Project Title
Media Catalog

Programming In Java
(CSE2006)

VIT®
BHOPAL
www.vitbhopal.ac.in

Name : Adhikari Viswa Teja
Reg n.o : 24MIM10138
Submitted to : Dr. Kamlesh Chandravanshi
Slot :A11+A12+A13+D11+D12

1. Introduction

The Media Catalog is a Java-based console application developed to manage a user's collection of movies and TV shows. It provides features such as adding new media entries, tracking watched status, monitoring show progress (season and episode), maintaining a watchlist, updating entries, and saving/loading information in CSV files.

The project demonstrates core Java concepts such as object-oriented programming (OOP), inheritance, polymorphism, file handling, and modular coding structure.

2. Problem Statement

Users often struggle to organize the movies and TV shows they watch. They may forget which ones they watched, the progress in long shows, or which movies they planned to watch later.
A simple digital system is needed to store movies and shows, track progress, and maintain a watchlist.

3. Objective

The objective of this project is to design and implement a structured Java-based application that enables users to efficiently manage their personal collection of movies and TV shows. The system aims to provide a simple and intuitive console interface where users can add, update, search, and remove media entries while also tracking watch status, watching progress, and maintaining a dedicated watchlist. Through the use of object-oriented programming principles, modular architecture, and CSV-based data persistence, the project seeks to demonstrate practical application of core Java concepts while delivering a functional, user-friendly media management solution.

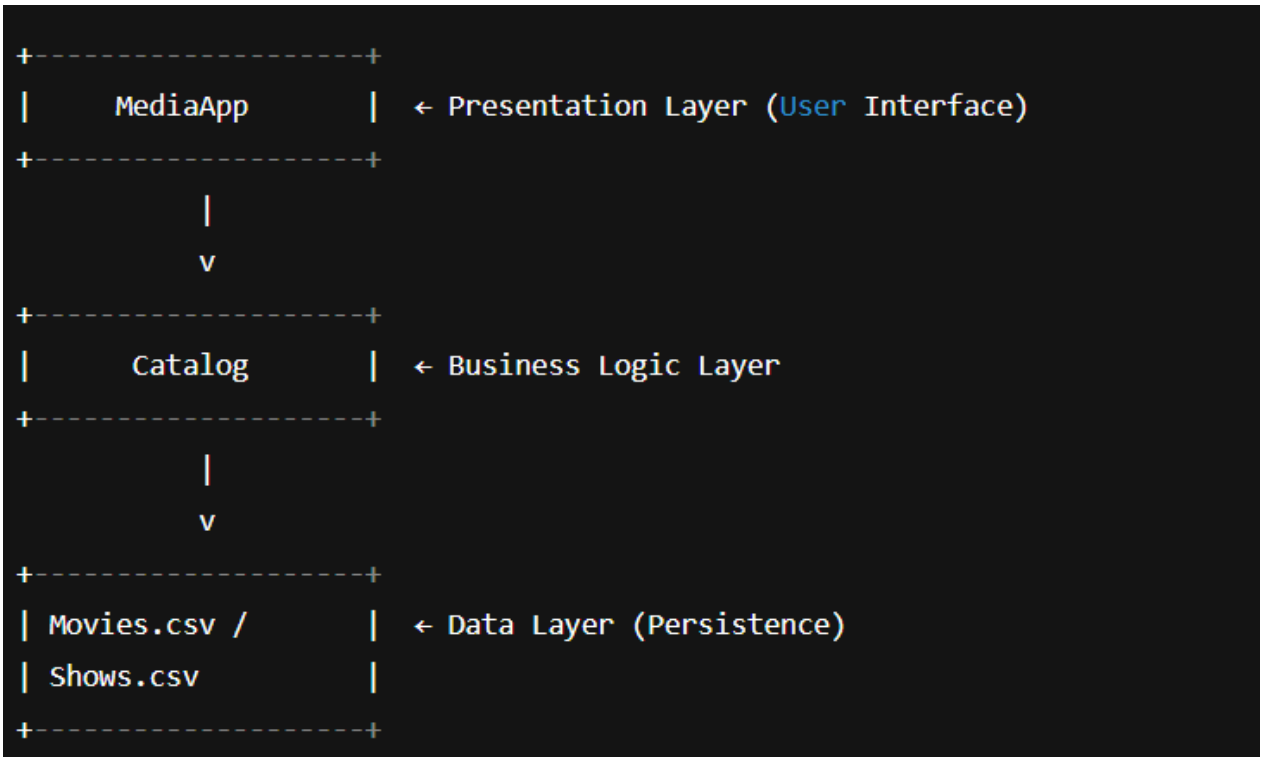
4. Functional Requirements

- **Add Movie :**
  Enter ID, title, year, genre, duration, Mark as watched OR add to watchlist
- **Add Show :**
  Enter ID, title, year, genre, seasons, episodes/season, Enter progress (season, episode) OR add to watchlist
- **List All Media :**
  Display all movies and shows stored in the system
- **Search Media :**
  Search by title or genre, Search applies to both movies and shows
- **Remove Media :**
  Delete a movie or show using its ID
- **Update Media :**
  Update year, genre, duration, seasons, episodes, Update watched/watching progress, Add/remove from watchlist
- **View Watchlist :**
  Display shows and movies marked “in watchlist”
- **Save & Exit :**
  Save all data to CSV files (movies.csv and shows.csv)

5. Non-Functional Requirements

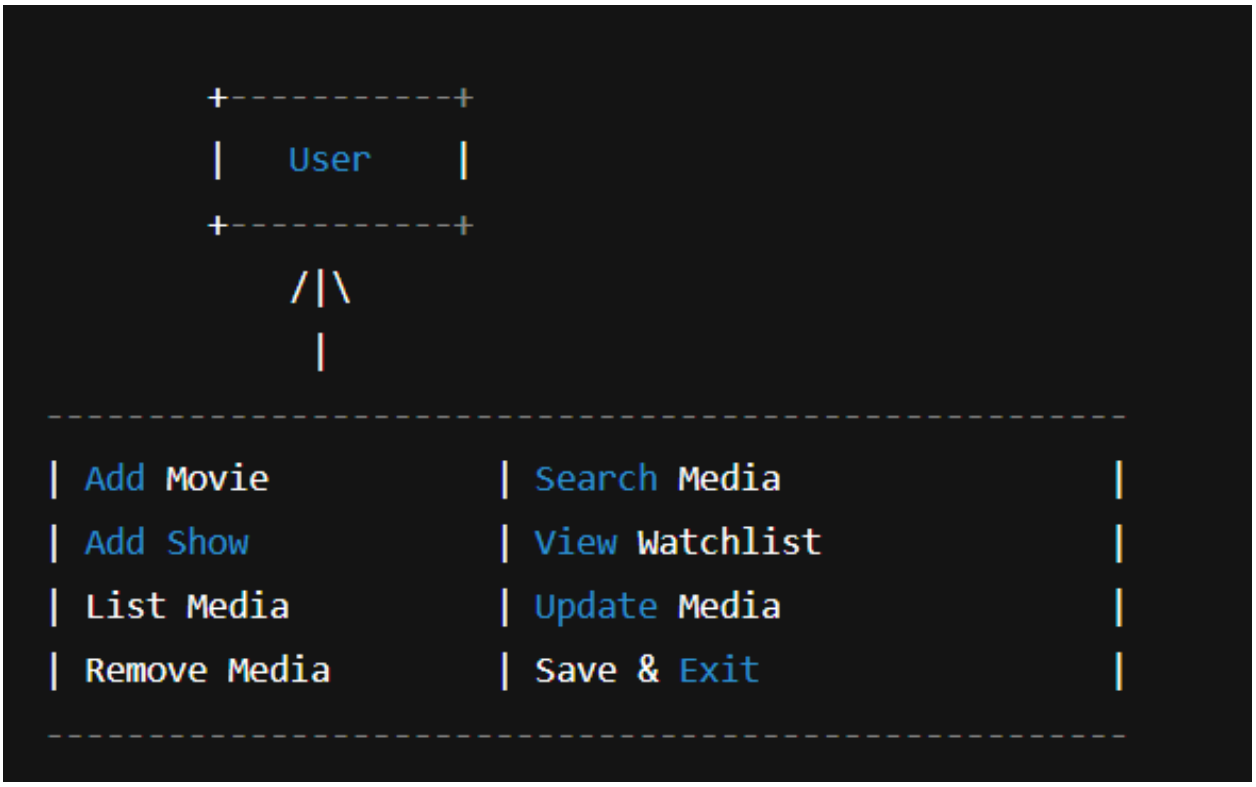
- **Usability :**
  Simple menu-based console interface for easy navigation.
- **Maintainability :**
  Code divided into packages (models, services, app) for modularity.
- **Reliability :**
  Input validation and error handling ensure stable operation.
- **Performance :**
  Uses lightweight ArrayList collections; operations execute instantly
- **Data Persistence:**
  CSV files allow data to be saved and restored across runs.

6. System Architecture

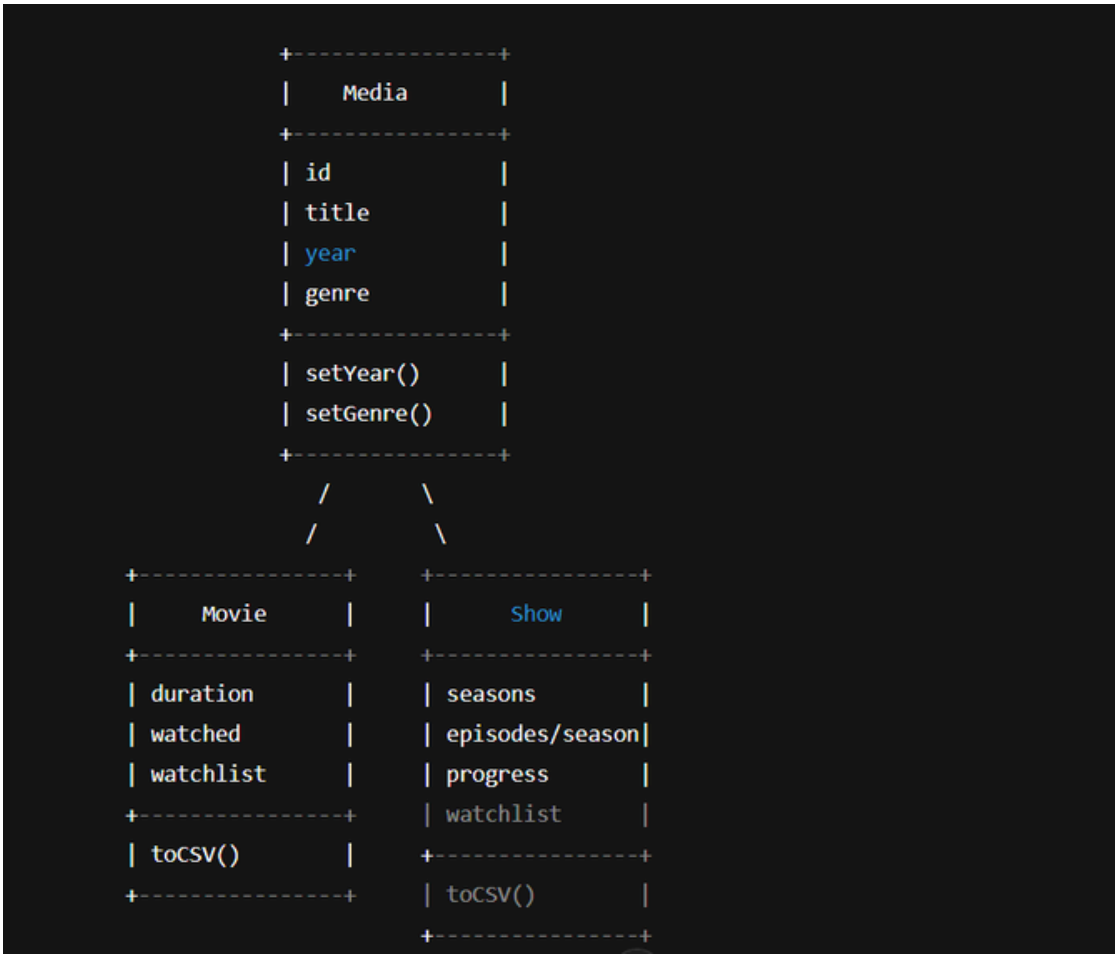


7. Design Diagrams

7.1 Use Case Diagram



7.2 Class Diagram



8. Design Decisions & Rationale

- **CSV Storage instead of Database**
  Lightweight, easy to parse, and ideal for beginner-level Java projects.
- **OOP-Based Model Desig**
  Media → Movie & Show hierarchy improves maintainability.
- **Modular Packages**
  models → data objects
  services → business logic
  app → user interface
  This separation follows clean-code principles.
- **Console UI**
  Provides universal compatibility and simplicity.

9. Implementation Details

- **Packages**
  models
  Media.java
  Movie.java
  Show.java

  services
  Catalog.java

  apps
  MediaApp.java

- **Key Concepts Used**
  Inheritance
  Polymorphism
  ArrayList collections
  File handling with Scanner, BufferedReader, BufferedWriter
  Exception handling
  Input validation

10. Screenshots

- **Menu**

```
PS C:\Coding stuff\JavaProject> java -cp out app.MediaApp
=== Welcome to Media Catalog ===

Menu:
1) Add Movie
2) Add Show
3) List All Media
4) Search (title/genre)
5) Remove by ID
6) Update Media
7) View Watchlist
8) Save & Exit
```

- **Add Movie**

```
Choice: 1
Enter ID: M001
Title: Spider Man
Year: 2002
Genre: Action, Drama
Duration (minutes): 120
Have you watched it? (y/n): y
Movie added.
```

- **Add Show**

```
Choice: 2
Enter ID: S001
Title: Stranger Things
Year: 2016
Genre: Thriller, Suspense, Gore, Horror, Comedy
Seasons: 5
Episodes per Season: 8
Have you watched any episodes? Enter as 'S E' (season episode), or leave blank:
Add to watchlist? (y/n): n
Show added.
```

- **List All Media**

```
Choice: 3

Movies:
Movie | ID:M001 | Spider Man (2002) | Action, Drama | 120 min | Watched
Movie | ID:M002 | Baahubali The beginning (2015) | Action, Drama | 150 min | Watched
Movie | ID:M003 | Inception (2011) | Action, SciFi, Thriller | 180 min | Not watched (In watchlist)

Shows:
Show  | ID:S001 | Stranger Things (2016) | Thriller, Suspense, Gore, Horror, Comedy | 5 seasons x 8 eps | Progress: Not started
Show  | ID:S002 | Game of Thrones (2011) | Action, Historical, Gore | 8 seasons x 8 eps | Progress: S1 E7
Show  | ID:S003 | Breaking Bad (2009) | Thriller, SciFi, Suspense | 5 seasons x 12 eps | Progress: S1 E4
```

- **Search Media**

```
Choice: 4
Enter query (title or genre): Baahubali the beginning

Movie Matches:
Movie | ID:M002 | Baahubali The beginning (2015) | Action, Drama | 150 min | Watched

Show Matches:
```

- **Remove Media**

```
Choice: 5
Enter ID to remove: S003
Removed successfully.
```

new media

```
Choice: 3

Movies:
Movie | ID:M001 | Spider Man (2002) | Action, Drama | 120 min | Watched
Movie | ID:M002 | Baahubali The beginning (2015) | Action, Drama | 150 min | Watched
Movie | ID:M003 | Inception (2011) | Action, SciFi, Thriller | 180 min | Not watched (In watchlist)

Shows:
Show  | ID:S001 | Stranger Things (2016) | Thriller, Suspense, Gore, Horror, Comedy | 5 seasons x 8 eps | Progress: Not started
Show  | ID:S002 | Game of Thrones (2011) | Action, Historical, Gore | 8 seasons x 8 eps | Progress: S1 E7
```

- **Update Media**

```
Choice: 6
Enter ID to update: S001
New Year (or blank):
New Genre (or blank):
New Seasons (or blank):
New Episodes/Season (or blank):
Update progress? Enter 'S E' (season episode) or blank to skip: 4 8
Show updated.
```

new media

```
Choice: 3

Movies:
Movie | ID:M001 | Spider Man (2002) | Action, Drama | 120 min | Watched
Movie | ID:M002 | Baahubali The beginning (2015) | Action, Drama | 150 min | Watched
Movie | ID:M003 | Inception (2011) | Action, SciFi, Thriller | 180 min | Not watched (In watchlist)

Shows:
Show  | ID:S001 | Stranger Things (2016) | Thriller, Suspense, Gore, Horror, Comedy | 5 seasons x 8 eps | Progress: S4 E8
Show  | ID:S002 | Game of Thrones (2011) | Action, Historical, Gore | 8 seasons x 8 eps | Progress: S1 E7
```

- **View Watchlist**

```
Watchlist - Movies:
Movie | ID:M003 | Inception (2011) | Action, SciFi, Thriller | 180 min | Not watched (In watchlist)

Watchlist - Shows:
```

- **Save & Exit**

```
Choice: 8
Saved. Goodbye!
```

data in movies.csv

movies.csv > data

```
M001,Spider Man,2002,Action\, Drama,120,true,false
M002,Baahubali The beginning,2015,Action\, Drama,150,true,false
M003,Inception,2011,Action\, SciFi\, Thriller,180,false,true
```

data in shows.csv

shows.csv > data

```
S001,Stranger Things,2016,Thriller\, Suspense\, Gore\, Horror\, Comedy,5,8,4,8,false
S002,Game of Thrones,2011,Action\, Historical\, Gore,8,8,1,7,false
```

11. Testing Approach

Test Case	Input	Expected Output	Status
Add Movie	Valid data	Movie added	Passed
Add Show	Progress S/E	Stored correctly	Passed
Search	Title = "Baahubali"	Movie found	Passed
Watchlist	Add & View	Display both lists	Passed
Update Movie	Change genre	Updated	Passed
Save & Exit	Exit program	Data saved	Passed

12. Challenges Faced

- Handling CSV escaping for titles with commas
- Designing update method for both movies and shows
- Managing watch progress (season/episode)
- Avoiding stale .class files during compilation
- Maintaining a clean, modular folder structure

13. Learnings & Key Takeaways

- Strong understanding of OOP concepts
- Experience with file handling and persistence
- Exposure to modular project structure
- Practice with UML diagrams and documentation
- Improved debugging and problem-solving skills

14. Future Enhancements

- Add JSON storage using Gson/Jackson
- Add sorting: by year, title, genre, watch status
- Add user profiles
- Add recommendations based on genre
- Implement GUI using JavaFX
- Add export to PDF/HTML

15. References

- Oracle Java Documentation
- Java File I/O tutorials
- StackOverflow (general troubleshooting)
- Course reference PDF