

# Development Workflow

- **Plan:**
  - Define workout data structure (strokes, sets, metrics).
  - Mock up UI in Figma (workout builder, history view).
- **Build MVP:**
  - Core features: Create/save workouts, view history, Apple Watch tracking.
  - Use SwiftUI and Core Data for offline storage, Firebase for cloud sync.
- **Test:**
  - Unit tests for workout logic (XCTest for Swift).
  - UI tests for usability (XCUITest).
  - Test on iPhone and Apple Watch simulators/devices.
- **Deploy:**
  - Use Xcode to publish to the App Store.
  - Automate with Fastlane for faster releases.
- **Monitor:**
  - Use Firebase Analytics for usage tracking and Sentry for crash reporting.

# Security

- **Authentication:** Firebase Auth for secure user login.
- **API Security:** Use HTTPS and validate inputs to prevent injection attacks.
- **Planned API's:**
- **API 1:** <https://thirdparty-api-documentation.swimsmember.org/#getting-started>
- **API 2:** [https://www.formswim.com/blogs/all/introducing-script-your-instant-workout-creator?srsId=AfmBOopv6GFO9jdfp1chKD-koY4kCC8DAIT0xw\\_4Vs3CcgBIEJX2ZH7k](https://www.formswim.com/blogs/all/introducing-script-your-instant-workout-creator?srsId=AfmBOopv6GFO9jdfp1chKD-koY4kCC8DAIT0xw_4Vs3CcgBIEJX2ZH7k)
- **OR**
- **API 2:** <https://help.validic.com/space/VCS/1681260696/Swimtag+API+Integration+for+Developers>
- **HealthKit:** Request user permission for HealthKit data access and comply with Apple's privacy rules.

# Example Workflow for User

- User opens the iOS app and creates a workout ("4x100 freestyle @ 1:30").
- User opens the iOS app and or saves a workout from a Masters Coach ("4x100 freestyle @ 1:30").
- Workout saves to Core Data (offline) and syncs to Firebase.
- Or user Selects from Premade "Workouts"
- Apple Watch tracks real-time metrics (laps, heart rate) during the swim.
- **Future**
- Web console (or future Android app) displays the workout and metrics, editable in real-time.
- Data syncs across all platforms via Firebase.

## Recommended Tech Stack

- **iOS App:** Swift, SwiftUI, Core Data, WatchKit, HealthKit.
- **Backend:** Firebase (Firestore, Auth, Cloud Messaging).
- **Future Android/Web:** Flutter for unified codebase, Firebase Hosting for web.
- **Tools:** Xcode, Figma, GitHub, Fastlane, Sentry.

## Learning Resources

- **Swift/SwiftUI:** Apple's SwiftUI tutorials ([developer.apple.com](https://developer.apple.com))
- **Firebase:** Firebase iOS SDK docs ([firebase.google.com](https://firebase.google.com)).
- **WatchKit/HealthKit:** Apple Developer docs for WatchKit and HealthKit integration.
- **Design:** Figma's swim-specific templates or Material Design for inspiration.

## 11. Next Steps

- **Prototype:** Build a SwiftUI app with a simple workout builder and Firebase sync.
- **Test:** Focus on iOS and Apple Watch usability with real swimmers.
- **Expand:** Add Android/web later using Flutter, reusing the Firebase backend.

Possible Names:

**SwimForge** - Suggests building and crafting personalized workouts.

**AquaPulse** - Captures the energy and rhythm of swimming.

**StrokeSync** - Emphasizes syncing workouts across devices and platforms.

**PoolPlan** - Simple and direct, focusing on workout planning for the pool.

**WaveMaker** - Conveys creating impact and progress in the water.

**LapLab** - A nod to experimenting and tracking laps in a structured way.

**SwimCraft** - Highlights the art of crafting swim routines.

**AquaFlow** - Evokes smooth, flowing swim sessions.

**TideTrack** - Suggests tracking progress with the rhythm of tides.

**SplashSet** - Fun and energetic, focusing on workout sets.