Felix Tien

Phone: 205-482-3933 | Email: felix.tien.takumu@gmail.com

Portfolio: https://www.felixtien.com/

SUMMARY

- With over 7 years of expertise, proficiently crafted mobile (iOS and Android) applications utilizing **Swift**, **Java**, and **Kotlin** programming languages.
- Knowledge of **OOP** principles including encapsulation, inheritance, and polymorphism.
- Proficient in leveraging various **Property Wrappers**, including **State Property Wrapper**, **Environment Object Property Wrapper**, etc, to enhance application architecture and data management.
- Demonstrated history of delivering intricate iOS applications, harnessing the power of **UIKit**, **Core Data**, and **Core Animation**.
- Profound familiarity with **Agile** methodologies, serving as a proficient **Scrum** Master, and directing comprehensive project lifecycles.
- Proficient in leveraging various **Jetpack** libraries, including **Data Binding Library**, **View Binding**, etc, to enhance application architecture and data management.
- Extensively experienced in crafting Android UI through adept utilization of **XML** layouts and **Jetpack Compose**, ensuring optimal design and functionality.
- Proficient in implementing both **MVC** and **MVVM** frameworks, with a proven track record of incorporating them into Android apps.
- Skilled in employing **Data Binding** and **Observable State Objects** to update UI elements efficiently.
- Demonstrated proficiency in manipulating **Firebase** service, including **Friestore Database**, **Firebase Authentication**, **Firebase Cloud Message**, etc.
- Proficient in utilizing **Postman** to collaborate with backend team API functionality and facilitate seamless data transfer between frontend and backend.
- Proficient in **REST** principles, successfully implementing **RESTful** APIs in Android using **Retrofit** and **URLSession** in iOS for backend **CRUD** operations and communications.
- Expertise in implementing **Google Material** design components within both Android **XML** and **Jetpack Compose** frameworks.
- Adept in generating color schemes via **Theme Builder** and implementing these color themes into Android projects.
- Proficient in version control systems and adeptly collaborating with colleagues through **GitHub**.
- Skilled in leveraging **JUnit** and **Mockito** for **Java** and **Kotlin** logic testing, adept in meticulous UI testing with **Espresso** for diverse Android apps

SKILLS

Languages	Swift, Java, Kotlin, XML
Client-side Tech Stack	UIKit, SwiftUI, CocoaPods, URLSession, Jetpack Compose, XML, Gradle, Retrofit, MVC/MVVM architecture
Database	SQLite (Core Data, Room), Cloud Firestore
Testing	XCTest, XCUITest, Postman, JUnit, Mockito, Espresso
Tools	XCode, Android Studio, IntelliJ, GitHub, VSCode, Jira

EXPERIENCE

Walmart Business Oct 2024 - Present
Android Developer Sunnyvale, CA

DCARD is a social media platform for college students to explore social networks. This app allowed each person to make one friend by getting a profile card every midnight. Our team was responsible for working REST network requests with the backend team and displaying DCARD posts efficiently.

Google Play Store link: [Walmart Business App - Apps on Google Play]

Responsibilities:

- Validating Functionality and Developing Documentation for Internal App Flags.
- Partnering with various Walmart teams to develop and enhance app features.
- Utilizing Android Layout Inspector to target specific files for debugging and usage analysis.
- Leveraging App Inspection to analyze network calls and JSON responses.
- Using the JIRA Agile Board to collaborate effectively with team members.
- Mocking network responses using Flipper for testing purposes.

Famrent Apr 2024 – Sep 2024
Android & iOS Developer Binghamton, NY

Famrent operates across both Android and iOS platforms, utilizing Jetpack Compose for Android and SwiftUI for iOS to streamline rental management processes efficiently. It provides landlords with a dashboard to track tenant payment history and rental income, while tenants can report issues and rate their landlords. The project was developed entirely from the ground up using SwiftUI, ensuring a modern, efficient, and scalable design.

Google Play Store link: Famrent App - Apps on Google Play

App Store link: Famrent App on the App Store

Responsibilities:

- Implemented **Google Auth**, **Apple Auth**, **Email Auth**, and **Phone Auth**, enabling seamless account creation and secure authentication for Famrent users.
- Leveraged **Firestore**, a NoSQL cloud database, as the backend to efficiently store and manage user data, notifications, and more.
- Utilized **Cloud Storage** to allow tenants to upload images of housing issues, providing landlords with visual context to address and resolve problems efficiently.
- Enabled **FCM v1 API** and employed **Push Notifications** in SwiftUI **AppDelegate** to deliver real-time updates in the Famrent app.
- Developed **Cloud Functions** using TypeScript to implement the backend logic for the Famrent project on the Firebase server.
- Applied **AdMob** API for banner, interstitial, and rewarded interstitial ads into the Famrent project, and deployed the **app-ads.txt** file to the Portfolio domain.
- Created multilingual support in the Android app by defining translation keys for four languages in a **JSON** file, allowing users to select their preferred language.
- Leveraged **Retrofit** and **URLSession** for **RESTful** API interactions and utilized **Dispatchers.IO** and **DispatchQueue** to handle remote data retrieval on the main thread asynchronously.
- Controlled asynchronous functions using **async/await** and utilized **@MainActor** to enhance the efficiency of Firebase SDK integration.
- Managed version control for development by effectively utilizing **GitHub** and deploying relevant data to designated **Repositories**.
- Prepared the Famrent privacy policy and produced app preview screenshots for the published Famrent app on the **Google Play Console** and **App Store Connect**.
- Set up a membership subscription in **Google Play Console** and **App Store Connect**, integrate it with **RevenueCat**, and configure API keys to link the subscription service to **Compose** and **SwiftUI**.

JLPT N1-N5 Level Oct 2023 – Mar 2024
Android & iOS Developer Birmingham, AL

The JLPT N1-N5 mobile applications are tailored for students preparing for the Japanese Language Proficiency Test. These apps are divided into four distinct versions corresponding to the proficiency levels and are available on both Android and iOS platforms. The Android versions utilize XML-based architecture, integrating Compose View to leverage Jetpack Compose, while the iOS versions are predominantly developed using SwiftUI.

JLPT N1 Level Google Play Store link: JLPT N1 Level App on the App Store

JLPT N1 Level App Store link: JLPT N1 Level App on the App Store

JLPT N2 Level Google Play Store link: JLPT N2 Level App on the App Store

JLPT N2 Level App Store link: JLPT N2 Level App on the App Store

JLPT N3 Level Google Play Store link: JLPT N3 Level App on the App Store

JLPT N3 Level App Store link: JLPT N3 Level App on the App Store

JLPT N4-N5 Level Google Play Store link: JLPT N4-N5 Level App on the App Store

JLPT N4-N5 Level App Store link: JLPT N4-N5 Level App on the App Store

Responsibilities:

- Used **XML** and **SwiftUI** to design Android and iOS UI and UX and published the app project to **Google Play Console** and **App Store Connect**.
- Utilized **Retrofit** and **URLRequest** to make RESTFull API requests and employed **@GET** and **@Path** in Android and **JSONDecoder()** in iOS to parse the **JSON** file.
- Implemented the **Room Database** and **Core Data**, SQLite libraries, for efficient local data storage in Android and iOS applications.
- Proficient in implementing the **MVVM** architectural pattern to design and develop robust and scalable software solutions.
- Leveraged **SharedPreferences** and **User Default** for managing small data storage and user preferences in Android and iOS applications.
- Integrated **Jetpack Compose View** within XML enables seamless interaction between XML-based projects and composable functions.
- Enhanced the **makeUIView** and **updateUIView** methods to define and update View Objects from **UIKit** within the **SwiftUI** project.

DCARD Dec 2017 – Aug 2020
iOS Developer Taipei, Taiwan

DCARD is a social media platform for college students to explore social networks. This app allowed each person to make one friend by getting a profile card every midnight. Our team was

responsible for working REST network requests with the backend team and displaying DCARD posts efficiently.

App Store link: Dcard App on the App Store

Responsibilities:

- Designed and administered user interfaces for iOS applications using **UIKit** and **Swift**, ensuring smooth and engaging user experiences.
- Leveraged the **MVC** pattern to optimize state management, thereby improving system stability and performance.
- Implemented iOS Pub-Sub library **NotificationCenter** to dynamically update the user interface as users interact with the application professionally.
- Employed responsive user interfaces for iOS applications using **Auto Layout**, a key component of the **UIKit** framework.
- Utilized **NSLayoutConstraint** and **NSLayoutAnchor** proficiently to meticulously define the dimensions and positioning of UI elements.
- Advanced proficiency in implementing efficient and dynamic user interfaces using **CollectionView** in Android app development.

Additional tools employed in the project:
XCTest for unit testing, **XCUITest** for UI testing, **GitHub** for version control, **Jira** for project management, and **URLSession** for network requests.

StreetVoice Aug 2015 – Sep 2017
Junior iOS Developer Taipei, Taiwan

StreetVoice is a leading music library where creators upload their works, and users can bookmark music and subscribe to creators. Our team was tasked with the implementation of the music playback functionality, and integrating data storage capabilities with Could Firestore for the music database.

App Store link: StreetVoice App on the App Store

Responsibilities:

- Experienced in developing responsive user interfaces using **UIKit** layout library in conjunction with **Swift** code for iOS applications.
- Utilized **TableView**'s smooth scrolling and cell-reusing capabilities to ensure optimal performance, even with large music playlists.
- Leveraged the **MVC** architectural pattern to oversee state variables, facilitating precise and efficient real-time updates of user interactions.
- Employed the **AVAudioSession** and **AVAudioPlayer** components to facilitate background music playback within the application.

- Implemented the **Core Data** SQLite database to enable users to bookmark music selections professionally.
- Utilized **UIStackView** to adeptly organize user interface elements either horizontally or vertically, optimizing layout efficiency within iOS applications.

Additional tools employed in the project:
XCTest for unit testing, **XCUITest** for UI testing, **GitHub** for version control, **Jira** for project management, and **Could Firestore** for remote database.

Bahamut Game Community Jul 2013 – Jun 2015
Junior Android Developer Taipei, Taiwan

Bahamut Game Community provides a comprehensive platform for gamers to access news, create content, connect with friends, and communicate across various gaming genres from different people. In this project, our team stored and retrieved data from Could Firestore and implemented pagination to display huge gaming news to users.

Google Play Store link: Bahamut Game Community App - Apps on Google Play

Responsibilities:

- Proficient in crafting adaptive user interfaces utilizing **XML** layout files alongside **Java** programming for Android applications.
- Skilled in applying **MVC** principles to iOS projects, implementing clear separation of concerns between the model, view, and controller components.
- Incorporated **SharePreference** to store user preferences such as device language, preferred gaming genre, etc, in key-value pairs.
- Implemented queries to filter **Collections** and stored **DocumentSnapshots** for each page from **Cloud Firestore** to implement **Pagination**.
- Utilized **GitHub** for collaborative efforts with colleagues, facilitating seamless version control and efficient cooperation in project development.
- Employed **Jira** project management, issue tracking, user stories, and sprints within **Agile** frameworks.

Additional tools employed in the project:
JUnit for unit testing, **Mockito** for integration testing, and **Espresso** for UI testing.