint Planning, and Backlog refinement.
- Proficient in testing on devices or emulators using Jenkins CI server for continuous integration, ensuring code quality and reliability.
- Skilled in various architecture and design patterns such as MVC, MVP, MVVM, as well as Singleton, Command, Facade, and Proxy.
- Proficient in leveraging cellular connectivity for remote monitoring, control, and data transfer in IoT and M2M applications on the Android platform.
- Experienced with GSM/GPRS modules for SMS-based applications, such as vehicle tracking and security systems, with potential extension to LTE/5G technologies.
- Skilled in wireless communication projects using Bluetooth for home automation, robotics control, and similar applications.
- Experienced with IntelliJ, Eclipse, and Android Studio IDEs, proficient in coding in Java and Kotlin.
- Expertise in designing well-crafted user interfaces following Google Material Design Guidelines, utilizing mockups and wireframes in Balsamiq and Mockito.
- Proficient in various source control tools including BitRise.io, SVN, Git, BitBucket, and SourceTree.

Technical Skills

Programming Languages: Java, Kotlin, JavaScript, jQuery, Python, PHP, C/C++, HTML/CSS, Bootstrap, Flutter

Databases: SQLite, MongoDB, Realm, Room, OrmLite

IDE: Eclipse, Android Studio, Firebase, IntelliJ, Visual Studio

Design Standards: Material Design, Figma, Zeplin

Design Patterns: MVP, MVC, MVVM, MVI, Clean code architecture

Development: Algorithm Writing, API Development, Server-Side Programming, Continuous Integration, Agile Scrum

Continuous Integration: Jenkins CI, Travis CI, Bitrise.io, TeamCity

Version Control: Git, GitHub, GitLab, SVN, Bitbucket, SourceTree, Mercurial

RESTful Web Services: REST, SOAP, XML, JSON, GSON

Threading: Loopers, Loaders, AsyncTask, Intent Service, RxJava, Coroutines
 Testing: Mockito, Junit, Espresso, Robotium, Test Fairy, LeakCanary, Firebase Crash Reporting, Fabric Crashlytics
 Dependency Injection: Hilt, Dagger 2, IcePick, Butter Knife
 Suites: Firebase, JetPack, Fabric, Compose
 Frameworks: Reactive-X (RxJava, RxAndroid, RxCache)
 Integration and Threading: Volley, Retrofit, okhttp
 Media: ExoPlayer, Glide, Picasso
 Misc Android Tools: Schematic, SmartTV, Certificate Pinning, MonkeyRunner, Bluetooth Low Energy, SyncAdapters, CircleCI, VidEffects, Push Notifications, Kickflip, SpongyCastle, Parse, Flurry, Twitter, FloatingActionButton, Espresso, Fresco, Moshi, Jenkins, UI Automator, Parceler, Dependency Injection, EventBus, Dagger, Crashlytics, Mixpanel, Material Dialogs, Marshmallow, Loaders, Jetpack, JobScheduler, ParallaxPager, XmlPullParser, Google Cloud Messaging.

Work Experience

SENIOR ANDROID MOBILE APP DEVELOPER

Jun 2023-Present: Etsy, Brooklyn, NY

<https://play.google.com/store/apps/details?id=com.etsy.android> HYPERLINK

"app:%20https://play.google.com/store/apps/details?id=com.etsy.android&hl=en&gl=US"&

HYPERLINK

"app:%20https://play.google.com/store/apps/details?id=com.etsy.android&hl=en&gl=US"hl=en

HYPERLINK

"app:%20https://play.google.com/store/apps/details?id=com.etsy.android&hl=en&gl=US"&

HYPERLINK

"app:%20https://play.google.com/store/apps/details?id=com.etsy.android&hl=en&gl=US"gl=US

Summary: Part of the tech department working to bring up new and innovative ideas and concepts and trying to apply them to existing technology. Led the team to develop a new recommendation system within the e-commerce app to provide personalized product suggestions to users based on their browsing history, purchase behavior, and preferences.

- Introduced Jetpack compose Server Driven UI capabilities and migrated XML layouts
- Implemented Clean code architecture with MVI Reactive Architecture Pattern to perform Unidirectional Flow
- Optimized Jetpack Compose UIs for performance by utilizing techniques like memorization and state management, side effects and StateFlow
- Collaborated with designers to translate Figma mockups into Jetpack Compose-based user interfaces
- Wrote unit and integration tests for Jetpack Compose-based components to ensure reliability using Semantic API
- Perform incremental migration from RestFul to GraphQL with Apollo Client
- Integrated core Android Jetpack components for efficient development.
- Utilized Kotlin Multiplatform's shared code capabilities to build the core chatbot logic, natural language processing models, and conversational flows that could be reused across the Android and iOS codebases.
- Configured Okta's built-in multi-factor authentication for additional security.
- Implement robust authentication mechanisms on the login screen using Biometric API.
- Implement user data security through database authentication, user ID validation, and real-time updates to Google Fire store tables.
- Added WorkManager to project to periodically sync data with a server and send logs, analytics to backend services.
- Integrate Coroutines Flow API with Retrofit, okhttp, NavGraph, deep links, paging to handle streams.
- Combine multiple Flows for login, analytics, security, and navigation layers in MVI architecture.
- Use Firebase Realtime Database for data synchronization and Firebase Remote Config to update configuration.
- Pair program to modularize the existing customer payment authentication flow.
- Integrate an additional external SDK which helped authenticate customer's payment details.
- Worked closely with UI/UX designers to implement recommendation features seamlessly into the app's interface.
- Incorporated Splunk MINT SDK to collect crash, performance, usage data and send it to Cloud servers

- Developed features using Kotlin programming language for modern Android development.
- Integrated Hilt for dependency injection to manage app dependencies efficiently.
- Utilized Google Maps SDK for location-based features and services.
- Integrated Lottie for adding rich animations to the app's user interface.
- Integrated Stripe Payment SDK for secure payment processing within the app.

SR. ANDROID MOBILE APP DEVELOPER

Aug 2022-Jun 2023: TD Bank, Cherryhill, NJ

<https://play.google.com/store/apps/details?id=com.tdbank>

Summary: Enhanced Security Features by implementing advanced security measures such as biometric authentication (fingerprint, face recognition) to ensure secure access to the banking app. Integrated multi-factor authentication (MFA) methods for user login to strengthen account security. Led the development of encryption techniques to secure sensitive user data stored within the app and during transmission.

- Code on Uncle Bob clean code architecture using MVVM in a TDD environment, code specially on Data for network config and Presentation Layer
- Spearheaded the integration of biometric authentication methods such as fingerprint and face recognition to provide users with secure access to the app.
- Utilized iOS development tools and languages, such as Swift and Objective-C, to build and maintain the app
- Led the development of multi-factor authentication (MFA) methods to reinforce account security, ensuring that user logins require additional verification steps for authentication.
- Oversaw the implementation of encryption techniques to safeguard sensitive user data stored within the app and during transmission over networks, mitigating the risk of unauthorized access or data breaches.
- Leveraged a range of development tools and libraries, including Android Jetpack, Firebase, Google Maps SDK, Picasso, ReactiveX, and others, to implement security features effectively.
- Incorporated advanced security-related libraries such as RootBeer and Mitek MiSnap to enhance the app's resilience against security threats and fraud attempts.
- Implemented reactive programming paradigms using libraries like Kotlin Flow Coroutines API to streamline asynchronous operations and improve responsiveness in security-related functionalities.
- Conducted code reviews and performance optimizations to identify and address security vulnerabilities, memory leaks, and performance bottlenecks.
- Implemented multi-dex support and resource optimization strategies using AndroidX Multi-Dex Library to ensure smooth performance on devices with varying hardware capabilities.
- Worked closely with quality assurance (QA) engineers to define test cases and conduct thorough testing of security features, including unit tests, integration tests, and security penetration testing.
- Ensured seamless integration of security features with existing app functionalities, minimizing disruption to the user experience while enhancing security.
- Collaborated with backend developers to integrate server-side security protocols and validate data integrity across client-server communications.
- Utilized permission to run foreground services, enhancing user experiences by enabling essential background tasks to be performed without interruption
- Solely integrate code to include Airship Performance Analytics library for Segment Integration and Azure Event Hubs Integration Worked in DevOps environment with Agile methodologies, Daily stands, 2 weeks sprints

SR. FLUTTER CROSS PLATFORM APP DEVELOPER

Apr 2021- Aug 2022: MGM Resorts International, Las Vegas, Nevada

<https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts> [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US) & [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US) & [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US) & [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.mgmresorts.mgmresorts&hl=en&gl=US) & [HYPERLINK](#)

Summary: The MGM Rewards app is the official mobile app from MGM Resorts International, designed to make navigating their properties and managing reservations easier.

- Code Implement the BLoC (Business Logic Component) pattern with RxDart's Observables and Streams, along with the Provider package.
- Used RxDart to manage complex data transformations, error handling, and event-driven interactions
- Used RxDart's testing utilities, such as the MockObserver and TestScheduler, to simulate and verify the behavior of Observables and Streams.
- Implemented get_it package as a dependency injection framework for BLoC to register BLoC instances
- Applied DevTools to visualize the widget tree, identify layout issues, and diagnose UI performance problems.
- Utilised Sentry Flutter plugin to monitor and debug layout issues in production app.
- Used go_router package for management and maintenance of the application's navigation structure.
- Collaborate with designers to create and implement create modular and reusable Rive animations
- Included widgets like AppBar, Drawer, BottomNavigationBar, and a wide range of UI elements t
- Automate regression geolocation (testing from various geographical locations) and fixed flaky tests
- Worked on configuring Circle CI pipelines with Bitbucket as VCS and automate app publishing process, JIRA tool and Confluence for documentation.
- Utilized the http_certificate_pinning Flutter package to implement certificate pinning in the app. Define StatelessWidget or StatefulWidget classes that represent the different screens or views.
- For Material Design, use the MaterialPageRoute class, while for Cupertino, use the CupertinoPageRoute class.
- Implemented testing strategies, including screenshot testing with Flutter
- Customized the accessibility of app using the Semantics widget

MID LEVEL ANDROID APP DEVELOPER

Jan 2020-Mar 2021: MetLife Inc., New York City, New York

<https://play.google.com/store/apps/details?id=com.metlifeapps.metlifeus>

HYPERLINK

"<https://play.google.com/store/apps/details?id=com.metlifeapps.metlifeus&hl=en&gl=US>"&

HYPERLINK

"<https://play.google.com/store/apps/details?id=com.metlifeapps.metlifeus&hl=en&gl=US>"[hl=en](https://play.google.com/store/apps/details?id=com.metlifeapps.metlifeus&hl=en&gl=US)

HYPERLINK

"<https://play.google.com/store/apps/details?id=com.metlifeapps.metlifeus&hl=en&gl=US>"&

HYPERLINK

"<https://play.google.com/store/apps/details?id=com.metlifeapps.metlifeus&hl=en&gl=US>"[gl=US](https://play.google.com/store/apps/details?id=com.metlifeapps.metlifeus&hl=en&gl=US)

Summary: As a Sr. Android App Developer, I led the implementation of healthcare-specific features, such as electronic health record (EHR) integration, medication management, telemedicine capabilities, appointment scheduling, and health tracking functionalities. Our team also implemented robust security measures to protect sensitive patient data, including encryption, secure authentication methods, and adherence to healthcare data privacy regulations.

- Used Android Jetpack to architect the app in a modular and maintainable way, ensuring scalability and robustness.
- Applied Android Architecture Components, including LiveData, ViewModel, and Room, were utilized to implement the MVVM (Model-View-ViewModel) architecture pattern.
- Worked on an Agile team consisting of Android developers, software designers/ programmers, API developers, QS specialists, and business process specialists.
- Worked on an existing large codebase in MVVM architecture with data-binding implementation.
- Used Android Jetpack VersionedParcelable for passing data between different components of the app, such as activities and fragments, while maintaining compatibility across different versions of the app.
- Leveraged Android Jetpack Annotations, such as @NonNull and @Nullable, to annotate methods, parameters, and fields to improve code readability and maintainability.
- Used RxKotlin in conjunction with RxAndroid, and RxBinding libraries to make the app multithreaded and perform asynchronous operations.
- Implemented new features in Kotlin, eliminating null pointer exceptions.
- Used RecyclerView for displaying lists of data efficiently, and ConstraintLayout facilitated building complex layouts with a flat view hierarchy, improving performance and reducing nesting.
- Used AndroidX Activity to implement various lifecycle-aware components and handle activity lifecycle events effectively, ensuring proper management of resources and states throughout the app's lifecycle.
- Used Activity Saved State to preserve the UI state of activities, such as scroll positions, text inputs, and selected items, across device rotations and other configuration changes, providing a seamless user experience.

- Used Android Jetpack Media to implement features like audio/video recording, playback, and streaming within the healthcare app, enabling functionalities like telemedicine and patient education.
- Ensured the application's adherence to best practices in healthcare app development, security, and privacy regulations.
- Collaborated with cross-functional teams to meet project goals and timelines effectively.
- Provided technical guidance and mentorship to junior developers.
- Participated in code reviews, testing, and debugging processes to maintain code quality and stability.
- Contributed to the continuous improvement of development processes and methodologies.

ANDROID APP DEVELOPER

Nov 2017-Jan 2020: Liberty Mutual Insurance Company, Boston, MA

<https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual> [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US)
[&](#) [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US)
[hl=en](#) [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US)
[&](#) [HYPERLINK](#)
["https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US"](https://play.google.com/store/apps/details?id=com.lmig.pm.internet.mobile.android.libertymutual&hl=en&gl=US)
[gl=US](#)

Summary: Worked with the team to streamline the claims processing workflow. This feature can enable users to report insurance claims directly from their smartphones by capturing photos, videos, and relevant documents. It can also facilitate real-time communication with insurance agents, track claim status, and provide updates on claim settlements.

- Developed the app using Android SDK, coding in both Java and Kotlin using Android Studio.
- Transformed ideas into code with Java and Kotlin, implemented Kotlin coroutines, and used Retrofit for back-end integrations.
- Implemented an MVP UI pattern as the basic layer architecture of the app and transitioned dependency injection from Dagger 1 to Dagger 2.
- Used Singleton, Proxy, Visitor, Decorator, Facade, Builder, and Interpreter Design Patterns
- Managed code review sessions to always have high-quality (reusable, simple, documented) code, Unit testing in Junit, and automated testing with Robolectric.
- Displayed images using the Picasso library.
- Participated in all stages of software development including defining the product vision, gathering requirements, software system design, coding, testing, release, and support.
- Implemented RESTful call to receive JSON-based response and parsed with GSON to display data.
- Configured Git and Bitbucket servers as the software version control tools.
- Used GitLab for continuous integration and tested using tools and devices.
- Avoided memory leaks and ANR using Leak Canary and memory monitor tools and used Crashlytics for troubleshooting.
- Turned technical specs into Android multi-panel Activities created using Fragments for better form factor adaptability.
- Improved Android Keystore program to store cryptographic keys in a container and protect key material from unauthorized use
- Applied Data binding library to include observable data objects, customize generated binding classes and support two-way data binding
- Used RxJava 2 for most of our asynchronous processes to simplify threading.
- Cached data from the network and stored in OrmLite local database to display in graphical format using MPAndroidChart library.
- Included PDFViewer Android library for displaying PDF documents with animation and zoom effects.
- Created Mockito and Espresso unit test cases.

ANDROID DEVELOPER

Jan 2015-Oct 2017: Walgreens, Deerfield, IL

<https://play.google.com/store/apps/details?id=com.usablenet.mobile.walgreen>

Summary: Collaborated with QA engineers to develop and execute comprehensive testing strategies, including functional testing, performance testing, and security testing, to validate the reliability and accuracy of healthcare applications.

- Implemented native Android solutions that communicate with RESTful web services.
- Collaborated with QA teams.
- Implemented VOIP functionality using native Android and VoCal VoIP SDK which uses standard algorithms for VOIP processing in a modular design.
- Implemented functionality with Google Cloud Messaging and Android Push Notifications.
- Worked on secure authorization using Authorization and Authentication tokens to restrict access.
- Responsible for business logic based on discrete modules with design patterns that varied with functionality (singleton, builder, proxy, visitor,)
- Developed modular workflows in the Android app using Activities and Fragments.
- Implemented Dexguard to protect the Android app from reverse engineering and hacking.
- Used GitHub private repositories with standard workflows as a version control software.
- The Android team used an Agile development process with daily scrums and bi-weekly Sprints.
- Refactored code to reduce the size of the code base and create more efficient, reusable code.
- Supported the design, development, testing, and implementation of the application.
- Consumed back-end services for communications through RxJava and Retrofit.
- Worked in different phases of the project cycle including development and testing.
- Use of SQLite database schema to persistent data storage for every schedule.
- Used Dagger library to apply binding in Android views.
- Worked on menu items to have a smooth transition while creating respective activities and fragments.
- Implemented a collapsing toolbar in a drawer layout to collapse the image on scrolling up.

Education

Bachelor's in computer science from San Francisco State