

Trường Đại học Công nghệ - ĐHQGHN
Khoa Công nghệ thông tin

BÀI TẬP LỚN: PHÂN TÍCH & THIẾT KẾ HƯỚNG ĐỐI TƯỢNG

Giảng viên: Đặng Đức Hạnh



IDENTIFY DESIGN ELEMENTS ỨNG DỤNG QUẢN LÝ RẠP PHIM

Ngày: 21/04/2024

Chuẩn bị bởi: Nhóm 6

Mục lục

Lịch sử sửa đổi	3
1. Giới thiệu	4
1.1. Mục đích	4
1.2. Đối tượng dự kiến và đề xuất cách đọc	4
1.3. Phạm vi dự án	5
1.4. Tài liệu tham khảo	5
2. Sơ đồ bối cảnh hệ thống con	6
2.1. Hệ thống con PaymentSystem	6
2.1.1. Sơ đồ	6
2.1.2. Mô tả giao diện hệ thống con	6
2.2. Hệ thống con NavigationSystem	7
2.2.1. Sơ đồ	7
2.2.2. Mô tả giao diện hệ thống con	7
2.3. Hệ thống con SuggestionSystem	8
2.3.1. Sơ đồ	8
2.3.2. Mô tả giao diện hệ thống con	8
2.4. Hệ thống con ManagementSystem	9
2.4.1. Sơ đồ	9
2.4.2. Mô tả giao diện hệ thống con	9
3. Xác định các phần tử thiết kế	9
4. Đóng gói các phần tử thiết kế	11
5. Các lớp kiến trúc và phụ thuộc	12
5.1. Biểu đồ quan hệ phụ thuộc giữa các lớp kiến trúc	12
5.2. Mô tả các lớp kiến trúc	12
6. Các gói và phụ thuộc	13
6.1 Biểu đồ quan hệ phụ thuộc giữa các gói	13
6.2. Mô tả các gói	14
Phụ lục I. Từ điển thuật ngữ	16

Lịch sử sửa đổi

Họ tên	Thời gian	Lý do sửa đổi	Phiên bản
Trung	21/04/2024	Khởi tạo mẫu tài liệu	1.0.0
Dương	27/04/2024	Bổ sung NavigationSystem	1.0.1
Trung	29/04/2024	Bổ sung ManagementSystem	1.0.2
hmtu	01/05/2024	Bổ sung PaymentSystem	1.0.3
Đạt	04/05/2024	Bổ sung SuggestionSystem	1.0.4
hmtu	08/05/2024	Xác định các phân tử thiết kế	1.1.0
nanhtu	08/05/2024	Xác định các cơ chế thiết kế	1.1.1
Đạt	08/05/2024	Xác định các gói và phụ thuộc	1.1.2

1. Giới thiệu

1.1. Mục đích

Đây là một báo cáo về chủ đề Phân tích và thiết kế hướng đối tượng của nhóm 6 về việc xác định các phần tử thiết kế.

Bản báo cáo được viết dựa theo định dạng tài liệu “IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications”. Nội dung của bản báo cáo đã được nhóm thảo luận và thống nhất thông qua buổi họp ngày 21/04/2024 (sử dụng Google Meets) và các cuộc thảo luận sau đó (qua Messenger).

Tài liệu này được dùng để phân tích các tương tác của các lớp phân tích để xác định các phần tử thiết kế của hệ thống.

1.2. Đối tượng dự kiến và đề xuất cách đọc

Vai trò kiến trúc sư phần mềm: Kiến trúc sư phần mềm đóng vai trò lãnh đạo và điều phối các hoạt động kỹ thuật và tạo tác trong suốt dự án. Kiến trúc sư phần mềm thiết lập cấu trúc tổng thể cho từng khung nhìn kiến trúc: phân chia khung nhìn, nhóm các phần tử và tạo giao diện giữa các nhóm chính này. Do đó, khác với các vai trò khác, khung nhìn của kiến trúc sư phần mềm là một khía cạnh về chiều rộng đối lập với chiều sâu.

Các đối tượng mà báo cáo này hướng đến bao gồm:

- Người quản lý dự án: Người thực hiện quản lý dự án và phản hồi về chất lượng của hệ thống. Người quản lý dự án nên đọc toàn bộ tài liệu để lập kế hoạch và phân công các công việc.
- Nhà phát triển: Người thực hiện nhiệm vụ phát triển hệ thống từ đầu vào là bản thiết kế và tài liệu để tạo thành đầu ra là một phiên bản có thể chạy được. Nhà phát triển phải đọc toàn bộ tài liệu để thực hiện đúng theo hệ thống đề ra.
- Người viết tài liệu: Người sẽ viết tài liệu trong tương lai (các báo cáo, biên bản). Người viết tài liệu nên đọc để hiểu phần sơ đồ Use Case Main.

Tài liệu này gồm 5 phần chính:

- **Phần 1 Sơ đồ bối cảnh hệ thống con:** Thiết kế hệ thống con được sử dụng để đóng gói hành vi bên trong một “package”, nó cung cấp giao diện rõ ràng và chính thức, và nó theo quy ước sẽ không tiết lộ bất kỳ nội dung nội bộ nào của mình. Nó được sử dụng như một đơn vị của hành vi trong hệ thống, cung cấp khả năng đóng gói hoàn toàn các tương tác của một số lớp và/hoặc các hệ thống con. Khả năng "encapsulation" - đóng gói của các thiết kế hệ thống con

đổi lập với khả năng của Artifact: Thiết kế gói - Design Package, không nhận ra giao diện và có thể hiển thị nội dung được đánh dấu là 'công khai'. Các gói được sử dụng chủ yếu để quản lý cấu hình và tổ chức mô hình, trong đó các hệ thống con cung cấp thêm ngữ nghĩa hành vi.

- **Phần 2 Xác định các phần tử thiết kế:** Các lớp phân tích được làm mịn thành các phần tử mô hình thiết kế (các lớp thiết kế, các gói và các hệ thống con).
- **Phần 3 Đóng gói các phần tử thiết kế:** Đóng gói là một cơ chế có mục đích chung để tổ chức các phần tử thành các nhóm và nó cung cấp khả năng tổ chức mô hình đang được phát triển.
- **Phần 4 Các lớp kiến trúc và phụ thuộc:** Phân lớp cung cấp một phân vùng logic của hệ thống con thành một số nhóm với quy tắc nhất định về cách các mối quan hệ có thể được hình thành giữa các lớp. Ngoài ra, việc phân lớp sẽ là một cách để hạn chế các phụ thuộc giữa các hệ thống con và cho kết quả là hệ thống được kết nối lỏng lẻo hơn và do đó dễ bảo trì hơn.
- **Phần 5 Các gói và phụ thuộc:** Một thiết kế gói và nội dung của nó là trách nhiệm của duy nhất của vai trò người thiết kế. Các phần tử trong gói này có thể phụ thuộc vào các phần tử có trong các gói khác, điều này dẫn đến sự phụ thuộc giữa các gói. Các phụ thuộc gói có thể được sử dụng như một công cụ để phân tích khả năng phục hồi của mô hình thiết kế: một mô hình với các gói phụ thuộc chéo thì có khả năng phục hồi để thay đổi thấp.

1.3. Phạm vi dự án

Ứng dụng *Cinemagic* được xây dựng như một phương tiện đưa ra gợi ý dinh dưỡng cho người dùng.

Ứng dụng sẽ được phát triển dưới dạng một ứng dụng Web. Người dùng cuối là khách hàng gặp khó khăn trong việc lựa chọn khẩu phần dinh dưỡng, sẽ sử dụng ứng dụng thông qua trình duyệt Web Browser (Chrome, Edge, Firefox,...). Người dùng có thể xem đề xuất về khẩu phần dinh dưỡng phù hợp với bản thân dựa trên vị trí địa lý, tình trạng sức khỏe, khẩu vị, tôn giáo, độ tuổi, ngân sách tài chính,...

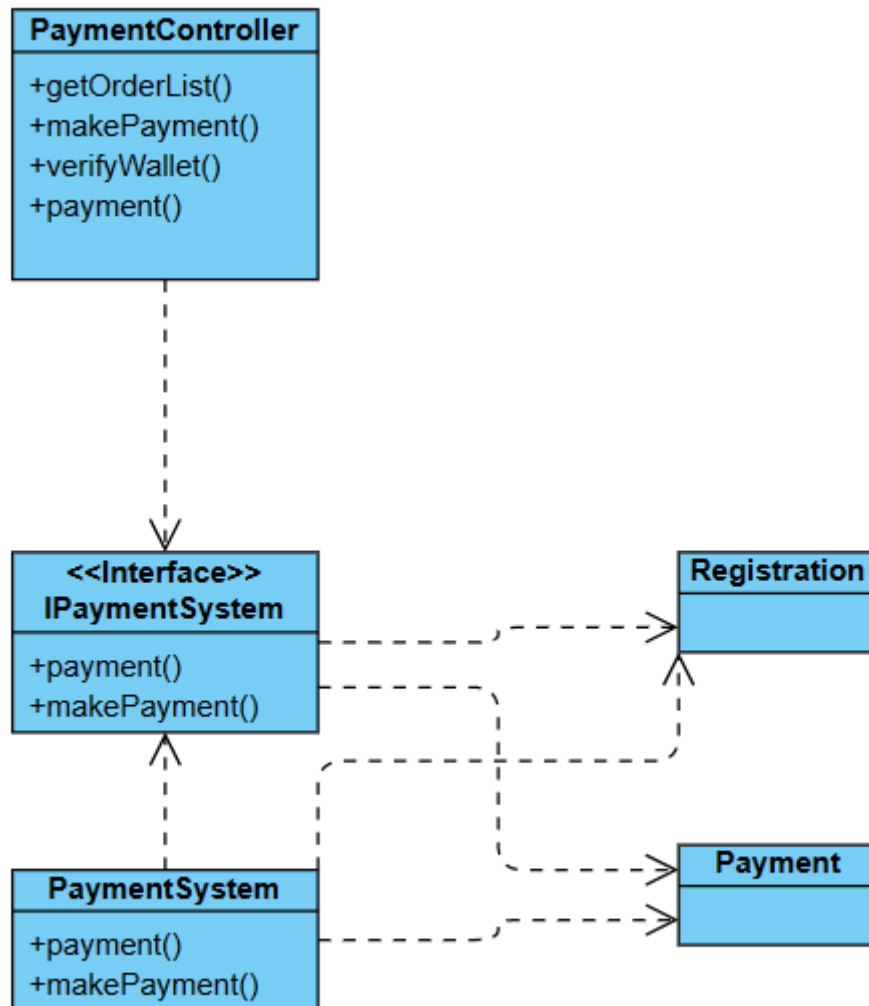
1.4. Tài liệu tham khảo

- [1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
- [2] Slide môn học Phân tích và thiết kế hướng đối tượng do giảng viên cung cấp.
- [3] Từ điển thuật ngữ của Ứng dụng *Quản lý rạp chiếu phim Cinemagic*.
- [4] Use Case Model của Ứng dụng *Quản lý rạp chiếu phim Cinemagic*.

2. Sơ đồ bối cảnh hệ thống con

2.1. Hệ thống con PaymentSystem

2.1.1. Sơ đồ



Hình 0: Sơ đồ hệ thống con BankSystem

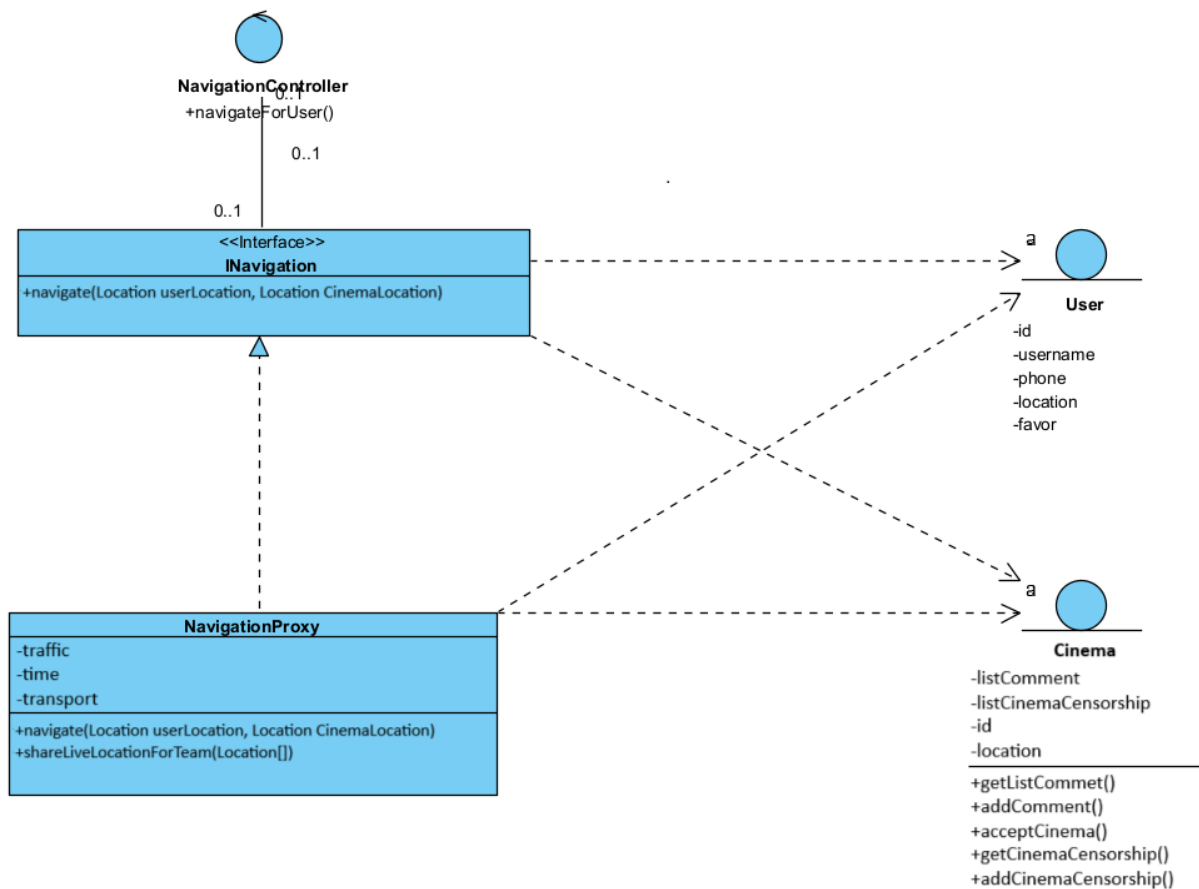
2.1.2. Mô tả giao diện hệ thống con

IBankSystem: Đóng gói các giao tiếp liên quan đến thanh toán hoá đơn của người dùng trực tiếp trên *Ứng dụng Cinemagic*.

- `makePayment()`: Yêu cầu thanh toán tiền vé sự kiện
- `payment()`: thực hiện giao dịch thanh toán vé

2.2. Hệ thống con NavigationSystem

2.2.1. Sơ đồ



Hình 1: Sơ đồ hệ thống con NavigationSystem

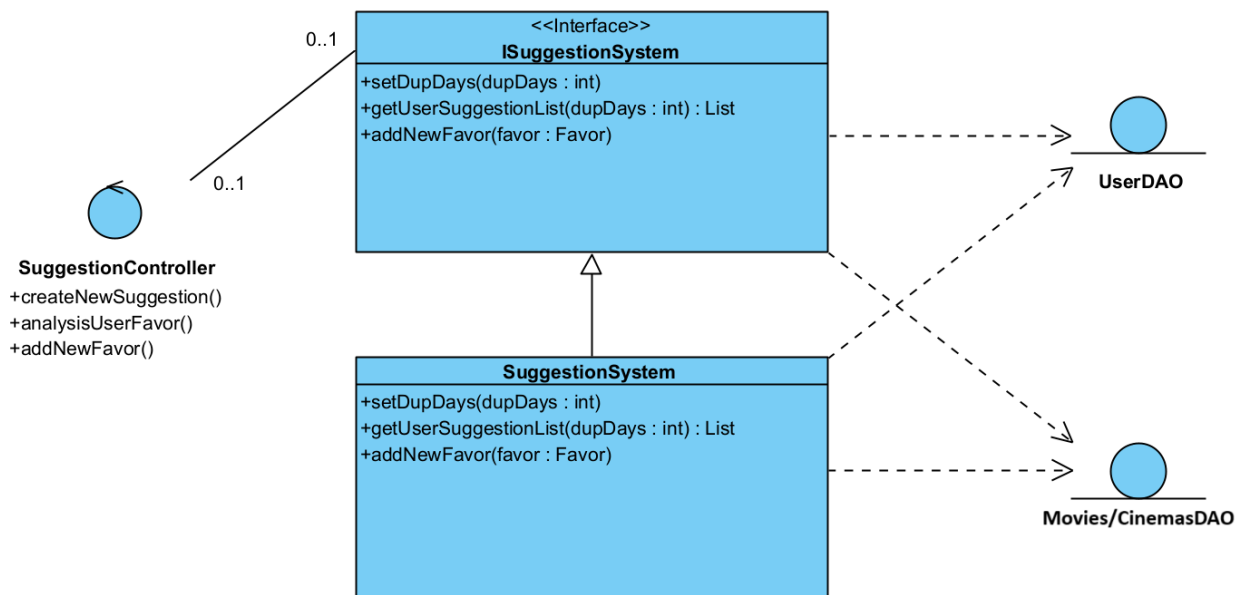
2.2.2. Mô tả giao diện hệ thống con

INavigation: Đóng gói các giao tiếp liên quan đến việc điều hướng, dẫn đường cho người dùng tới rạp phim đã chọn trong *Ứng dụng Cinemagic*.

- `navigate(Location userLocation, Location cinemaLocation)`: điều hướng cho người dùng, nhận đầu vào là vị trí người dùng và vị trí rạp phim muốn đến.
- `traffic`: tình trạng giao thông theo thời gian thực.
- `time`: thời gian đến địa điểm dự kiến.
- `transport`: phương tiện di chuyển của người dùng.

2.3. Hệ thống con SuggestionSystem

2.3.1. Sơ đồ



Hình 2: Sơ đồ hệ thống con SuggestionSystem

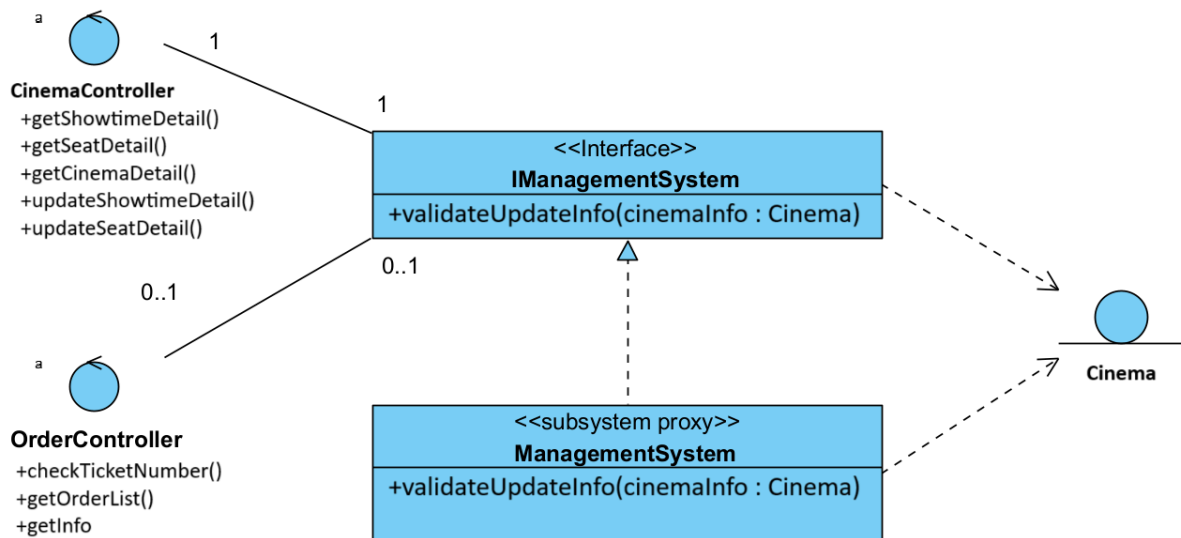
2.3.2. Mô tả giao diện hệ thống con

ISuggestionSystem: Đóng gói các giao tiếp liên quan đến gợi ý danh sách phim trong Ứng dụng Cinemagic.

- setDupDays(dupDays : int): Chỉnh sửa thời gian cho phép lặp lại một phim.
- getUserSuggestionList(dupDays : int) : List: Trả về danh sách các phim gợi ý cho một người dùng.
- addNewFavor(favor : Favor): Thêm phim vào danh sách không gợi ý hoặc yêu thích.

2.4. Hệ thống con ManagementSystem

2.4.1. Sơ đồ



Hình 4: Sơ đồ hệ thống con ManagementSystem

2.4.2. Mô tả giao diện hệ thống con

IManagementSystem: Đóng gói giao tiếp liên quan đến phê duyệt yêu cầu thay đổi thông tin rạp phim, yêu cầu của tất cả chủ rạp phim được gửi cho quản trị viên phê duyệt.

- `validateUpdateInfo(CinemaInfo: Cinema)`: Gửi thông tin thay đổi của rạp phim cho quản trị viên hoặc hệ thống ngoài phê duyệt.

3. Xác định các phần tử thiết kế

STT	Lớp phân tích	Phần tử thiết kế
1	searchFormViewer	searchFormViewer
2	searchController	searchController
3	Movie	Movie
4	Cinema	Cinema
5	user	user

6	userListViewer	userListViewer
7	userListController	userListController
8	LoginForm	LoginForm
9	userController	userController
10	suggestViewer	suggestViewer
11	suggestController	suggestController
12	cinemaManageViewer	cinemaManageViewer
13	order	order
14	orderController	orderController
15	cinemaUI	cinemaViewer
16	mapController	mapController
17	cart	cart

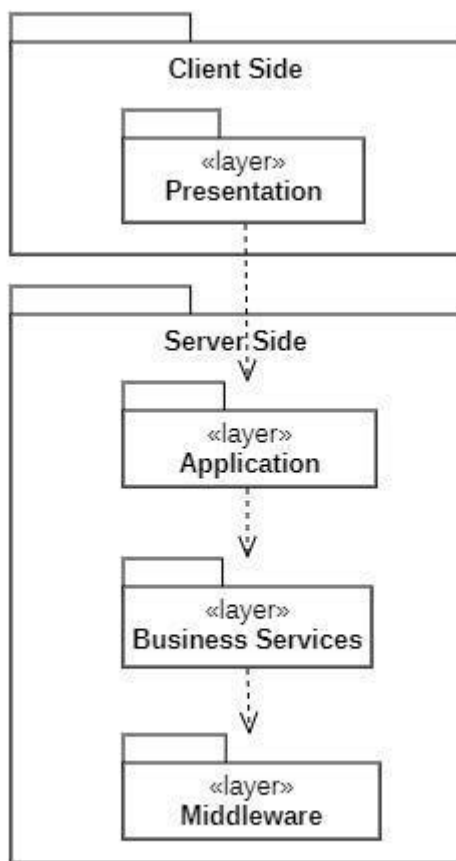
4. Đóng gói các phần tử thiết kế

STT	Phần tử thiết kế	Gói
1	searchFormViewer	Presentation
2	searchController	Application::User Activities
3	Movie	Business Service::Domain::Movie
4	cinema	Business Service::Domain::cinema
5	user	Business Service::Domain::User
6	userListViewer	Presentation
7	userListController	Application::User Activities
8	LoginForm	Presentation
9	UserController	Application::Admin Activities
10	suggestViewer	Presentation
11	suggestController	Application::User Activities
12	cinemaManageViewer	Presentation
13	order	Business Service::Domain::Order
14	orderController	Application::User Activities
15	cinemaUI	Presentation
16	mapController	Application::Maps
17	cart	Business Services::Domain::Order
18	UserDB	Business Services::Persistence
19	cinemaDB	Business Services::Persistence
20	PaymentSystem	Business Services::Payment
21	NavigationSystem	Business Services::Navigation
22	SuggestionSystem	Business Services::Suggestion

23	ManagementSystem	Business Services::Management
----	------------------	-------------------------------

5. Các lớp kiến trúc và phụ thuộc

5.1. Biểu đồ quan hệ phụ thuộc giữa các lớp kiến trúc



Hình 6: Biểu đồ quan hệ phụ thuộc giữa các lớp kiến trúc

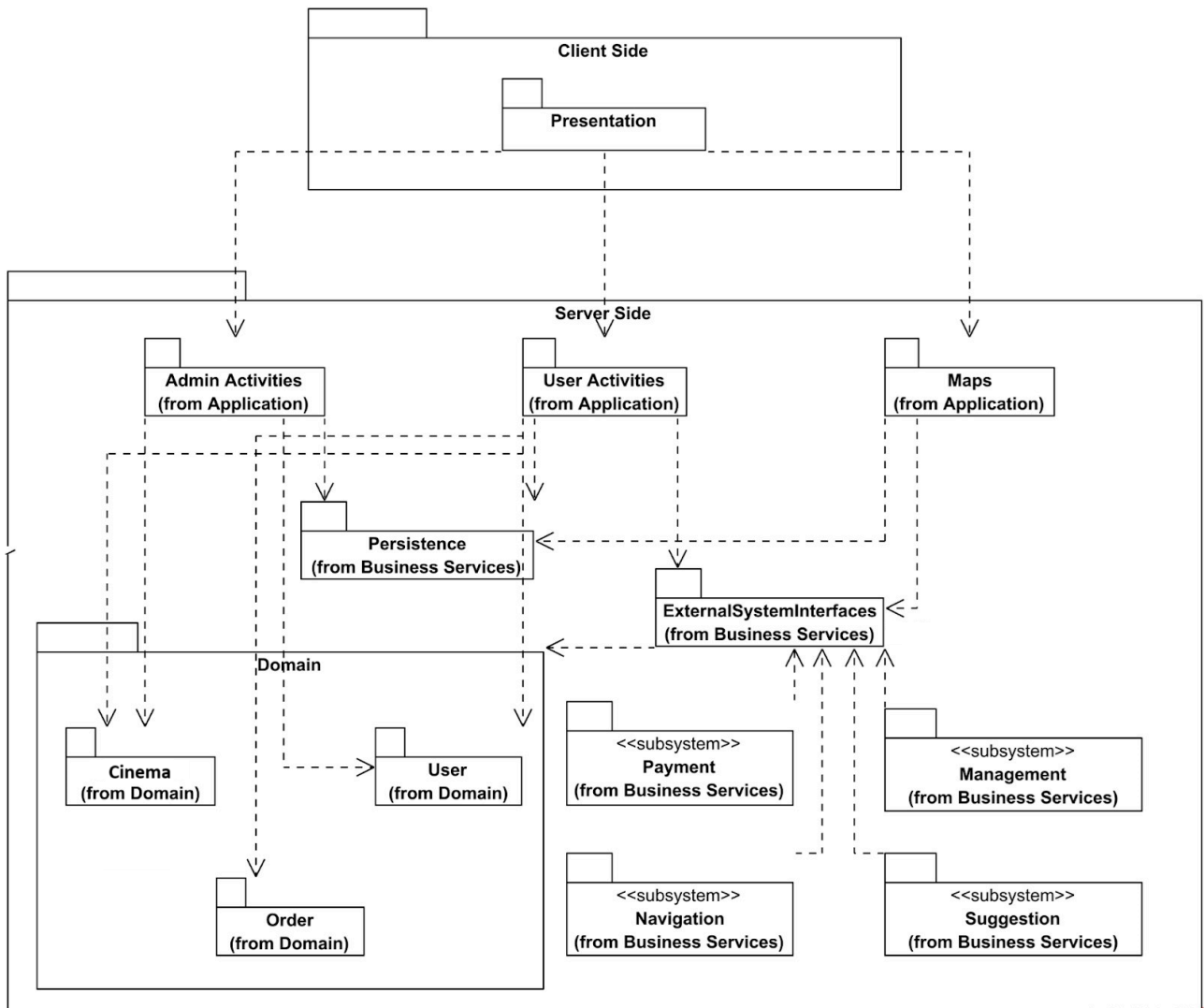
5.2. Mô tả các lớp kiến trúc

- Client Side: Lớp này là lớp mà nơi người dùng truy cập vào ứng dụng. Server layer chấp nhận yêu cầu thông qua kết nối internet từ client layer và chuyển các yêu cầu này đến tác nhân thích hợp. Máy chủ sẽ phản hồi kết quả từ tác nhân trở lại lớp người dùng. Trong trường hợp này, người dùng chỉ đơn giản là một trình duyệt.
- Presentation: Chứa các lớp cho mỗi biểu mẫu mà các tác nhân sử dụng để giao tiếp với Hệ thống.
- Server Side: Server layer hỗ trợ nhiều ứng dụng máy chủ khác nhau, trong đó “ứng dụng” bao gồm cả các trang web tĩnh. Máy chủ mạng sẽ biết trạng thái của máy chủ có tồn tại hay không. Thông thường, các máy chủ thường được quản lý bởi máy chủ mạng, tuy nhiên chương trình ứng dụng có thể đảm nhận một phần trách nhiệm này.

- Application: Chứa các lớp ứng dụng của các phần tử thiết kế cho chức năng xử lý chính của hệ thống.
- Business Services: Chứa các lớp để cung cấp các lớp hệ thống cho mục đích bảo trì.
- Middleware: Cung cấp các tiện ích và nền tảng dịch vụ độc lập.

6. Các gói và phụ thuộc

6.1 Biểu đồ quan hệ phụ thuộc giữa các gói



Hình 7: Biểu đồ quan hệ phụ thuộc giữa các gói

6.2. Mô tả các gói

Sơ đồ này mô tả cách các gói trên cùng một lớp kiến trúc phụ thuộc vào nhau và phụ thuộc vào các gói từ các lớp kiến trúc khác. Một phụ thuộc tồn tại giữa hai gói nếu thay đổi đối với một gói có thể gây ra thay đổi cho gói kia. Sơ đồ cũng cho thấy khả năng hiển thị - visibility có thể được xác định.

- Client Side: Là lớp mà nơi người dùng truy cập vào ứng dụng.
- Presentation: Chứa các lớp cho mỗi biểu mẫu mà các tác nhân sử dụng để giao tiếp với hệ thống.
- Server Side: Server layer hỗ trợ nhiều ứng dụng máy chủ khác nhau, trong đó “ứng dụng” bao gồm cả các trang web tĩnh.
- Admin Activities (from Application): Chứa các phần tử thiết kế để hỗ trợ cho các chức năng của quản trị viên hệ thống như hỗ trợ xác thực người dùng, phản hồi báo cáo, quản lý người dùng, quản lý rạp phim, xác thực rạp phim trên hệ thống.
- User Activities (from Application): Chứa các phần tử thiết kế để hỗ trợ cho các chức năng của người dùng hệ thống như xem gợi ý phim, chọn rạp phim, báo cáo, quản lý thông tin cá nhân.
- Domain: Gói này chứa các phần tử thiết kế để hỗ trợ về mặt người dùng, rạp phim, quản trị viên, .
- User (from Domain): Chứa tất cả các phần tử thiết kế cho các lớp xác thực người dùng ngoại trừ quản trị viên.
- Order (from Domain): Chứa tất cả các phần tử thiết kế cho các yêu cầu, dịch vụ lớp để hỗ trợ những thao tác liên quan đến đơn hàng.
- Movie (from Domain): Chứa tất cả các phần tử thiết kế cho các yêu cầu, dịch vụ lớp để hỗ trợ những thao tác liên quan đến phim.
- cinema (from Domain): Chứa tất cả các phần tử thiết kế cho các yêu cầu, dịch vụ lớp để hỗ trợ những thao tác liên quan đến rạp phim.
- Persistence (from Business Services): Gói này chứa các gói dữ liệu để đảm bảo tính nhất quán của dữ liệu. Bốn toán tử: thêm, sửa, xóa, cập nhật là bốn chức năng chính được thực hiện trong các ứng dụng cơ sở dữ liệu.
- ExternalSystemInterfaces (from Business Services): Các lớp truy cập hệ thống bên ngoài được phân vùng vào gói này.
- <subsystem> PaymentSystem (from Business Services): Đóng gói các giao tiếp liên quan đến thanh toán của người dùng trực tiếp trên hệ thống.
- <subsystem> NavigationSystem (from Business Services): Đóng gói các giao tiếp liên quan đến việc điều hướng, dẫn đường tới rạp phim của hệ thống.
- <subsystem> SuggestionSystem (from Business Services): Đóng gói các giao tiếp liên quan đến việc gợi ý phim của hệ thống.

- <subsystem> ManagementSystem (from Business Services): Đóng gói các giao tiếp liên quan đến việc quản lý rạp phim, quản lý đặt vé của hệ thống.

Phụ lục I. Từ điển thuật ngữ

- **Lớp phân tích:** Một lớp được sử dụng để mô hình hóa tương tác giữa môi trường xung quanh của hệ thống, tính kiểm soát, lượng thông tin và hành vi liên kết trong hệ thống (lớp ranh giới, lớp điều khiển, lớp thực thể).
- **Lớp kiến trúc:** Lớp được xây dựng, phát triển và liên quan đến sự phân chia hợp lý các thành phần và chức năng trong hệ thống.
- **Sơ đồ bối cảnh:** Một sơ đồ bối cảnh là một sơ đồ luồng dữ liệu, chỉ có một tiến trình trung tâm lớn bao gồm tất cả mọi thứ trong phạm vi của hệ thống. Nó cho thấy cách hệ thống sẽ nhận và gửi các luồng dữ liệu đến các thực thể bên ngoài có liên quan.
- **Phụ thuộc:** Tồn tại giữa hai yếu tố nếu thay đổi một yếu tố có thể gây ra thay đổi cho yếu tố khác.
- **Phần tử thiết kế:** Các lớp phân tích được làm mịn thành các phần tử mô hình thiết kế (các lớp thiết kế, các gói và các hệ thống con).
- **Giao diện (interface):** Giao diện là một phần tử mô hình, được định nghĩa là một tập các hành vi (một tập các hoạt động) được cung cấp bởi một phần tử mô hình (cụ thể là một lớp, hệ thống con hay một thành phần).
- **Lớp (layer):** Một nhóm các lớp (classes), các gói (packages) hoặc hệ thống con (subsystems) cùng chịu trách nhiệm về một khía cạnh chính của hệ thống.
- **Gói (package):** Một cơ chế có mục đích chung để tổ chức các phần tử thành các nhóm. Nó cung cấp khả năng tổ chức mô hình đang được phát triển. Một gói được thể hiện dưới dạng một thẻ thư mục.
- **Hệ thống con (subsystem):** Được sử dụng như một đơn vị hành vi trong hệ thống, nó cung cấp khả năng đóng gói hoàn toàn các tương tác của một số lớp và/ hoặc hệ thống con.
- **Khả năng hiển thị (visibility):** Cho phép chỉ định cách các gói khác có thể truy cập vào các phần tử được sở hữu bởi các gói. Có ba loại visibility là: Public (+), Protect (#) và Private (-).