

1 Executive Summary

Project Name: SmartNotes – AI-Powered Study Assistant

Objective: Develop an intelligent note-taking Android application using Google Gemini AI to enhance active learning through automated summarization and flashcard generation.

Target Users:

- Students
- Researchers
- Professionals
- Lifelong learners

2 Problem Statement

2.1 Current Challenges

- Information overload from lengthy notes
- Manual summarization is time-consuming
- Passive learning methods
- Disorganized notes across platforms
- Inefficient study sessions

2.2 Market Gap

Existing apps lack AI-driven study features. SmartNotes bridges this gap by combining structured note management with intelligent AI processing.

3 Proposed Solution

3.1 Note Management System

- CRUD operations
- Category-based organization
- Real-time search
- Rich text formatting
- Automatic timestamps

3.2 AI-Powered Features

Smart Summarization

- Condenses notes (500+ words)
- Preserves key concepts
- Multiple summary lengths

Flashcard Generation

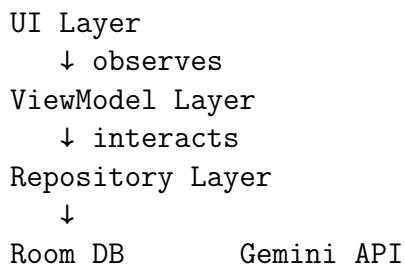
- Automatic Q&A extraction
- Multiple formats
- Spaced repetition hints

3.3 User Experience

- Material Design 3
- Smooth animations
- Offline-first approach
- Dark/Light themes

4 Technical Architecture

Architecture Pattern: MVVM



4.1 Technology Stack

Frontend:

- Kotlin 1.9+
- Android Jetpack
- Material Design 3
- View Binding

Backend/Data Layer:

- Room Database
- Retrofit
- Coroutines + Flow
- LiveData

5 Database Schema

5.1 Note Entity

```
@Entity(tableName = "notes")
data class Note(
    @PrimaryKey(autoGenerate = true)
    val id: Int = 0,
    val title: String,
    val content: String,
    val category: String,
    val createdAt: Long,
    val updatedAt: Long,
    val summary: String?,
    val isFavorite: Boolean
)
```

5.2 Flashcard Entity

```
@Entity(tableName = "flashcards")
data class Flashcard(
    @PrimaryKey(autoGenerate = true)
    val id: Int = 0,
    val noteId: Int,
    val question: String,
    val answer: String,
    val createdAt: Long
)
```

6 API Integration

Gemini AI Endpoint:

POST <https://generativelanguage.googleapis.com/v1beta/models/gemini-pro:generate>

Key Features:

- Rate limiting
- Retry logic

- Local caching
- Graceful error handling

7 User Interface Design

7.1 Screen Flow

Splash → Home → Add/Edit Note
→ AI Summary
→ Flashcards

8 Key Features & Development Phases

Phase	Features
Phase 1	CRUD, Room DB, MVVM, Search
Phase 2	Gemini API, Summaries, Flashcards
Phase 3	Optimization, Offline mode, Testing

9 Success Metrics

Technical:

- 60fps UI
- ≤ 2 s AI response
- $\geq 99.5\%$ crash-free

User Experience:

- 30–40% study time reduction
- $\geq 85\%$ flashcard relevance

10 Security & Privacy

- Encrypted local storage
- Secure API key handling
- No third-party analytics
- GDPR-compliant deletion

11 Testing Strategy

- Unit Tests (70%)
- Integration Tests (20%)
- UI Tests (10%)

12 Future Enhancements

- Cloud Sync
- Voice Notes
- Study Analytics
- Collaboration
- Multi-language support

13 Budget & Resources

Estimated Cost: \$0 (Free tier services)

14 Project Timeline

- Week 1–3: Core Features
- Week 4–5: AI Integration
- Week 6–8: Optimization & Testing

15 Learning Outcomes

- Advanced Kotlin
- MVVM Architecture
- API Integration
- AI-powered apps

16 Deliverables

- Source Code
- APK File
- Documentation
- Test Reports

17 Conclusion

SmartNotes demonstrates modern Android development with AI integration, clean architecture, and user-centric design. It is a production-ready, portfolio-grade project.