

1. System Introduction

[System Name]

Musical Reservation System

[System Description]

1. 티켓 예매/환불(사용자)

고객이 pc 또는 휴대폰에서 뮤지컬을 예약하기 위해 예약을 진행하려는 사이트에 들어가 원하는 뮤지컬을 고른다. 뮤지컬을 선택하면 뮤지컬 이름, 공연장 이름이 안내된다. 관람일, 관람 시간대, 티켓 수를 고른 후 해당하는 날짜와 시간대의 좌석 정보를 고른다. 티켓의 정보와 함께 옵션에 대한 선택이 끝나면 결제가 진행된다. 결제화면에서는 결제하려는 티켓 정보를 확인한 후 개인정보와 카드 정보를 입력하여 결제를 진행한다.

고객은 개인정보가 검증되면 예매내역을 확인할 수 있으며 예매내역이 있으면, 관람 당일 이전의 예매내역에 대해 환불(또는 변경을 위한 환불)을 할 수 있다. 고객은 수수료를 제외한 최종 티켓을 환불 받는다.

2. 뮤지컬 관리

관리자는 관리자모드로 들어가서 뮤지컬 객체를 삭제하거나 추가할 수 있다. 또한 뮤지컬의 좌석에 대한 상태를 변경할 수 있다.

[Vision]

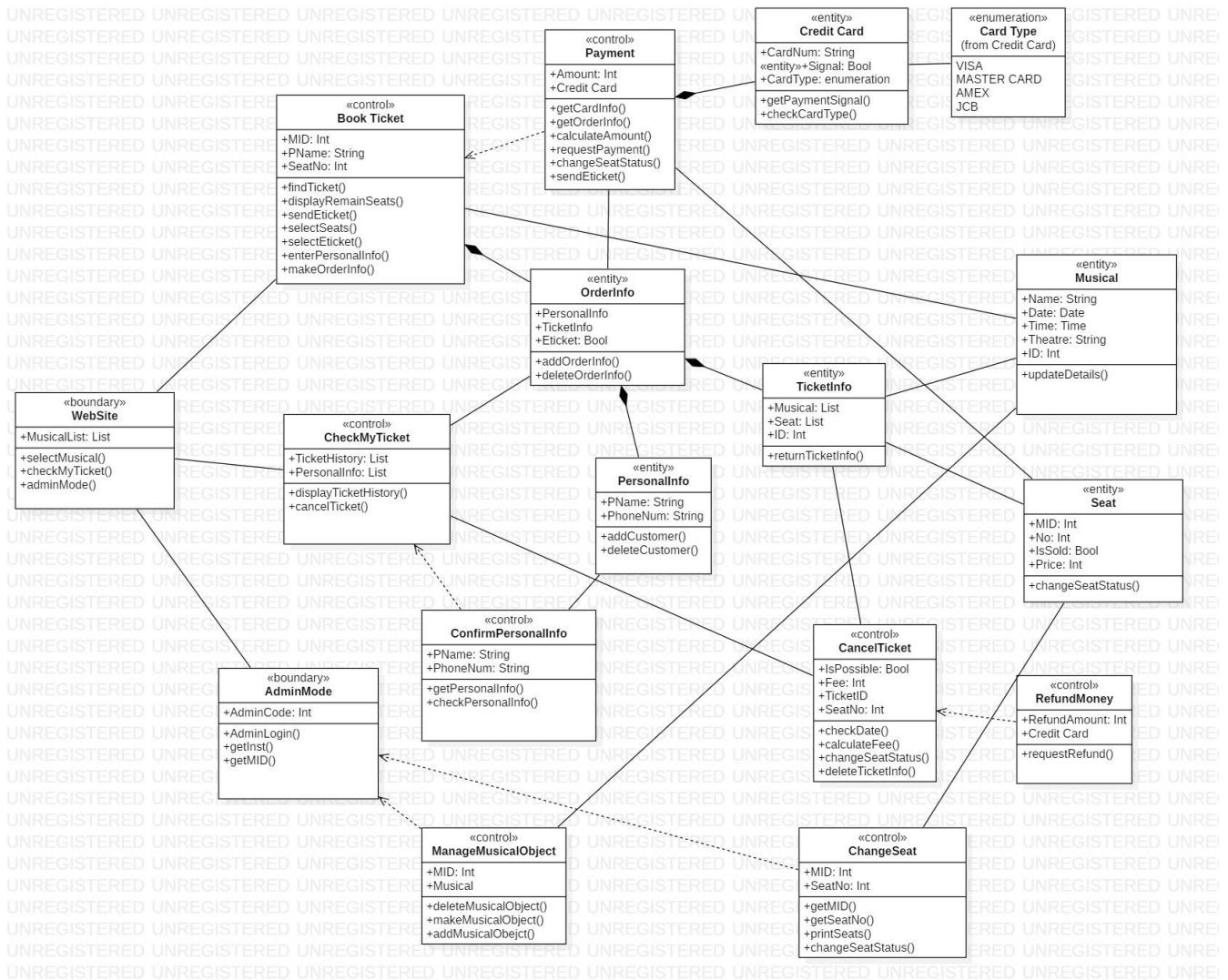
뮤지컬 티켓 예약을 위한 런던 뮤지컬 예매 사이트 시스템 설계

[Scope]

런던에서 상영하는 뮤지컬을 예매할 수 있는 사이트로 카드 결제만 가능하며 티켓 수량은 포함하지 않는다.

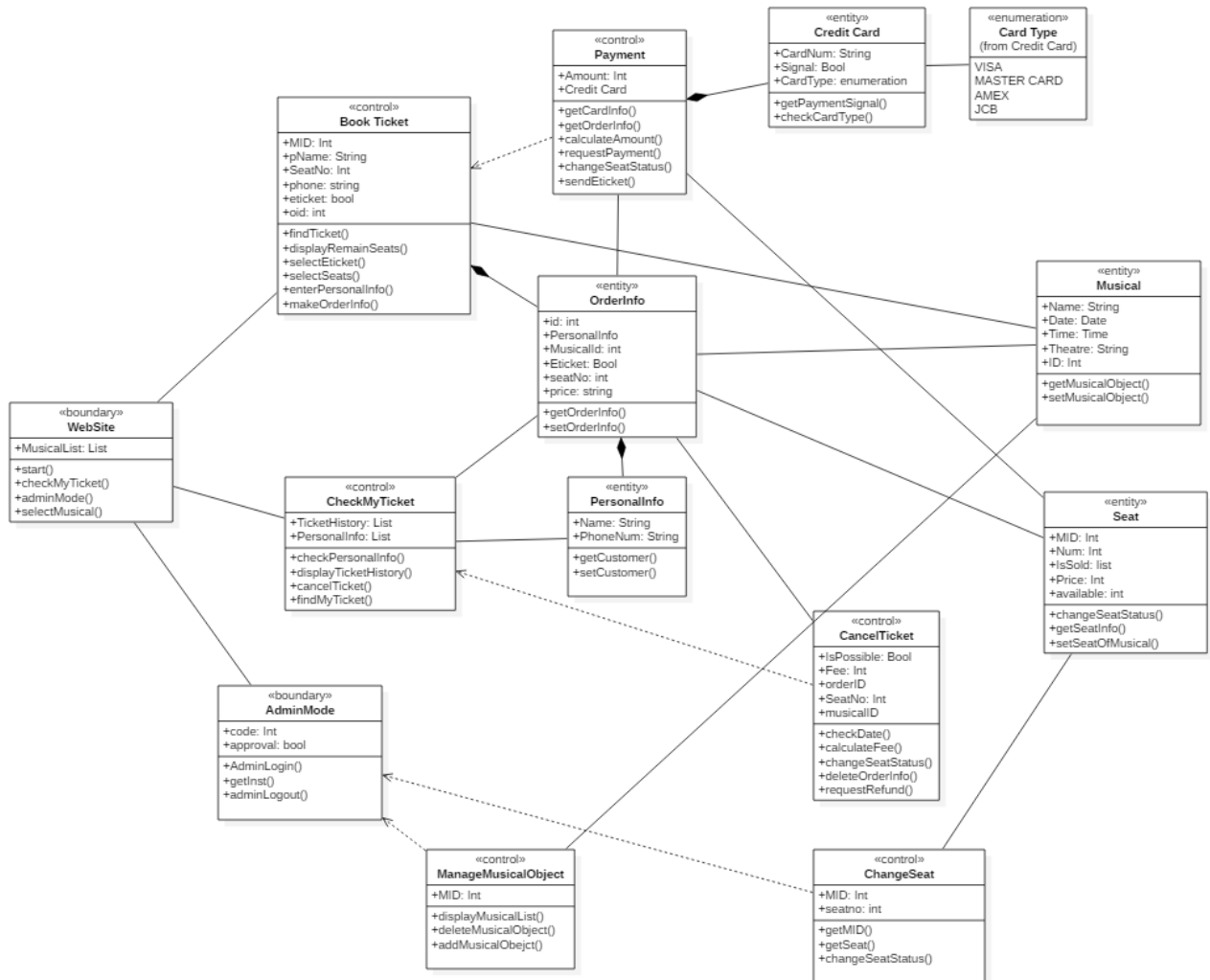
2. Class Diagram

[Before: in Hw4]



클래스가 불필요하게 나누어져 있어 RefundMoney 클래스 처럼 함수로 구현 가능한 기능이 클래스로 나와있는 경우도 있었다. 또한 entity 클래스에서 중요한 역할을 하는 get과 set 함수들이 부족하다. 또한 CancelTicket과 CheckMyTicket이 단순히 연관 관계로 이어져 있다.

[After: Refined in Hw5]



TicketInfo 클래스를 삭제하고 OrderInfo에 seatNo과 price Attribute로 추가하였다.

ConfirmPersonalInfo라는 클래스는 checkPersonalInfo() 함수로 축소하여 checkMyTicket 클래스로 넣었다.

CancelTicket에 dependency 관계인 RequestRefund 클래스를 삭제하여 CancelTicket에 함수로 넣었다.

CancelTicket 클래스는 CheckMyTicket에 dependency 관계로 수정하였다.

각 entity 클래스에 필요에 따라 get과 set기능을 함수를 추가하였다.

3. Design Refinement Explanation

Before 단계	After 단계	적용한 설계 개념	Architecture Design Rationale	기타
Musical(entity class)	Musical(entity class)	High Cohesion 적용	Get과 Set 기능을 하는 함수를 추가하여 외부 클래스에서 해당 클래스를 참조하여 사용할 때 불필요한 함수를 구현하지 않고 이식성 있게 쓸수 있다.	
Website(boundary class)	Website(boundary class)에 start 함수	Low Coupling 적용 Singleton Pattern	Website에서 start함수와 기타의 옵션선택 클래스들을 분리하여 Model-View를 분리시켰다. 또한 website 클래스는 인스턴스가 하나만 존재한다.	
Book Ticket(controller class)	Book Ticket(controller class)	Controller class 구현 Creator	OrderInfo를 만드는 Creator 클래스이며, controller 역할을 하여 Payment까지 dependency 관계로 연결시켰다.	
ManageMusicalObject(controller class)	ManageMusicalObject(controller class)	Controller class 구현 Pure Fabrication High cohesion	뮤지컬 객체를 관리하는 기능을 제공하는 역할을 한곳으로 모아 구현하였다.	