D79 LMS - Application Architecture

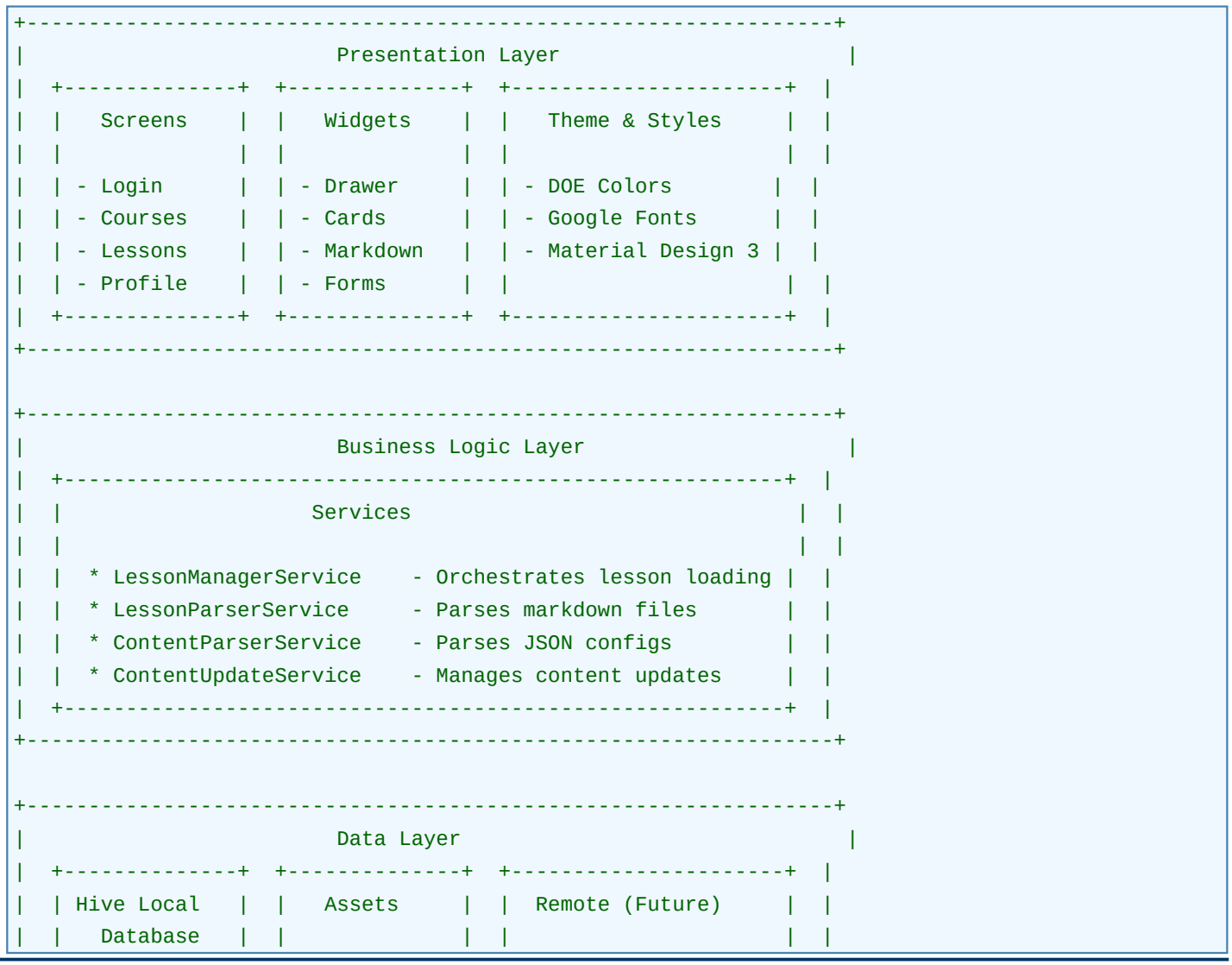
Technical Documentation

Overview

D79 LMS is a Flutter-based Learning Management System designed for GED/HSE education. The application follows a **layered architecture** pattern with offline-first capabilities, focusing on accessibility, modern UI/UX, and interactive learning experiences.

Table of Contents

Architecture Diagram



```
|  |              |  | - Markdown   |  | - Content Updates    |  |
|  | - Courses    |  | - JSON       |  | - User Sync          |  |
|  | - Modules    |  | - Images     |  | - Progress Tracking  |  |
|  | - Lessons    |  | - Videos     |  |                      |  |
|  | - Progress   |  | - Translations| |                      |  |
|  +--------------+  +--------------+  +----------------------+  |
+-------------------------------------------------------------------+
```

Core Principles

1. **Offline-First Architecture**
- All course content stored locally using Hive database
- Lessons and materials accessible without internet connection
- Periodic sync for content updates when online

2. **Modular Design**
- Feature-based folder structure
- Separation of concerns (UI, Business Logic, Data)
- Reusable components and services

3. **Scalability**
- Support for multiple content types (Markdown, Interactive Forms, Videos)
- Extensible lesson format system
- Easy addition of new GED subjects

4. **Accessibility**
- DOE-compliant color scheme (Blue #003366, Gold #FFD700)
- Text scaling support
- Bilingual support (English/Spanish)
- Clear visual hierarchy

Layer Structure

Presentation Layer

Located in: `lib/features/` and `lib/core/screens/`, `lib/core/widgets/`

Responsibilities:
- Display UI components
- Handle user interactions
- Navigate between screens
- Apply theming and styling

Key Components:

```
features/
+-- auth/
|   +-- screens/
|       +-- login_screen.dart          # Authentication UI
+-- courses/
|   +-- screens/
|   |   +-- course_list_screen.dart    # Browse all courses
|   |   +-- course_details_screen.dart # Course overview
|   |   +-- module_details_screen.dart # Module content
|   |   +-- lessons_list_screen.dart   # Lesson listing
|   |   +-- lesson_viewer_screen.dart  # Markdown lessons
|   |   +-- enhanced_lesson_viewer_screen.dart # Interactive lessons
|   +-- widgets/
|       +-- fillable_content_viewer.dart # Form widgets
+-- profile/
|   +-- screens/
|       +-- profile_screen.dart         # User profile & progress
+-- about/
|   +-- screens/
|       +-- about_screen.dart           # About & portfolio
+-- privacy/
    +-- screens/
        +-- privacy_screen.dart         # Privacy policy
```

Business Logic Layer

Located in: `lib/core/services/`

Responsibilities:
- Orchestrate data operations
- Parse and transform content
- Manage application state
- Handle content updates

Key Services:

1. **LessonManagerService**

```
dart
Purpose: Central orchestrator for lesson loading
- Loads both markdown and structured lessons
- Determines lesson type (markdown vs. interactive)
- Manages lesson metadata extraction
```

```
- Coordinates between different parsers
```

2. **LessonParserService**

```
dart
Purpose: Parse markdown lesson files
- Load markdown files from assets
- Extract metadata (title, objectives, reading time)
- Convert to Lesson objects
- Handle 34 math lesson files
```

3. **ContentParserService**

```
dart
Purpose: Parse structured JSON lesson configs
- Load JSON configuration files
- Extract fillable fields
- Parse interactive elements
- Support multiple field types
```

4. **ContentUpdateService**

```
dart
Purpose: Manage weekly content updates (Future)
- Download content ZIP files
- Extract and install updates
- Preserve user progress
- Handle offline caching
```

Data Layer

Located in: `lib/core/models/` and Local Storage

Responsibilities:
- Define data structures
- Persist data locally
- Manage assets
- Handle remote sync (future)

Data Models:

Core Models (Hive-enabled)

```
dart
@HiveType(typeId: 0)
class Course {
  - id, title, description
  - modules, instructor
```

```
  - icon, color (DOE theme)
  - duration, level, category
}

@HiveType(typeId: 1)
class Module {
  - id, title, description
  - courseId, order
  - contents, assignments
}

@HiveType(typeId: 5)
class Lesson {
  - id, title
  - content (markdown)
  - learningObjectives
  - estimatedReadingTime
  - isCompleted
}

@HiveType(typeId: 7)
class LessonContent {
  - id, lessonId
  - type (ContentType enum)
  - content, metadata
  - fillableFields
  - order, isCompleted
}

@HiveType(typeId: 8)
class FillableField {
  - id, label
  - fieldType (FieldType enum)
  - placeholder, validationRule
  - studentAnswer, correctAnswer
  - isRequired, isCorrect
}

@HiveType(typeId: 2)
class User {
  - id, username, email
  - fullName, role
  - enrolledCourses
}
```

Data Flow

Lesson Loading Flow

```
1. User Opens Lessons Screen

2. LessonsListScreen calls LessonManagerService.loadAllLessons()

3. LessonManagerService orchestrates:
   +--> _loadMarkdownLessons() -> LessonParserService
   |    +--> Reads AssetManifest.json
   |    +--> Finds all .md files in directory
   |    +--> Loads each file content
   |    +--> Extracts metadata (title, objectives)
   |    +--> Returns List<Lesson>
   |
   +--> _loadStructuredLessons()
        +--> Finds lesson_X_config.json files
        +--> Parses JSON with ContentParserService
        +--> Creates LessonContent objects
        +--> Returns List<Lesson>

4. Combine and sort all lessons

5. Display in ListView with progress tracking

6. User taps lesson -> Navigate to LessonViewerScreen

7. Display lesson.content (already loaded)

8. User marks complete -> Save to Hive
```

User Progress Flow

```
User Actions -> Local Hive Storage -> Progress Calculation -> UI Update

- Complete lesson
- Submit assignment
- Take quiz
- View content

Stored in Hive boxes:
- lessons (completion status)
- assignments (submissions)
- user (overall progress)

Progress calculated in real-time

Displayed on:
- Profile screen
- Course cards
- Module headers
```

Key Features

1. **Multi-Format Lesson Support**

Markdown Lessons
- 34 HSE Math lessons stored as `.md` files
- Rich text formatting with `flutter_markdown`
- Embedded images and tables
- Estimated reading time calculation

Interactive Lessons
- JSON-configured structured content
- Fillable forms and text inputs
- Multiple choice questions
- True/False exercises
- Progress tracking per field

2. **Offline-First Content**

Current Implementation:
- All content bundled with app
- Hive local database for user data
- No network required for learning

Future Enhancement (ContentUpdateService):

```
Weekly Update Flow:
1. Check for updates on app launch
2. Download content ZIP from server
3. Extract to local storage
4. Merge with existing content
5. Preserve user progress
6. Notify user of new content
```

3. **Bilingual Support**

```
assets/translations/
+-- en.json    # English translations
+-- es.json    # Spanish translations

Using easy_localization package:
- Runtime language switching
- Formatted strings with parameters
- Accessibility support
```

4. **Progress Tracking**

```
dart
User Progress Metrics:
- Lessons completed / Total lessons
- Module completion percentage
- Course progress visualization
- Time spent on content
- Assignment scores

Visualization:
- Linear progress bars
- Circular progress indicators
- Color-coded status (DOE colors)
- Achievement badges (future)
```

Technology Stack

Frontend Framework
- **Flutter 3.32.0** - Cross-platform UI framework
- **Dart 3.8.0** - Programming language
- **Material Design 3** - Design system

State Management
- **StatefulWidget** - Local component state
- **Provider** - App-wide state (ready for expansion)
- **Hive** - Reactive local database

Local Storage
- **Hive 2.2.3** - NoSQL local database
- **SharedPreferences** - Simple key-value storage
- **PathProvider** - File system access

Content Rendering
- **flutter_markdown 0.6.23** - Markdown rendering
- **flutter_svg 2.0.9** - Vector graphics
- **video_player 2.8.2** - Video playback
- **chewie 1.7.5** - Video player UI

UI Enhancement
- **google_fonts 6.1.0** - Custom typography (Poppins)
- **url_launcher 6.2.4** - External links
- **easy_localization 3.0.7** - Internationalization

Code Generation
- **build_runner 2.4.15** - Code generation runner
- **hive_generator 2.0.1** - Hive adapter generation

Future Additions
- **http 1.1.0** - Network requests (for content updates)
- **archive 3.3.0** - ZIP extraction (for content updates)

Module Breakdown

Core Modules

`lib/core/`
Central shared functionality

```
core/
+-- models/          # Data models (Hive-enabled)
+-- services/        # Business logic services
+-- screens/         # Shared screens (splash, main)
+-- widgets/         # Reusable components (drawer, cards)
+-- theme/           # App theming (DOE colors, fonts)
```

`lib/features/`
Feature-based organization

```
features/
+-- auth/            # Authentication & login
+-- courses/         # Course browsing & lessons
+-- profile/         # User profile & progress
+-- settings/        # App settings
+-- help/            # Help & support
+-- about/           # About & portfolio
+-- privacy/         # Privacy policy
```

Content Organization

```
assets/
+-- lessons/
|   +-- math_lesson1/
|       +-- HSE MATH_ Q.1 Apply number sense concepts.md
|       +-- HSE MATH_ Q.1 Apply number sense concepts (1).md
|       +-- ... (34 total files)
|       +-- lesson_1_config.json    # Interactive lesson config
|       +-- lesson_2_config.json
+-- translations/
|   +-- en.json
```

```
|   +-- es.json
+-- images/          # Course thumbnails, icons
+-- videos/          # Educational videos
+-- documents/       # PDF resources
```

GED Course Structure

The app is designed around the **4 GED/HSE Subject Areas**:

```
1. Mathematical Reasoning
   +-- Module 1: HSE Mathematical Reasoning (34 lessons)
   |   +-- Apply number sense concepts
   |   +-- Measurement and geometry
   |   +-- Data and statistics
   +-- Module 2: Algebra and Geometry

2. Reasoning Through Language Arts
   +-- Module 3: Reading Comprehension
   +-- Module 4: Writing & Grammar

3. Social Studies
   +-- Module 5: Civics & Government
   +-- Module 6: History & Economics

4. Science
   +-- Module 7: Life & Physical Science
   +-- Module 8: Earth & Space Science
```

Each course follows the same pattern:

- **Modules** -> **Lessons** -> **Assignments**
- Progress tracking at all levels
- Mixed content types (reading, video, interactive)

Design Patterns

1. **Service Locator Pattern**

Services are accessed statically for simplicity:

```
dart
LessonManagerService.loadAllLessons()
ContentParserService.parseLessonConfig()
```

2. **Repository Pattern** (Implicit)

Hive boxes act as repositories:

```
dart
final coursesBox = Hive.box<Course>('courses');
final lessonsBox = Hive.box<Lesson>('lessons');
```

3. **Factory Pattern**

Model creation from different sources:

```
dart
Lesson.fromMarkdownPath(...)
LessonContent.fromJson(...)
```

4. **Strategy Pattern**

Different lesson types handled differently:

```
dart
enum LessonType { markdown, structured }
```

Deployment Architecture

Current: Standalone Mobile App

```
User Device
+-- Flutter App (APK/IPA)
+-- Bundled Content
+-- Local Hive Database
```

Future: Hybrid Architecture

```
+-----------------+
|  Mobile App     |
|  - Core Content |
|  - Hive DB      |
+--------+--------+
         |
          (Weekly Sync)
+-----------------+
|  Content Server |
|  - New Lessons  |
|  - Updates      |
|  - Media Files  |
+-----------------+
```

Deployment Targets:

- Android (Google Play)
- iOS (App Store - future)
- Vercel (Web version - future)

Security & Privacy

Data Privacy

- All user data stored locally on device
- No data transmission to external servers (currently)

- FERPA and NYC DOE compliant
- Clear privacy policy included in app

Content Protection
- Lessons bundled in app assets
- No external API keys required
- Offline-first prevents data leaks

Authentication
- Simple local authentication
- Prepare for OAuth 2.0 (future)
- Role-based access control ready

Performance Considerations

Optimization Strategies

1. **Lazy Loading**
   - Lessons loaded on-demand
   - Images cached automatically by Flutter
   - Videos streamed, not preloaded

2. **Efficient Rendering**
   - ListView.builder for long lists
   - Markdown parsed once, cached
   - Minimal widget rebuilds

3. **Database Optimization**
   - Hive is extremely fast (NoSQL)
   - Indexed access by keys
   - Lazy box opening

4. **Asset Management**
   - Compressed images (WebP when possible)
   - Optimized markdown files
   - Minimal JSON configs

Testing Strategy

Planned Test Coverage

```
tests/
```

```
+-- unit/
|   +-- models/           # Model tests
|   +-- services/         # Service logic tests
|   +-- utils/            # Utility function tests
+-- widget/
|   +-- screens/          # Screen widget tests
|   +-- components/       # Component widget tests
+-- integration/
    +-- flows/            # End-to-end user flows
```

Future Enhancements

Phase 1: Content Expansion (Q1 2025)
- [ ] Complete all 4 GED subject areas
- [ ] Add video lessons
- [ ] Include practice quizzes
- [ ] PDF study guides

Phase 2: Enhanced Interactivity (Q2 2025)
- [ ] Drag-and-drop exercises
- [ ] Audio lessons
- [ ] Flashcard system
- [ ] Progress gamification

Phase 3: Cloud Integration (Q3 2025)
- [ ] User authentication (OAuth)
- [ ] Cloud progress sync
- [ ] Teacher dashboard
- [ ] Analytics & reporting

Phase 4: Social Features (Q4 2025)
- [ ] Discussion forums
- [ ] Peer study groups
- [ ] Live tutoring integration
- [ ] Achievement sharing

Development Guidelines

Code Organization
- Follow Flutter style guide
- Use feature-first folder structure
- Keep screens under 500 lines

- Extract reusable widgets

Naming Conventions
- Files: `snake_case.dart`
- Classes: `PascalCase`
- Variables: `camelCase`
- Constants: `SCREAMING_SNAKE_CASE`

Documentation
- Document all services with `///` comments
- README for each major feature
- Architecture diagrams in `/docs`

Contact & Support

Developer: Javier Jaramillo
Purpose: Portfolio & Family Education Project
Compliance: NYC DOE, FERPA
License: Private/Educational Use

Acknowledgments

This application is built with love for:
- **Cris** (Wife)
- **Sofia** (Daughter)
- **Mateo** (Son)

Showcasing professional experience in:
- Mobile Development (Flutter/Dart)
- Education Technology
- Offline-First Architecture
- Accessibility & Compliance

Last Updated: October 16, 2025
Version: 1.0.0
Flutter Version: 3.32.0