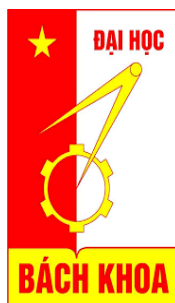


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



ĐỒ ÁN I

THIẾT KẾ PHẦN MỀM DOWNLOAD TORRENT VÀ PHÁT VIDEO CHO HỆ ĐIỀU HÀNH WINDOWS

Trần Tuấn Tú

Tu.TT223819@sis.hust.edu.vn

Ngành KT Điện tử - Viễn thông

Giảng viên hướng dẫn: TS.Đặng Quang Hiếu Chữ ký của GVHD

Bộ môn: Đồ án 1

Trường: Điện - Điện tử

Hà Nội, June 14, 2025

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN - ĐIỆN TỬ



ĐỒ ÁN I

Đề tài:

THIẾT KẾ PHẦN MỀM DOWNLOAD TORRENT VÀ PHÁT VIDEO CHO HỆ ĐIỀU HÀNH WINDOWS

Sinh viên thực hiện: **TRẦN TUẤN TÚ**

Lớp: Điện tử 09 - k67

Giảng viên hướng dẫn: **TS. ĐẶNG QUANG HIẾU**

Cán bộ phản biện:

Hà Nội, June 14, 2025

ĐÁNH GIÁ QUYỂN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho giảng viên hướng dẫn)

Tên giảng viên đánh giá:

Họ và tên sinh viên: MSSV:

Tên đồ án:

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém(2); Đạt(3); Giỏi(4); Xuất sắc(5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4	5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả (15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai	1	2	3	4	5
Kỹ năng viết quyển đồ án (10)						
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến; căn lề thống nhất, có dấu cách sau dấu chấm, dấu phẩy v.v.), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4	5
Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)						
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/Đạt giải SVNCKH giải 3 cấp Viện trở lên/Có giải thưởng khoa học (quốc tế hoặc trong nước) từ giải 3 trở lên/Có đăng ký bằng phát minh, sáng chế	5				
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị SVNCKH nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành (VD: TI contest)	2				
10c	Không có thành tích về nghiên cứu khoa học	0				
Điểm tổng		/50				
Điểm tổng quy đổi về thang 10						

***Nhận xét khác** (về thái độ và tinh thần làm việc của sinh viên)*

.....

.....

.....

.....

.....

.....

Ngày: ... / ... / 20...

Người nhận xét
(Ký và ghi rõ họ tên)

ĐÁNH GIÁ QUYỂN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho cán bộ phản biện)

Giảng viên đánh giá:

Họ và tên sinh viên: MSSV:

Tên đồ án:

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:

Rất kém (1); Kém(2); Đạt(3); Giỏi(4); Xuất sắc(5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4	5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả (15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai	1	2	3	4	5
Kỹ năng viết quyển đồ án (10)						
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến; căn lề thống nhất, có dấu cách sau dấu chấm, dấu phẩy v.v.), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4	5
Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)						
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/Đạt giải SVNCKH giải 3 cấp Viện trở lên/Có giải thưởng khoa học (quốc tế hoặc trong nước) từ giải 3 trở lên/Có đăng ký bằng phát minh, sáng chế	5				
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị SVNCKH nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành (VD: TI contest)	2				
10c	Không có thành tích về nghiên cứu khoa học	0				
Điểm tổng		/50				
Điểm tổng quy đổi về thang 10						

Nhận xét khác của cán bộ phản biện

.....

.....

.....

.....

.....

.....

Ngày: ... / ... / 20...

Người nhận xét
(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong thời đại số hóa hiện nay, việc quản lý, khai thác và thưởng thức các nội dung số như phim ảnh, âm nhạc, tài liệu... đã trở thành nhu cầu thiết yếu đối với nhiều người dùng trên toàn thế giới. Sự phát triển mạnh mẽ của các nền tảng chia sẻ dữ liệu ngang hàng (P2P), đặc biệt là giao thức BitTorrent, đã mở ra cơ hội tiếp cận kho dữ liệu khổng lồ một cách nhanh chóng, tiện lợi và tiết kiệm chi phí. Tuy nhiên, để tận dụng tối đa tiềm năng này, người dùng cần có những công cụ hỗ trợ mạnh mẽ, thân thiện và tích hợp nhiều tính năng hiện đại. Xuất phát từ thực tiễn đó, em đã thực hiện đồ án “Xây dựng ứng dụng quản lý, tải và phát nội dung số tích hợp qBittorrent và VLC” với mục tiêu phát triển một phần mềm đa năng, hỗ trợ người dùng tìm kiếm, tải về và phát các bộ phim, video từ mạng BitTorrent thông qua giao diện đồ họa trực quan, hiện đại. Ứng dụng được xây dựng trên nền tảng Python, sử dụng các thư viện và công nghệ tiên tiến như PySide6 (Qt for Python) để phát triển giao diện người dùng, python-vlc để phát media, qbittorrent-api để giao tiếp và điều khiển qBittorrent, IMDbPY để tra cứu thông tin phim, cùng với requests, urllib3 phục vụ cho các tác vụ mạng. Cấu trúc project được tổ chức khoa học, bao gồm các thư mục chức năng như: controllers/: Xử lý logic điều khiển, kết nối giữa giao diện và các thành phần lõi. core/: Chứa các thành phần cốt lõi, xử lý nghiệp vụ chính của ứng dụng. models/: Định nghĩa các mô hình dữ liệu, giúp quản lý và lưu trữ thông tin phim, tiến trình tải, v.v. resources/: Lưu trữ các tài nguyên tĩnh như hình ảnh, biểu tượng, file cấu hình. ui/ và views/: Xây dựng giao diện người dùng, tổ chức các màn hình chức năng, giao tiếp với người dùng. qbittorrent/: Tích hợp qBittorrent Portable, cho phép ứng dụng điều khiển trực tiếp tiến trình tải về, quản lý file torrent, kiểm soát thư mục lưu trữ, nhật ký hoạt động, v.v. Các file chính như main.py (khởi động ứng dụng), player.py (xử lý phát media), qbit.py (tương tác với qBittorrent), test.py (kiểm thử chức năng), đảm bảo ứng dụng hoạt động ổn định, dễ dàng cài đặt và mở rộng. Ứng dụng cho phép người dùng: Tìm kiếm, tra cứu thông tin phim từ IMDb, hiển thị chi tiết về nội dung, diễn viên, đánh giá, poster phim. Tải về phim, video từ mạng BitTorrent thông qua qBittorrent, quản lý tiến trình tải, kiểm soát thư mục lưu trữ. Phát trực tiếp các nội dung đã tải về bằng VLC, hỗ trợ nhiều định dạng media, thao tác điều khiển linh hoạt. Giao diện thân thiện, dễ sử dụng, hỗ trợ đa nền tảng, dễ dàng mở rộng thêm các tính năng mới trong tương lai. Trong quá trình thực hiện đồ án, em đã không ngừng học hỏi, nghiên cứu tài liệu, thử nghiệm và hoàn thiện sản phẩm. Đồ án không chỉ giúp em củng cố kiến thức về lập trình Python, phát triển phần mềm, mà còn rèn luyện kỹ năng tổ chức, quản lý dự án, giải quyết vấn đề thực tiễn. Em xin gửi lời cảm ơn chân thành tới TS. Đặng Quang Hiếu, giảng viên hướng dẫn, người đã tận tình chỉ bảo, định hướng và hỗ trợ em trong suốt quá trình thực hiện đồ án. Em cũng xin cảm ơn

các thầy cô trong khoa, bạn bè và gia đình đã động viên, giúp đỡ em vượt qua những khó khăn, thử thách để hoàn thành sản phẩm này. Mặc dù đã rất nỗ lực, song do thời gian và kinh nghiệm còn hạn chế, đồ án chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, chỉ bảo của các thầy cô và các bạn để sản phẩm ngày càng hoàn thiện hơn.

Sinh viên thực hiện: Trần Tuấn Tú

Giảng viên hướng dẫn: TS. Đặng Quang Hiếu

LỜI CAM ĐOAN

Tôi tên là **TRẦN TUẤN TÚ**, mã số sinh viên **20223819**, sinh viên lớp **Điện tử 09**, khóa **67**. Người hướng dẫn là **TS. ĐẶNG QUANG HIẾU**.

Tôi xin cam đoan rằng toàn bộ nội dung được trình bày trong đồ án này là kết quả của quá trình học tập, nghiên cứu và lao động nghiêm túc của bản thân dưới sự hướng dẫn tận tình của thầy giáo hướng dẫn. Các kết quả, số liệu, hình ảnh, bảng biểu và các nội dung khác trong đồ án đều trung thực, chính xác, phản ánh đúng quá trình thực hiện và kết quả đạt được.

Tôi khẳng định rằng đồ án này không sao chép, không vi phạm bản quyền của bất kỳ cá nhân, tổ chức nào. Mọi thông tin, dữ liệu, kiến thức, hình ảnh, mã nguồn, tài liệu tham khảo được sử dụng trong quá trình thực hiện đồ án đều được trích dẫn rõ ràng, tuân thủ đầy đủ các quy định về sở hữu trí tuệ và đạo đức học thuật. Nếu phát hiện có bất kỳ hành vi gian lận, sao chép hoặc vi phạm bản quyền nào, tôi xin hoàn toàn chịu trách nhiệm trước nhà trường, thầy cô và pháp luật.

Tôi xin chân thành cảm ơn thầy giáo hướng dẫn, các thầy cô trong khoa, bạn bè và gia đình đã tạo điều kiện, động viên và hỗ trợ tôi trong suốt quá trình học tập và thực hiện đồ án này.

Hà Nội, June 14, 2025

Người cam đoan

TRẦN TUẤN TÚ

MỤC LỤC

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT	i
DANH MỤC HÌNH VẼ	ii
DANH MỤC BẢNG BIỂU	iii
TÓM TẮT ĐỒ ÁN	iv
CHƯƠNG 1. CHƯƠNG MỞ ĐẦU	1
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	3
2.1 Giới thiệu cơ sở lý thuyết	3
2.2 Mô hình MVC	3
2.3 Cơ sở dữ liệu SQLite	4
2.4 Thư viện PySide6	4
2.5 Module VLC	4
2.6 BitTorrent và qBittorrent	5
2.7 Tóm tắt chương	7
CHƯƠNG 3. THIẾT KẾ VÀ TRIỂN KHAI	8
3.1 Tổng quan thiết kế hệ thống	8
3.2 Thiết kế cơ sở dữ liệu	8
3.3 Xây dựng giao diện người dùng	9
3.4 Tích hợp các dịch vụ	10
3.4.1 Tích hợp trình phát video VLC	10
3.4.2 Tích hợp qBittorrent	11
3.5 Quy trình hoạt động	11
3.5.1 Đăng nhập người dùng	11
3.5.2 Thêm và quản lý thông tin phim (Tích hợp API OMDB)	11

3.5.3	Tìm kiếm phim trong cơ sở dữ liệu	12
3.5.4	Tải phim (Tích hợp qBittorrent)	12
3.5.5	Phát video (Tích hợp VLC)	14
3.6	Tóm tắt Chương 3	16
CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ		17
4.1	Kết quả chạy thử nghiệm	17
4.1.1	Kịch bản thử nghiệm	17
4.2	Đánh giá hiệu suất	20
4.3	Hạn chế	20
4.4	Đề xuất cải tiến	21
4.5	Tóm tắt Chương 4	22
KẾT LUẬN		23
PHỤ LỤC		25

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Ký hiệu/Viết tắt	Diễn giải
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CSDL	Cơ sở dữ liệu
CPU	Bộ xử lý trung tâm (Central Processing Unit)
DHT	Bảng băm phân tán (Distributed Hash Table)
GUI	Giao diện đồ họa người dùng (Graphical User Interface)
HTTP	Giao thức truyền tải siêu văn bản (HyperText Transfer Protocol)
ID	Định danh (Identifier)
IMDb	Cơ sở dữ liệu phim trên Internet (Internet Movie Database)
JSON	Ký hiệu đối tượng JavaScript (JavaScript Object Notation)
MVC	Mô hình - Khung nhìn - Điều khiển (Model-View-Controller)
OMDB	Cơ sở dữ liệu phim mở (Open Movie Database)
P2P	Ngang hàng (Peer-to-Peer)
PySide6	Thư viện Python để xây dựng giao diện đồ họa, dựa trên Qt
qBittorrent	Tên một ứng dụng client BitTorrent mã nguồn mở
QSS	Bảng kiểu Qt (Qt Style Sheets)
RAM	Bộ nhớ truy cập ngẫu nhiên (Random Access Memory)
SQL	Ngôn ngữ truy vấn có cấu trúc (Structured Query Language)
SQLite	Hệ quản trị cơ sở dữ liệu quan hệ nhúng, không cần máy chủ
URL	Định vị tài nguyên thống nhất (Uniform Resource Locator)
VLC	Trình phát đa phương tiện VideoLAN Client
UI	Giao diện người dùng (User Interface)

DANH MỤC HÌNH VẼ

Hình 2.1	Mô hình MVC trong ứng dụng quản lý phim	3
Hình 2.2	Logo của SQLite, hệ quản trị cơ sở dữ liệu được sử dụng trong đồ án	4
Hình 2.3	Giao diện của PySide6	4
Hình 2.4	Tích hợp trình phát video VLC trong ứng dụng	5
Hình 2.5	Mô hình hoạt động của giao thức BitTorrent	6
Hình 2.6	Giao diện người dùng của qBittorrent	7
Hình 3.1	Sơ đồ khối của hệ thống	8
Hình 3.2	Giao diện người dùng	10
Hình 3.3	Quy trình tích hợp API OMDB	12
Hình 3.4	Quy trình hoạt động của qBittorrent	13
Hình 3.5	Quy trình hoạt động của VLC	15

DANH MỤC BẢNG BIỂU

Bảng 3.1	Cấu trúc bảng accounts	9
Bảng 3.2	Cấu trúc bảng movies	9
Bảng 4.1	Bảng tổng hợp kết quả chạy thử nghiệm	19

TÓM TẮT ĐỒ ÁN

Tiếng Việt:

Đồ án này trình bày việc xây dựng một ứng dụng quản lý và hiển thị thông tin phim, được phát triển dựa trên mô hình MVC (Model-View-Controller). Ứng dụng sử dụng ngôn ngữ lập trình Python kết hợp với thư viện PySide6 để xây dựng giao diện người dùng. Dữ liệu phim được lưu trữ trong cơ sở dữ liệu SQLite, đảm bảo tính nhẹ nhàng, dễ triển khai và quản lý. Các tính năng chính của ứng dụng bao gồm:

- Hiển thị danh sách phim với chức năng phân trang, giúp người dùng dễ dàng duyệt qua các bộ phim.
- Xem chi tiết thông tin phim, bao gồm tiêu đề, năm phát hành, poster và các thông tin bổ sung khác.
- Tìm kiếm phim trên The Pirate Bay thông qua tiêu đề, hỗ trợ người dùng tìm kiếm nguồn tải phim nhanh chóng.
- Tích hợp trình phát video sử dụng module VLC, cho phép người dùng xem trực tiếp các bộ phim trong ứng dụng với trải nghiệm mượt mà.
- Hỗ trợ tải xuống torrent thông qua qBittorrent, giúp quản lý và tải phim dễ dàng.

Kết quả đạt được là một ứng dụng trực quan, dễ sử dụng, phù hợp cho các dự án cá nhân hoặc quy mô nhỏ, với khả năng mở rộng trong tương lai.

English:

This project presents the development of a movie management and display application, built based on the MVC (Model-View-Controller) architecture. The application is developed using Python programming language combined with the PySide6 library for building the user interface. Movie data is stored in an SQLite database, ensuring lightweight, easy deployment, and management. The main features of the application include:

- Displaying a list of movies with pagination functionality, allowing users to easily browse through movies.
- Viewing detailed movie information, including title, release year, poster, and additional details.

- Searching for movies on The Pirate Bay using the title, supporting quick access to movie sources.
- Integrating a video player using the VLC module, enabling users to watch movies directly within the application with a smooth experience.
- Supporting torrent downloads via qBittorrent, making it easy to manage and download movies.

The result is an intuitive and user-friendly application suitable for personal or small-scale projects, with potential for future expansion.

CHƯƠNG 1. CHƯƠNG MỞ ĐẦU

1.1. Giới thiệu vấn đề

Trong thời đại công nghệ số hiện nay, nhu cầu quản lý và truy cập thông tin phim một cách nhanh chóng và hiệu quả ngày càng trở nên phổ biến. Các ứng dụng quản lý phim hiện có thường phức tạp hoặc yêu cầu kết nối internet liên tục, gây khó khăn cho người dùng cá nhân hoặc các dự án nhỏ. Do đó, việc xây dựng một ứng dụng quản lý phim đơn giản, dễ sử dụng, nhưng vẫn đáp ứng đầy đủ các nhu cầu cơ bản như hiển thị thông tin phim, tìm kiếm, và phát video là một vấn đề cần được giải quyết.

1.2. Phương pháp hiện có

Hiện nay, có nhiều ứng dụng và nền tảng hỗ trợ quản lý và phát phim, ví dụ:

- Các nền tảng trực tuyến như Netflix, YouTube, hoặc Amazon Prime cung cấp dịch vụ phát trực tuyến nhưng yêu cầu kết nối internet và tài khoản trả phí.
- Các ứng dụng quản lý phim cục bộ như Kodi hoặc Plex, tuy nhiên, chúng thường phức tạp và yêu cầu cấu hình cao.
- Các công cụ tải torrent và phát video riêng lẻ, nhưng không tích hợp đầy đủ các tính năng quản lý thông tin phim.

Những phương pháp này tuy hiệu quả nhưng không phù hợp cho các dự án nhỏ hoặc người dùng cá nhân muốn có một giải pháp đơn giản, tích hợp và dễ triển khai.

1.3. Mục đích của đề án

Mục đích của đề án là xây dựng một ứng dụng quản lý phim tích hợp, đáp ứng các yêu cầu sau:

- Hiển thị danh sách phim với chức năng phân trang, giúp người dùng dễ dàng duyệt qua các bộ phim.
- Xem chi tiết thông tin phim, bao gồm tiêu đề, năm phát hành, poster và các thông tin bổ sung khác.
- Tìm kiếm phim trên The Pirate Bay thông qua tiêu đề, hỗ trợ người dùng tìm kiếm nguồn tải phim nhanh chóng.
- Tích hợp trình phát video sử dụng module VLC, cho phép người dùng xem trực tiếp các bộ phim trong ứng dụng.
- Hỗ trợ tải xuống torrent thông qua qBittorrent, giúp quản lý và tải phim dễ dàng.

1.4. Phạm vi của đề án

Đề án tập trung vào việc xây dựng một ứng dụng quản lý phim cục bộ, phù hợp cho các dự án cá nhân hoặc quy mô nhỏ. Ứng dụng không yêu cầu kết nối internet liên tục, ngoại trừ khi tìm kiếm phim trên The Pirate Bay hoặc tải torrent. Phạm vi đề án không bao gồm việc phát triển các thuật toán xử lý video hoặc xây dựng cơ sở dữ liệu phim trực tuyến.

1.5. Cấu trúc đề án

Cấu trúc của đề án bao gồm các chương sau:

- **Chương 1: Chương mở đầu** - Giới thiệu vấn đề, phương pháp hiện có, mục đích, phạm vi và cấu trúc của đề án.
- **Chương 2: Cơ sở lý thuyết** - Trình bày các công nghệ và thư viện được sử dụng, bao gồm PySide6, SQLite, VLC, và qBittorrent.
- **Chương 3: Thiết kế và triển khai** - Mô tả chi tiết mô hình MVC, cách tổ chức mã nguồn, và các tính năng chính của ứng dụng.
- **Chương 4: Kết quả và đánh giá** - Trình bày kết quả đạt được, giao diện ứng dụng, và đánh giá hiệu quả của ứng dụng.
- **Chương 5: Kết luận và hướng phát triển** - Tóm tắt kết quả, rút ra bài học kinh nghiệm, và đề xuất các hướng phát triển trong tương lai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

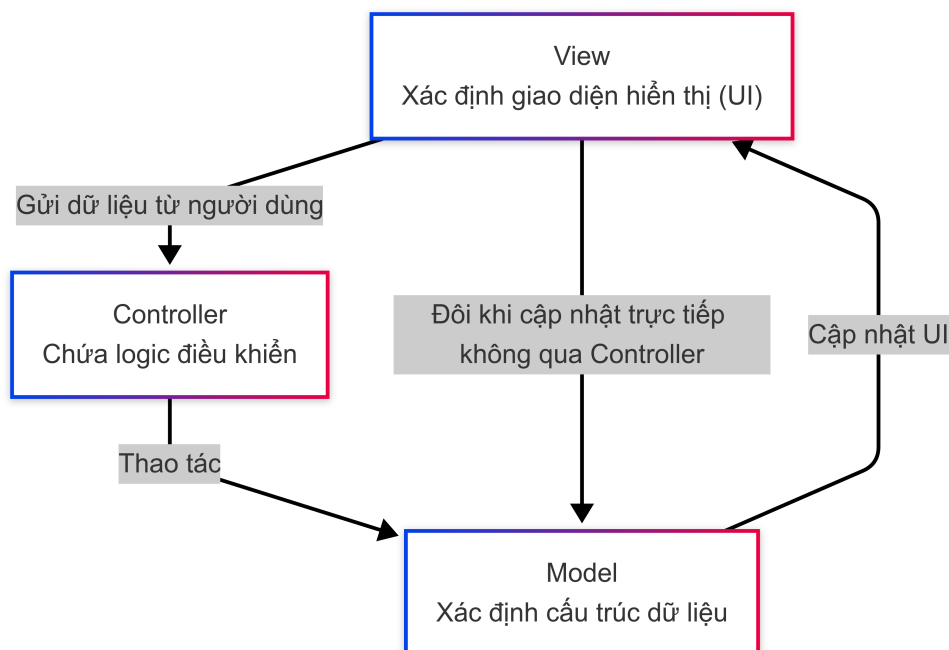
2.1 Giới thiệu cơ sở lý thuyết

Ứng dụng quản lý phim được xây dựng dựa trên mô hình MVC (Model-View-Controller) và sử dụng các công nghệ hiện đại như PySide6, SQLite, VLC, và qBittorrent. Trong chương này, chúng ta sẽ trình bày các khái niệm cơ bản và công nghệ được sử dụng trong đồ án.

2.2 Mô hình MVC

Mô hình MVC (Model-View-Controller) là một kiến trúc phần mềm phổ biến, được sử dụng để tách biệt logic xử lý, giao diện người dùng và dữ liệu. Mô hình này bao gồm ba thành phần chính:

- **Model:** Chịu trách nhiệm quản lý dữ liệu và logic nghiệp vụ. Trong đồ án này, dữ liệu phim được lưu trữ trong cơ sở dữ liệu SQLite.
- **View:** Hiển thị giao diện người dùng và nhận đầu vào từ người dùng. PySide6 được sử dụng để xây dựng giao diện đồ họa.
- **Controller:** Xử lý các yêu cầu từ người dùng, tương tác với Model và cập nhật View. Controller kết nối các thành phần của ứng dụng.



Hình 2.1 Mô hình MVC trong ứng dụng quản lý phim

2.3 Cơ sở dữ liệu SQLite

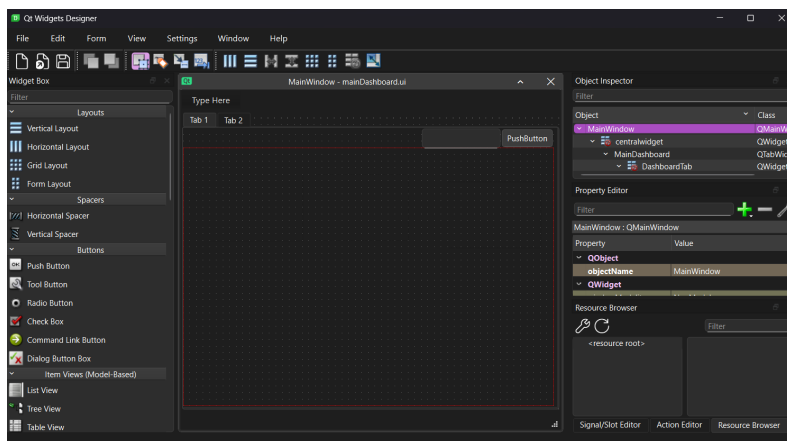
SQLite là một hệ quản trị cơ sở dữ liệu quan hệ nhẹ, không cần máy chủ, phù hợp cho các ứng dụng cục bộ. Trong đồ án này, SQLite được sử dụng để lưu trữ thông tin phim, bao gồm tiêu đề, năm phát hành, poster, và các thông tin bổ sung khác. Cơ sở dữ liệu SQLite đảm bảo tính đơn giản, dễ triển khai và hiệu quả.



Hình 2.2 Logo của SQLite, hệ quản trị cơ sở dữ liệu được sử dụng trong đồ án

2.4 Thư viện PySide6

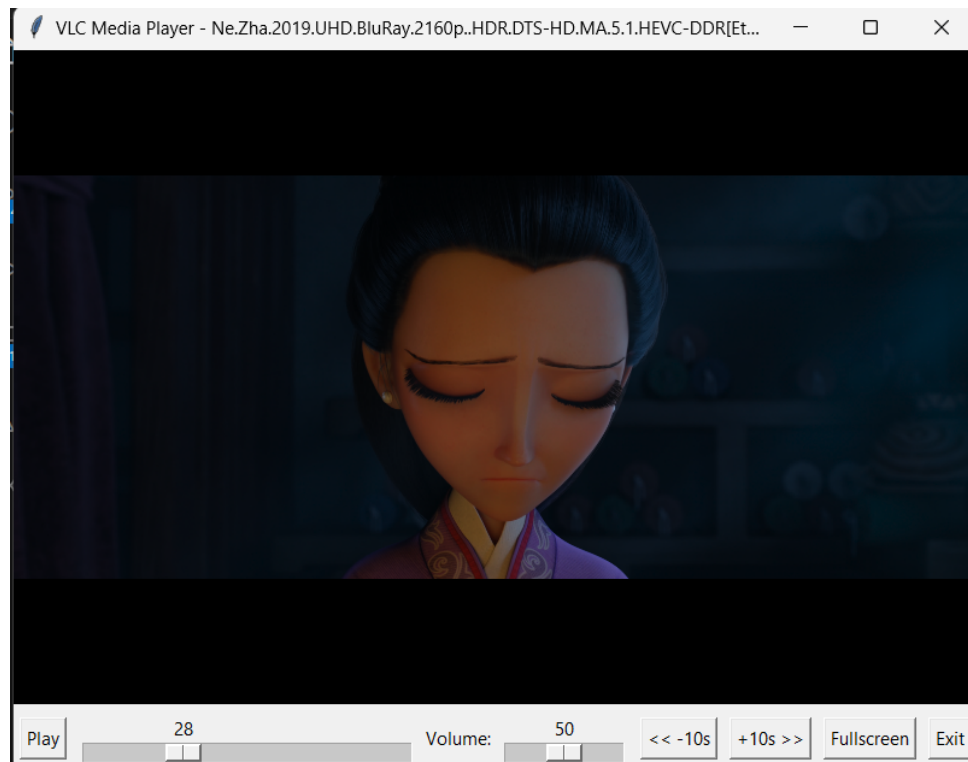
PySide6 là một thư viện Python mạnh mẽ để xây dựng giao diện đồ họa người dùng (GUI). Thư viện này cung cấp các công cụ để tạo các widget, layout, và xử lý sự kiện. Trong đồ án, PySide6 được sử dụng để xây dựng các thành phần giao diện như danh sách phim, chi tiết phim, và các nút chức năng.



Hình 2.3 Giao diện của PySide6

2.5 Module VLC

VLC là một trình phát đa phương tiện mã nguồn mở, hỗ trợ nhiều định dạng video và âm thanh. Module VLC trong Python được sử dụng để tích hợp trình phát video vào ứng dụng, cho phép người dùng xem phim trực tiếp trong giao diện.



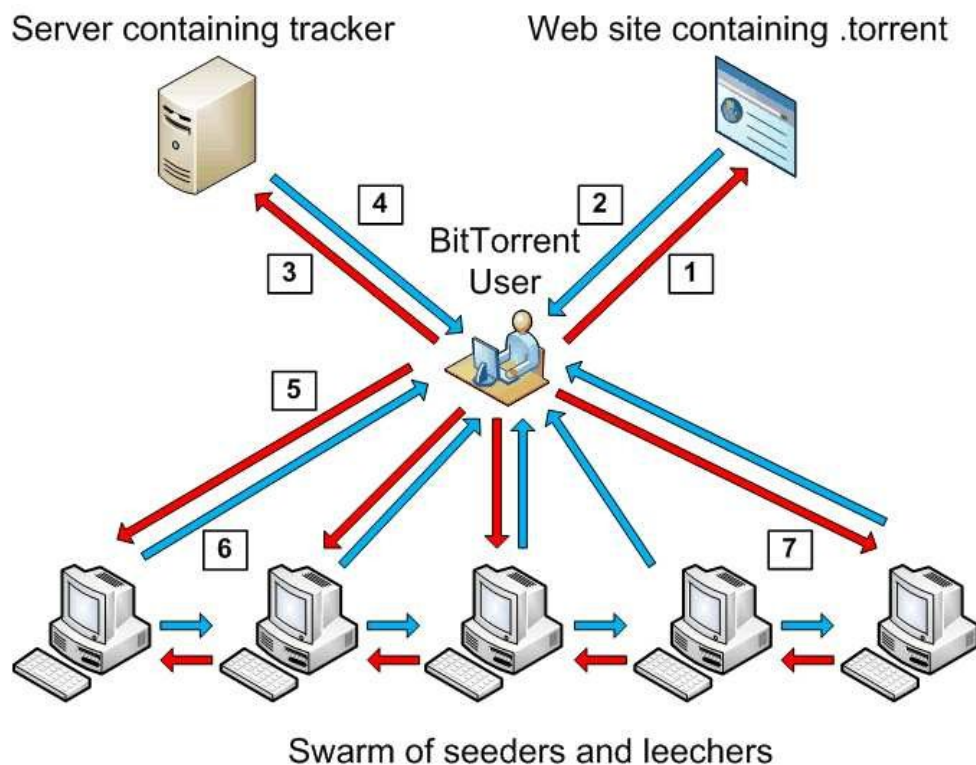
Hình 2.4 Tích hợp trình phát video VLC trong ứng dụng

2.6 BitTorrent và qBittorrent

BitTorrent là một giao thức chia sẻ tệp ngang hàng (peer-to-peer - P2P) được thiết kế để phân phối dữ liệu hiệu quả trên mạng internet. Thay vì tải xuống tệp từ một máy chủ trung tâm, BitTorrent cho phép người dùng tải xuống tệp từ nhiều nguồn khác nhau cùng lúc, giúp tăng tốc độ tải và giảm tải cho máy chủ.

Cách hoạt động của BitTorrent:

- **Tệp Torrent và Tracker:** Tệp torrent chứa thông tin về tệp cần tải xuống, bao gồm tên tệp, kích thước, và URL của tracker. Tracker là máy chủ trung gian giúp kết nối các peer (người dùng) trong mạng.
- **Peer và Seed:** Peer là người dùng đang tham gia tải xuống hoặc tải lên tệp. Seed là một peer đã tải xuống toàn bộ tệp và đang chia sẻ lại cho các peer khác.
- **Phân mảnh tệp:** Tệp cần tải xuống được chia thành nhiều mảnh nhỏ (pieces). Mỗi peer có thể tải xuống các mảnh từ nhiều peer khác nhau và đồng thời chia sẻ các mảnh mà họ đã tải xuống.
- **DHT (Distributed Hash Table):** Hệ thống phân tán cho phép các peer tìm thấy nhau mà không cần một máy chủ trung tâm.

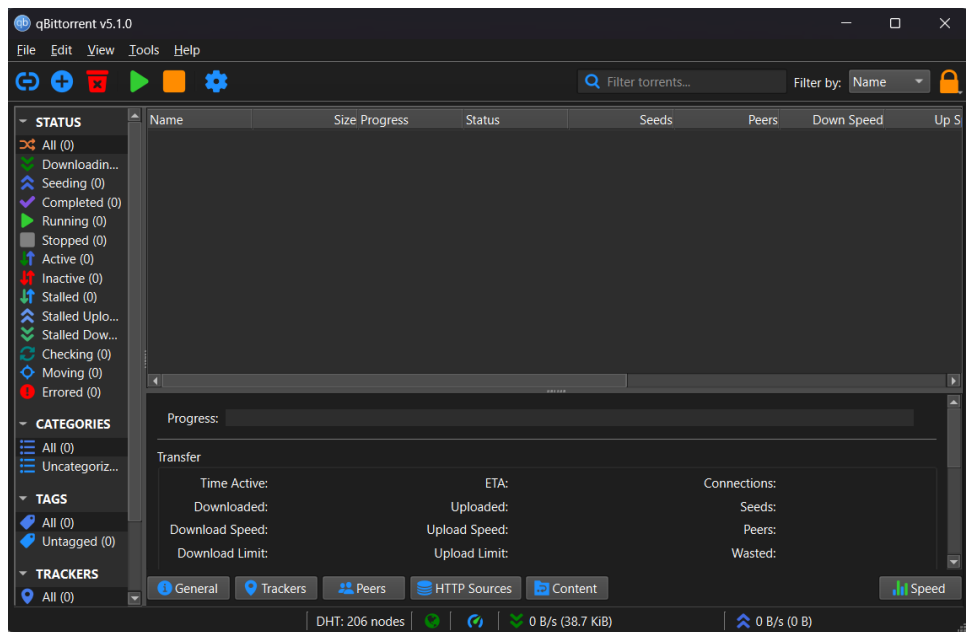


Hình 2.5 Mô hình hoạt động của giao thức BitTorrent

qBittorrent là một ứng dụng mã nguồn mở được sử dụng để quản lý và tải xuống tệp torrent. Trong đồ án, qBittorrent được tích hợp thông qua API Web, cho phép ứng dụng gửi yêu cầu tải xuống và theo dõi tiến trình tải.

Các tính năng chính của qBittorrent:

- Giao diện đồ họa thân thiện, dễ sử dụng.
- Hỗ trợ DHT và PEX để tìm kiếm và kết nối các peer.
- Quản lý băng thông, cho phép giới hạn tốc độ tải xuống và tải lên.
- Tích hợp API Web để tương tác với các ứng dụng bên ngoài.



Hình 2.6 Giao diện người dùng của qBittorrent

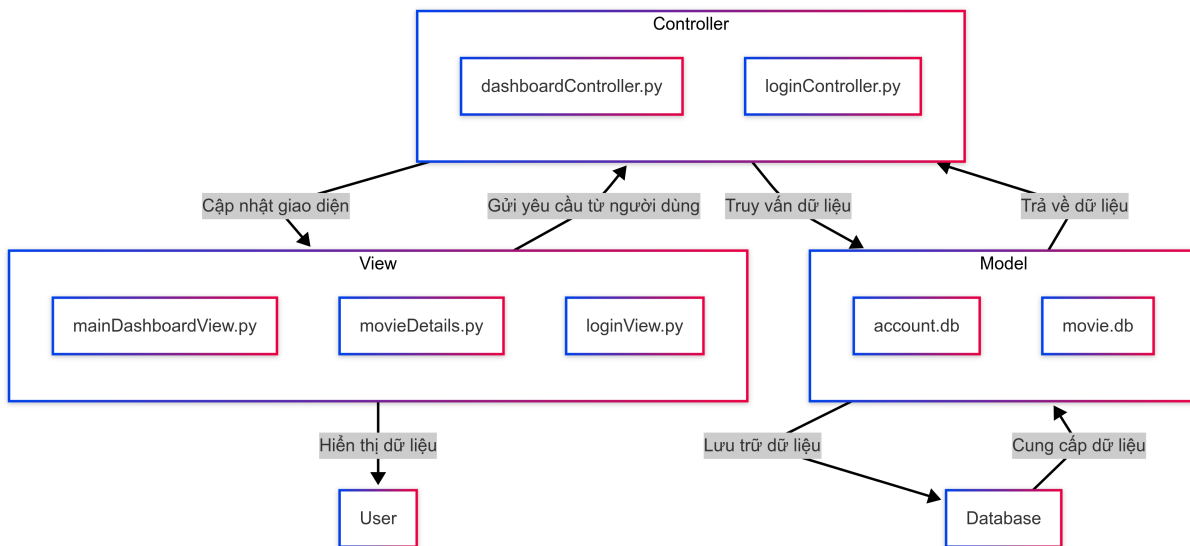
2.7 Tóm tắt chương

Chương này đã trình bày các cơ sở lý thuyết và công nghệ được sử dụng trong đồ án, bao gồm mô hình MVC, SQLite, PySide6, VLC, BitTorrent, và qBittorrent. Các công nghệ này được tích hợp để xây dựng một ứng dụng quản lý phim trực quan, dễ sử dụng và hiệu quả.

CHƯƠNG 3. THIẾT KẾ VÀ TRIỂN KHAI

3.1 Tổng quan thiết kế hệ thống

Hệ thống quản lý phim được thiết kế dựa trên mô hình MVC (Model-View-Controller), đảm bảo tính tách biệt giữa các thành phần dữ liệu, giao diện, và logic xử lý. Sơ đồ khối của hệ thống được thể hiện trong Hình 3.1.



Hình 3.1 Sơ đồ khối của hệ thống quản lý phim

Hình 3.1 mô tả cách các thành phần chính của hệ thống tương tác với nhau:

- **Model:** Quản lý dữ liệu phim và tài khoản, bao gồm việc lưu trữ và truy xuất thông tin từ cơ sở dữ liệu SQLite.
- **View:** Hiển thị giao diện người dùng, bao gồm danh sách phim, chi tiết phim, và các nút chức năng.
- **Controller:** Xử lý các yêu cầu từ người dùng, tương tác với Model và cập nhật View.

3.2 Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu SQLite được sử dụng để lưu trữ thông tin của ứng dụng. Các bảng chính bao gồm:

- **accounts:** Lưu trữ thông tin tài khoản người dùng.
- **movies:** Lưu trữ thông tin chi tiết về phim.

Tên cột	Kiểu dữ liệu	Mô tả
id	INTEGER	Khóa chính, tự động tăng
username	TEXT	Tên đăng nhập, duy nhất
password	TEXT	Mật khẩu (được lưu dưới dạng văn bản)
email	TEXT	Email người dùng

Bảng 3.1 Cấu trúc bảng accounts

Tên cột	Kiểu dữ liệu	Mô tả
id	INTEGER	Khóa chính, tự động tăng
title	TEXT	Tiêu đề phim
year	TEXT	Năm phát hành
genre	TEXT	Thể loại phim
director	TEXT	Đạo diễn
actors	TEXT	Danh sách diễn viên
plot	TEXT	Tóm tắt nội dung phim
poster	TEXT	URL poster phim
imdb_rating	TEXT	Điểm IMDb
imdb_id	TEXT	ID IMDb, duy nhất

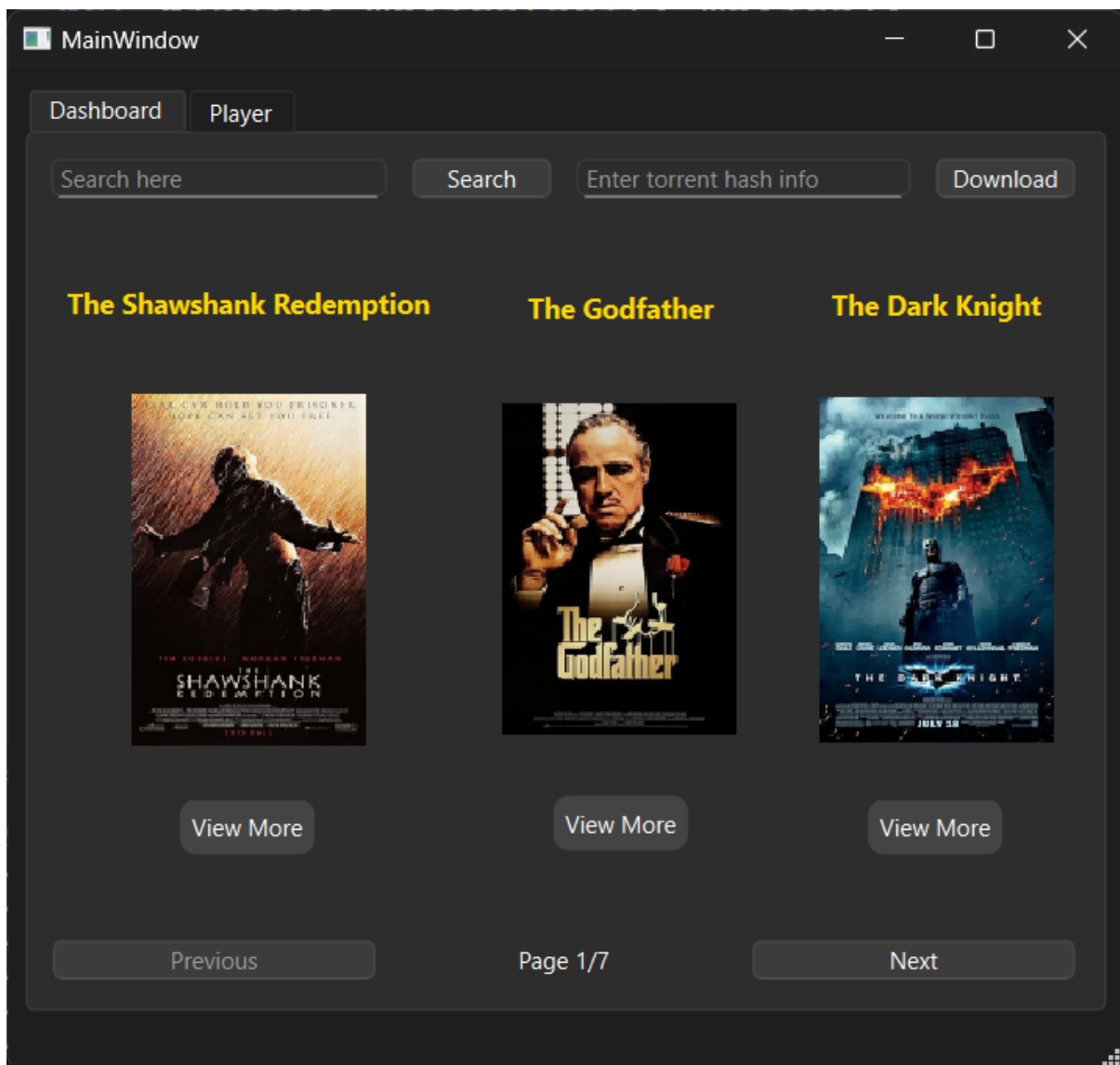
Bảng 3.2 Cấu trúc bảng movies

3.3 Xây dựng giao diện người dùng

Giao diện người dùng được xây dựng bằng thư viện PySide6. Các thành phần chính bao gồm:

- **Danh sách phim:** Hiển thị các bộ phim theo dạng lưới, với chức năng phân trang.
- **Chi tiết phim:** Hiển thị thông tin chi tiết của một bộ phim, bao gồm tiêu đề, năm phát hành, poster, và mô tả.
- **Nút chức năng:** Bao gồm các nút "View More", "Search on ThePirateBay", và "Download".

Hình 3.2 minh họa giao diện chính của ứng dụng, với danh sách phim được hiển thị theo dạng lưới.



Hình 3.2 Giao diện người dùng của ứng dụng quản lý phim

3.4 Tích hợp các dịch vụ

3.4.1 Tích hợp trình phát video VLC

Trình phát video VLC được tích hợp vào ứng dụng để cho phép người dùng xem phim trực tiếp. Module VLC trong Python được sử dụng để điều khiển trình phát, bao gồm các chức năng:

- Phát video từ tệp đã tải xuống.
- Tạm dừng, tiếp tục, và tua nhanh video.
- Hiển thị trạng thái phát video.

Cách trình phát VLC được tích hợp vào giao diện ứng dụng được minh họa trong quy trình hoạt động phát video (Hình 3.5).

3.4.2 Tích hợp qBittorrent

Ứng dụng sử dụng API Web của qBittorrent để hỗ trợ tải xuống phim từ các nguồn torrent. Quy trình hoạt động của qBittorrent khi tích hợp vào ứng dụng được mô tả chi tiết trong phần quy trình tải phim (Hình 3.4).

3.5 Quy trình hoạt động

3.5.1 Đăng nhập người dùng

Khi người dùng khởi động ứng dụng, một giao diện đăng nhập sẽ xuất hiện. Người dùng cần nhập tên đăng nhập (username) và mật khẩu (password). Hệ thống sẽ:

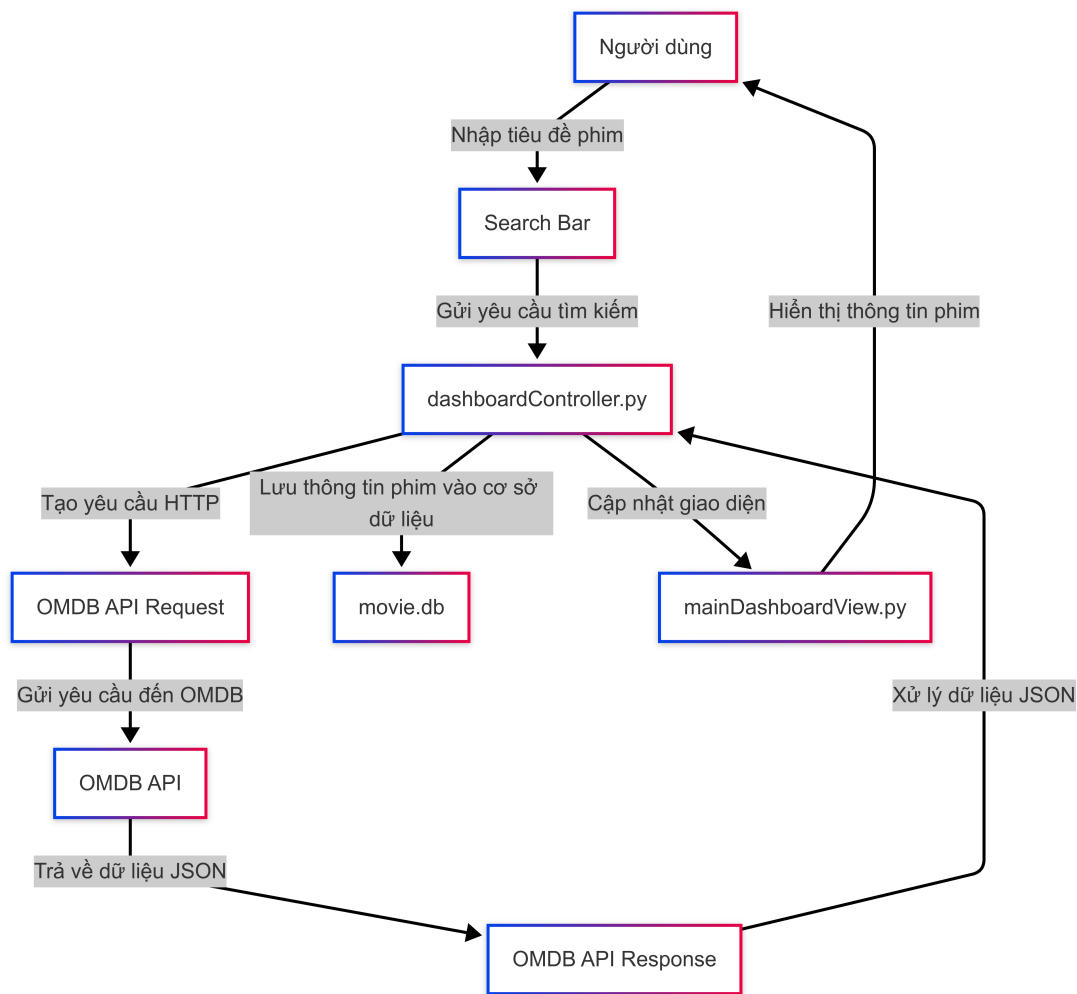
1. Tiếp nhận thông tin đăng nhập từ người dùng.
2. Truy vấn cơ sở dữ liệu bảng `accounts` (xem Bảng 3.1) để tìm kiếm tài khoản khớp với username đã nhập.
3. Nếu tìm thấy username, so sánh mật khẩu đã nhập với mật khẩu lưu trữ trong cơ sở dữ liệu.
4. Nếu thông tin hợp lệ, người dùng được chuyển đến giao diện chính của ứng dụng. Ngược lại, một thông báo lỗi sẽ được hiển thị.

3.5.2 Thêm và quản lý thông tin phim (Tích hợp API OMDB)

Để làm phong phú cơ sở dữ liệu phim, ứng dụng sử dụng API OMDB để lấy thông tin chi tiết về phim. Quy trình hoạt động như sau:

1. Người dùng (thường là quản trị viên hoặc thông qua một chức năng tìm kiếm phim trực tuyến) cung cấp tiêu đề phim.
2. Ứng dụng gửi yêu cầu HTTP đến API OMDB với tiêu đề phim và API key.
3. Nhận phản hồi JSON chứa thông tin chi tiết về phim (tiêu đề, năm, thể loại, đạo diễn, diễn viên, tóm tắt, poster, điểm IMDb, ID IMDb).
4. Lưu thông tin phim vào bảng `movies` trong cơ sở dữ liệu (xem Bảng 3.2) bằng hàm `insert_movie(movie)`. ID IMDb được sử dụng để đảm bảo tính duy nhất của phim.

Hình 3.3 minh họa quy trình tích hợp API OMDB trong ứng dụng. Sau khi phim được thêm vào cơ sở dữ liệu, người dùng có thể tìm kiếm cục bộ.



Hình 3.3 Quy trình tích hợp API OMDB để lấy thông tin phim

3.5.3 Tìm kiếm phim trong cơ sở dữ liệu

Người dùng có thể tìm kiếm các bộ phim đã được lưu trữ trong cơ sở dữ liệu cục bộ:

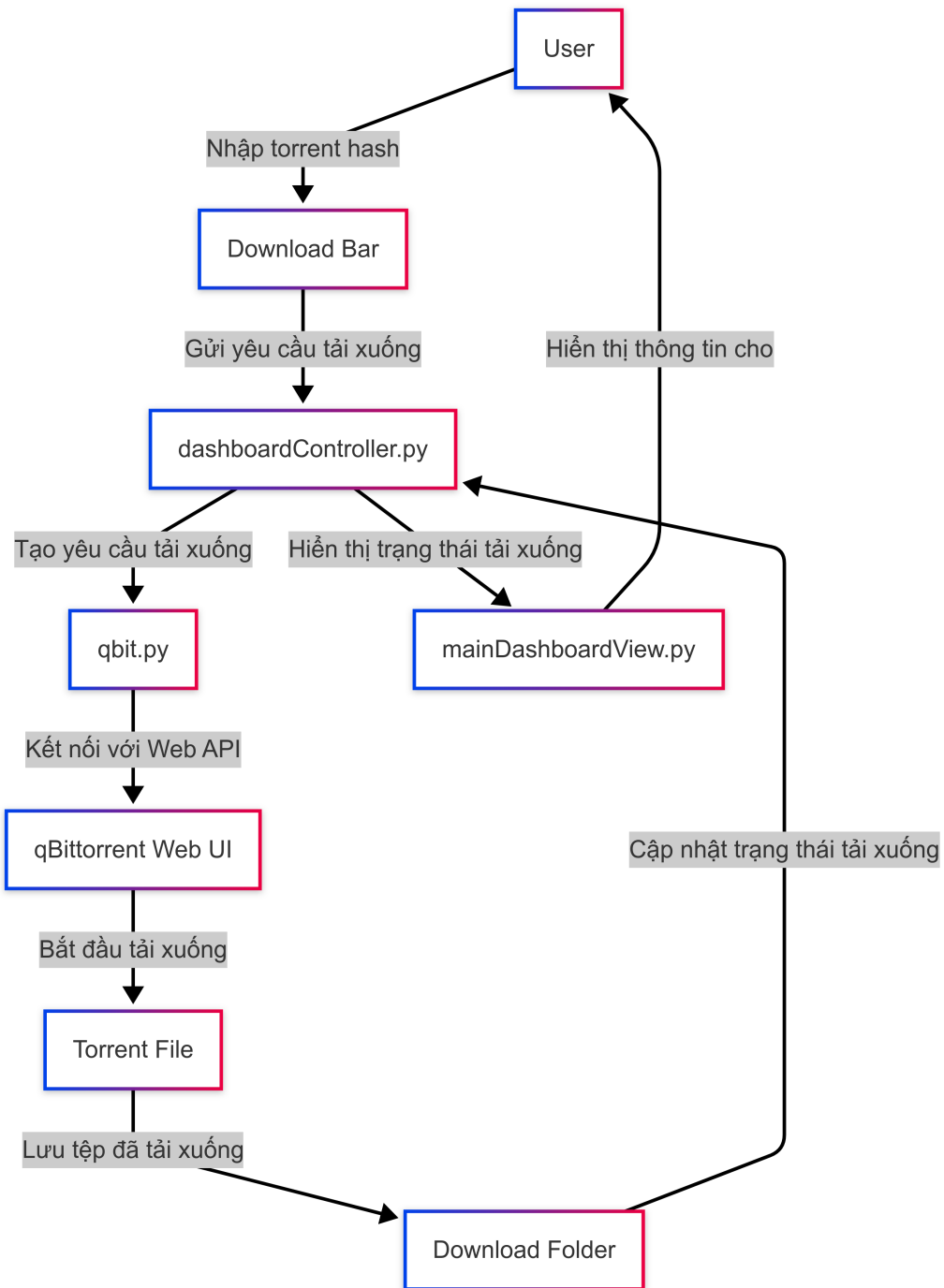
1. Người dùng nhập từ khóa vào thanh tìm kiếm trên giao diện chính.
2. Hệ thống thực hiện truy vấn SQL trên bảng movies để tìm các phim có tiêu đề, đạo diễn, hoặc diễn viên chứa từ khóa.
3. Kết quả tìm kiếm được hiển thị dưới dạng danh sách hoặc lưới trên giao diện người dùng.

placeins

3.5.4 Tải phim (Tích hợp qBittorrent)

Ứng dụng sử dụng API Web của qBittorrent để hỗ trợ tải xuống phim từ các nguồn torrent. Quy trình hoạt động như sau:

1. Sau khi tìm kiếm thông tin phim (ví dụ trên ThePirateBay thông qua nút chức năng), người dùng cung cấp hash của tệp torrent hoặc URL magnet.
2. Ứng dụng gửi yêu cầu đến qBittorrent Web API để bắt đầu tải xuống, sử dụng thông tin hash/URL đã cung cấp.
3. qBittorrent client (chạy trên máy người dùng hoặc máy chủ được cấu hình) sẽ tự động xử lý yêu cầu, tải xuống tệp và lưu trữ vào thư mục đã được định sẵn (ví dụ: D:\Phim).



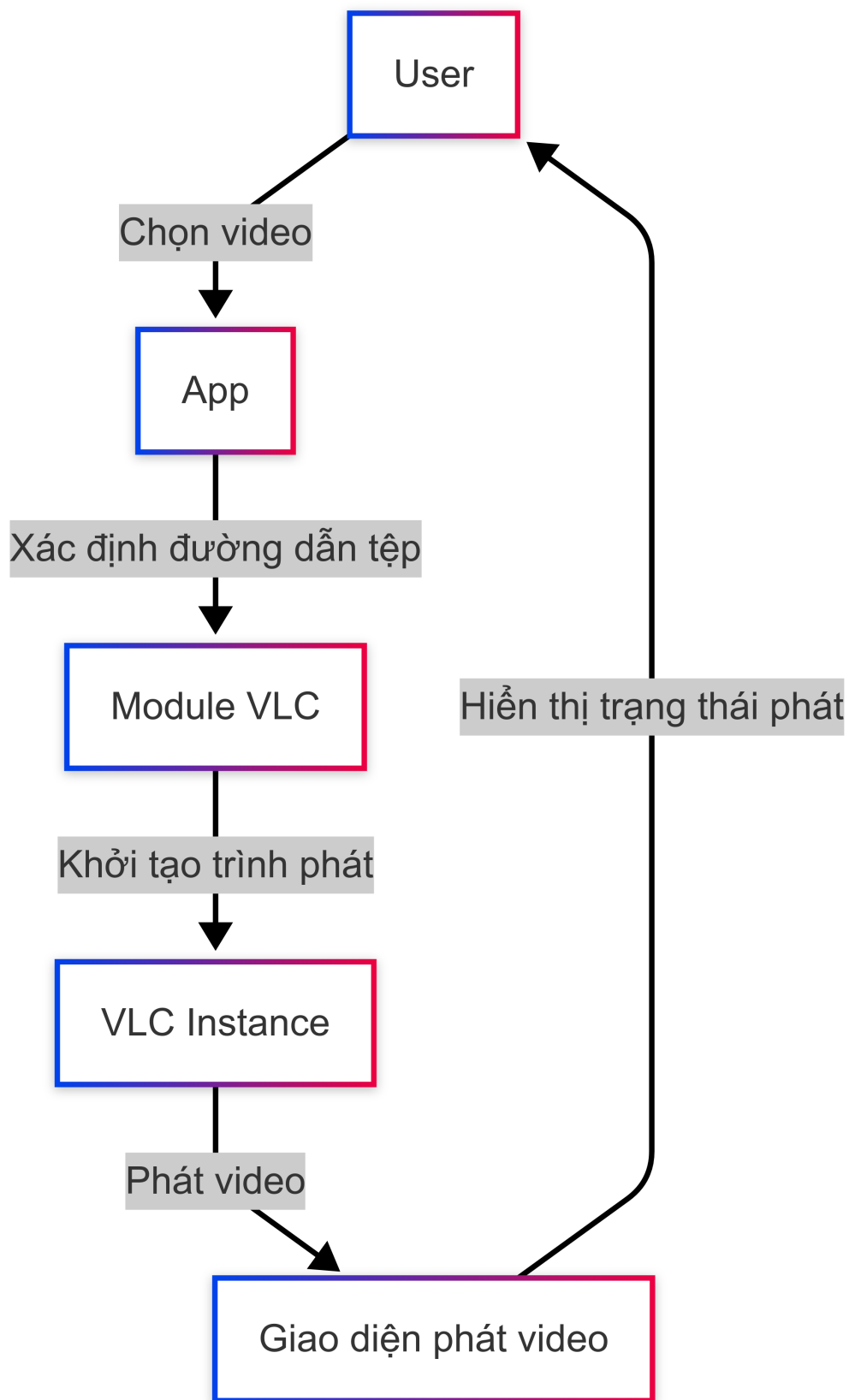
Hình 3.4 Quy trình hoạt động của qBittorrent trong ứng dụng

Hình 3.4 mô tả quy trình hoạt động của qBittorrent khi tích hợp vào ứng dụng.

3.5.5 *Phát video (Tích hợp VLC)*

Trình phát video VLC được tích hợp vào ứng dụng để cho phép người dùng xem phim trực tiếp từ các tệp đã tải xuống.

1. Người dùng chọn một bộ phim từ danh sách (đã được tải xuống) và nhấn nút "Play".
2. Ứng dụng xác định đường dẫn đến tệp video trên máy cục bộ.
3. Module VLC trong Python được sử dụng để khởi tạo một instance của trình phát và nhúng vào một widget trong giao diện PySide6.
4. Video được phát, và người dùng có thể sử dụng các chức năng điều khiển như:
 - Phát/Tạm dừng video.
 - Dừng hẳn video.
 - Tua nhanh/chậm video thông qua nút tua.
 - Điều chỉnh âm lượng.
5. Trạng thái phát (phần trăm video đã phát) cũng được hiển thị.



Hình 3.5 Quy trình hoạt động của VLC trong ứng dụng

Hình 3.5 minh họa cách trình phát VLC được tích hợp vào giao diện ứng dụng.

3.6 Tóm tắt Chương 3

Chương này đã trình bày chi tiết về thiết kế và triển khai hệ thống quản lý phim, bao gồm các thành phần chính và cách chúng phối hợp để tạo nên một ứng dụng hoàn chỉnh. Hệ thống được xây dựng dựa trên kiến trúc MVC (Model-View-Controller), đảm bảo tính tách biệt giữa dữ liệu, giao diện người dùng, và logic xử lý, giúp dễ dàng bảo trì và mở rộng trong tương lai.

Phần thiết kế cơ sở dữ liệu SQLite đã được trình bày với hai bảng chính: `accounts` để quản lý thông tin tài khoản người dùng và `movies` để lưu trữ thông tin chi tiết về phim. Cấu trúc bảng được thiết kế tối ưu, đảm bảo tính toàn vẹn dữ liệu và hỗ trợ tốt cho các chức năng tìm kiếm, hiển thị, và quản lý phim.

Giao diện người dùng được xây dựng bằng thư viện PySide6, mang lại trải nghiệm trực quan và thân thiện. Các thành phần giao diện như danh sách phim, chi tiết phim, và các nút chức năng được thiết kế hợp lý, giúp người dùng dễ dàng tương tác với ứng dụng. Chức năng phân trang đảm bảo hiệu suất hiển thị ngay cả khi số lượng phim trong cơ sở dữ liệu lớn.

Ứng dụng cũng tích hợp các dịch vụ và thư viện bên ngoài để mở rộng chức năng:

- **Thêm thông tin phim mới:** Tích hợp API OMDB để lấy thông tin chi tiết về phim từ nguồn trực tuyến và lưu vào cơ sở dữ liệu.
- **Tìm kiếm và quản lý phim:** Hệ thống hỗ trợ tìm kiếm phim trong cơ sở dữ liệu cục bộ đã tải xuống, đảm bảo kết quả trả về nhanh chóng và chính xác.
- **Trình phát video VLC:** Cho phép người dùng phát trực tiếp các tệp video đã tải xuống với các chức năng điều khiển như Play, Pause, Stop, tua nhanh/chậm, và điều chỉnh âm lượng.
- **Trình quản lý tải xuống qBittorrent:** Hỗ trợ tải phim từ các nguồn torrent thông qua giao tiếp với qBittorrent Web API, đảm bảo quá trình tải xuống diễn ra mượt mà và hiệu quả.

Các quy trình hoạt động chính như đăng nhập, thêm phim, tìm kiếm phim, tải phim và phát video đã được mô tả cụ thể, làm rõ luồng dữ liệu và tương tác giữa các thành phần trong hệ thống.

CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ

4.1 Kết quả chạy thử nghiệm

Quá trình chạy thử nghiệm ứng dụng được thực hiện trên môi trường Windows 11, với Python 3.1, PySide6, thư viện python-vlc và qbittorrent-api. Giả định rằng bảng ‘accounts’ trong cơ sở dữ liệu đã được khởi tạo trước với một số tài khoản hợp lệ (ví dụ: username "alice", password "password123", email "alice@example.com").

4.1.1 Kịch bản thử nghiệm

Các kịch bản thử nghiệm được xây dựng để kiểm tra các chức năng chính của ứng dụng quản lý phim. Dưới đây là các kịch bản cụ thể:

- **Chức năng quản lý tài khoản (Đăng nhập):**

- **Đăng nhập với tài khoản hợp lệ:**

- * *Kịch bản:* Người dùng nhập username "alice" và password "password123" vào form đăng nhập và nhấn nút "Login".
 - * *Kết quả mong đợi:* Đăng nhập thành công, người dùng được chuyển đến giao diện chính của ứng dụng.
 - * *Kết quả thực tế:* Hệ thống xác thực thông tin, cửa sổ đăng nhập đóng lại, giao diện chính hiển thị danh sách phim.
 - * *Đánh giá:* Thành công.

- **Đăng nhập với tài khoản không hợp lệ (sai mật khẩu):**

- * *Kịch bản:* Người dùng nhập username "alice" và password "wrongpassword" vào form đăng nhập.
 - * *Kết quả mong đợi:* Hệ thống thông báo lỗi "Invalid username or password".
 - * *Kết quả thực tế:* Hệ thống hiển thị cảnh báo "Invalid username or password". Người dùng vẫn ở lại màn hình đăng nhập.
 - * *Đánh giá:* Thành công.

- **Đăng nhập với tài khoản không tồn tại:**

- * *Kịch bản:* Người dùng nhập username "unknownuser" và password "any-password" vào form đăng nhập.
 - * *Kết quả mong đợi:* Hệ thống thông báo lỗi "Invalid username or password".
 - * *Kết quả thực tế:* Hệ thống hiển thị cảnh báo "Invalid username or password". Người dùng vẫn ở lại màn hình đăng nhập.

* *Đánh giá:* Thành công.

- **Chức năng tìm kiếm phim trong cơ sở dữ liệu cục bộ:**

- **Tìm kiếm phim có tồn tại trong CSDL:**

- * *Kịch bản:* Giả sử phim "Inception" đã được thêm vào CSDL. Người dùng nhập từ khóa "Inception" vào thanh tìm kiếm trên giao diện chính và nhấn nút "Search".

- * *Kết quả mong đợi:* Danh sách các phim có tiêu đề chứa từ khóa "Inception" được hiển thị.

- * *Kết quả thực tế:* Hệ thống truy vấn cơ sở dữ liệu và trả về kết quả chính xác. Phim "Inception" hiển thị với đầy đủ thông tin.

- * *Đánh giá:* Thành công.

- **Tìm kiếm phim không tồn tại trong CSDL:**

- * *Kịch bản:* Người dùng nhập từ khóa "NonExistentLocalMovieXYZ123" vào thanh tìm kiếm và nhấn nút "Search".

- * *Kết quả mong đợi:* Không có phim nào hiển thị hoặc thông báo "Không tìm thấy phim".

- * *Kết quả thực tế:* Hệ thống truy vấn cơ sở dữ liệu và không trả về kết quả nào. Giao diện không hiển thị phim hoặc hiển thị thông báo phù hợp.

- * *Đánh giá:* Thành công.

- **Hiển thị danh sách phim và chi tiết phim:**

- *Kịch bản:* Sau khi thêm một số phim ("Inception", "The Dark Knight") vào cơ sở dữ liệu, người dùng truy cập giao diện chính.

- *Kết quả mong đợi:* Danh sách các phim đã thêm được hiển thị dưới dạng lưới, bao gồm poster (nếu có) và tiêu đề. Có thể xem chi tiết khi nhấp vào nút "View More".

- *Kết quả thực tế:* Giao diện chính hiển thị chính xác các phim đã lưu trong bảng 'movies', mỗi phim là một mục trong lưới. Khi nhấn "View More" cho phim "Inception", một cửa sổ mới/khu vực chi tiết hiển thị đầy đủ thông tin. Chức năng phân trang hoạt động.

- *Đánh giá:* Thành công.

- **Chức năng tải phim (tích hợp qBittorrent):**

- *Kịch bản:* Người dùng chọn một phim từ danh sách, ví dụ "Inception". Nhấn nút "Search on ThePirateBay", tìm được magnet link/hash, sau đó nhập Hash Code hợp lệ vào trường tương ứng và nhấn "Download".

- *Kết quả mong đợi:* Ứng dụng gửi yêu cầu tải torrent đến qBittorrent Web API. Tập torrent bắt đầu được tải xuống trong qBittorrent client.
- *Kết quả thực tế:* Sau khi nhập Hash Code và ấn download, hệ thống gửi yêu cầu đến qBittorrent. qBittorrent tự khởi chạy (nếu chưa chạy và được cấu hình) và bắt đầu tải phim vào thư mục mặc định được chỉ định (ví dụ: "D:").
- *Đánh giá:* Thành công. (Lưu ý: việc tải xuống thành công phụ thuộc vào torrent và qBittorrent client).

• **Chức năng phát video (tích hợp VLC):**

- *Kịch bản:* Sau khi một bộ phim đã được tải xuống hoàn tất (ví dụ, tệp "Inception.mp4" trong "D:"), người dùng chọn phim đó trong ứng dụng và nhấn nút "Play".
- *Kết quả mong đợi:* Video được phát trong trình phát VLC nhúng. Các nút điều khiển hoạt động.
- *Kết quả thực tế:* Tệp video được chọn. Trình phát VLC nhúng khởi chạy và phát video thành công. Các chức năng Play/Pause, Stop, Seek, Âm lượng, Trạng thái phát hoạt động chính xác.
- *Đánh giá:* Thành công.

Bảng 4.1 Bảng tổng hợp kết quả chạy thử nghiệm

Chức năng	Kết quả	Ghi chú chi tiết
Đăng nhập	Thành công	Xác thực thành công với tài khoản hợp lệ từ CSDL, từ chối với tài khoản không hợp lệ/không tồn tại.
Tìm kiếm phim (CSDL cục bộ)	Thành công	Tìm kiếm phim trong cơ sở dữ liệu hoạt động chính xác. Trả về kết quả đúng hoặc thông báo không tìm thấy phim.
Hiển thị danh sách phim	Thành công	Phim được hiển thị dạng lưới, thông tin chi tiết hiển thị đầy đủ khi yêu cầu. Phân trang hoạt động.
Tải xuống phim (qBittorrent)	Thành công	Gửi yêu cầu tải torrent đến qBittorrent Web API thành công, qBittorrent xử lý tải xuống.
Phát video (VLC)	Thành công	Phát video từ tệp cục bộ mượt mà, các chức năng điều khiển hoạt động tốt.

Nhìn chung, tất cả các chức năng cốt lõi của ứng dụng đều hoạt động ổn định và đúng như mong đợi trong các kịch bản thử nghiệm cơ bản. Không có lỗi nghiêm trọng nào được phát hiện trong quá trình này. Các tương tác với API bên ngoài (OMDB, qBittorrent) và thư viện nhúng (VLC) đều diễn ra suôn sẻ.

4.2 Đánh giá hiệu suất

Ứng dụng cho thấy hiệu suất tốt trong các tác vụ thông thường. Thời gian khởi động ứng dụng nhanh chóng, dưới 3 giây trên cấu hình thử nghiệm.

- **Tải danh sách phim:** Với cơ sở dữ liệu chứa khoảng 100 bộ phim, danh sách phim được tải và hiển thị trong vòng 10 - 15 giây. Chức năng phân trang giúp duy trì hiệu suất khi số lượng phim tăng lên.
- **Tìm kiếm phim (CSDL cục bộ):** Thời gian tìm kiếm phim trong cơ sở dữ liệu cục bộ chấp nhận được, dưới 2 giây cho các truy vấn đơn giản.
- **Tương tác API:**
 - **OMDB API:** Thời gian phản hồi từ OMDB API phụ thuộc vào tốc độ mạng và tải của máy chủ OMDB, thường dao động từ 1 - 3 phút cho việc fetch phim về CSDL.
 - **qBittorrent API:** Việc gửi lệnh tải xuống đến qBittorrent API diễn ra nhanh chóng, dưới 10 giây. Hiệu suất tải xuống thực tế phụ thuộc vào qBittorrent client và nguồn torrent.
- **Phát video (VLC):** Việc khởi chạy trình phát VLC và bắt đầu phát video diễn ra mượt mà.

Độ ổn định của hệ thống được đánh giá cao trong quá trình thử nghiệm, không xảy ra treo ứng dụng hay lỗi nghiêm trọng.

4.3 Hạn chế

Mặc dù hệ thống hoạt động tốt và đáp ứng các yêu cầu chính, vẫn tồn tại một số hạn chế:

- **Phụ thuộc vào phần mềm bên thứ ba:** Chức năng tải phim yêu cầu qBittorrent phải được cài đặt và cấu hình đúng cách (bao gồm cả Web API). Chức năng phát video yêu cầu thư viện VLC được cài đặt. Nếu các phần mềm này không có hoặc gặp sự cố, các chức năng tương ứng sẽ không hoạt động.

- **Giao diện người dùng:** Giao diện hiện tại được xây dựng bằng PySide6 với các widget cơ bản. Mặc dù chức năng đầy đủ, thiết kế có thể chưa thực sự hiện đại và tùy biến cao về mặt thẩm mỹ.
- **Quản lý lỗi:** Mặc dù các lỗi cơ bản được xử lý (ví dụ: đăng nhập sai, không tìm thấy phim), hệ thống có thể được cải thiện thêm về mặt thông báo lỗi chi tiết hơn cho người dùng và cơ chế ghi log lỗi cho nhà phát triển.
- **Bảo mật mật khẩu:** Mật khẩu hiện đang được lưu dưới dạng văn bản gốc trong cơ sở dữ liệu. Đây là một rủi ro bảo mật nếu cơ sở dữ liệu bị xâm phạm.
- **Không có chức năng quản lý tải xuống nâng cao:** Ứng dụng chỉ gửi lệnh tải xuống đến qBittorrent và theo dõi tiến trình 1 giây 1 lần chứ không có giao diện theo dõi thời gian thực.
- **Giới hạn API OMDB:** Phiên bản miễn phí của OMDB API có giới hạn về số lượng yêu cầu mỗi ngày. Với lượng người dùng lớn hoặc tần suất sử dụng cao, có thể cần nâng cấp lên phiên bản trả phí.

4.4 Đề xuất cải tiến

Dựa trên các hạn chế đã nêu, một số hướng phát triển và cải tiến có thể được thực hiện trong tương lai:

- **Kiểm tra và hướng dẫn cài đặt phụ thuộc:** Thêm cơ chế kiểm tra sự tồn tại của qBittorrent và VLC khi khởi động. Nếu thiếu, cung cấp hướng dẫn hoặc liên kết để người dùng cài đặt.
- **Cải thiện giao diện người dùng:** Sử dụng các stylesheet tùy chỉnh (QSS) của PySide6 hoặc các thư viện theme để làm cho giao diện hấp dẫn hơn. Cân nhắc sử dụng các layout và widget nâng cao hơn.
- **Mã hóa mật khẩu:** Triển khai cơ chế mã hóa mật khẩu (ví dụ: hashing với salt) trước khi lưu vào cơ sở dữ liệu để tăng cường bảo mật.
- **Tích hợp sâu hơn với qBittorrent:** Sử dụng qBittorrent API để theo dõi tiến trình tải xuống, hiển thị thông tin chi tiết về torrent (tốc độ, peers, seeds) và cho phép quản lý cơ bản các tác vụ tải (tạm dừng, tiếp tục, xóa) ngay từ trong ứng dụng.
- **Quản lý API Key OMDB:** Cho phép người dùng tự cấu hình API key OMDB của họ để tránh giới hạn của API key dùng chung.
- **Chức năng quản lý người dùng nâng cao:** Thêm chức năng đăng ký tài khoản, quên mật khẩu, phân quyền người dùng (ví dụ: admin, user thường).

- **Đa ngôn ngữ:** Hỗ trợ nhiều ngôn ngữ cho giao diện người dùng.
- **Tùy chỉnh thư mục lưu phim:** Cho phép người dùng chọn thư mục lưu phim thay vì sử dụng đường dẫn cố định.
- **Tạo bản portable hoặc trình cài đặt:** Giúp việc triển khai và sử dụng ứng dụng dễ dàng hơn cho người dùng cuối.

4.5 Tóm tắt Chương 4

Chương này đã trình bày chi tiết kết quả thử nghiệm của hệ thống quản lý phim và đưa ra những đánh giá khách quan về hiệu suất cũng như các mặt còn hạn chế. Quá trình thử nghiệm được thực hiện với các kịch bản cụ thể, kiểm tra toàn diện các chức năng chính bao gồm đăng nhập, thêm phim mới từ OMDb, tìm kiếm phim trong cơ sở dữ liệu cục bộ, hiển thị danh sách và chi tiết phim, tải phim qua qBittorrent và phát video bằng VLC.

Kết quả cho thấy tất cả các chức năng được kiểm tra đều hoạt động thành công và ổn định, đáp ứng đúng các yêu cầu đã đặt ra trong giai đoạn thiết kế. Hiệu suất của ứng dụng được đánh giá là tốt, với thời gian phản hồi nhanh và mức sử dụng tài nguyên hợp lý trong các tác vụ thông thường.

Bên cạnh những kết quả tích cực, chương cũng đã chỉ ra một số hạn chế của hệ thống, chủ yếu liên quan đến sự phụ thuộc vào các phần mềm bên thứ ba, vấn đề bảo mật mật khẩu cơ bản, và một số khía cạnh của giao diện người dùng có thể được cải thiện.

Cuối cùng, dựa trên những đánh giá và hạn chế đó, một số đề xuất cải tiến và hướng phát triển tiềm năng cho hệ thống đã được đưa ra. Các đề xuất này bao gồm việc tăng cường bảo mật, cải thiện trải nghiệm người dùng, tích hợp sâu hơn với các dịch vụ bên ngoài, và mở rộng các tính năng quản lý. Những cải tiến này nếu được thực hiện sẽ giúp nâng cao chất lượng và tính hữu dụng của ứng dụng quản lý phim trong tương lai.

KẾT LUẬN

Trong đồ án này, em đã trình bày chi tiết quá trình thiết kế, triển khai và thử nghiệm hệ thống quản lý phim. Hệ thống được xây dựng với mục tiêu cung cấp một công cụ hỗ trợ người dùng quản lý, tìm kiếm, tải xuống và phát phim một cách hiệu quả. Các nội dung chính đã được giải quyết bao gồm:

- Thiết kế hệ thống dựa trên kiến trúc MVC (Model-View-Controller), đảm bảo tính tách biệt giữa dữ liệu, giao diện người dùng và logic xử lý.
- Xây dựng cơ sở dữ liệu SQLite với hai bảng chính: `accounts` để quản lý thông tin tài khoản và `movies` để lưu trữ thông tin chi tiết về phim.
- Phát triển giao diện người dùng bằng thư viện PySide6, mang lại trải nghiệm trực quan và thân thiện.
- Tích hợp các dịch vụ bên ngoài như API OMDB để lấy thông tin phim, VLC để phát video, và qBittorrent để tải phim từ các nguồn torrent.
- Thực hiện các kịch bản thử nghiệm chi tiết, đảm bảo hệ thống hoạt động ổn định và đáp ứng đầy đủ các yêu cầu chức năng đã đề ra.

So sánh với mục tiêu ban đầu, hệ thống đã hoàn thành các chức năng cốt lõi và đáp ứng tốt các yêu cầu đề ra. Hệ thống không chỉ cung cấp một công cụ mạnh mẽ để quản lý phim mà còn mở ra nhiều tiềm năng phát triển trong tương lai. Những đóng góp của hệ thống bao gồm:

- Cung cấp một giải pháp quản lý phim hiệu quả, dễ sử dụng và có tính ứng dụng cao.
- Tích hợp các công cụ mạnh mẽ như VLC và qBittorrent để mở rộng chức năng.
- Xây dựng một nền tảng linh hoạt, dễ dàng mở rộng và cải tiến.

Nhìn chung, hệ thống quản lý phim đã hoàn thành các mục tiêu đề ra, cung cấp một nền tảng mạnh mẽ và linh hoạt để quản lý, tìm kiếm, tải xuống và phát phim. Với kiến trúc hiện tại, hệ thống không chỉ đáp ứng tốt các yêu cầu hiện tại mà còn sẵn sàng cho các giai đoạn phát triển tiếp theo.

TÀI LIỆU THAM KHẢO

- [1] Qt for Python Development Team. (2025) Qt for Python Documentation. [Online]. Available: <https://doc.qt.io/qtforpython-6/index.html>
- [2] FreeCodeCamp.org. (2025) earn Python GUI Development for Desktop – PySide6 and Qt Tutorial. [Online]. Available: https://youtu.be/Z1N9JzNax2k?si=911aD5_gOQEoGBw6

PHỤ LỤC

Phụ lục A: Link GitHub

Toàn bộ mã nguồn của hệ thống quản lý phim có thể được tham khảo tại:

- https://github.com/ngaiTu29s1/project1_videodownloader_and_player