



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the core frontend architecture for **Streamly - Entertainment Streaming UI**. Streamly is a modern, responsive React-based interface for browsing, searching, and viewing entertainment media, utilizing advanced CSS and Tailwind for a seamless, futuristic user experience.

1. System Components

Component	Description / Responsibility
App	Root component, routing, global state provider
Header	Branding, navigation, search input
MediaGrid	Displays media items in a responsive grid
MediaCard	Shows individual media item details
MediaDetailModal	Modal for detailed media info and playback
SearchBar	Handles user search input and triggers search
Loader/Error	Shows loading spinners or error messages

2. Class/Interface Overview

Component/Class	Key Props/State	Relationships
App	-	Parent of all components
Header	onSearch(query)	Contains SearchBar
MediaGrid	mediaList, onSelect(mediaId)	Contains MediaCard(s)
MediaCard	media (object)	Child of MediaGrid
MediaDetailModal	media, onClose()	Shown by App/MediaGrid
SearchBar	value, onChange, onSubmit	Used in Header

Key Methods/Attributes Example:

```
// MediaGrid
props: {
  mediaList: MediaItem[],
  onSelect: (mediaId: string) => void
}

// MediaCard
```



```
props: {  
  media: MediaItem,  
  onClick: () => void  
}
```

3. Data Structure Overview

Model	Fields
MediaItem	id: string, title: string, imageUrl: string, description: string,
	genre: string[], year: number, rating: number, videoUrl: string
SearchState	query: string, results: MediaItem[], loading: boolean, error: string?

4. Algorithms / Logic

Media Search Flow (Pseudocode):

```
onSearch(query):  
  set loading = true  
  fetch(`/api/media?search=${query}`)  
    .then(response => set results)  
    .catch(error => set error)  
    .finally(() => set loading = false)
```

Responsive Grid Layout:

- Use Tailwind CSS grid utilities (`grid-cols-1 md:grid-cols-3 lg:grid-cols-5`)
- Adjusts columns based on screen size

5. Error Handling

Scenario	Handling Approach
API fetch failure	Show error message in Loader/Error component
No search results	Display "No results found" in MediaGrid
Image/video load error	Show fallback image or message in MediaCard/Modal
Invalid user input	Prevent search, show input validation message

End of Document