21 - Facility Booking Management System

Ao Liu Liangjun Song Vivian Lai

Use Case 1: Make a booking

Use Case 2: Receive email notification

Use Case 3: Override a booking

Design Pattern

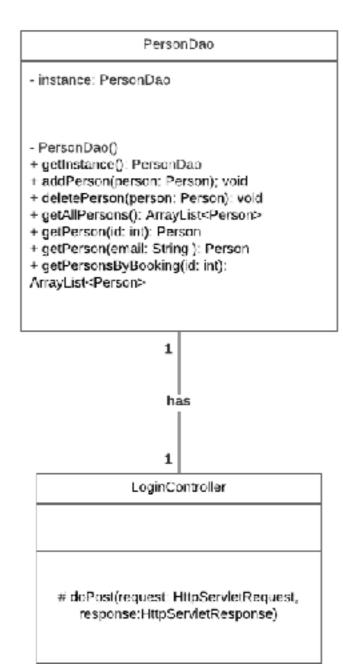
Singleton

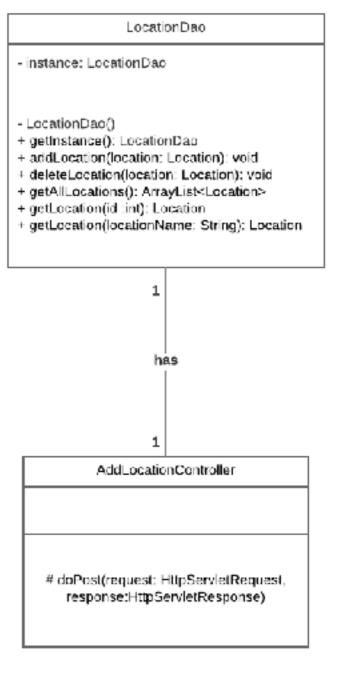
- What was applied?
 - Data Access Object (DAO): interface that provides access to database
- Why?
 - Should only have one instance of each DAO

BookingDao instance : BookingDao BookingDao() + getInstance(): BookingDao + addBooking(booking: Booking): int + deleteBooking(booking: Booking); void + getAllBookings(): ArrayList<Booking> + getBooking(id: int): Booking + getBookingsByPerson(int id): ArrayList<Booking> -+ getBookingsByLocation(id: int): ArrayList<Booking> updateBooking(booking: Booking): void has EditBookingController

doPost(request: HttpServletRequest,

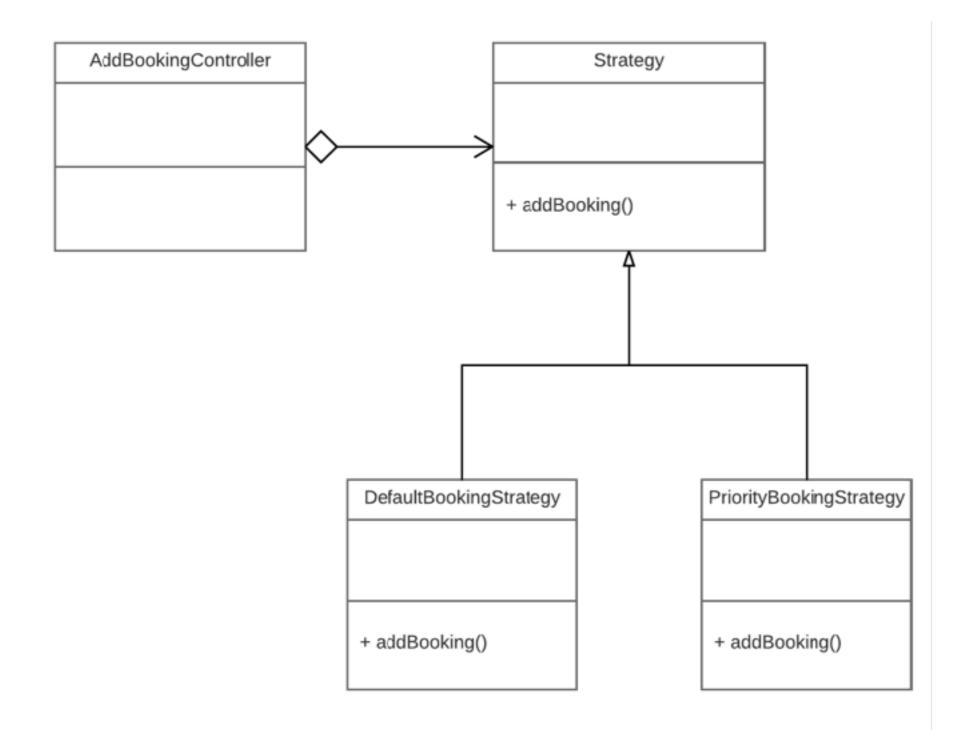
response:HttpServletResponse)





Strategy

- How could it be applied?
 - addBooking method
- Why?
 - Student and Admin have different "rights"



Thank you

Link to demo used during presentation:

http://tinyurl.com/ooad5448presentation-demo-vid