مینی پروژه درس برنامه نویسی پیشرفته

مستندات فاز دوم مینی پروژه: پیاده‌سازی سیستم پویا و هوشمند وفاداری کاربران

دکتر ثبوتی

بهار ۱۴۰۵

مقدمه

در فاز اول پروژه، ساختار پایه سامانه مدیریت سفارش آنلاین غذا شامل مدیریت رستوران‌ها، منوها، کاربران و فرآیند اصلی ثبت سفارش در محیط ترمینال (کنسول) پیاده‌سازی شد. هدف از فاز دوم، ارتقای این ساختار پایه به یک سیستم تجاری هوشمندتر و پویا تر از طریق پیاده‌سازی **«ماژول عضویت و وفاداری کاربران با سطوح کاربری پویا»** است.

در این فاز، تمرکز اصلی بر به‌کارگیری پیشرفته مفاهیم شی‌گرایی، به‌ویژه **Polymorphism**، مفهوم **Composition** و استفاده عملی از **الگوهای طراحی (Design Patterns)** مانند الگوهای وضعیت/استراتژی (State/Strategy) به جای شرط‌های متوالی (if-else) است. مطابق با محدودیت‌های ترم جاری، این فاز نیز به صورت کامل در **محیط متنی و ترمینال (کنسول)** پیاده‌سازی شده و تمامی داده‌های جدید باید به شکل پایدار در پایگاه داده MySQL کپسوله‌سازی و ذخیره شوند.

نکات کلی پروژه:

- دانشجویان می‌بایست پروژه را یک نفره انجام دهند.
- تمامی فازهای پروژه در مخزن گیت به صورت ***تدریجی*** آپدیت شوند.
- هرگونه مشابهت نامتعارف و یا عدم تسلط به کدهای تحویلی قابل پذیرش نیست و دانشجو نمره پروژه را از دست خواهد داد.
- پیاده‌سازی پروژه به تنهایی کافی نیست و نمره کسب شده در نمره ارائه ضرب خواهد شد، لذا انتظار می‌رود دانشجو به بخش‌های پروژه تسلط کافی داشته باشد.
- پروژه شامل نمرات اضافه خواهد بود که دانشجویان با مطالعه و تلاش بیشتر -در صورت تمایل- می‌توانند آن‌ها را کسب کنند.
- **پیاده‌سازی رابط گرافیکی برای هر دو فاز پروژه، شامل ۱۰ درصد نمره اضافه می‌شود.**

عملکرد و ویژگی‌های مورد انتظار از ماژول وفاداری

در این فاز، هر کاربر با نقش «مشتری» دیگر یک عضو ساده نیست، بلکه بر اساس رفتار، تعداد سفارشات و سابقه‌اش در یکی از **سطوح عضویت چهارگانه** قرار می‌گیرد. سیستم باید رفتارهای زیر را به طور دقیق پشتیبانی کند:

الف) سیستم امتیاز وفاداری (Loyalty Points)

- با ثبت هر سفارش موفق، مشتری بر اساس مبلغ خرید خود امتیاز جمع‌آوری می‌کند.
- در صورت لغو سفارش یا مرجوعی، امتیازات مربوط به آن سفارش از حساب کاربر کسر می‌شود.
- جمع‌آوری امتیازها محرک اصلی ارتقای سطح کاربر در سامانه است.

ب) مکانیزم ارتقا و تنزل سطح (Level Upgrading & Downgrading)

- **ارتقا:** هر زمان که مجموع امتیازات کاربر از حد نصاب تعیین‌شده برای هر سطح عبور کند، سیستم باید به صورت خودکار سطح کاربر را ارتقا داده و پیام تبریک مناسبی در کنسول نمایش دهد.
- **تنزل:** ادمین سیستم یا فرآیندهای دوره‌ای سامانه باید بتوانند در صورت عدم فعالیت یا کاهش سفارشات کاربر در یک بازه زمانی، سطح او را تنزل دهند. این بخش نیازمند مدیریت منطق تغییر وضعیت پویا در حافظه و دیتابیس است.

ج) جدول منطقی مشخصات و مزایای سطوح چهارگانه

سیستم باید چهار سطح زیر را با رفتارهای کاملاً متمایز پیاده‌سازی کند:

نام سطح (Level)	حد نصاب امتیاز	ضریب کسب امتیاز (Multiplier)	درصد تخفیف روی سفارش	وضعیت هزینه ارسال	مزایای جانبی
Normal	۰ (شروع اولیه)	1.0x	0%	عادی (کامل)	بدون مزایای ویژه
Silver	۱۰۰ امتیاز	1.2x	5%	کاهش هزینه برای سفارش‌های بالا	امکان دریافت کوپن اختصاصی
Gold	۳۰۰ امتیاز	1.5x	10%	کاهش هزینه ارسال (نیم‌بها)	دسترسی به پیشنهادها و آیتم‌های ویژه منو
VIP	۷۰۰ امتیاز	2.0x	15%	ارسال کاملاً رایگان	اولویت پردازش در صف سفارشات ادمین

د) قابلیت‌های جانبی ماژول

- **سیستم کوپن اختصاصی:** امکان تخصیص کوپن‌های ماهانه هوشمند به کاربران سطوح بالا (Silver: ۱ کوپن، Gold: ۱ کوپن، VIP: ۳ کوپن).
- **نشان‌های کاربری:** تخصیص نشان‌های تشویقی در منوی کنسول به کاربران (مانند *Frequent Buyer* برای خریدهای متوالی، *Night Customer* برای سفارش‌های اواخر شب).
- **تاریخچه ارتقا:** ذخیره و امکان مشاهده تاریخچه تغییرات سطوح کاربر.

الزامات شی‌گرایی، معماری و الگوهای طراحی (Design Patterns)

شما موظف هستید از پیاده‌سازی این سیستم با **ساختارهای شرطی طولانی و تو در تو if-else یا switch-case بر روی نام سطح** به شدت خودداری کنند. معماری فاز دوم باید کاملاً منطبق بر اصول SOLID و طراحی پاک شی‌گرا باشد:

- **به‌کارگیری الگوهای طراحی State یا Strategy:** رفتار مشتری (میزان تخفیف، هزینه ارسال، ضریب امتیاز) باید به یک اینترفیس یا کلاس انتزاعی (Abstract Class) به نام MembershipLevel واگذار شود. کلاس‌های متغیر NormalLevel، SilverLevel، GoldLevel و VipLevel این رفتارها را بازنویسی (Override) می‌کنند. کلاس Customer با داشتن یک اشاره‌گر به اینترفیس فوق، به صورت پویا در زمان اجرا رفتار خود را تغییر می‌دهد.
- **اصل Open/Closed (OCP):** معماری باید به گونه‌ای باشد که اگر در آینده سطح جدیدی به نام *Platinum* تعریف شد، بدون دستکاری در کدهای اصلی بخش سفارشات و مشتری، تنها با افزودن یک کلاس جدید سیستم توسعه یابد.
- **مدیریت ایمن حافظه:س** با توجه به تغییر لول‌ها در زمان اجرا، تخصیص و آزادسازی حافظه اشاره‌گرهای مربوط به لول‌های کاربری باید بدون هیچ‌گونه نشت حافظه (Memory Leak) مدیریت شود.

ذخیره‌سازی و توسعه پایگاه داده (MySQL)

تمامی کدهای جدید پایگاه داده باید مجدداً در کلاس‌های واسط دسترسی به داده (**Data Access Object - DAO**) کپسوله‌سازی شوند و هیچ کوئری مستقیمی نباید در منطق برنامه یا منوهای کنسول نوشته شود.

- ✓ **توسعه جداول:** جداول کاربران در دیتابیس باید به نحوی اصلاح یا افزوده شوند که فیلدهای points امتیاز فعلی، current_level سطح فعلی و جداول مربوط به تغییرات سطوح و کوپن‌ها را پشتیبانی کنند.
- ✓ **عملیات CRUD واسط:** کلاس‌های DAO باید بتوانند در هنگام بارگذاری یک نمونه از Customer، شیء وضعیت سطح مربوطه مثلا GoldLevel را بر اساس داده‌های دیتابیس به درستی نیو (new) کرده و به آبجکت مشتری متصل کنند. در هنگام ثبت سفارش نیز امتیاز جدید و تغییر احتمالی سطح باید در دیتابیس بروزرسانی (Update) شود.

تعامل با کاربر در محیط کنسول

منوهای متنی متناسب با این فاز باید دستخوش تغییرات زیر شوند:

- **منوی مشتری:** در منوی اصلی مشتری، در کنار نام و مشخصات وی، **نشان سطح کاربری فعلی مثلاً VIP Member**، میزان امتیاز فعلی و میزان امتیاز لازم برای ارتقا به سطح بعدی به صورت شفاف نمایش داده شود. هنگام نهایی‌سازی سبد خرید، باید فاکتور تفکیک‌شده شامل: قیمت پایه، میزان تخفیف سطح، هزینه ارسال متناسب با سطح و امتیاز دریافتی از این سفارش به صورت زنده چاپ شود.
- **منوی مدیر سامانه (سیستم):** گزینه‌های جدیدی برای مشاهده گزارش‌های مدیریتی به ادمین اضافه می‌شود؛ شامل:

۱. گزارش آماری تعداد کاربران موجود در هر سطح به عنوان مثال: چند کاربر VIP و چند کاربر Normal در سامانه وجود دارند.

۲. پنل تنظیمات مدیریتی برای اعمال تغییرات دستی بر روی سطح یا امتیاز یک کاربر خاص.

۳. مشاهده لاگ و تاریخچه تغییرات سطح کاربران.

مستندات و گزارش فنی مورد نیاز برای تحویل

دانشجو موظف است گزارش فنی فاز اول خود را به‌روزرسانی کرده و بخش‌های زیر را به آن اضافه کند:

- **نمودار کلاس‌ها (Class Diagram) به روز شده:** نمایش رابطه کلاس Customer با کلاس انتزاعی MembershipLevel و زیرکلاس‌های آن (رابطه Composition و Inheritance فاز دوم).
- **تشریح الگوی طراحی:** توضیح دقیق نحوه پیاده‌سازی پترن انتخاب شده (State/Strategy) و دلیل برتری آن نسبت به کدهای شرطی عادی.
- **طراحی دیتابیس جدید:** نمایش ساختار جداول دستخوش تغییر و نحوه نگهداری وضعیت سطوح پویا.
- **سناریوهای تست (Test Cases):** گزارش متنیِ خروجی کنسول برای حداقل ۳ سناریو: (۱) ثبت سفارش و ارتقای خودکار سطح، (۲) اعمال تخفیف و ارسال رایگان برای یک کاربر VIP ، (۳) تنزل سطح یک کاربر توسط ادمین.