BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

A blue and white logo

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: Xây dựng ứng dụng đặt vé xem phim**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Sinh Viên | Họ và Tên | Ngày Sinh | Lớp |
| 2151173808 | Lê Ngọc Khôi Nguyên | 08/09/2003 | 63KTPM1 |
| 2151173806 | Nguyễn Chính Nghĩa | 02/07/2003 | 63KTPM1 |
| 2151170539 | Phạm Minh Quang | 23/01/2003 | 63KTPM1 |

### 

**Hà Nội, năm 2024**

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

A blue and white logo

Description automatically generated

**BÀI TẬP LỚN**

**HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG**

**ĐỀ TÀI: Xây dựng ứng dụng đặt vé xem phim**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 2151173808 | Lê Ngọc Khôi Nguyên | 08/09/2003 |  |  |
| 2151173806 | Nguyễn Chính Nghĩa | 02/07/2003 |  |  |
| 2151170539 | Phạm Minh Quang | 23/01/2003 |  |  |

CÁN BỘ CHẤM THI 1 CÁN BỘ CHẤM THI 2

**Hà Nội, năm 2024**

**LỜI NÓI ĐẦU**

**Chào mừng bạn đến với Thủy Lợi Cinema – nơi trải nghiệm điện ảnh đỉnh cao!**

Với niềm đam mê mang lại trải nghiệm giải trí tuyệt vời cho mọi khán giả, Thủy Lợi Cinema tự hào giới thiệu ứng dụng đặt vé trực tuyến tiện ích, đáp ứng mọi nhu cầu xem phim của bạn chỉ trong vài bước đơn giản. Dù bạn là người yêu thích các bộ phim bom tấn Hollywood, say mê phim nghệ thuật, hay đơn giản là muốn dành thời gian thư giãn cùng gia đình và bạn bè, Thủy Lợi Cinema luôn sẵn sàng mang đến cho bạn những giây phút đáng nhớ nhất.

**Tại sao chọn Thủy Lợi Cinema?**

* **Đặt vé trực tuyến dễ dàng**: Với giao diện thân thiện và thao tác đặt vé nhanh chóng, bạn chỉ cần vài cú chạm là có ngay tấm vé xem phim yêu thích.
* **Lịch chiếu phim cập nhật liên tục**: Ứng dụng cung cấp lịch chiếu chi tiết cho các bộ phim đang hot, giúp bạn dễ dàng sắp xếp thời gian và lựa chọn suất chiếu phù hợp.
* **Thông tin phim chi tiết**: Mỗi bộ phim đều đi kèm trailer, tóm tắt nội dung, thông tin diễn viên và đánh giá từ người xem, giúp bạn có cái nhìn toàn diện trước khi quyết định.
* **Ưu đãi độc quyền**: Đăng ký làm thành viên để nhận ngay nhiều ưu đãi hấp dẫn, tích điểm đổi quà và cập nhật những khuyến mãi mới nhất chỉ dành cho khách hàng thân thiết của Thủy Lợi Cinema.

Ứng dụng Thủy Lợi Cinema là người bạn đồng hành lý tưởng cho mọi tín đồ điện ảnh. Chúng tôi không ngừng nỗ lực để nâng cao chất lượng dịch vụ, mang đến trải nghiệm xem phim phong phú, tiện lợi và hiện đại nhất.

**Hãy tải ngay ứng dụng Thủy Lợi Cinema và bắt đầu hành trình điện ảnh của bạn!**

**MỤC LỤC**

Catalog

[MỤC LỤC HÌNH ẢNH 5](#_Toc16137)

[CHƯƠNG 1. MÔ TẢ BÀI TOÁN 6](#_Toc9916)

[1.1. Giới thiệu 6](#_Toc19625)

[1.2. Chức năng chính 6](#_Toc30850)

[1.3. Yêu cầu phi chức năng 7](#_Toc15881)

[CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG 8](#_Toc27761)

[2.1. Phân tích yêu cầu: 8](#_Toc5551)

[2.2. Thiết kế hệ thống: 8](#_Toc10304)

[2.3. Triển khai: 26](#_Toc669)

[2.4. Vận hành và bảo trì: 27](#_Toc18054)

[CHƯƠNG 3. KẾT QUẢ THỰC HIỆN 28](#_Toc9793)

[3.1. Công nghệ đã sử dụng 28](#_Toc22369)

[3.2. Tiến độ thực hiện 28](#_Toc6909)

[3.3. Hình ảnh sản phẩm 30](#_Toc13156)

[1. Giao diện trang bắt đầu 30](#_Toc18413)

[2. Giao diện trang đăng kí tài khoản 31](#_Toc6853)

[3. Giao diện trang quản trị 32](#_Toc13264)

[4. Giao diện trang quản lí tài khoản 33](#_Toc11043)

[5. Giao diện trang quản lí phim 34](#_Toc22591)

[6. Giao diện trang thêm phim 35](#_Toc18527)

[7. Giao diện trang chủ của app 36](#_Toc16278)

[8. Giao diện trang tìm kiếm 37](#_Toc14684)

[9. Giao diện trang phim chi tiết 38](#_Toc19263)

[10. Giao diện trang đặt vé 39](#_Toc32674)

[11. Giao diện trang thanh toán 40](#_Toc11587)

[12. Giao diện trang thông tin vé 41](#_Toc26936)

[KẾT LUẬN 42](#_Toc14140)

[QUY ĐỊNH TRÌNH BÀY TRONG BÁO CÁO CÀI TẬP LỚN 43](#_Toc5685)

[*Cách đánh dấu câu:* 43](#_Toc29228)

[Hướng dẫn xếp tài liệu tham khảo 44](#_Toc14285)

**CHƯƠNG 1. MÔ TẢ BÀI TOÁN**

**1.1. Giới thiệu**

**1.2. Chức năng chính**

Ứng dụng Đặt vé xem phim có các chức năng sau:

* Đăng ký-Đăng nhập: Người dung có thể tạo tài khoản và đăng nhập bằng tài khoản đó để truy cập vào ứng dụng
* Hiển thị danh sách phim: Liệt kê tất cả phim hiện có
* Tìm kiếm phim: Cho phép người dùng tìm kiếm phim hiện có dựa trên từ khóa
* Đặt vé xem phim: Cho phép người dùng chọn phim, ngày giờ và vị trí chỗ ngồi và thanh toán để đặt vé xem phim
* Hiển thị thông tin đặt vé : Cho phép người dùng xem lại thông tin đặt vé sau khi đã đặt xong.
* Quản lý danh sách phim: Cho phép quản trị viên có thể thêm, sửa, xóa phim và các thông tin liên quan
* Quản lý tài khoản: Cho phép quản trị viên them, xóa tài khoản của người dùng
* Lưu trữ dữ liệu: Ứng dụng dung sharedpreferences với sqlite để lưu trữ dữ liệu.

Sharedpreferences để lưu trữ thông tin đăng nhập, còn dữ liệu app dung sqlite là chủ yếu.

**1.3. Yêu cầu phi chức năng**

* Dễ sử dụng: Giao diện cần rõ ràng, dễ hiểu và dễ sử dụng.
* Hiệu năng: Ứng dụng cần hoạt động nhanh chóng và hiệu quả
* Độ tin cậy: Dữ liệu cần được lưu trữ an toàn và không bị mất mát.

**CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG**

**2.1. Phân tích yêu cầu:**

**Xác định người dùng:**

* Người dùng cuối là bất kỳ ai muốn sử dụng ứng dụng để đặt vé xem phim
* Người dùng có thể có các mức độ am hiểu về máy tính khác nhau, do đó ứng dụng cần dễ sử dụng và thân thiện với người dùng.

**Thu thập yêu cầu:**

* Dựa trên mô tả bài toán, ta đã xác định được các chức năng chính và yêu cầu phi chức năng của ứng dụng.

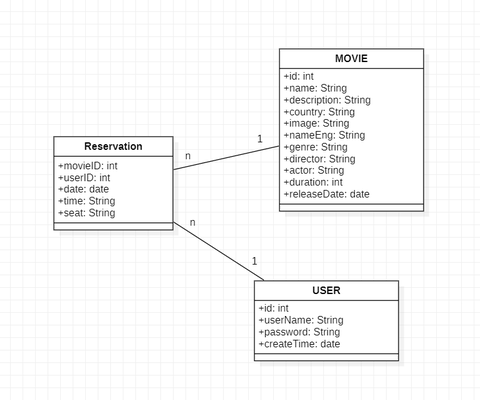
**Phân tích yêu cầu:**

* **Đăng ký:** Yêu cầu người dùng nhập tên tài khoản(bắt buộc), mật khẩu (bắt buộc), và nhập lại mật khẩu(Bắt buộc)
* **Đăng nhập:** Yêu cầu người dùng nhập tên tài khoản(bắt buộc), mật khẩu (bắt buộc) đã đăng ký trên hệ thống
* **Tìm kiếm phim:** Người dùng nhập từ khóa thông qua thanh tìm kiếm và sẽ nhận được kết quả là các phim có chứa từ khóa đó
* **Đặt vé:** Người dung chọn phim, ngày giờ và vị trí chỗ ngồi và bấm xác nhận để đặt v
* **Quản lý danh sách phim:** Quản trị viên phải truy cập vào tài khoản quản trị mới có thể sử dụng chức năng này. Quản trị viên có thể thêm phim vào danh sách bằng cách nhập tên phim (Bắt buộc) và các thông tin khác (Tùy chọn).Quản trị viên có thể xóa phim bằng cách bấm vào icon bên cạnh tên phim
* **Quản lý tài khoản:** Quản trị viên phải truy cập vào tài khoản quản trị mới có thể sử dụng chức năng này. Quản trị viên có thể thêm tài khoản vào danh sách bằng cách nhập tên tài khoản (Bắt buộc) và mật khẩu (Bắt buộc).Quản trị viên có thể xóa tài khoản bằng cách bấm vào icon bên cạnh tên tài khoản

**2.2. Thiết kế hệ thống:**

**Biểu đồ lớp cho mô hình miền:**

* Tập trung vào các thực thể nghiệp vụ và mối quan hệ giữa chúng.
* Thường không chứa các lớp liên quan đến giao diện người dùng, cơ sở dữ liệu, hay các chi tiết kỹ thuật khác.



**Biểu đồ lớp cho MVC:**

* Mô tả kiến trúc của ứng dụng, bao gồm các lớp trong Model, View và Controller.
* Thể hiện mối quan hệ giữa các lớp trong MVC và cách chúng tương tác với nhau.

**Thiết kế kiến trúc (MVC):**

* **Model:**
  + **Movie**: Lớp này đại diện cho một bộ phim, chứa các thuộc tính thông tin phim.

Các thuộc tính chính:

* id: Mã định danh duy nhất của phim.
* name: Tên phim.
* nameEng: Tên phim không dấu, giúp việc tìm kiếm dễ dàng hơn.
* image: Đường dẫn hình ảnh đại diện của phim.
* description: Mô tả nội dung phim.
* country: Quốc gia sản xuất phim.
* genre: Thể loại phim (hành động, tình cảm, ...).
* director: Đạo diễn của phim.
* actor: Danh sách diễn viên tham gia.
* duration: Thời lượng phim (phút).
* releaseDate: Ngày phát hành phim.
  + **Reservation**: Lớp này đại diện cho một đặt chỗ (vé) xem phim, lưu trữ thông tin về phim, người dùng và chi tiết đặt chỗ.

Các thuộc tính chính:

* id: Mã định danh duy nhất cho mỗi đặt chỗ.
* movieId: Mã định danh của phim được đặt.
* userId: Mã định danh của người dùng đặt vé.
* date: Ngày xem phim.
* time: Giờ chiếu phim.
* seat: Danh sách ghế ngồi được đặt.
  + **User**: Lớp này đại diện cho một người dùng trong hệ thống, lưu trữ thông tin đăng nhập và thời điểm tạo tài khoản.

Các thuộc tính chính:

* id: Mã định danh duy nhất của người dùng.
* username: Tên đăng nhập của người dùng.
* password: Mật khẩu của người dùng.
* time: Thời điểm tạo tài khoản, lưu dưới dạng chuỗi ngày tháng.
* **View:**
* activity\_add\_user: Hiển thị màn hình quản trị viên thêm tài khoản
* activity\_admin: Màn hình chính của quản trị viên
* activity\_begin: Màn hình chính trước khi đăng nhập
* activity\_edit\_movie: Màn hình quản trị viên nhập thông tin phim để thêm
* activity\_main: Màn hình chính sau khi đăng nhập
* activity\_manage\_movie: Hiển thị danh sách phim hiện có đối với quản trị viên
* activity\_manage\_user: Hiển thị danh sách tài khoản hiện có đối với quản trị viên
* activity\_movie: Hiển thị thông tin phim sau khi người dung bấm vào
* activity\_payment: Màn hình thanh toán
* activity\_search: Thanh tìm kiếm trên màn hình chính
* activity\_seat: Màn hình chọn ngày giờ, chỗ ngồi để đặt vé
* activity\_sign\_in: Màn hình đăng nhập tài khoản
* activity\_sign\_up: Màn hình đăng ký tài khoản
* activity\_view\_ticket: Màn hình hiển thị lại thông tin vé sau khi đặt thành công
* image\_slider\_layout\_item: Tự động trượt qua lại giữa các phim trên màn hình chính
* **Controller:**
* BeginActivity: Lớp này chịu trách nhiệm cho màn hình khởi động của ứng dụng. Mục đích chính của lớp này là kiểm tra xem người dùng đã mở ứng dụng lần đầu tiên hay chưa và điều hướng đến màn hình đăng nhập nếu ứng dụng đã được mở trước đó
* MainActivity: Lớp này chịu trách nhiệm cho giao diện chính của ứng dụng, hiển thị các danh mục phim nổi bật và cung cấp chức năng tìm kiếm phim.
* MovieActivity: Lớp này hiển thị thông tin chi tiết về một bộ phim và cung cấp tùy chọn để người dùng đặt vé.
* PaymentActivity: Lớp này quản lý màn hình thanh toán, hiển thị thông tin chi tiết của việc đặt vé bao gồm phim, ngày, giờ, ghế ngồi, người đặt, và tổng số tiền.
* SearchActivity: Lớp này quản lý màn hình tìm kiếm phim, cho phép người dùng nhập tên phim và xem danh sách kết quả tìm kiếm.
* SearchAdapter: Lớp này là một RecyclerView.Adapter, quản lý hiển thị kết quả tìm kiếm phim trong RecyclerView của SearchActivity.
* SeatActivity: Lớp này quản lý màn hình chọn ghế ngồi, cung cấp cho người dùng khả năng chọn ngày, giờ chiếu, và ghế ngồi trước khi tiến hành thanh toán.
* SeatAdapter: Lớp này là một RecyclerView.Adapter giúp hiển thị danh sách ghế ngồi trong RecyclerView của SeatActivity, cho phép người dùng chọn ghế.
* SeatSelection: Lớp này đại diện cho một ghế ngồi trong rạp chiếu phim, lưu trữ thông tin về trạng thái của ghế (đã chọn, đã đặt).
* SignInActivity: Là một lớp AppCompatActivity chịu trách nhiệm cho việc xử lý màn hình đăng nhập của ứng dụng. Nó cho phép người dùng nhập tên đăng nhập và mật khẩu, đồng thời thực hiện kiểm tra và xử lý xác thực người dùng.
* SignUpActivity: Là một lớp AppCompatActivity giúp người dùng đăng ký tài khoản mới trong ứng dụng. Nó cho phép người dùng nhập tên đăng nhập, mật khẩu, và xác nhận mật khẩu.
* SliderMovieAdapter: Là một lớp SliderViewAdapter giúp hiển thị danh sách phim trong một slider (trình chiếu hình ảnh). Nó cho phép người dùng cuộn qua các hình ảnh phim và nhấp vào từng hình để xem chi tiết phim.
* TimeAdapter: là một lớp RecyclerView.Adapter chịu trách nhiệm hiển thị danh sách thời gian có sẵn cho người dùng chọn. Mỗi mục trong danh sách được hiển thị dưới dạng một thẻ (CardView) có chứa thời gian.
* TimeSelection: đại diện cho một lựa chọn thời gian trong ứng dụng. Nó lưu trữ thông tin về thời gian và trạng thái đã chọn hay chưa của lựa chọn đó.
* ViewTicketActivity: Chịu trách nhiệm hiển thị thông tin chi tiết của vé đã đặt, bao gồm tên phim, ngày, giờ, ghế ngồi, người đặt và tổng số tiền. Lớp này cũng tạo mã QR chứa thông tin vé.

|  |
| --- |
| **Dựa trên kiến trúc MVC đã chọn, ta có thể xác định các lớp sau:**   * **Lớp Users:**   + **Thuộc tính:** id, username, password, time   + **Phương thức:** * Constructor: Khởi tạo đối tượng User với các thông tin cần thiết để đại diện cho người dùng. * Getters/Setters: Cung cấp các phương thức truy xuất và cập nhật giá trị của các thuộc tính. * **Lớp Movie:**   + **Thuộc tính:** id, name, nameEng, image, description, country, genre, director, actor, duration, releaseDate   + **Phương thức:** * Constructor: Khởi tạo đối tượng Movie với các thuộc tính cần thiết để đại diện cho một bộ phim. * toString(): Trả về chuỗi mô tả đối tượng Movie. * Getters/Setters: Cung cấp các phương thức truy xuất và cập nhật giá trị của các thuộc tính trong lớp. * **Lớp Reservation:**   + **Thuộc tính:** id, movieId, userId, date, time,seat   + **Phương thức:** * Constructor: Khởi tạo đối tượng Reservation với các thuộc tính cần thiết để đại diện cho một đặt chỗ. * Getters/Setters: Cung cấp các phương thức truy xuất và cập nhật giá trị của các thuộc tính trong lớp. * **Lớp TaskView:**   + **Phương thức:** * displayMenu(): hiển thị menu chính. * displayTasks: hiển thị danh sách công việc. * getInput: lấy input từ người dùng. * showMessage: hiển thị thông báo. * **Lớp BeginActivity:**   + **Thuộc tính:** * buttonGetStarted: Nút bắt đầu cho người dùng khi mở ứng dụng lần đầu tiên.   + **Phương thức:** * onCreate(Bundle savedInstanceState): Phương thức này thiết lập giao diện cho màn hình bắt đầu, kiểm tra trạng thái mở ứng dụng lần đầu của người dùng. Nếu người dùng đã mở ứng dụng trước đó, nút bắt đầu sẽ ẩn đi, và sau một khoảng thời gian ngắn, người dùng sẽ được tự động chuyển đến màn hình đăng nhập. * moveToSignIn(): Điều hướng người dùng đến màn hình đăng nhập. * **Lớp MovieActivity:**   + **Thuộc tính:** * filmImage: Ảnh đại diện của phim. * description, country, genre, director, actor, duration, release: Thông tin chi tiết về phim như mô tả, quốc gia, thể loại, đạo diễn, diễn viên, thời lượng và ngày phát hành. * buyTicket: Nút để đặt vé cho phim.   + **Phương thức:** * onCreate(Bundle savedInstanceState): Khởi tạo giao diện chi tiết của phim, truy xuất dữ liệu phim từ cơ sở dữ liệu dựa trên id phim được truyền từ màn hình trước. Nếu id không hợp lệ, người dùng sẽ được thông báo và quay lại. Phương thức này cũng thiết lập các giá trị chi tiết phim vào các thành phần tương ứng. * buyTicket.setOnClickListener(...): Sự kiện nhấn vào nút buyTicket, điều hướng người dùng đến màn hình SeatActivity để chọn ghế và đặt vé. * **Lớp MainActivity:**   + **Thuộc tính:** * sliderViewRandom, sliderViewTop, sliderViewUpcoming: Các SliderView hiển thị danh sách phim ngẫu nhiên, phim hàng đầu, và phim sắp chiếu. * itemClickListener: Một ItemClickListener được sử dụng để bắt sự kiện khi người dùng nhấn vào một phim cụ thể..   + **Phương thức:** * onCreate(Bundle savedInstanceState): Khởi tạo giao diện chính của ứng dụng, thiết lập các SliderView với dữ liệu phim được lấy từ cơ sở dữ liệu. Cung cấp tính năng tìm kiếm khi người dùng nhấn vào ô tìm kiếm. * setupSliderView(SliderView sliderView): Cấu hình hoạt ảnh và hiệu ứng cho SliderView, bao gồm kiểu chuyển động, màu sắc của chỉ báo, và thời gian chuyển đổi giữa các phim. * itemClickListener.onMovieClick: Sự kiện nhấn vào một phim, chuyển người dùng đến màn hình MovieActivity để xem chi tiết về phim đó. * **Lớp PaymentActivity:**   + **Thuộc tính:** * film, date, time, seats, booker, total: Các TextView hiển thị thông tin chi tiết về phim, ngày, giờ chiếu, số ghế, tên người đặt và tổng chi phí. * confirm: Nút xác nhận thanh toán. * databaseHelper: Đối tượng truy cập cơ sở dữ liệu, giúp lấy thông tin phim.   + **Phương thức:** * onCreate(Bundle savedInstanceState): Thiết lập giao diện thanh toán, lấy dữ liệu từ Intent như phim, ngày, giờ, ghế ngồi, và tổng chi phí để hiển thị cho người dùng. Nếu movie không tồn tại, ứng dụng sẽ quay lại màn hình trước đó. Phương thức cũng lấy tên người đặt từ thông tin người dùng. * confirm.setOnClickListener: Sự kiện nhấn vào nút confirm, xác nhận giao dịch thành công và kết thúc hoạt động, trả về kết quả RESULT\_OK. * **Lớp SearchActivity:**   + **Thuộc tính:** * search: Ô nhập để người dùng nhập từ khóa tìm kiếm phim. * searchRecyclerView: RecyclerView để hiển thị kết quả tìm kiếm dưới dạng danh sách phim. * adapter: Đối tượng SearchAdapter để quản lý dữ liệu và hiển thị danh sách kết quả tìm kiếm. * databaseHelper: Đối tượng giúp truy xuất cơ sở dữ liệu, hỗ trợ tìm kiếm phim theo tên.   + **Phương thức:** * onCreate: Khởi tạo giao diện tìm kiếm, bao gồm EditText để người dùng nhập từ khóa và RecyclerView để hiển thị kết quả. Phương thức này cũng thiết lập bàn phím xuất hiện tự động khi vào màn hình. * search.addTextChangedListener: Theo dõi các thay đổi trong ô tìm kiếm, mỗi khi người dùng nhập ký tự mới, phương thức này sẽ tìm kiếm phim theo tên từ cơ sở dữ liệu và cập nhật kết quả trên RecyclerView. Nếu không có kết quả nào, RecyclerView sẽ hiển thị trống. * **Lớp SearchAdapter:**   + **Thuộc tính:** * context: Ngữ cảnh của ứng dụng, được sử dụng để truy cập tài nguyên. * listener: Đối tượng MovieClickListener, lắng nghe sự kiện khi người dùng nhấn vào một mục phim trong danh sách. * list: Danh sách phim được tìm kiếm, hiển thị trên RecyclerView.   + **Phương thức:** * renewItems(List<Movie> list): Cập nhật danh sách phim tìm kiếm và làm mới RecyclerView khi có kết quả mới. * onCreateViewHolder(@NonNull ViewGroup parent, int viewType): Tạo và trả về một ViewHolder cho từng mục trong danh sách tìm kiếm. * onBindViewHolder(@NonNull SeatAdapterVH holder, int position): Liên kết dữ liệu phim với ViewHolder, thiết lập hình ảnh và tên phim cho từng mục, đồng thời cài đặt sự kiện nhấn vào phim để chuyển đến trang chi tiết phim. * getItemCount(): Trả về số lượng phim trong danh sách tìm kiếm. * **Lớp SeatActivity:**   + **Thuộc tính:** * timeAdapter, seatAdapter: Adapter để hiển thị danh sách giờ chiếu và danh sách ghế ngồi có thể chọn. * timeRecyclerView, seatRecyclerView: RecyclerView để hiển thị danh sách giờ chiếu và ghế ngồi. * total: Hiển thị tổng chi phí của ghế đã chọn. * buyTicket: Nút xác nhận để mua vé. * id, date, time, totalValue: ID của phim, ngày chiếu, giờ chiếu và tổng chi phí của ghế đã chọn. * databaseHelper: Đối tượng hỗ trợ truy xuất cơ sở dữ liệu để lưu trữ và lấy dữ liệu liên quan đến đặt vé.   + **Phương thức:** * onCreate: Thiết lập giao diện, bao gồm RecyclerView để chọn giờ chiếu và ghế ngồi, đồng thời khởi tạo chức năng chọn ngày từ DatePicker. Khi người dùng chọn ghế và nhấn nút mua vé, ứng dụng sẽ chuyển đến màn hình thanh toán với các thông tin đã chọn. * initSeats: Khởi tạo danh sách ghế ngồi cho một suất chiếu cụ thể. Nếu có ghế nào đã được đặt, ghế đó sẽ được đánh dấu là không thể chọn. * updateTotal: Cập nhật tổng chi phí hiển thị khi người dùng chọn hoặc bỏ chọn ghế. * onActivityResult: Xử lý kết quả trả về từ PaymentActivity. Khi thanh toán thành công, phương thức này lưu thông tin đặt vé vào cơ sở dữ liệu và chuyển đến màn hình ViewTicketActivity. * **Lớp SeatAdapter:**   + **Thuộc tính:** * context: Ngữ cảnh của ứng dụng, dùng để truy cập tài nguyên. * listener: Đối tượng SeatClickListener, lắng nghe sự kiện khi người dùng chọn ghế. * list: Danh sách các ghế có thể chọn trong phòng chiếu.   + **Phương thức:** * renewItems: Cập nhật danh sách ghế khi có sự thay đổi và làm mới RecyclerView. * getSelectedCount(): Trả về số lượng ghế đã được chọn. * getSelectedSeats(): Trả về danh sách vị trí của các ghế đã được chọn. * onCreateViewHolder: Tạo và trả về một ViewHolder cho từng mục ghế trong danh sách. * onBindViewHolder: Liên kết dữ liệu ghế với ViewHolder, thiết lập trạng thái ghế (đã đặt, đã chọn, chưa chọn) và màu sắc phù hợp, đồng thời cài đặt sự kiện nhấn vào ghế để người dùng chọn hoặc bỏ chọn ghế * **Lớp SeatSelection:**   + **Thuộc tính:** * seat: Vị trí của ghế ngồi trong rạp chiếu. * isSelected: Trạng thái ghế có được người dùng chọn hay không. * isBooked: Trạng thái ghế đã được đặt trước đó hay chưa.   + **Phương thức:** * SeatSelection(int seat, boolean isSelected, boolean isBooked): Constructor để khởi tạo một ghế với vị trí và trạng thái ban đầu. * getSeat(), setSeat(int seat): Truy xuất và thiết lập vị trí của ghế. * isSelected(), setSelected(boolean selected): Truy xuất và thiết lập trạng thái đã chọn của ghế. * isBooked(), setBooked(boolean booked): Truy xuất và thiết lập trạng thái đã đặt của ghế. * **Lớp SignInActivity:**   + **Thuộc tính:** * signUp: Nút để chuyển đến màn hình đăng ký. * signIn: Nút để thực hiện đăng nhập. * editTextUsername: Trường nhập tên đăng nhập. * editTextPassword: Trường nhập mật khẩu. * rememberMe: Checkbox để lưu thông tin đăng nhập của người dùn.   + **Phương thức:** * onCreate(Bundle savedInstanceState): Phương thức này được gọi khi hoạt động được tạo ra. Nó thiết lập giao diện, gán các view từ layout, và cài đặt các sự kiện nhấn cho nút đăng nhập và đăng ký. Nếu có thông tin người dùng đã lưu, nó sẽ tự động điền vào các trường nhập liệu * signIn():Phương thức này được gọi khi người dùng nhấn nút đăng nhập * **Lớp SignUpActivity:**   + **Thuộc tính:** * editTextUsername: Trường nhập tên đăng nhập cho người dùng. * editTextPassword: Trường nhập mật khẩu. * editTextConfirmPassword: Trường nhập để xác nhận mật khẩu. * buttonSignUp: Nút để thực hiện đăng ký.   + **Phương thức:** * onCreate(Bundle savedInstanceState): Phương thức này được gọi khi hoạt động được tạo ra. Nó thiết lập giao diện cho màn hình đăng ký, gán các view từ layout và cài đặt sự kiện nhấn cho nút đăng ký. * signUp():Phương thức này được gọi khi người dùng nhấn nút đăng ký * **Lớp SliderMovieAdapter:**   + **Thuộc tính:** * context: Ngữ cảnh của ứng dụng, được sử dụng để truy cập tài nguyên và các thành phần khác. * listener: Đối tượng ItemClickListener, lắng nghe sự kiện khi người dùng nhấp vào một phim. * movies: Danh sách chứa các đối tượng Movie, đại diện cho các bộ phim sẽ được hiển thị trong slider.   + **Phương thức:** * SliderMovieAdapter: Constructor để khởi tạo đối tượng SliderMovieAdapter với ngữ cảnh và listener cho sự kiện nhấp * renewItems: Cập nhật danh sách phim trong slider với danh sách mới và thông báo cho adapter để làm mới hiển thị. * deleteItem: Xóa một phim ở vị trí chỉ định trong danh sách và làm mới hiển thị. * addItem: Thêm một phim mới vào danh sách và làm mới hiển thị. * onCreateViewHolder: Tạo và trả về một ViewHolder cho mỗi mục trong slider bằng cách nạp layout từ file XML image\_slider\_layout\_item. * onBindViewHolder: Liên kết dữ liệu phim với ViewHolder. Chuyển đổi đường dẫn hình ảnh thành bitmap và thiết lập hình ảnh cho ImageView. Cài đặt sự kiện nhấp cho mỗi mục để gọi onMovieClick từ listener. * Convert: Phương thức tĩnh chuyển đổi một đối tượng Bitmap thành chuỗi mã hóa Base64. Phương thức này có thể được sử dụng để lưu trữ hoặc truyền tải hình ảnh. * getCount():Trả về số lượng phim hiện có trong danh sách, xác định số lượng mục trong slider. * **Lớp TimeAdapter:**   + **Thuộc tính:** * context: Ngữ cảnh của ứng dụng, được sử dụng để truy cập tài nguyên và các thành phần khác. * listener: Đối tượng ItemTimeClickListener, lắng nghe sự kiện khi người dùng nhấp vào một thời gian cụ thể. * list: Danh sách chứa các đối tượng TimeSelection, đại diện cho các thời gian có sẵn.   + **Phương thức:** * TimeAdapter: Constructor để khởi tạo đối tượng TimeAdapter với ngữ cảnh và listener cho sự kiện nhấp. Danh sách thời gian ban đầu được lấy từ Const.availableTime. * renewItems: Cập nhật danh sách thời gian trong adapter với danh sách mới và thông báo cho adapter để làm mới hiển thị. * onCreateViewHolder: Tạo và trả về một ViewHolder cho mỗi mục thời gian bằng cách nạp layout từ file XML item\_time. * onBindViewHolder: Liên kết dữ liệu thời gian với ViewHolder. Thiết lập văn bản cho TextView thời gian và thay đổi màu nền của CardView tùy thuộc vào trạng thái đã chọn hay chưa. Cài đặt sự kiện nhấp cho mỗi mục, để khi người dùng nhấp vào, nó sẽ xóa tất cả các mục đã chọn trước đó, đánh dấu mục hiện tại là đã chọn và gọi phương thức onTimeClick từ listener. * getItemCount():Trả về số lượng thời gian có sẵn từ Const.availableTime * clearSelected: Đặt trạng thái đã chọn cho tất cả các mục thời gian thành false. Nếu tham số notify là true, adapter sẽ được thông báo để làm mới hiển thị. * **Lớp TimeSelection:**   + **Thuộc tính:** * time: Một chuỗi (String) lưu trữ giá trị thời gian. * isSelected: Một biến kiểu boolean cho biết liệu lựa chọn thời gian này có được chọn hay không.   + **Phương thức:** * TimeSelection: Constructor khởi tạo đối tượng TimeSelection với thời gian được chỉ định. Trạng thái isSelected được đặt mặc định là false. * TimeSelection: Constructor này cho phép khởi tạo đối tượng với cả thời gian và trạng thái đã chọn. * getTime():Phương thức này trả về giá trị thời gian của đối tượng. * setTime: Phương thức này cho phép người dùng thay đổi giá trị thời gian. * isSelected():Phương thức này trả về trạng thái đã chọn (true hoặc false) của lựa chọn. * setSelected: Phương thức này cho phép người dùng thay đổi trạng thái đã chọn của lựa chọn thời gian. * **Lớp ViewTicketActivity:**   + **Thuộc tính:** * film: TextView hiển thị tên phim. * date: TextView hiển thị ngày xem phim. * time: TextView hiển thị thời gian chiếu. * seats: TextView hiển thị số ghế đã đặt. * booker: TextView hiển thị tên người đặt vé. * total: TextView hiển thị tổng số tiền vé. * qrImage: ImageView hiển thị mã QR của vé. * databaseHelper: Đối tượng DatabaseHelper để tương tác với cơ sở dữ liệu..   + **Phương thức:** * onCreate: Phương thức này được gọi khi activity được tạo. Nó thiết lập layout từ file XML activity\_view\_ticket và cấu hình các thành phần UI. Sử dụng EdgeToEdge để hiển thị toàn màn hình. Lấy thông tin phim từ cơ sở dữ liệu dựa trên ID được truyền qua Intent. Nếu không tìm thấy phim, activity sẽ kết thúc. Thiết lập các giá trị cho các TextView dựa trên dữ liệu nhận được từ Intent và thông tin người dùng. Gọi phương thức createQr để tạo mã QR cho vé. * createQr: Tạo mã QR chứa thông tin vé bao gồm ID phim, tên phim, ngày, ghế, và tên người đặt. Sử dụng thư viện QRGEncoder để tạo mã QR. Màu sắc của mã QR được thiết lập là đỏ cho nền đen và trắng cho mã. Bitmap của mã QR được thiết lập cho ImageView để hiển thị. * **Lớp AddUserActivity:**   + **Thuộc tính:** * editTextUsername: EditText để nhập tên đăng nhập của người dùng mới. * editTextPassword: EditText để nhập mật khẩu cho người dùng mới. * addUser: Button để kích hoạt hành động thêm người dùng khi người dùng nhấn vào.   + **Phương thức:** * onCreate: Phương thức khởi tạo Activity, thiết lập giao diện người dùng với bố cục activity\_add\_user và cài đặt các yếu tố EdgeToEdge để cải thiện bố cục toàn màn hình. Các phần tử giao diện như EditText và Button được ánh xạ tới các biến tương ứng, đồng thời, sự kiện nhấn Button được gán để gọi phương thức addUser(). * addUser(): Xử lý việc thêm người dùng mới. * **Lớp AdminActivity:**   + **Thuộc tính:** * user: Button cho phép admin truy cập vào giao diện quản lý người dùng. * movie: Button cho phép admin truy cập vào giao diện quản lý phim.   + **Phương thức:** * onCreate(Bundle savedInstanceState): Phương thức khởi tạo Activity, thiết lập giao diện người dùng với bố cục activity\_admin, đồng thời kích hoạt chế độ EdgeToEdge để tối ưu hiển thị toàn màn hình. * **Lớp EditMovieActivity:**   + **Thuộc tính:** * selectImage: Button cho phép admin mở thư viện ảnh để chọn ảnh bìa phim. * movieImage: ImageView hiển thị ảnh bìa phim đã chọn. * save: Button để lưu thông tin phim vào cơ sở dữ liệu. * Các EditText khác: Bao gồm name, description, country, genre, director, actor, duration, releaseDate để admin nhập thông tin phim. * imagePath: Đường dẫn đến ảnh bìa phim đã chọn. * databaseHelper: Đối tượng trợ giúp quản lý dữ liệu cho phép truy cập cơ sở dữ liệu. * mGetContent: ActivityResultLauncher giúp mở thư viện ảnh của thiết bị và lấy về ảnh bìa phim.   + **Phương thức:** * onCreate: Phương thức khởi tạo, cài đặt giao diện, ánh xạ các thành phần UI và xử lý sự kiện nhấn nút chọn ảnh và nút lưu. * initAdd(): Phương thức thiết lập giao diện và xử lý sự kiện cho trường hợp thêm mới phim. * initUpdate: Chuẩn bị cho trường hợp cập nhật, tuy nhiên phần này chưa được triển khai chi tiết. * **Lớp ManageUserActivity:**   + **Thuộc tính:** * userRecyclerView: RecyclerView hiển thị danh sách người dùng trong cơ sở dữ liệu. * userAdapter: UserAdapter quản lý dữ liệu người dùng và hiển thị dữ liệu này lên RecyclerView. * databaseHelper: Đối tượng DatabaseHelper để truy xuất danh sách người dùng từ cơ sở dữ liệu. * addUser: Button cho phép chuyển sang màn hình thêm người dùng mới.   + **Phương thức:** * onCreate: Phương thức khởi tạo hoạt động. Cài đặt giao diện và ánh xạ các thành phần UI. Thiết lập RecyclerView với DividerItemDecoration để thêm đường viền ngăn cách giữa các mục người dùng. Sử dụng userAdapter để hiển thị danh sách người dùng lấy từ databaseHelper. Nút AddUser: Khi nhấn, nút này sẽ chuyển đến AddUserActivity để thêm tài khoản người dùng mới. * onResume(): Khi ManageUserActivity quay trở lại hoạt động, danh sách người dùng được cập nhật tự động bằng cách lấy dữ liệu từ cơ sở dữ liệu và gọi renewItems() trên userAdapter. * **Lớp MovieAdapter:**   + **Thuộc tính:** * context: Ngữ cảnh của ứng dụng, giúp truy xuất tài nguyên và các thao tác liên quan đến DatabaseHelper. * list: Danh sách các đối tượng Movie đại diện cho các bộ phim sẽ được hiển thị.   + **Phương thức:** * renewItems: Cập nhật danh sách các bộ phim và làm mới RecyclerView. * onCreateViewHolder: Tạo và trả về ViewHolder cho từng mục trong danh sách phim. * onBindViewHolder: Liên kết dữ liệu của mỗi bộ phim với ViewHolder, bao gồm hiển thị tên phim, ảnh bìa và xử lý sự kiện nhấn vào nút xóa. holder.delete.setOnClickListener: Khi người dùng nhấn vào biểu tượng xóa, bộ phim tương ứng sẽ bị xóa khỏi cơ sở dữ liệu thông qua DatabaseHelper, đồng thời bị loại bỏ khỏi danh sách hiển thị. * getItemCount(): Trả về số lượng bộ phim trong danh sách. * **Lớp UserAdapter:**   + **Thuộc tính:** * context: Ngữ cảnh của ứng dụng, dùng để truy cập vào tài nguyên của ứng dụng và thao tác với cơ sở dữ liệu. * list: Danh sách các đối tượng User đại diện cho các tài khoản người dùng sẽ được hiển thị.   + **Phương thức:** * renewItems: Cập nhật danh sách người dùng với dữ liệu mới và làm mới RecyclerView. * onCreateViewHolder: Tạo và trả về ViewHolder cho từng mục người dùng trong danh sách. * onBindViewHolder: Liên kết dữ liệu của người dùng với ViewHolder, bao gồm hiển thị tên người dùng và ngày tạo tài khoản, cũng như xử lý sự kiện xóa người dùng.   holder.delete.setOnClickListener: Khi người dùng nhấn vào biểu tượng xóa, UserAdapter gọi phương thức deleteUser() từ DatabaseHelper để xóa người dùng khỏi cơ sở dữ liệu. Đồng thời, người dùng cũng sẽ bị loại khỏi danh sách hiển thị.   * getItemCount(): Trả về số lượng người dùng trong danh sách. |

**Thiết kế cơ sở dữ liệu:**

* SQLite: Được sử dụng để lưu trữ dữ liệu cục bộ trên thiết bị, bao gồm thông tin về người dùng, phim, và các đặt vé.
* SQLiteOpenHelper: Lớp này được dùng để tạo và quản lý cơ sở dữ liệu SQLite, với các phương thức như onCreate và onUpgrade để tạo và cập nhật bảng.
* ContentValues và SQLiteDatabase: Được sử dụng để chèn, cập nhật và xóa dữ liệu trong các bảng của SQLite.

**2.3. Triển khai:**

* **Viết code:** Sử dụng Java để xây dựng các lớp trong mô hình MVC, bao gồm các lớp cho mô hình (dữ liệu), giao diện, và điều khiển.

Model:

* Tạo các lớp User, Movie, và Reservation để đại diện cho các thực thể trong cơ sở dữ liệu với các thuộc tính và phương thức tương ứng.
* Cài đặt lớp DatabaseHelper để thao tác với SQLite, sử dụng SQLiteOpenHelper để tạo và quản lý các bảng User, Movie, và Reservation.
* Sử dụng ContentValues và SQLiteDatabase để thêm, sửa, xóa và truy vấn dữ liệu từ cơ sở dữ liệu.

View:

* Thiết kế các Activity như MainActivity, MovieActivity, và PaymentActivity để hiển thị dữ liệu cho người dùng và xử lý tương tác trên giao diện người dùng.
* Sử dụng XML để xây dựng giao diện người dùng cho các màn hình khác nhau.

Controller:

* Cài đặt các lớp điều khiển chính trong các Activity như MainActivity và MovieActivity để xử lý logic ứng dụng.
* Kết nối View và Model bằng cách gọi các phương thức của DatabaseHelper từ các Activity, cho phép truy xuất dữ liệu từ cơ sở dữ liệu và hiển thị trên giao diện người dùng.
* **Kiểm thử:** Viết các test case để kiểm tra các chức năng của ứng dụng, đảm bảo ứng dụng hoạt động đúng theo yêu cầu.

**2.4. Vận hành và bảo trì:**

* Cài đặt và triển khai:
  + Hướng dẫn người dùng cách chạy ứng dụng bằng máy ảo.
* Bảo trì: Sửa lỗi phát sinh, cập nhật chức năng mới (nếu có) và cải thiện hiệu năng của ứng dụng.

**CHƯƠNG 3. KẾT QUẢ THỰC HIỆN**

**3.1. Công nghệ đã sử dụng**

* Ngôn ngữ lập trình: Java
* Công cụ: Android Studio
* Thư viện: (tùy chọn) có thể sử dụng các thư viện hỗ trợ đọc/ghi file, xử lý dữ liệu,

**3.2. Tiến độ thực hiện**

Link github tới dự án: https://github.com/Nghiazzz/CSE441\_PROJECT

**Hướng dẫn các bước đã thực hiện:**

**B1. Tạo dự án mới:**

* Mở Android Studio
* Chọn "Create New Project" và thiết lập các tùy chọn:
* Ngôn ngữ lập trình: Chọn Java.
* Cấu hình Android SDK: Chọn phiên bản SDK phù hợp
* Tên dự án: Đặt tên dự án
* Vị trí lưu trữ: Chọn thư mục lưu trữ dự án.
* Nhấn "Finish" để hoàn tất tạo dự án..

**B2. Tạo các package:**

Tạo các package sau:

* + model: Chứa các lớp User, Movie, và Reservation.
  + database: Chứa lớp DatabaseHelper để quản lý cơ sở dữ liệu SQLite.
  + controller: Chứa các Activity như MainActivity, MovieActivity, và PaymentActivity.
  + view: Chứa các giao diện và màn hình hiển thị..

**B3. Tạo các lớp:**

* Trong mỗi package, click chuột phải và chọn "New" -> "Java Class" để tạo các lớp tương ứng.
* Cài đặt các thuộc tính và phương thức cho từng lớp dựa trên thiết kế đã phân tích.

**B4. Cấu hình thư viện SQLite:**

* SQLite được tích hợp sẵn trong Android SDK, không cần cài đặt thêm thư viện ngoài.
* Cài đặt lớp DatabaseHelper:
* Trong lớp DatabaseHelper, kế thừa SQLiteOpenHelper và ghi đè các phương thức onCreate và onUpgrade.
* Tạo các bảng User, Movie, Reservation với câu lệnh CREATE TABLE để lưu trữ dữ liệu ứng dụng.

**B5. Viết code:**

* Bắt đầu viết code cho từng lớp, thực hiện các chức năng của ứng dụng:
  + User, Movie, Reservation (trong package model): Cài đặt các thuộc tính và phương thức cơ bản, như thông tin người dùng, phim, và đặt vé.
  + DatabaseHelper (trong package database): Cài đặt các phương thức để thêm, cập nhật, xóa và truy vấn dữ liệu trong cơ sở dữ liệu.
  + MainActivity, MovieActivity, PaymentActivity,..: Cài đặt các logic xử lý yêu cầu của người dùng, điều phối các lớp khác để hoàn thành chức năng:
  + MainActivity: Hiển thị danh sách phim.
  + MovieActivity: Hiển thị thông tin chi tiết của phim và xử lý đặt vé.
  + PaymentActivity: Xử lý thanh toán và xác nhận vé.
  + Giao diện XML (trong package res/layout): Thiết kế giao diện người dùng cho từng Activity, đảm bảo trải nghiệm trực quan và dễ sử dụng.

**B6. Cấu hình và triển khai cơ sở dữ liệu SQLite:**

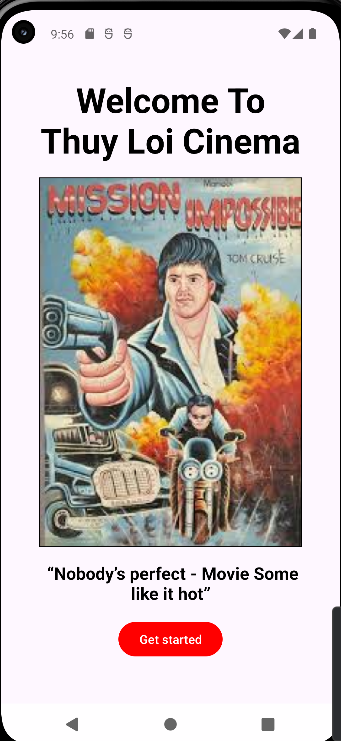
* Trong DatabaseHelper, cài đặt các phương thức như addUser, addMovie, addReservation để thực hiện các thao tác CRUD.
* Đảm bảo kết nối User và Movie với Reservation qua các khóa ngoại để duy trì toàn vẹn dữ liệu.

**B7. Chạy và kiểm thử:**

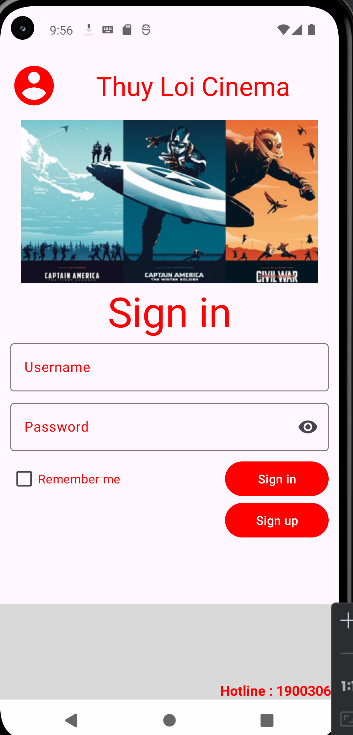
* Kết nối điện thoại hoặc sử dụng trình giả lập Android.
* Nhấn Run trong Android Studio để khởi chạy ứng dụng.

# **3.3. Hình ảnh sản phẩm**

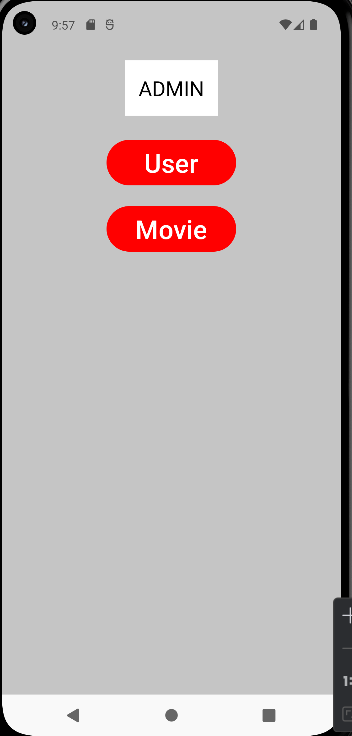
## 1. Giao diện trang bắt đầu

****

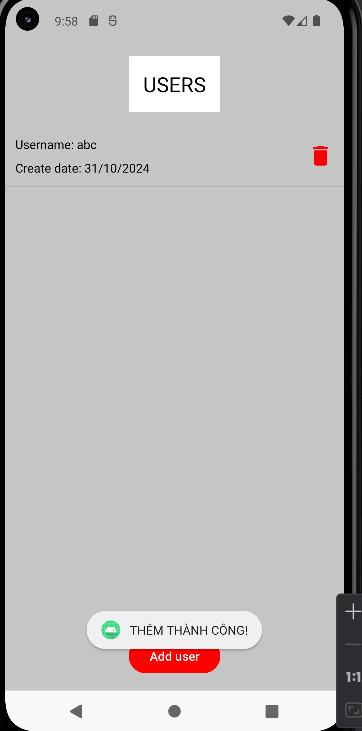
## 2. Giao diện trang đăng kí tài khoản

****

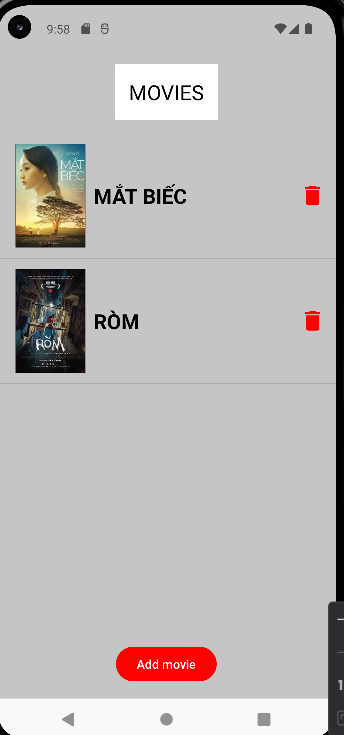
## 3. Giao diện trang quản trị

****

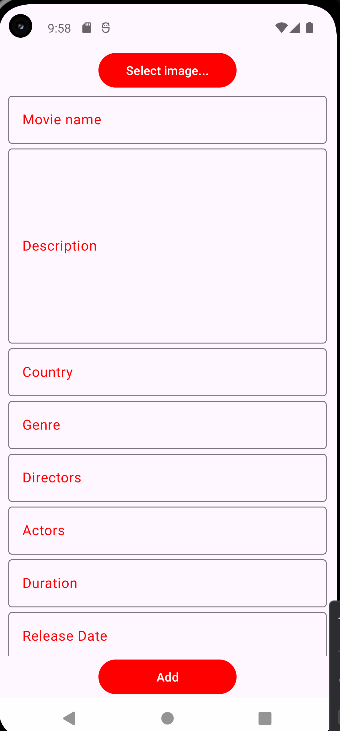
## 4. Giao diện trang quản lí tài khoản

****

## 5. Giao diện trang quản lí phim

****

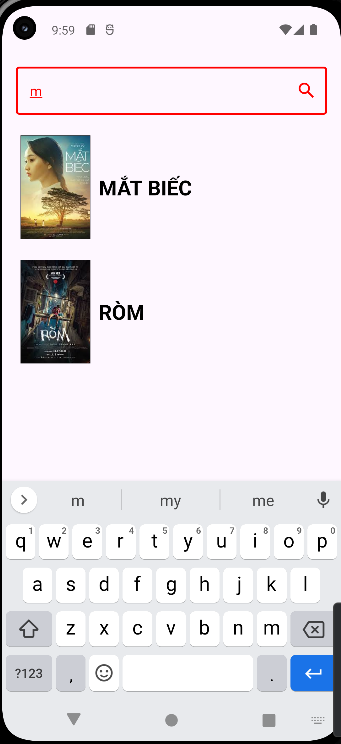
## 6. Giao diện trang thêm phim

****

## 7. Giao diện trang chủ của app

****

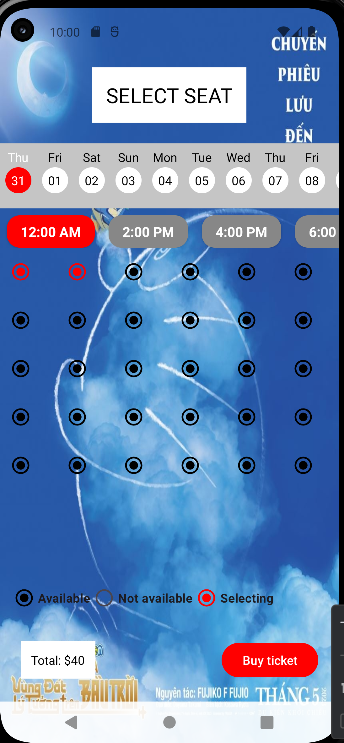
## 8. Giao diện trang tìm kiếm

****

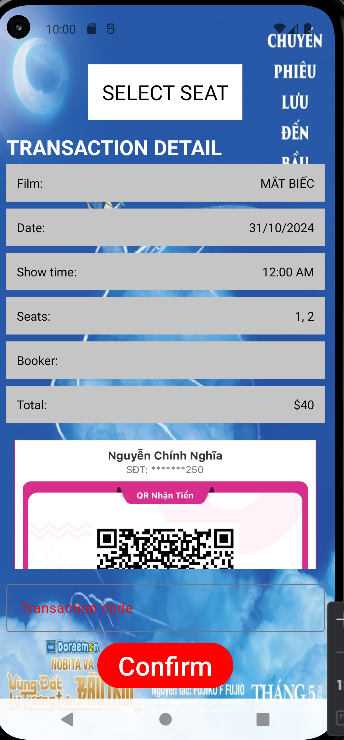
## 9. Giao diện trang phim chi tiết

****

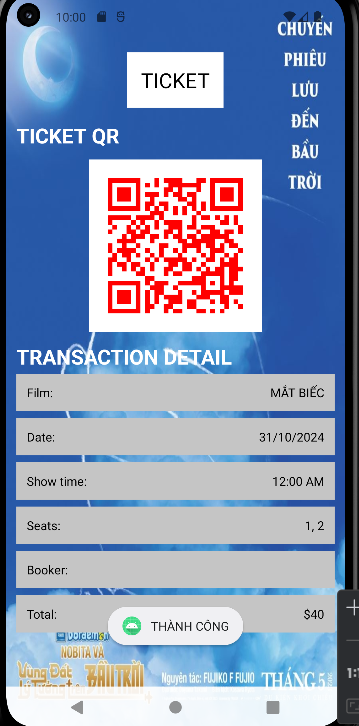
## 10. Giao diện trang đặt vé

****

## 11. Giao diện trang thanh toán

****

## 12. Giao diện trang thông tin vé

****

**KẾT LUẬN**

Ưu điểm của dự án ứng dụng đặt vé xem phim này là ứng dụng được thiết kế theo mô hình MVC, giúp mã nguồn dễ bảo trì và mở rộng. Việc sử dụng cơ sở dữ liệu SQLite giúp lưu trữ dữ liệu cục bộ hiệu quả, đặc biệt phù hợp với các ứng dụng không đòi hỏi kết nối internet liên tục. Điều này mang lại trải nghiệm liền mạch cho người dùng khi có thể truy cập thông tin ngay cả khi không có kết nối mạng.

Tuy nhiên, nhược điểm của ứng dụng là bị hạn chế trong khả năng mở rộng quy mô dữ liệu và chức năng do sử dụng SQLite, vốn chỉ phù hợp cho lưu trữ dữ liệu cục bộ. Nếu ứng dụng cần quản lý dữ liệu lớn hơn hoặc tích hợp tính năng đa người dùng, hệ thống sẽ gặp khó khăn trong việc đồng bộ và đảm bảo tính toàn vẹn của dữ liệu. Ngoài ra, ứng dụng hiện tại chỉ đáp ứng các chức năng cơ bản và có thể thiếu các tính năng nâng cao, chẳng hạn như tích hợp các cổng thanh toán đa dạng, gửi thông báo đẩy cho người dùng hoặc cung cấp thông tin thời gian thực về suất chiếu.

Trong hướng phát triển tương lai, ứng dụng có thể được nâng cấp để sử dụng một cơ sở dữ liệu đám mây như Firebase hoặc MySQL để hỗ trợ tính năng đồng bộ hóa và đa người dùng. Ngoài ra, ứng dụng có thể tích hợp thêm các API để cung cấp thông tin suất chiếu chính xác từ các rạp chiếu phim trên toàn quốc. Các tính năng như thông báo đẩy, nhận xét và đánh giá phim, cũng như thanh toán qua nhiều phương thức khác nhau sẽ giúp nâng cao trải nghiệm người dùng, đáp ứng nhu cầu ngày càng đa dạng của thị trường.

**QUY ĐỊNH TRÌNH BÀY TRONG BÁO CÁO CÀI TẬP LỚN**

- Bài tập lớn được in trên một mặt giấy trắng khổ A4 (210 x 297mm), dày lớn hơn 30 trang, nhỏ hơn 100 trang, không kể hình vẽ, bảng biểu, đồ thị và danh mục tài liệu tham khảo.

- Phần nội dung trình bày trong bài tập lớn sử dụng Font chữ **Times New Roman** cỡ **13**, hệ soạn thảo Microsoft Word; mật độ chữ bình thường, không được nén hoặc kéo dãn khoảng cách giữa các chữ; dãn dòng đặt ở chế độ **1,5 lines**; lề trên **2,0 cm**; lề dưới **2,0 cm**; lề trái **2,5 cm**, lề phải **2,0 cm**. Số trang được đánh ở giữa, phía dưới trang giấy.

- Cách ghi trích dẫn tài liệu tham khảo: Cuối đoạn trích dẫn đánh số thứ tự tài liệu tham khảo (ví dụ: [1]: tham khảo tài liệu số 1; [3,4,8]: tham khảo 3 tài liệu số 3, 4, 8).

- Tuyệt đối không được tẩy, xoá, sửa chữa trong bài tập lớn.

- Quy cách trình bày nội dung

|  |  |  |  |
| --- | --- | --- | --- |
| **Đề mục** | **Cỡ chữ** | **Định dạng** | **Canh lề trang** |
| Tên chương | 14 | In hoa, đậm | Giữa |
| Tên tiểu mục mức 1 | 13 | Chữ thường, đậm | Trái |
| Tên tiểu mục mức 2 | 13 | Chữ thường, đậm, nghiêng | Trái |
| Tên tiểu mục mức 3 | 13 | Đánh chỉ mục bằng chữ cái thường  a), ...  b), ..... | Trái |
| Nội dung | 13 | Chữ thường (Normal) | Đều hai bên |
| Nội dung bảng (table) | 12 | Normal | Giữa ô |
| Tên bảng | 12 | Chữ thường, nghiêng | Giữa, trên bảng |
| Tên hình | 12 | Chữ thường, nghiêng | Giữa, dưới hình |
| Tài liệu tham khảo | 12 | APA style | Chú thích bên dưới |

***Cách đánh dấu câu****:*

Các dấu: : , . ; ) } ] ! ? ” được gõ ngay sau ký tự cuối cùng (không khoảng cách), và gõ 1 phím cách (space) sau chúng. Sau các dấu: “{ ( [ không gõ dấu cách.

***\* Cách đánh số các tiểu đề mục nhiều nhất là 3 mức và không lùi sang phải***

Hướng dẫn xếp tài liệu tham khảo

1. Tài liệu tham khảo xếp theo thứ tự trích dẫn trong bài tập lớn.
2. Tài liệu tham khảo là sách, luận án, báo cáo phải ghi đầy đủ các thông tin sau:

* Tên các tác giả hoặc cơ quan ban hành (không có dấu ngăn cách)
* (năm xuất bản), (đặt trong ngoặc đơn, dấu phẩy sau ngoặc đơn)
* Tên sách, luận án hoặc báo cáo, (in nghiêng, dấu phẩy cuối tên)
* Nhà xuất bản, (dấu phẩy cuối tên nhà xuất bản)
* Nơi xuất bản, (dấu chấm kết thúc tài liệu tham khảo).

Tài liệu tham khảo là bài báo trong tạp chí, bài trong một cuốn sách… ghi đầy đủ các thông tin sau:

* Tên các tác giả (không có dấu ngăn cách);
* (Năm công bố), (đặt trong ngoặc đơn, dấu phẩy sau ngoặc đơn)
* “Tên bài báo”, (đặt trong ngoặc kép, không in nghiên, dấu phẩy cuối tên)
* Tên tạp chí hoặc tên sách, (in nghiên, dấu phẩy cuối tên)
* Tập (không có dấu ngăn cách)
* (Sổ), (đặt trong ngoặc đơn, dấu phẩy sau ngoặc đơn)
* Các số trang, (gạch ngang giữa hai chữ số, dấu chấm kết thúc)

Cần chú ý những chi tiết về trình bày nêu trên. Nếu tài liệu dài hơn một dòng thì nên trình bày sau cho từ dòng thứ hai lùi vào so với dòng thứ nhất 1 cm để phần tài liệu tham khảo được rõ ràng và dễ theo dõi.

Ví dụ:

1. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.
2. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.
3. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.