مینی پروژه درس برنامه نویسی پیشرفته

مستندات مینی پروژه

دکتر ثبوتی

بهار ۱۴۰۵

مقدمه

هدف این درس یادگیری برنامه‌نویسی با زبان‌های سطح بالا و مفاهیم برنامه‌نویسی شیءگرا، همراه با پیاده‌سازی رابط کاربری ساده با استفاده از فریم‌ورک‌های این زبان‌ها است. به طور خاص در این درس زبان C++ تدریس می‌شود و فریم‌ورک مورد نظر نیز Qt می‌باشد که ابزار مناسبی برای ایجاد نرم‌افزارهای کاربرپسند است. پیاده‌سازی دقیق پروژه در کنار فراگیری مباحث نظری، یادگیری عمیق مفاهیم درس را تضمین می‌کند.

اما توجه داشته باشید: با توجه به شرایط فعلی این ترم و قطعی اینترنت، امکان دانلود، نصب و استفاده از فریم‌ورک Qt وجود ندارد. بنابراین، پیاده‌سازی رابط کاربری گرافیکی حذف شده و این پروژه **صرفاً به صورت متنی و در محیط ترمینال (کنسول)** پیاده‌سازی خواهد شد تا تمرکز دانشجویان تماماً بر روی درک و اجرای صحیح معماری شیءگرایی قرار گیرد.

نکات کلی پروژه:

- دانشجویان می‌بایست پروژه 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.

بخش اول: هدف و ماهیت پروژه

هدف این پروژه، طراحی و پیاده‌سازی یک سامانه مدیریت سفارش غذای آنلاین مشابه سرویس‌هایی مانند اسنپ‌فود است. تمرکز پروژه بر استفاده صحیح از اصول شیءگرایی در C++، مدیریت حافظه، طراحی معماری مناسب و تعامل با پایگاه داده MySQL است. سامانه باید نماینده یک پلتفرم واقعی سفارش غذا باشد و از نظر رفتار، امکانات و گردش کاری مشابه یک سرویس تجاری عمل کند، هرچند در محیط کنسولی اجرا می‌شود. دانشجو باید بر اساس نیازهای عملکردی سیستم، ساختار کلاس‌ها، روابط میان آن‌ها و نحوه مدیریت داده‌ها را خودش طراحی کند.

بخش دوم: عملکرد مورد انتظار از سامانه

سامانه باید قابلیت مدیریت چندین رستوران را داشته باشد. هر رستوران با مجموعه مشخصی از داده‌ها مدل می‌شود که عبارتند از:

- **شناسه یکتا (ID):** عددی منحصر به فرد برای هر رستوران
- **نام رستوران:** رشته متنی حاوی نام رستوران
- **نشانی دقیق:** اطلاعات محل شامل آدرس کامل
- **وضعیت رستوران:** فعال یا غیرفعال (برای امکان مدیریت نمایش در سامانه)
- **زمان تقریبی آماده‌سازی سفارش:** به دقیقه، عدد صحیح مثبت که نمایانگر مدت زمان استاندارد برای آماده‌سازی یک سفارش در آن رستوران است
- **شماره تماس:** رشته متنی شامل شماره تلفن
- **توضیحات تکمیلی:** شرح کوتاه یا توضیحات دلخواه درباره رستوران

هر رستوران باید یک منوی اختصاصی داشته باشد که شامل مجموعه‌ای از آیتم‌های قابل سفارش است. هر آیتم منو، چه غذا و چه نوشیدنی، باید دارای مشخصات زیر باشد:

- **شناسه آیتم:** عدد یا رشته منحصر به فرد
- **نام آیتم:** نام قابل نمایش برای مشتری
- **شرح آیتم:** توضیحات کوتاه راجع به آیتم
- **قیمت پایه:** عدد اعشاری (بعنوان قیمت اصلی بدون محاسبات اضافی)
- **نوع آیتم:** مشخص می‌کند آیتم غذا، نوشیدنی، یا سایر موارد است (برای چندریختی کاربرد دارد)
- **ویژگی‌های مخصوص آیتم:** متغیرهایی مانند حجم (برای نوشیدنی)، زمان پخت (برای غذا) یا سایر پارامترهای مرتبط که بر قیمت نهایی یا نمایش تاثیر دارند
- **وضعیت موجود بودن:** فعال یا غیرفعال برای کنترل نمایش در منو

در سامانه باید امکان مدیریت کاربران با نقش‌های مختلف فراهم باشد:

- **مشتریان:** قادر به ثبت سفارش، مشاهده منوها و رستوران‌ها، مدیریت سبد خرید، اصلاح یا حذف آیتم‌های انتخاب‌شده در سبد، و نهایی کردن سفارش هستند. مشتری همچنین باید بتواند تاریخچه سفارش‌های خود را مشاهده و وضعیت هر سفارش را در قالب وضعیت‌های مختلف مثل "درحال آماده‌سازی" یا "تحویل داده شده" پیگیری کند.
- **مدیران رستوران:** می‌توانند اطلاعات رستوران خود را مشاهده و ویرایش کنند، منوی رستوران را مدیریت نمایند (اضافه کردن، حذف یا ویرایش آیتم‌ها)، و سفارش‌های دریافت‌شده را مرور کرده و وضعیت سفارش‌ها را بروزرسانی کنند.
- **مدیر سامانه (سیستم):** توانایی ثبت رستوران‌های جدید، فعال/غیرفعال کردن رستوران‌ها، و مشاهده گزارش‌های کلی شامل آمار فروش، تعداد سفارش‌ها و میزان فعالیت کاربران را دارد.

فرآیند ثبت سفارش‌ها باید به شکل زیر دقیق انجام شود: مشتری پس از ورود و انتخاب رستوران، منوی آن رستوران را مشاهده می‌کند، آیتم‌های مورد نظر را به سبد خرید اضافه می‌کند، در هر لحظه مجموع مبلغ سفارش قابل مشاهده است، سپس مراحل پرداخت و نهایی‌سازی سفارش اتفاق می‌افتد. بعد از ثبت سفارش، سیستم باید وضعیت آن سفارش را ذخیره و به مدیر رستوران برای رسیدگی اطلاع دهد. امکان مشاهده و تغییر وضعیت سفارش توسط مدیر رستوران و مشاهده وضعیت سفارش توسط مشتری فراهم است.

بخش سوم: الزامات شی‌گرایی و معماری

پروژه باید استفاده صحیحی از اصول کپسوله‌سازی داشته باشد، به این معنا که داده‌ها نباید مستقیماً قابل دسترسی باشند و باید از طریق متدهای مشخص مدیریت شوند. وراثت باید در بخش‌هایی استفاده شود که در آن نوعی دسته‌بندی طبیعی وجود دارد و رفتار مشترک میان چند نوع شیء قابل مشاهده است. دانشجو باید خودش تشخیص دهد چه جایی وراثت معنا دارد و چه جایی ترکیب یا تجمیع مناسب‌تر است. در بخش‌هایی که اشیاء مختلف رفتار متفاوتی دارند اما تحت یک نوع عمومی قابل مدیریت هستند، باید از چندریختی استفاده شود. به طور طبیعی در این پروژه حداقل منوی رستوران و آیتم‌های مختلف آن نقطه مناسبی برای استفاده از چندریختی است، اما دانشجو آزاد است در محل‌های دیگری نیز از این قابلیت استفاده کند. برنامه باید از لحاظ مدیریت حافظه کاملاً ایمن باشد و تمام اشیائی که با اشاره‌گر ایجاد می‌شوند حتماً به‌درستی آزاد شوند.

بخش چهارم: ذخیره‌سازی و استفاده از MySQL

سامانه باید تمامی داده‌ها را در پایگاه داده MySQL ذخیره کند و به نحوی طراحی شود که تعامل مستقیم بخش‌های مختلف برنامه با کوئری‌های SQL وجود نداشته باشد تا از پراکندگی کدهای پایگاه داده و سختی نگهداری جلوگیری شود. برای این منظور باید یک یا چند کلاس واسط (مانند کلاس DataObjectAccess) طراحی شود که نقش پل بین پایگاه داده و منطق برنامه را ایفا کند. این کلاس‌ها مسئول اجرای کوئری‌ها، خواندن داده‌ها، و برگرداندن اطلاعات به شکل آبجکت‌های مناسب به بخش‌های دیگر برنامه هستند. این ساختار باعث می‌شود اصل کپسوله‌سازی حفظ شود و هیچ بخش دیگری در برنامه مستقیماً با پایگاه داده تعامل نداشته باشد.

کلاس‌های واسط باید به گونه‌ای طراحی شوند که عملیات CRUD (ساخت، خواندن، به‌روزرسانی، حذف) را به طور کامل پوشش دهند و بتوانند داده‌ها را در قالب کلاس‌های دامنه (مثلاً کلاس‌های Restaurant، MenuItem، Order و غیره) بارگذاری و ذخیره کنند. این روش باعث می‌شود بخش‌های مختلف برنامه فقط با آبجکت‌های خودکار تعامل داشته باشند و کد پایگاه داده متمرکز و قابل مدیریت باقی بماند. علاوه بر این، این ساختار امکان سفارشی‌سازی ارتباط با پایگاه داده، تغییر در ساختار دیتابیس و تست جایگزین را تسهیل می‌کند.

بنابراین، دانشجو باید نقش هر کلاس داده را مشخص کند و بین وظایف تعامل با پایگاه داده و منطق برنامه جداسازی کامل انجام شود. دسترسی غیرمستقیم از طریق کلاس‌های واسط الزامی است و بخش‌های بالاتر برنامه نباید کوئری مستقیم ارسال کنند یا داده خام دریافت نمایند.

بخش پنجم: رفتار کنسولی و تعامل با کاربر

بعد از اجرای برنامه، صفحه ورود (Login) به سیستم به کاربر نمایش داده می‌شود که می‌تواند نقش خود را انتخاب کند: مشتری، مدیر رستوران یا مدیر سامانه. ورود موفق به نقش، منوی مخصوص آن نقش را به کاربر نمایش می‌دهد و بسته به این نقش دسترسی‌ها و گزینه‌های قابل مشاهده متفاوت است.

اگر کاربر نقش **مشتری** را انتخاب کند، منوی اصلی او ابتدا فهرستی از رستوران‌های فعال را می‌بیند. برای هر رستوران، نام، نشانی و زمان تقریبی آماده‌سازی سفارش نمایش داده می‌شود. مشتری می‌تواند یک رستوران را انتخاب و وارد منوی آن شود که در آنجا فهرستی از آیتم‌های منوی رستوران به همراه مشخصات کامل مثل نام، توضیح کوتاه، قیمت و وضعیت موجود بودن به مشتری نمایش داده می‌شود. مشتری می‌تواند آیتم‌های دلخواه را به سبد خرید خود اضافه کند، تعداد هر آیتم را تغییر داده یا آیتمی را حذف کند. مجموع قیمت سبد خرید در هر لحظه به صورت زنده نمایش داده می‌شود و پس از اتمام انتخاب‌ها، مشتری می‌تواند سفارش را نهایی کند. پس از ثبت سفارش، شماره سفارش و وضعیت فعلی برای مشتری نمایش داده می‌شود. مشتری همچنین می‌تواند وارد صفحه "تاریخچه سفارش‌ها" شود و لیستی از سفارش‌های پیشین خود به همراه وضعیت هر کدام را مشاهده نماید.

اگر کاربر نقش **مدیر رستوران** داشته باشد، پس از ورود به منوی مدیریتی رستوران، اطلاعات رستوران شامل نام، آدرس، وضعیت و زمان آماده‌سازی نمایش داده می‌شود و امکان ویرایش آن‌ها را دارد. منوی مدیریت آیتم‌های رستوران به وی ارائه می‌شود تا بتواند آیتم‌های جدید اضافه کند، آیتم‌های موجود را ویرایش کرده (تغییر قیمت، توضیح، وضعیت موجود بودن)، یا آیتمی را حذف نماید. همچنین، مدیر دسترسی به لیستی از سفارش‌های جاری رستوران را دارد که شامل شماره سفارش، وضعیت سفارش و اطلاعات مشتری است. مدیر رستوران می‌تواند هر سفارش را انتخاب کرده و وضعیت آن را به مراحل مختلف مانند "در حال آماده‌سازی"، "آماده ارسال" یا "تحویل داده شده" تغییر دهد تا این تغییرات در سامانه ذخیره شود و برای مشتری نیز قابل مشاهده باشد.

اگر کاربر نقش **مدیر سامانه** را انتخاب کند، منوی مدیریتی گسترده‌تری برای نظارت کلی روی سامانه به وی نمایش داده می‌شود. این منو شامل گزینه‌هایی برای ثبت رستوران‌های جدید با تمام اطلاعات لازم، فعال یا غیرفعال کردن رستوران‌های موجود، و مشاهده گزارش‌های کلی مانند مجموع فروش، تعداد سفارش‌ها، تعداد کاربران فعال و سایر نمودارها و آمارهای مدیریتی است.

در تمامی منوها، سیستم باید ورودی‌های نامعتبر را تشخیص دهد و با پیام‌های مناسب کاربر را راهنمایی کند. خطاهای احتمالی باید کنترل شده باشند تا برنامه کرش نکند و تجربه کاربری روان و بدون خطا ارائه گرد.

علاوه بر پیاده‌سازی کامل پروژه، هر دانشجو موظف است یک گزارش فنی از پروژه تهیه کند که در آن طراحی سیستم، معماری برنامه، مفاهیم شی‌گرایی، طراحی پایگاه داده و روند پیاده‌سازی بخش‌های مختلف توضیح داده شده باشد.

- در گزارش باید هدف پروژه و قابلیت‌های اصلی سامانه توضیح داده شوند
- معماری کلی برنامه و نحوه جداسازی بخش‌های مختلف سیستم بایدشرح داده شود
- کلاس‌های اصلی پروژه و ارتباط بین آن‌ها باید معرفی شوند
- نحوه استفاده از مفاهیم شی‌گرایی مانند کپسوله‌سازی، وراثت و چندریختی باید توضیح داده شود
- ساختار پایگاه داده و جداول اصلی سیستم باید توضیح داده شوند
- نحوه ارتباط برنامه با MySQL و استفاده از کلاس‌های واسط (DAO) بایدشرح داده شود
- روند ثبت سفارش و مدیریت سفارش‌ها در سیستم باید توضیح داده شود
- نحوه مدیریت خطاها و حافظه در پروژه باید بیان شود
- چند سناریوی تست و خروجی آن‌ها باید در گزارش قرار گیرد
- نمودار اصلی پروژه که Class Diagram است باید داخل گزارش قرار داده شود
- چند تصویر از اجرای برنامه و منوهای مختلف باید در گزارش قرار گیرد
- گزارش باید تحلیلی و منظم باشد و صرفاً شامل کپی کدها نباشد