ARCHANA PRABHU

prabhuarc27@gmail.com | (973) 626-8877 | California | LinkedIn | Github | Portfolio

Profile

- 7 years of hands-on professional experience as a full stack Software Engineer with proficiency in developing user facing mobile and web applications.
- Advanced proficiency with modern mobile development frameworks such as Jetpack Compose, Dagger / Hilt, Flows, Channels and Kotlin Coroutines.
- Skilled in designing and implementing responsive web components using semantic HTML, Javascript, ReactJS and NodeJS.
- **Play Store Mobile App:** Microsoft Teams Android, **Teams web application:** Microsoft Teams Web
- Expertise in application performance optimization for low battery, memory and bandwidth constraints.
- Experience in developing backend services using Typescript, NodeJS, Python, Apache Kafka, Azure Data Explorer, Geneva Azure Monitor and on Azure DevOps pipeline.
- Expertise in implementing prototypes of networking protocols like HTTP2 (okhttp) and QUIC (Cronet) for Android network performance analysis.
- Published "**Network Intrusion Detection using Sequence models**" paper in IEEE Xplore, 2019

Skills

Programming languages: Kotlin, Java, Javascript, Python

Android Libraries and frameworks: Retrofit, Volley, OkHttp, Dagger, Glide, Bolt, Butterknife, Databinding, RxJava2, Mockito, Firebase, MaterialUI, Compose, SQLite, Room, Gradle, Maven, Protocol Buffer, Moshi, GSON

Web front end technology: HTML, Javascript, ReactJS, Typescript, Vue.js

Backend technology: Apache Kafka, NodeJs (kafkajs), RabbitMQ, JWT

Database: MySQL, PostgresQL, Redis, SQLite

IDE: Android Studio, Visual Studio Code, Eclipse

Testing & Debugging tools: Espresso, Roboelectric, Monkey, AppCrawler, Junit, Mockito, Perfetto, SysTracing, Charles Proxy, Testcontainers (for Kafka consumer), Cypress, Jest

Experience

Microsoft Bangalore, India
Software Engineer 2 Sep. 2021– Nov. 2024

1. Search UI for Microsoft Teams Android (Kotlin, Java)

- Implemented a **TabLayout** combined with **ViewPager2** to enable smooth horizontal scrolling and dynamic loading of search contexts, such as chats, files, and channels.
- Designed the search interface with **ConstraintLayout** to create adaptive layouts that responded dynamically to screen sizes and orientations. Used **MotionLayout** for handling smooth transitions between expanded and collapsed states of the search bar, optimizing usability for phones and tablets.
- Built a **RecyclerView-based suggestion list** that dynamically populated real-time search results as the user typed.
- Implemented **DiffUtil** for efficient updates to the suggestions, minimizing unnecessary re-renders and improving performance on resource-constrained devices.
- Developed custom **CardView-based components** to display context-aware previews of search results, such as chat message snippets or file names with icons.

2. **Activity Feed search and filter support for Microsoft Teams Web (React, Typescript, GraphQL, Vue.js)**
   - Designed and implemented **reusable** React components for filtering and searching items based on user-defined parameters (read, un-read, types of notifications).
   - Developed state management solutions using React **hooks** like **useMemo**, **useCallback**, **useEffect** to prevent unnecessary re-renders.
   - Leveraged **Context API** to propagate data to handle filter states and prevent prop-drilling.
   - Integrated **GraphQL** queries to fetch and filter activity feed data efficiently.
   - Implemented **responsive** UI elements ensuring optimal performance across different devices and screen sizes.
   - Collaborated with UX/UI designers to ensure the filter and search functionalities were intuitive and aligned with Teams' design guidelines

3. **Simple Collab Chat experience for Microsoft Teams Android (Kotlin, Java)**
   - Developed a highly responsive and dynamic chat interface with features like **message threading**, **real-time updates**, and **quick file sharing**.
   - Enhanced performance by optimizing **RecyclerView** for seamless scrolling through chat history, ensuring smooth interactions even with large datasets.
   - Incorporated robust **state management** and lifecycle handling to deliver consistent user experiences during network fluctuations and app transitions.
   - Built interactive UI components such as **custom toolbars**, **floating action buttons**, and **action sheets** using Kotlin and Jetpack Compose for modern Android design.

4. **Priority framework for thread management and coroutine migration (Threading, Kotlin Coroutine)**
   - Redesigned the network layer to prioritize user-initiated actions over background tasks, ensuring real-time responsiveness for critical operations like sending messages or loading active screens.
   - Integrated an **exponential backoff strategy** for network retries, preventing redundant requests and reducing strain during unstable connectivity scenarios.
   - Developed a robust backpressure policy to manage incoming and outgoing network requests, leveraging **RxJava** or **Kotlin coroutines** to maintain system stability under high loads.
   - Leveraged **SharedFlow** for broadcasting network state changes (e.g., retries, success, failures) to multiple UI components in real-time, and **Channels** for handling one-to-one communication between the network layer and worker threads, ensuring streamlined request processing and user feedback loops.

5. **Message Delivery Latency (MDL) Service (Kafka, NodeJS)**
   - Implemented a service using **Node.js** and **Kafka** to track end-to-end message delivery latency across multiple devices and services.
   - Utilized **Kafka** for reliable and efficient handling of message events, ensuring scalability and fault tolerance in latency tracking pipelines.
   - Collected and pushed latency data to **Kusto clusters** for storage and analysis, enabling detailed reporting and identification of bottlenecks in the messaging workflow.
   - Ensured that the system seamlessly handled latency tracking across different device platforms, including Web, Android and iOS, without performance degradation.

6. **Glassjar Service for Microsoft Chat, Contact and Calendar service (Typescript, Express.js)**
   - Designed and implemented a robust Glassjar service to facilitate scenario execution by **mocking network calls, eliminating the need for live service dependencies**.
   - Supported multiple test account types (EDU, Enterprise, Consumer, Shifts) with tailored configurations for each.
   - Utilized a **seed input mechanism** to populate user data such as chats and channel information, ensuring fresh and isolated test environments for every execution.
   - Developed features for **dynamic API response generation,** simulating live service behavior, and supporting custom test scenarios like failure simulations and latency handling.
   - Integrated the service with CI pipelines to streamline automated testing workflows.

Microsoft Bangalore, India
Software Engineer June, 2019– August, 2021

1. Multi-Account Support for Microsoft Teams Web (React, Redux, Typescript, CSS, Javascript)

- Designed and implemented a **multi-account switcher UI**, allowing users to easily toggle between different accounts and tenants from a single interface, improving accessibility and usability.
- Utilized **Redux for global state management**, ensuring seamless transitions between multiple accounts without affecting the overall performance or data integrity.
- Created **context-aware UI components** that dynamically adjusted based on the active account or tenant, ensuring users only saw relevant data for their current context.
- Implemented custom **animations and transitions** to enhance the experience of switching between accounts and tenants, ensuring smooth and intuitive interactions.
- Worked closely with backend teams to implement **secure session management**, ensuring that each account or tenant context was properly isolated while still providing a unified user experience.

2. Azure DevOps CI pipeline test integration (Python, Linux)

- Mentored an intern, imparting best practices in Android UI development and led the integration of **Crawler Tests** into the CI/CD pipeline.
- Crawler / Monkey tests enabled early detection of performance bottlenecks, particularly in UI transitions in Android app, ensuring a seamless experience for users.
- Built a latency regression detection system that analyzed latency in UI scenarios during CI builds, with a threshold of 100ms delays. This ensured that any performance degradation in UI responsiveness was detected and addressed immediately.

3. Improvement of Microsoft Teams App Vitals (Kotlin, Java, C++)

- Investigated and resolved the most frequently occurring native crash (**SIGSEGV**) within the **Conscrypt Library**'s SSL layer, preventing app freezes during sensitive network operations and improving user trust in the app.
- Conceptualized and implemented a **Preheat Framework** to schedule non-critical tasks, leveraging **onPreDrawListener** in the **ViewTreeObserver**. This ensured that UI elements rendered with maximum speed and fluidity before loading additional background data. The change significantly reduced visible load times for users.
- Developed a deep understanding of **Android Runtime (ART)** optimizations and implemented **Baseline Profiles** for Android devices. The effort reduced app startup time by **27%** during updates and decreased **Janks** (frame rendering delays) on the chat screen by **16%**, improving app responsiveness during high-stress scenarios like app launches.
- Spearheaded the integration of **R8 optimizer** in full mode with obfuscation for the Microsoft Teams legacy app. This reduced the app size by **12 MB**, improved cold and warm startup times by **20%**, and ensured better memory management for devices with limited resources.
- Designed and built a **CachingRecyclerView** capable of handling multi-bind blocks with higher per-item caching. This innovation reduced **Janks** by **20%** and minimized view inflation time by **90%**, providing a buttery-smooth scrolling experience even on resource-intensive pages like chat lists and activity feeds.

4. Network call reliability and performance analysis (Nginx, Python, Java)

- Designed and developed an algorithm for devising network quality in real time using **exponential moving average algorithm.**
- Developed **dynamic UI elements** that adapted based on real-time network conditions. For example, implemented low-data mode banners, optimized file upload UIs, and adaptive image compression, ensuring a consistent user experience across varying connectivity levels.
- Enhanced HTTP reliability using **adaptive socket timeouts in OkHttp**, reducing timeouts and improving responsiveness for high-priority user actions like messaging and file uploads.

Microsoft Bangalore, India
Software Engineering Intern April, 2018–July, 2018

1. Enhanced Camera UI for Microsoft Teams (Kotlin, Java)

- Integrated **Office Lens SDK** capabilities, including **text conversion**, **edge detection**, and **image processing algorithms**, into the Microsoft Teams camera interface. This allowed users to extract information from documents, whiteboards, and other surfaces directly within the app, significantly broadening the camera's functionality.
- Designed and implemented **real-time preview overlays** to provide instant feedback to users while scanning documents or detecting edges. These overlays visually highlighted detected edges, providing an intuitive and engaging experience that minimized user errors.
- Developed advanced gesture-based controls for the camera UI, including **pinch-to-zoom**, **tap-to-focus**, and **drag-to-adjust** functionalities. These interactions made the feature more intuitive and increased accessibility for non-technical users.
- **User Engagement and Adoption:** Collaborated with UX designers to ensure a visually appealing and user-centric interface. The enhancements drove a **25% increase in feature usage**, showcasing the positive impact of an intuitive and robust camera experience.

2. Voice Activated commands with Cortana (Kotlin, Java)

- Integrated the **Cortana SDK** to enable voice-activated commands for navigating Microsoft Teams activities. This allowed users to execute commands such as opening chat windows, scheduling meetings, and searching for content using natural language, improving accessibility and convenience.
- Designed and implemented an intuitive **voice command interface**, featuring **real-time suggestions** and prompts to guide users through available voice actions. This reduced the cognitive load on users, especially during complex workflows.
- Developed robust mechanisms to **handle unrecognized commands** gracefully, providing real-time feedback and alternative suggestions. This ensured a seamless experience and minimized user frustration.
- Focused on enhancing productivity by integrating Cortana commands with critical Teams features such as messaging, calls, and calendar management. These **voice-driven workflows** reduced interaction complexity and improved accessibility, particularly for users with limited mobility.

Tripler Bangalore, India
Android Developer May, 2017–Jan, 2018

1. Feature-Rich Timeline UI

- Designed and developed a dynamic, Instagram-like **timeline UI** for a media-centric mobile app, where users could easily scroll through and interact with posts, photos, and videos.
- Implemented **smooth transitional animations** that triggered as users navigated between posts, providing a seamless visual experience. The UI was optimized for high performance, ensuring fast load times even with heavy media content.
- Integrated **gesture-driven interactions**, such as pinch-to-zoom for images and swipe actions to like, comment, or share posts, enhancing the app's usability and user engagement.

2. Integrated Google Maps with Custom Markers

- Enhanced the app's interactivity by **integrating Google Maps SDK** to display location-based media.
- Developed custom-designed markers and overlays to pinpoint and highlight media content, allowing users to easily explore media tagged with specific geographic locations.
- Incorporated **dynamic clustering** of nearby markers to improve performance and ensure an intuitive browsing experience when zooming in or out of the map.
- The integration provided users with a richer, more immersive experience, enabling them to interact with location-tagged content seamlessly.

3. **Delightful Animations and Micro-Interactions:**
   - Focused on delivering a polished and engaging user experience by incorporating **motion-driven animations** and **gesture recognition** for intuitive navigation.
   - Implemented a variety of animations, such as **swipe-to-dismiss** for media posts, **zoomable media previews**, and **touch-based contextual menus** that appeared with a subtle delay, creating a smooth and responsive interface.
   - These micro-interactions helped refine the overall user experience, making navigation and content interaction effortless while providing users with delightful feedback through every action.

Projects

1. **DiceDB (Go)**
   - Developed **DiceDB**, a Redis-inspired database designed for real-time applications, emphasizing scalability and high availability.
   - Built using **GoLang**, it supports Redis-compatible APIs, ensuring seamless integration with existing systems. DiceDB employs a horizontally scalable caching mechanism capable of managing and querying large datasets efficiently across distributed environments.

2. **Chores Android App (Kotlin, Jetpack Compose)**
   - Created an Android app using **Kotlin** and **Jetpack Compose** to help families stay organized by tracking groceries, assigning tasks, and managing schedules.
   - The app features a **ScrollView calendar** for a clear visual representation of family members' schedules and integrates geolocation tracking for real-time member updates.
   - Designed with a user-friendly interface, it promotes better collaboration and task allocation within households, leveraging modern Android development practices for a seamless experience. Link

3. **Network Intrusion Detection (Machine Learning, Python)**
   - Implemented a machine learning-based intrusion detection system to analyze and identify network traffic anomalies using the **NSL-KDD** benchmark dataset.
   - This project compared the performance of models like **Nonsymmetric Deep Autoencoders (NDAE)**, **Recurrent Neural Networks (RNNs)**, **Gated Recurrent Units (GRUs)**, and **Long Short-Term Memory (LSTM)** networks.
   - The system provided insights into the effectiveness of various approaches, helping improve detection rates for unauthorized access or malicious activities in network environments.

4. **Surveillance Video Anomaly Detector (Machine Learning, Python)** - Designed a real-time system for detecting anomalies in surveillance video feeds using machine learning techniques. Leveraged **C3D**, a modified 3D convolutional neural network based on the **Caffe framework**, to train, test, and fine-tune models for identifying unusual activities. Link

5. **Image Captioning (Machine Learning, Python)** - Developed a system to automatically generate descriptive captions for images using deep learning. Utilized pretrained models like **VGG16**, **ResNet50**, and **InceptionV3** combined with **Python**, **Keras**, and **TensorFlow** to extract visual features and map them to meaningful text. Link

Education

Bangalore University, University Visvesvaraya College of Engineering Bangalore, India
Bachelor of engineering in Computer Science, GPA Jul. 2019

- Coursework: Operating Systems, Algorithms, Discrete Mathematics, Machine Learning, Distributed Systems
- Honors: K.V Kumaraswamy Gold Medal