

AI-Driven Mobile Engineering: Flutter, LLMs and Intelligent Features Training Course

28 December 2026 – 1 January 2027
Istanbul

AGILE LEADERS
Training Center

AI-Driven Mobile Engineering: Flutter, LLMs and Intelligent Features Training Course

Ref: 2257_264950 **Date:** 28 December 2026 – 1 January 2027 **Location:** Istanbul **Fees:** 4500 Euro

Overview

The AI-Driven Mobile Engineering: Flutter, LLMs and Intelligent Features Training Course is a one-week technical course for software engineers and mobile architects seeking to deploy generative models and machine learning pipelines into cross-platform applications. Modern mobile development demands responsive interfaces alongside capabilities like offline semantic search, structured generative output, and on-device inference without degrading battery performance or runtime latency. Participants analyze client-side inference engines, streaming client-server architectures, and local vector retrieval. Agile Leaders Training Center provides this AI-driven mobile engineering course to establish robust architectural patterns for enterprise mobile environments.

Who Should Attend

- Engineers responsible for cross-platform user interface architecture, reactive state management, and asynchronous network orchestration.
- Mobile software developers tasked with integrating cloud-hosted large language model APIs and streaming event responses.
- Software architects designing local data persistence, edge inference pipelines, and mobile security boundaries.
- Technical leads evaluating on-device machine learning frameworks and model optimization for mobile runtimes.

The course assumes participants already build production mobile applications using Dart and Flutter, leaving out fundamental object-oriented programming syntax, basic widget composition, and mobile design theory.

Departments and Industries

The course provides architectural frameworks and implementation patterns applicable across multiple technology-driven operational environments.

- Mobile application engineering departments in retail banking and financial technology
- Digital patient engagement and telehealth development teams in healthcare
- Supply chain and field operations technology units in commercial logistics
- E-commerce engineering and consumer software product divisions
- Enterprise workforce enablement and internal software delivery groups

Learning Objectives

By the end of this course, participants will be able to:

- Build reactive Flutter interfaces that stream structured responses from large language model endpoints.
- Apply on-device inference runtimes using LiteRT and ONNX Runtime Mobile for offline predictive capabilities.
- Integrate client-side retrieval-augmented generation patterns using SQLite-based vector extensions and embedding models.
- Evaluate model quantization profiles and execution delegates to optimize mobile hardware utilization.
- Diagnose security exposures across mobile local storage, API credential handshakes, and user prompt validation boundaries.

AGILE LEADERS
Training Center

Course Agenda

Day 1: Mobile AI Foundations and Streaming Model Integration

- Dart isolates and background asynchronous concurrency for compute tasks
- OpenAI and Anthropic streaming client implementations in Flutter
- Structured JSON schema enforcement and prompt template orchestration
- Token stream parsing using the BLoC state management pattern
- Rate limiting, network retry policies, and backoff mitigation with Dio interceptors

Day 2: On-Device Machine Learning and Runtime Acceleration

- LiteRT model runtime bindings using Dart Foreign Function Interface
- ONNX Runtime Mobile execution delegates across Android NNAPI and iOS Core ML
- INT8 and FP16 quantization evaluation for on-device parameter constraints
- Background sensor and image preprocessing pipelines using the Flutter Camera plugin
- Memory allocation profiling and thermal throttling mitigation using Flutter DevTools

Day 3: Client-Side Vector Storage and Retrieval-Augmented Generation

- Local text embedding generation using quantized mobile embedding models
- Vector similarity search implementation using sqlite-vec in SQLite databases
- Chunking strategies and metadata filtering for mobile document indexing
- Hybrid retrieval pipelines combining BM25 keyword matching with dense embeddings
- Context window management and prompt payload compaction for mobile bandwidth limits

Day 4: Mobile Security, Privacy Boundaries and Performance Optimization

- Zero-trust API credential management using Flutter Secure Storage and Hardware Keystores
- Prompt injection filtering and client-side PII redacting algorithms
- Battery drain and network radio profiling using Android Studio Profiler and Xcode Instruments
- Offline-first cache synchronization architectures using Hive and SQLite
- Guardrail verification workflows for AI output validation and error fallbacks

Day 5: Applied Engineering Practice and Capstone Integration

- Architectural refactoring exercise: migrating cloud inference workflows to local on-device runtimes
- Performance tuning exercise: eliminating frame drops and interface jank in streaming interfaces
- Security audit exercise: remediating prompt injection risks and credential leaks in mobile codebases
- Production release readiness review using structured mobile deployment checklists
- Capstone exercise: building an intelligent mobile assistant with local vector search and streaming generation

Practical Exercises

Participants complete hands-on technical labs to reinforce implementation concepts across typical mobile architectures.

- Suggested activity: Implement a streaming conversational interface using Flutter BLoC, Server-Sent Events, and JSON validation.
- Suggested activity: Deploy a quantized text classification model on-device using LiteRT and measure inference latency against target thresholds.
- Suggested activity: Configure a local sqlite-vec database to perform semantic retrieval over simulated field service documentation.
- Suggested activity: Conduct a dynamic thermal and memory profiling audit during continuous local model inference sessions.

FAQs

Who should attend the AI-Driven Mobile Engineering: Flutter, LLMs and Intelligent Features Training Course, and what background is required?

Mobile engineers, software architects, and technical team leads who maintain production Flutter applications will benefit most from this course. Participants must possess hands-on proficiency in Dart programming and asynchronous state management. The course does not cover beginner mobile UI development, foundational Dart syntax, or backend infrastructure administration.

How does this AI-driven mobile engineering course differ from a general mobile app development course?

General mobile development courses focus on building graphical layouts, native platform integrations, and REST API connectivity. In contrast, this course concentrates on integrating on-device machine learning runtimes, streaming large language model integrations, edge vector databases, and managing hardware performance constraints unique to artificial intelligence workloads.

Why should mobile developers integrate local LLMs and on-device inference instead of relying solely on cloud APIs?

On-device inference eliminates cloud latency, reduces network bandwidth consumption, lowers API subscription costs, and guarantees data privacy by keeping sensitive user inputs on the physical device. Local runtimes also enable critical offline functionality in operational field environments where cellular connectivity is intermittent or unavailable.

What tools and frameworks are covered in the AI-driven mobile engineering course?

The course utilizes Flutter and Dart alongside on-device runtimes including LiteRT and ONNX Runtime Mobile. Participants work with local database tools such as sqlite-vec and Hive, cloud generative APIs, Flutter DevTools, Android Studio Profiler, and Xcode Instruments to monitor performance and resource consumption.

Conclusion

Participants complete the course with an enterprise-ready intelligent mobile architecture blueprint tailored to production Flutter applications. This artifact provides concrete reference implementations for streaming generative models, on-device inference, and offline vector retrieval across mobile platforms. Engineering teams use this blueprint to accelerate feature delivery, reduce cloud infrastructure dependency, and maintain strict data privacy standards across user devices.