دانشگاه فردوسی مشهد / دانشکده علوم ریاضی

مشخصات پروژه برنامه‌سازی پیشرفته

سیستم مدیریت هتل

Hotel Management System (HMS)

نیمسال تحصیلی: ۱۴۰۴-۱۴۰۵

زبان پیاده‌سازی: Java

⚠ این پروژه یک سیستم شبیه‌سازی‌شده است

۱. داستان و سناریوی کامل پروژه

۱.۱ داستان پروژه

تصور کنید شما به‌عنوان توسعه‌دهنده ارشد نرم‌افزار استخدام شده‌اید تا سیستم مدیریت یک هتل چندطبقه را از صفر طراحی و پیاده‌سازی کنید. این هتل «هتل گرند پرشیا» نام دارد و دارای ۵۰ اتاق از چهار نوع مختلف است. مدیریت هتل از شما خواسته یک سیستم کامل کنسولی طراحی کنید که تمام عملیات روزانه هتل را پوشش دهد.

۱.۲ وضعیت فعلی هتل (مشکل)

در حال حاضر هتل تمام کارها را کاغذی انجام می‌دهد و با مشکلات زیر روبرو است:

- رزرو اتاق‌ها به‌صورت دستی انجام می‌شود و اشتباهات زیادی رخ می‌دهد (مثلاً یک اتاق دو بار رزرو می‌شود)
- اطلاعات مهمانان در دفترهای کاغذی نگه داشته می‌شود و گم می‌شود
- محاسبه صورت‌حساب نهایی با تخفیف‌ها و مالیات به‌صورت دستی انجام می‌شود
- هیچ سیستم گزارش‌گیری وجود ندارد و مدیر نمی‌داند اتاق‌های خالی کدام‌اند
- کارکنان مختلف به اطلاعات یکدیگر دسترسی دارند و امنیت وجود ندارد

۱.۳ راه‌حل مورد نیاز

شما باید یک سیستم نرم‌افزاری بنویسید که تمام این مشکلات را حل کند. این سیستم باید:

- فقط در حافظه (RAM) کار کند — نه پایگاه‌داده، نه فایل (مگر امتیاز اضافه)
- از طریق کنسول (CLI) با کاربر تعامل داشته باشد
- چندین نقش کاربری با دسترسی‌های متفاوت داشته باشد
- تمام قوانین تجاری هتل را پیاده‌سازی کند

۲. کاربران سیستم — نقش‌ها و دسترسی‌ها

۲.۱ توضیح کامل هر نقش

الف) SuperAdmin — مدیر ارشد سیستم

این شخص بالاترین سطح دسترسی را دارد. فقط یک نفر SuperAdmin در سیستم وجود دارد و حساب آن به‌صورت خودکار هنگام راه‌اندازی سیستم ساخته می‌شود:

مقدار	ویژگی
admin	نام کاربری
admin۱۲۳	رمز عبور

خودکار هنگام اجرای برنامه	ایجاد
همه بخش‌ها بدون محدودیت	دسترسی

کارهایی که SuperAdmin می‌تواند انجام دهد:

- ساختن حساب HotelManager
- مشاهده و تغییر تنظیمات سیستم (قیمت پایه اتاق‌ها، ضریب فصلی، و...)
- مشاهده گزارش‌های کامل مالی هتل
- حذف یا غیرفعال کردن حساب‌های کاربری
- مشاهده تمام لاگ‌های سیستم

ب) HotelManager — مدیر هتل

توسط SuperAdmin ساخته می‌شود. مسئول مدیریت روزانه هتل است:

- اضافه کردن و حذف اتاق‌ها از سیستم
- ساختن حساب Receptionist
- مشاهده تمام رزروها، گزارش اشغال، و گزارش مالی
- تغییر وضعیت اتاق‌ها (مثلاً قرار دادن در وضعیت MAINTENANCE)
- تأیید یا رد درخواست‌های تعمیرات

ج) Receptionist — منشی / پذیرش

توسط HotelManager ساخته می‌شود. کارهای روزمره رزرواسیون را انجام می‌دهد:

- ثبت رزرو جدید برای مهمانان
- انجام Check-In مهمان (تبدیل رزرو به اقامت فعال)
- انجام Check-Out و صدور فاکتور نهایی
- جستجوی اتاق‌های خالی بر اساس تاریخ و نوع
- ثبت درخواست‌های خدماتی مهمانان (مینی‌بار، تأسیسات، و...)

د) Guest — مهمان

خود مهمان از طریق ثبت‌نام حساب می‌سازد. دسترسی محدودتری دارد:

- جستجوی اتاق‌های خالی
- مشاهده رزروهای خودش
- درخواست لغو رزرو (با احتساب جریمه)
- مشاهده تاریخچه اقامت‌ها
- مشاهده سطح عضویت و تخفیف‌های فعال

۳. ماژول‌های پروژه — توضیح دقیق چیزی که باید پیاده‌سازی شود

ماژول ۱: مدیریت اتاق‌ها (Room Management)

چه کلاس‌هایی باید بنویسید؟

یک کلاس انتزاعی Room بنویسید که پایه تمام اتاق‌هاست. این کلاس باید فیلدهای زیر را داشته باشد:

توضیح	نام فیلد	نوع داده
شماره اتاق (مثل ۱۰۱، ۲۰۵)	roomNumber	String
نوع اتاق: STANDARD / DELUXE / SUITE / PENTHOUSE	type	RoomType (Enum)
وضعیت: AVAILABLE / RESERVED / OCCUPIED / MAINTENANCE / OUT_OF_SERVICE	status	RoomStatus (Enum)
قیمت پایه اتاق به ریال	basePrice	double
شماره طبقه	floorNumber	int
حداکثر ظرفیت نفر	capacity	int

سپس چهار کلاس فرزند پیاده‌سازی کنید:

ویژگی‌های اضافه	ضریب قیمت	کلاس
اتاق معمولی با سرویس پایه	۱.۰ × (پایه)	StandardRoom
بالکن + مینی‌بار رایگان	۱.۵ ×	DeluxeRoom
جکوزی + سالن پذیرایی جداگانه	۲.۵ ×	Suite
طبقه آخر + سرویس VIP + ویو پانوراما	۵.۰ ×	PentHouse

فرمول محاسبه قیمت نهایی هر شب

این فرمول را باید در متد calculatePrice() پیاده‌سازی کنید:

```
قیمت نهایی = قیمت پایه × ضریب نوع اتاق × ضریب فصلی × ضریب تعداد مهمان × (۱ - تخفیف عضویت)
```

توضیح	مقادیر ممکن	پارامتر
بر اساس تاریخ ورود	۱.۵ (تابستان/اوج)، ۱.۲ (بهار/پاییز)، ۰.۸ (زمستان/کم‌رونق)	ضریب فصلی

هر مهمان اضافه ۲۰٪ افزایش	۱ نفر=۱.۰، ۲ نفر=۱.۲، ۳+نفر=۱.۴	ضریب تعداد مهمان
بر اساس سطح عضویت مهمان	BRONZE=۰٪، SILVER=۵٪، GOLD=۱۰٪، PLATINUM=۱۵٪	تخفیف عضویت

خطاهایی که باید پرتاب کنید

- اگر کسی سعی کرد اتاق در وضعیت MAINTENANCE را رزرو کند: RoomNotAvailableException
- اگر کسی بدون دسترسی وارد بخشی شد: AccessDeniedException
- اگر شماره اتاق وجود نداشت: RoomNotFoundException

ماژول ۲: موتور رزرو (Reservation Engine)

چرخه کامل یک رزرو را بفهمید:

هر رزرو از لحظه ساخت تا تسویه‌حساب، مراحل زیر را طی می‌کند:

چه اتفاقی می‌افتد	وضعیت اتاق	وضعیت رزرو	مرحله
اتاق قفل می‌شود، کد رزرو تولید می‌شود	RESERVED	PENDING	ساخت رزرو
پرداخت پیش‌پرداخت ثبت می‌شود	RESERVED	CONFIRMED	تأیید پرداخت
مهمان وارد اتاق می‌شود	OCCUPIED	ACTIVE	Check-In
فاکتور نهایی صادر، اتاق آزاد می‌شود	AVAILABLE	COMPLETED	Check-Out
جریمه محاسبه، مبلغ مسترد می‌شود	AVAILABLE	CANCELLED	لغو

قانون جریمه لغو — این را باید دقیقاً پیاده‌سازی کنید:

مثال	جریمه	زمان لغو نسبت به تاریخ ورود
رزرو ۱,۰۰۰,۰۰۰ ریال → جریمه ۱,۰۰۰,۰۰۰ ریال	۱۰۰٪ مبلغ کل	کمتر از ۲۴ ساعت مانده
رزرو ۱,۰۰۰,۰۰۰ ریال → جریمه ۵۰۰,۰۰۰ ریال	۵۰٪ مبلغ کل	بین ۱ تا ۳ روز مانده
استرداد کامل مبلغ	بدون جریمه	بیشتر از ۳ روز مانده

الگوی Observer برای اطلاع‌رسانی — این یک الزام است:

وقتی یک اتاق آزاد می‌شود (Check-Out یا لغو رزرو)، سیستم باید به‌صورت خودکار به اولین نفری که در لیست انتظار آن اتاق است اطلاع دهد. برای این کار باید Observer Pattern پیاده‌سازی کنید:

- Interface: RoomObserver با متد onRoomAvailable(String roomNumber)
- کلاس WaitlistManager این اینترفیس را پیاده‌سازی می‌کند
- وقتی اتاق آزاد شد، تمام observer ها notify می‌شوند

ماژول ۳: مدیریت مهمانان و سیستم عضویت

پروفایل مهمان چه اطلاعاتی دارد؟

توضیح	نوع	فیلد
نام کامل مهمان	String	fullName
کد ملی (۱۰ رقم، یکتا در سیستم)	String	nationalId
شماره تماس	String	phone
BRONZE / SILVER / GOLD / PLATINUM	Enum	membershipLevel
تعداد کل اقامت‌های قبلی (برای ارتقای سطح)	int	totalStays
لیست همه رزروهای این مهمان	<List<Reservation	reservationHistory

قانون ارتقای خودکار سطح عضویت:

هر بار که مهمان Check-Out می‌کند، سیستم باید به‌صورت خودکار بررسی کند آیا سطح عضویت او باید ارتقا یابد:

امتیاز ویژه	تخفیف	شرط ارتقا	سطح عضویت
-	بدون تخفیف	پیش‌فرض (همه با این شروع می‌کنند)	BRONZE
-	۵٪ تخفیف	۵ اقامت یا بیشتر	SILVER
پارکینگ رایگان	۱۰٪ تخفیف	۱۵ اقامت یا بیشتر	GOLD
ارتقا به بهترین اتاق موجود	۱۵٪ تخفیف	۳۰ اقامت یا بیشتر	PLATINUM

ماژول ۴: سیستم مالی (Financial System)

فرمول کامل محاسبه فاکتور — هیچ چیزی را فراموش نکنید:

مثال	نحوه محاسبه	آیتم	ردیف
۵۰۰,۰۰۰ × ۳ = ۱,۵۰۰,۰۰۰	قیمت شبانه × تعداد شب‌ها	هزینه اقامت	۱
+ ۲۵۰,۰۰۰	جمع سفارشات (مینی‌بار، لاندری، و...)	هزینه خدمات	۲
− ۱۷۵,۰۰۰ (۱۰٪ GOLD)	کسر از جمع اقامت + خدمات	تخفیف عضویت	۳
+ ۱۵,۷۵۰	۱٪ از مبلغ پس از تخفیف	عوارض شهری	۴
+ ۱۴۱,۷۵۰	۹٪ از مبلغ پس از تخفیف	ارزش افزوده	۵
۱,۷۳۲,۵۰۰ ریال			جمع کل

پرداخت قسطی — الزامی است:

- سیستم باید پرداخت جزئی (Partial Payment) را پشتیبانی کند
- هر بار که مهمان مبلغی پرداخت کرد، مانده بدهکاری به‌روز شود
- در Check-Out نمی‌توان اگر بدهی معوق وجود داشته باشد خارج شد
- تمام تراکنش‌های مالی باید با تاریخ و ساعت ثبت شوند

ماژول ۵: خدمات جانبی (Services)

انواع خدماتی که مهمان می‌تواند سفارش دهد:

نکات	قیمت نمونه	نوع خدمت
باید لیست آیتم‌ها قابل تعریف باشد	متغیر بر اساس آیتم	مینی‌بار (اقلام مصرفی)
فقط در ساعات کاری	۵۰,۰۰۰ ریال	نظافت فوری اتاق (Extra Cleaning)
رایگان برای GOLD به بالا	۸۰,۰۰۰ ریال/روز	پارکینگ اضافه
باید به HotelManager اطلاع داده شود	رایگان	درخواست تعمیرات اتاق
باید رسید ثبت شود	بر اساس منو	سرویس اتاق (Room Service)

ماژول ۶: گزارش‌گیری (Reporting)

گزارش‌هایی که سیستم باید تولید کند (همه در کنسول چاپ شوند):

محتوا	مخاطب	نام گزارش	کد گزارش
تعداد خالی/پر/تعمیر به تفکیک نوع	همه	وضعیت لحظه‌ای اتاق‌ها	R-۰۱
درصد اشغال هر روز	Manager, SuperAdmin	گزارش اشغال روزانه	R-۰۲
درآمد به تفکیک نوع اتاق	Manager, SuperAdmin	گزارش درآمد ماهانه	R-۰۳
بر اساس تعداد رزرو	Manager	محبوب‌ترین اتاق‌ها	R-۰۴
مهمانانی که بدهی معوق دارند	Manager, SuperAdmin	لیست بدهکاران	R-۰۵
همه اقامت‌های یک مهمان خاص	Receptionist, Manager	تاریخچه مهمان	R-۰۶

ماژول ۷: لاگ‌گذاری و امنیت (Security & Logging)

هر عملیات مهم در سیستم باید در یک لاگ ثبت شود. ساختار هر لاگ:

```
[2024-11-15 14:32:05] [INFO] [User: receptionist1] [Action: CHECK_IN] [Room: 205]
[Guest: ALI AHMADI]

[2024-11-15 14:35:12] [ERROR] [User: guest2] [Action: ACCESS_DENIED] [Attempted:
Admin Panel]
```

رویدادهایی که باید لاگ شوند:

- ورود و خروج هر کاربر
- هر عمل CRUD روی رزرو (ساخت، تغییر، لغو)
- هر Check-In و Check-Out
- هر تراکنش مالی
- هر تلاش برای دسترسی غیرمجاز

۴. معماری کلاس‌ها — آنچه باید پیاده‌سازی کنید

۴.۱ سلسله‌مراتب ارث‌بری

توضیح	نوع	کلاس / اینترفیس
پایه همه کاربران - فیلد name و username	Abstract Class	Person (Abstract)
اضافه کردن password و role	→ Abstract Class Person	User (Abstract)
membershipLevel، totalStays، reservationHistory	Class → User	Guest (Concrete)
employeeId، hireDate	→ Abstract Class User	Staff (Abstract)
shiftType (صبح/عصر/شب)	Class → Staff	Receptionist (Concrete)
departmentName	Class → Staff	HotelManager (Concrete)
systemConfigAccess = true	Class → Staff	SuperAdmin (Concrete)
basePrice، status، roomNumber، capacity	Abstract Class	Room (Abstract)
هر کدام ضریب قیمت خود را override می‌کنند	→ Concrete Room	StandardRoom، DeluxeRoom، Suite، PentHouse
dates، room، guest، reservationId، status	Class	Reservation (Concrete)
balance، payments، itemizedBill	Class	Invoice (Concrete)

۴.۲ اینترفیس‌های اجباری

چه کلاس‌هایی پیاده‌سازی می‌کنند	متدهای اجباری	اینترفیس
Reservation، Invoice	applyDiscount، calculateTotal()	Billable
Room، Guest، Reservation	search(String query) → List<T>، filter(Predicate<T>) → List<T>	Searchable<T>
ReservationEngine	registerObserver()، removeObserver()، notifyObservers()	Notifiable

Invoice، Report	exportToText() → String (جدول ASCII)	Exportable

۴.۳ Enum های اجباری

مقادیر	نام Enum
AVAILABLE, RESERVED, OCCUPIED, MAINTENANCE, OUT_OF_SERVICE	RoomStatus
STANDARD, DELUXE, SUITE, PENTHOUSE	RoomType
PENDING, CONFIRMED, ACTIVE, COMPLETED, CANCELLED	ReservationStatus
BRONZE, SILVER, GOLD, PLATINUM	MembershipLevel
SUPER_ADMIN, HOTEL_MANAGER, RECEPTIONIST, GUEST	UserRole
PEAK (تابستان، ضریب ۱.۵)، NORMAL (بهار/پاییز، ۱.۲)، OFF (زمستان، ۰.۸)	Season

۵. Exception های سفارشی — الزامی است

باید Exception های زیر را بنویسید و در جاهای مناسب throw کنید:

کِی throw می‌شود	نام Exception
رزرو اتاق در وضعیت غیر AVAILABLE	RoomNotAvailableException
اتاق با شماره وارد شده وجود ندارد	RoomNotFoundException
تلاش برای عمل غیرمجاز بر اساس نقش	AccessDeniedException
تداخل تاریخ با رزرو دیگری	ReservationConflictException
مبلغ پرداختی کمتر از حداقل لازم	InsufficientPaymentException
مهمان با کد ملی وارد شده پیدا نشد	GuestNotFoundException
تاریخ خروج قبل از تاریخ ورود است	InvalidDateRangeException

۶. سناریوهای نمونه برای تست — چه چیزی باید کار کند

سناریو ۱: ثبت‌نام مهمان جدید و اولین رزرو

نتیجه مورد انتظار	عمل	گام
حساب Guest با سطح BRONZE ساخته می‌شود	مهمان نام کاربری 'ali.tehrani' را ثبت می‌کند	۱
لیست اتاق‌های DeluxeRoom موجود نمایش داده می‌شود	مهمان اتاق‌های خالی از نوع DELUXE برای ۱۰–۱۵ مهر جستجو می‌کند	۲
رزرو با کد R-۰۰۰۰۱ در وضعیت PENDING ساخته می‌شود، اتاق RESERVED می‌شود	مهمان اتاق ۲۰۵ را برای ۲ نفر رزرو می‌کند	۳
رزرو به ACTIVE تبدیل می‌شود، اتاق OCCUPIED می‌شود	منشی Check-In را انجام می‌دهد	۴
خدمت به رزرو اضافه می‌شود	مهمان یک قلم مینی‌بار ۴۵,۰۰۰ ریالی سفارش می‌دهد	۵
فاکتور دقیق با اقامت + مینی‌بار + مالیات چاپ می‌شود	منشی Check-Out انجام می‌دهد	۶

سناریو ۲: لغو رزرو با جریمه

نتیجه مورد انتظار	عمل	گام
وضعیت CONFIRMED است	رزرو ۲ روز دیگر Check-In دارد	۱
سیستم هشدار می‌دهد: جریمه ۵۰٪ اعمال می‌شود	مهمان درخواست لغو می‌کند	۲
رزرو CANCELLED می‌شود، ۵۰٪ استرداد می‌شود، اتاق AVAILABLE می‌شود	مهمان تأیید می‌کند	۳
سیستم به‌صورت خودکار پیام ارسال می‌کند (Observer)	نفر بعدی در لیست انتظار	۴

سناریو ۳: ارتقای خودکار سطح عضویت

وضعیت	گام
تخفیف ۵٪ دریافت می‌کند	مهمان 'sara.k' اکنون ۱۴ اقامت دارد (سطح SILVER)
سیستم totalStays را ۱۵ می‌کند	Check-Out پانزدهم انجام می‌شود

شرط GOLD برقرار است → سطح به GOLD تغییر می‌کند	سیستم checkMembershipUpgrade() را صدا می‌زند
«تبریک! شما به سطح GOLD ارتقا یافتید. از این پس پارکینگ رایگان دارید»	پیام ارتقا نمایش داده می‌شود

۷. نمونه تعامل کنسول — UI باید اینطور باشد

بخش‌های مختلف منو باید به‌صورت واضح و زیبا نمایش داده شوند. مثال:

```
╔═══════════════════════════════════════════════╗
║═══════════════════════════════════════════════╕
╒═══════════════════════════════════════════════╕
╔═══════════════════════════════════════════════╗
╔═══════════════════════════════════════════════╗
╔═══════════════════════════════════════════════╗
```

۸. معیارهای ارزیابی(۳ نمره)

معیار کلیدی	ماژول
ارث‌بری صحیح، Enum ها، فرمول قیمت	ماژول ۱: کلاس‌بندی اتاق‌ها
چرخه کامل، جریمه لغو، Observer Pattern	ماژول ۲: موتور رزرو
پروفایل کامل، ارتقای خودکار عضویت	ماژول ۳: مدیریت مهمانان
فاکتور دقیق، مالیات، پرداخت قسطی	ماژول ۴: سیستم مالی
ثبت سفارشات، اضافه شدن به فاکتور	ماژول ۵: خدمات جانبی
حداقل ۴ گزارش کار کند	ماژول ۶: گزارش‌گیری
کنترل دسترسی + ثبت رویدادها	ماژول ۷: لاگ و امنیت
Generic ها، اینترفیس‌ها	کیفیت کد و OOP

امتیاز اضافه (۱ نمره)

توضیح	امتیاز اضافه
داده‌ها بعد از بستن برنامه حفظ شوند	ذخیره‌سازی در فایل (I/O)
پیاده‌سازی با Swing یا JavaFX	رابط کاربری گرافیکی

۹. محدودیت‌ها و الزامات فنی

- زبان: Java
- **این پروژه ۳ نمره اصلی و ۱ نمره اضافه رو شامل میشه**
- **درصورتی که مشخص شه کد کامل توسط هوش مصنوعی نوشته شده هیچ نمره ای تعلق نمی گیره**
- **در صورت عدم شرکت در ارایه انلاین نمره ای تعلق نمی گیرد**
- ساختار: فقط OOP خالص — هیچ ابزار فریم‌ورک یا کتابخانه خارجی مجاز نیست
- کنسول: تمام تعاملات از طریق Scanner و System.out
- Generic: حداقل یک Generic Class یا Interface پیاده‌سازی شود
- Exception: همه Exception های سفارشی تعریف‌شده باید استفاده شوند

۱۰. تحویلی‌ها

- سورس کد
- فایل داکیونت با دستور اجرا و توضیح تمام فیلد ها و کلاس های پیاده شده
- ارایه پروژه به صورت انلاین

نکته مهم برای دانشجویان

این پروژه یک سیستم واقعی تجاری را شبیه‌سازی می‌کند.

هدف اصلی: یاد گرفتن اینکه چطور یک سیستم پیچیده را با اصول OOP صحیح طراحی و پیاده‌سازی کنید. پیش از نوشتن کد، ابتدا دیاگرام کلاس‌ها را طراحی کنید. سپس از پایین‌ترین سطح (Enum ها و کلاس‌های ساده) به بالا پیش بروید.(صرفا برای اینکه در پیاده سازی راحتر باشید نیازی به ارسال نداره)

موفق باشید! 🏢