

Building Cross-Platform MAUI Apps with Bluetooth, GPS, and Device Sensors

Duration: 5 days

Prerequisites: Knowledge of Mobile Application Development

Day 1 – Advanced MAUI Structure & Layouts

Objectives:

- Understand MAUI's single-project architecture and platform-specific customization
- Implement advanced navigation patterns including Shell and deep linking
- Design adaptive and performance-optimized layouts

Topics:

1. **MAUI Project Architecture**
 - Single project structure
 - Platform-specific code integration (Android/iOS/Windows)
 - Conditional compilation for platform logic
2. **MAUI Pages & Navigation**
 - Shell navigation and modal navigation
 - Deep linking and navigation parameters
 - Managing navigation state
3. **Advanced Layout Patterns**
 - Complex Grid layouts
 - FlexLayout for adaptive design
 - Performance tips for complex layouts

Labs & Hands-On:

- Create a Shell-based app with nested navigation
 - Build a responsive multi-column dashboard using FlexLayout/Grid
-

Day 2 – Controls, MVVM Enhancements & Styling

Objectives:

- Implement advanced MAUI control patterns for dynamic UI rendering
- Apply MVVM best practices with dependency injection and async commands
- Create dynamic themes and style resources for cross-platform consistency

Topics:

1. **Advanced Control Usage**
 - Custom controls with ControlTemplate
 - DataTemplateSelectors for dynamic UI rendering
 - Advanced event-to-command binding
2. **MVVM Best Practices**
 - Dependency injection in MAUI MVVM
 - ViewModel communication patterns (MessagingCenter, WeakReferenceMessenger)

- Async commands with cancellation support
3. **Advanced Styling & Theming**
 - Merged resource dictionaries for app-wide theming
 - Dynamic theme switching (light/dark)
 - Performance considerations in styles

Labs & Hands-On:

- Implement a dynamic dashboard with DataTemplateSelector
 - Add light/dark mode switching with live theme change
-

Day 3 – Data Presentation & API Integration

Objectives:

- Build high-performance list-based UIs with advanced CollectionView features
- Integrate APIs with caching, pagination, and offline support
- Implement real-time data updates using SignalR and push notifications

Topics:

1. **CollectionView Mastery**
 - Virtualization & lazy loading for large datasets
 - Grouped & hierarchical data presentation
 - Interactive items (swipe gestures, context menus)
2. **API Integration & Data Management**
 - Optimized HttpClient usage
 - API response caching & offline sync patterns
 - Handling API pagination & search filtering
3. **Real-Time Data in MAUI**
 - Using SignalR for live updates
 - Push notifications integration (Firebase)

Labs & Hands-On:

- Build a grouped CollectionView with swipe-enabled items
 - Implement a paginated API client with caching and offline mode
-

Day 4 – Local Storage & Hardware Integration

Objectives:

- Manage local storage using SQLite with encryption and transactions
- Implement Bluetooth LE device communication for sensor data exchange
- Integrate GPS and device sensors for location-based functionality

Topics:

1. **SQLite Advanced Usage**
 - ORM patterns with SQLite-net

- Transactions & bulk operations
 - Data encryption for security
2. **Bluetooth LE Integration**
 - Advanced BLE features: GATT profiles, characteristics
 - Handling multiple device connections
 - Streaming and parsing BLE data
 3. **GPS & Sensor Integration**
 - High-accuracy GPS data retrieval
 - Background location tracking & geofencing
 - Integrating accelerometer and gyroscope

Labs & Hands-On:

- Store encrypted data locally in SQLite
 - Connect to a BLE device and stream sensor data
 - Implement background GPS tracking with geofencing alerts
-

Day 5 – Toolkit, Security, Optimization & Deployment

Objectives:

- Use CommunityToolkit.MVVM for faster and cleaner MVVM development
- Secure apps with SSL pinning and secure storage
- Optimize app performance and automate deployment workflows

Topics:

1. **CommunityToolkit.MVVM Advanced**
 - Source generators for commands & properties
 - AsyncRelayCommand with exception handling
 - ValueConverters & Behaviors for complex UI logic
2. **App Security & Performance**
 - Secure storage of sensitive data
 - SSL pinning for secure API communication
 - Profiling and memory leak detection
3. **Optimized Deployment**
 - App trimming & linking for reduced size
 - Platform-specific build optimizations
 - Continuous deployment with App Center / GitHub Actions

Labs & Hands-On:

- Refactor app using CommunityToolkit advanced features
- Implement SSL pinning for API security
- Automate build and deployment to a test device