

Study Notes & Quiz Android Application

Name: Musayab Raza

Reg No: 23-ARID-579

1. Feature List / Functional Requirements

1. User can create, edit, and delete study notes
2. Notes are saved locally using Room database
3. User can create quizzes from notes
4. Track quiz progress and scores
5. Display quiz history
6. Save UI state during screen rotation
7. Restore data after app is killed by OS
8. Background/foreground state preservation
9. Low-memory process death recovery
10. Simple dashboard for notes and quizzes

2. Feasibility Study (Resources & APIs)

Requirement	Android Resource / API / Library
Notes CRUD	Room Database, SQLite
Quiz Creation	RecyclerView, ViewModel
Progress Tracking	SharedPreferences
UI Layout	XML Layouts, ConstraintLayout
State Preservation	ViewModel, onSaveInstanceState
Background Handling	LifecycleObserver
Process Death Recovery	SavedStateHandle
Navigation	Jetpack Navigation Component
Async Operations	Coroutines
Persistence	Room + SharedPreferences

3. Technical Developer Guide (Kotlin + XML)

Selected Features:

- Create & Save Notes
- Quiz Progress Tracking
- State Preservation

Feature 1: Create & Save Notes

Use Room database to store notes. Define Entity, DAO, and Database class. Use ViewModel to interact with UI.

```
@Entity  
data class Note(  
    @PrimaryKey(autoGenerate = true) val id:Int,  
    val title:String,  
    val content:String  
)
```

Feature 2: Quiz Progress Tracking

Store quiz score and completion status using SharedPreferences for quick access.

```
val prefs = getSharedPreferences("quiz", MODE_PRIVATE)  
prefs.edit().putInt("score", score).apply()
```

Feature 3: State Preservation

Use ViewModel and onSaveInstanceState to preserve UI data during rotation and SavedStateHandle for process death recovery.

```
override fun onSaveInstanceState(outState: Bundle) {  
    outState.putString("draft", text)  
    super.onSaveInstanceState(outState)  
}
```

4. Implementation Summary

The mini project is implemented using Kotlin and XML. Notes are stored using Room database, quiz scores using Shared Preferences, and lifecycle-safe state handling is achieved using ViewModel and SavedStateHandle. The project is pushed to GitHub with proper README and screenshots.

Submission Checklist:

- GitHub repository link
- Screenshots (Code, Emulator Output, Repo)
- README file
- This PDF documentation