



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the core implementation details for **EduNotes**, a React-based note-taking app for students. EduNotes enables users to create, edit, and organize notes with support for media attachments and tagging.

1. System Components

Component	Description	Key Responsibilities
NoteList	Displays list of notes	Fetch, filter, and render notes
NoteEditor	Create/edit note interface	Input, validate, and save notes
NoteView	Detailed view of a single note	Render note content and media
TagManager	Manage tags for notes	Add, remove, and filter by tags
MediaUploader	Attach media to notes	Upload and preview media files
App (Root)	Main app container and routing	State management, navigation

2. Class/Interface Overview

Class/Component	Relationships	Key Methods/Props
NoteList	Uses NoteView , TagManager	notes , onSelectNote , filterTags
NoteEditor	Uses MediaUploader	note , onSave , onCancel
NoteView	Receives note prop	note , onEdit , onDelete
TagManager	Used by NoteList , Editor	tags , onAddTag , onRemoveTag
MediaUploader	Used by NoteEditor	mediaFiles , onUpload , onRemove

Example: NoteEditor Props

```
<NoteEditor
  note={selectedNote}
  onSave={handleSave}
  onCancel={handleCancel}
/>
```

3. Data Structure Overview



Model	Fields
Note	id , title , content , tags[] , media[] , createdAt , updatedAt
Tag	id , name
Media	id , url , type (image, pdf, etc.), noteId

Note Example:

```
{
  "id": "n1",
  "title": "Biology Notes",
  "content": "Cell structure...",
  "tags": ["biology", "exam"],
  "media": [{ "id": "m1", "url": "...", "type": "image" }],
  "createdAt": "2024-06-01T10:00:00Z",
  "updatedAt": "2024-06-01T10:05:00Z"
}
```

4. Algorithms/Logic

Note Creation/Editing Flow (Pseudocode):

```
onSaveNote(noteData):
  if noteData.title is empty:
    showError("Title required")
    return
  if editing:
    update existing note in state
  else:
    add new note to state
  close editor
```

Tag Filtering:

```
onTagSelect(tag):
  filteredNotes = notes.filter(note => tag in note.tags)
  display(filteredNotes)
```

5. Error Handling

Scenario	Handling Approach
Missing note title/content	Show validation error message
Media upload failure	Show upload error, allow retry



Tag name duplicate	Prevent add, show warning
Note not found (view/edit)	Show “Note not found” message
Network/storage failure	Show error, retry or fallback to local

End of Document