گزارش کار پروژه Online Shop

درس برنامه نویسی پیشرفته

هدیه عباسیان ابیانه

99222069

برای قسمت Shop باید یک نام ، یک آدرس وب و یک شماره پشتیبانی داشته باشیم در کلاس Online shop این سه ویژگی را لحاظ می کنیم و داده های ورودی (حساب های کاربری) ایجاد کرده که این اطلاعات را در کلاس LoginDatas ذخیره کرده ایم و در آن با استفاده از کلاس Admin لیستی ایجاد کرده و و آنرا به LoginDatas اضافه می کنیم و همچنین برای مجموع سود کسب شده ویژگی profit را ایجاد و return می کنیم

حال از کلاس Inventory که از قبل ایجاد کرده ایم استفاده کرده و آبجکت inventory را تشکیل می دهیم حال constractur از کلاس OnlineShop تعریف می کنیم که متشکل از :

String name, String website, LoginDatas loginDatas, Admin mainAdmin است .

* در کلاس LoginDatas از کلاس IAllUser برای ایجاد لیستی از تمام کاربران ایجاد می کنیم و آنرا در Users قرار می دهیم و با آبجکتی از IAllUser یعنی user تشکیل می دهیم آنرا به لیست users اضافه می کنیم

* در کلاس Inventory باید لیستی از محصولات درست کنیم برا اینکار قبلا کلاس Product را ایجاد کرده و از آن لیست درست می کنیم و اسم آنرا products می گذاریم همچنین آبجکتی از کلاس Product ایجاد کرده و نامش را product گذاشته و آنرا به لیست products اضافه می کنیم .

در قسمت User برای کاربر باید لیستی از سفارشات قرار دهیم برای این کار از کلاس Order که از قبل آنرا تشکیل داده استفاده کرده و لیست را ایجاد می کنیم.

حال از کلاس UserDashboard که آنرا توضیح خواهم داد، آبجکت dashboard را ایجاد و return می کنیم و همچنین برای تشکیل سبد خرید از کلاس ShoppingCart که انرا شرح خواهم داد استفاده کرده و از آن آبجکتی به نام Cart تشکیل داده و return می کنیم

حال برای لیست محصولات خریداری شده لیستی از کلاس Product به نام purchasedProducts ایجاد می کنیم و همچنین متدی برای ایجاد صفحه نمایی ایجاد کرده به نام showProfile که در آن username و مقدار پول در کیف پول نمایش داده می شود.

* در کلاس UserDashboard به کاربر و فروشگاه نیازمندیم به همین دلیل پس دو ویژگی User user و OnlineShop shop را ایجاد و در این کلاس قرار داده

از متد showDashboard برای نشان دادن این داشبورد استفاده می کنیم که شامل 6 آپشن است که فایل ذکر شده آنهارو پرینت گرفته و با استفاده از switch به آن دسترسی پیدا می کنیم و جزییات پیاده سازی آپشن ها در فایل نوشته شده است

* در کلاس ShoppingCart (برای ایجاد سبد خرید) لیستی از محصولات یعنی Product ایجاد کرده و نام آنرا products می گذاریم .و همچنین ویژگی ای برا جمع محصولات (double total) و تعداد آن ها (int itemCount) تعریف می کنیم و همچنین ویژگی OnlineShop shop .

از متد emptyCart برای خالی بودن سبد استفاده می کنیم در این صورت total و itemCount را برابر صفر قرار می دهیم

از متد removeProduct برای حذف محصولات از سبد استفاده می کنیم که در آن total و itemCount را یکی یکی کم می کند.

از متد addProduct برای اضافه کردن محصولات به سبد استفاده می کنیم که در آن total و itemCount را یکی یکی اضافه می کند.

اینترفیس IAllUser برای دسترسی به تمام کاربران ایجاد کرده که شامل ویژگی Username, password, dashboard است و متد showProfile برای نمایش صفحه نمایی.

برای قسمت Registration از کلاس Authentication برای احراز هویت استفاده می کنیم از متد Boolean isRegistered با پارامتر username استفاده کرده که اگر true باشد ثبت آن چک شود

آبجکت user را از کلاس IAllUser تعریف کرده که اگر null user نباشد،true برگرداند در غیر اینصورت false.

همچنین این کار را برای متدی Boolean Login انجام می دهیم .

برای قسمت Admin کلاس آن را تعریف کرده که از اینترفیس IAllUser ارث بری کرده و دو ویژگی username و password و dashboard از کلاس AdminDashboard را به آن اضافه می کنیم برای اینکه مدیران جدید را بتواند اضافه کند.

حال costractur آن را هم تعریف می کنیم.

قسمت معامله دراین بخش را در قسمت امتیازی توضیح خواهم داد.

درقسمت Seller این کلاس از کلاس WalletHolder که خود از اینترفیس IAllUser ارث بری کرده، ارث بری می کند.

متد Boolean getCertified را تعریف کرده چون فروشندگان باید ایتدا از یک مدیر مجوز دریافت کنند تا اجازه فروش داشته باشند که در صورت درست بودن متد آن را تایید کند، حال constractur این کلاس را تعریف کرده که شامل OnlineShop String username, String password,Wallet wallet , Boolean isCertified shop, است.

همچنین متدی برا نمایش پروفایل آنها و داشبورد تعریف شده است.

در قسمت Wallet ، یک ویژگی مقدار پول نهایی در کیف پول به نام balance تعریف می کنیم.

متد addMoney را با پارامتر amount تعریف کرده که در این متد مقدار balance به amount اضافه می شود و در متد deductMoney عکس ان را انجام می دهیم برای کاهش مقدار پول در کیف پول.

متدی دیگری هم تعریف کرده که اگر مقدار balance کمتر از amount کمتر باشد Insufficient balance را پرینت کند.

در قسمت Product ویژگی های name , price و Quantity را به آن اضافه کرده که constractur آن علاوه بر این ویژگی ها Seller seller هم دارد .

آرایه ای برای ایجاد نظرات ایجاد کرده و در نهایت تمام این ویژگی هارا در متد return toString می کنیم .

5 دسته بندی برای محصولات ایجاد کردم که عبارت است از :

BooksProduct

FashionProduct

PersonalCareProduct

PharmecyProduct

ElectronicsProduct

چالش‌های پیاده‌سازی:

برای پیاده‌سازی پروژه مشکلات زیادی وجود داشتند که در ادامه به برخی از آن‌ها و نحوه مواجه شدن با آن‌ها می‌پردازیم.

۱.ارتباط بین کلاس‌ها

با توجه به ماهیت پروژه و لزوم پیاده‌سازی یک فروشگاه آنلاین, لازم بود که کلاس‌ها با هم ارتباط زیادی داشته باشند, برای مثال کاربرها باید بتوانند در لیست محصولات جست‌وجو کنند و همچنین فروشندگان نیز باید بتوانند کالاهای جدید اضافه کنند. برای اینکه این ارتباط به وجود بیاید, به طور کلی دو راه وجود داشت؛ ترکیب همه این کلاس‌ها در یک کلاس که باعث پیچیدگی بیش از حد پیاده‌سازی می‌شد و یا طراحی یک ارتباط بین کلاس‌ها با استفاده از UML, به همین دلیل ابتدا زمان لازم برای طراحی کلاس‌های موجود اختصاص داده شد و سعی شد یک نقشه کلی از روابط بین کلاس‌ها ایجاد شود و سپس با پیاده‌سازی این نقشه و تصمیم‌گیری در مورد کلاس‌ها و اینترفیس‌ها, این چالش برطرف شد.

۲.ورود کاربران

با توجه به چالش قبل, به طور کلی ۳ نوع کاربر می‌توانستند وارد سیستم فروشگاه شوند: ۱.ادمین‌ها ۲.فروشنده‌ها ۳.خریدارها برای اینکه سیستم بتواند تشخیص بدهد کسی که در تلاش است وارد شود چه ماهیتی دارد دو راه‌حل موجود بود؛ چک کردن نوع کلاس کاربر در کلاس IntroPage در همان زمان ورود و یا استفاده از Interface منظور رعایت Seperation of concerns, به زبان ساده, در پیاده‌سازی انتخاب شده, لازم نیست کلاس IntroPage بررسی کند شخصی که در حال ورود است از نوع ادمین است یا از نوع کاربر عادی, بلکه با استفاده از اینترفیس IAllUser, به همه کاربران متدی اختصاص داده شده که در صورت ورود موفق صدا زده می‌شود و به توجه به نوع کاربر, داشبورد مربوطه نمایش داده می‌شود.

۳.درخواست‌های دو طرفه به ادمین

در پروژه, برای بخش‌های مختلفی از جمله بخش امتیازی, لازم بود کاربران درخواستی به ادمین ارسال کنند و منتظر پاسخ آن بمانند. در ادامه به ۳ مورد از این موارد اشاره می‌کنیم:

درخواست تایید خرید کاربر: کاربر خریدار پس از تایید سفارش خود باید منتظر بماند تا ادمین نیز سفارش او را تایید کند و در صورت تایید شدن بررسی می‌شود که کاربر موجودی لازم را دارد یا خیر و همچنین مبلغی از خرید به حساب فروشگاه واریز می‌شود.

۴.درخواست احراز هویت فروشنده

فروشندگانی که به تازگی به سیستم اضافه شدند, باید قبل از شروع فروش درخواستی را برای ادمین ارسال کنند که شامل نام و آدرس آن‌هاست و در صورت تایید, می‌توانند فعالیت خود را شروع کنند.

درخواست افزایش موجودی کاربر: کاربران می‌توانند در پنل خود درخواستی را ثبت و ارسال نمایند که در صورت تایید, موجودی آن‌ها را به مقدار مشخصی افزایش خواهد داد.

برای پیاده‌سازی این موارد, از کلاس‌های کمکی استفاده شد. برای مثال, هرگاه کاربر درخواست پرداخت سبد خرید خود را ثبت کند این تراکنش فقط `پردازش` می‌شود , به این معنی که به تراکنش‌های در لیست انتظار ادمین اضافه می‌شود؛ سپس هر موقع که ادمین تصمیم بگیرد تراکنش‌های پردازش شده را `اجرا` کند, پول از حساب کاربر کسر خواهد شد. برای این پیاده‌سازی از دو کلاس Transaction و TransactionHandler استفاده شده است.

۵.ثبت نظر برای خریدها

در ابتدا, هنگامی که کاربران خرید خود را برای پردازش به ادمین ارسال می‌کردند کالا به لیست کالاهای خریداری شده اضافه می‌شد و کاربر می‌توانست نظر خود را ثبت نماید اما این موضوع ایده‌آل نبود برای این‌که امکان رد درخواست توسط ادمین وجود داشت, برای حل این مشکل در کلاس Transaction و در تابع مربوط به انجام‌نهایی تراکنش, با دسترسی به لیست خرید کاربر, تنها پس از انجام نهایی تراکنش محصول به لیست کاربر اضافه می‌شد تا این مشکل برطرف شود.

۶.فیلدهای متفاوت محصولات

محصول‌های متفاوت در فروشگاه مشخصات متفاوتی داشتند برای مثال محصول کتاب دارای فیلد نویسنده بود و محصول لباس این فیلد را نداشت و بجای آن دارای فیلد سایز بود که برای حل این مشکل کلاسی به نام Product با تمام فیلدهای مشترک ایجاد شد و سپس کلاس‌های فرزند با ارث‌بری از آن فیلدهای دیگری را نیز اضافه کردند.