

گزارش کار پروژه ی فایلال آنلاین شاپ

یاسمین میرعبداله (99222104)

اول از همه در صفحه ی یا main منو را نوشتم که شامل لاگین ، رجیستر، پشتیبانی و خارج شدن از سایت بود.

برای حساب های کاربری کلاس های ادمین ، خریدار و فروشنده یک کلاس پدر به نام users تعریف کردم برای این که این سه کلاس ویژگی های مشابهی با هم داشتند. همچنین برای مشخص کردن نقش آن ها چون ثابت بود از enum استفاده کردم.

برای کلاس shop ویژگی های فایلالی را تعریف کردم و تمام متد هایی که در این کلاس تعریف شده static هستند برای این که کلاس shop همان آنلاین شاپ ما هستند و از آن نمونه ای ساخته نمیشود.

برای حساب های کاربری لیست محصولات و لیست تمام سفارشات از arrayList استفاده کردم چون مثلاً برای حساب های کاربری میخوامم موجوداتی با "کلاس" user در آن ها ذخیره کنم و آرایه ها خیلی به در نمیخورد.

در حین ثبت حساب کاربری یک نمونه ی جدید از کلاس خریدار یا فروشنده ساخته میشد و در لیست حساب کاربری shop ثبت میشد.

برای حساب کاربری ادمین یک نمونه از ادمین ساخته میشد ولی وارد لیست کاربرهای shop نمیشد بلکه وارد لیست تعریف شده در کلاس ادمین به نام Request for Admin Account میشد که با توجه به متد هایی که در کلاس ادمین تعریف شده ادمین دسترسی به این درخواست ها داشته و میتواند این درخواست ها را رد یا قبول بکند . در صورت قبولی ادمین های قدیمی این نمونه از مدیر تازه ایجاد شده وارد حساب های کاربری شاپ میشود و به قابلیت های مدیران دست پیدا میکنند.

مشکل : بعد از قبول درخواست وقتی لاگین میکنم هنوز اسم مدیر جدید رو نمشناسه و راحلی پیدا نکردم .

کاربران میتوانند اطلاعات شخصی خود را ویرایش کنند برای این قابلیت از کپسوله سازی و متد set... استفاده کردم.

برای محصولات من از کلاس پدر product استفاده کردم که نام ، قیمت ، مقدار موجودی ، نام فروشنده و لیست (arrayList) از نظرات را دارد از arraylist استفاده کردم چون مشتریان در حین خرید میتوانند نظر خود را به این لیست نظرات اضافه کنند. همچنین 5 دسته بندی فرزند که علاوه بر اطلاعات و متد ها product اطلاعات اضافه تری هم دارند اضافه کردم.

همچنین برای مشخص کردن دسته بندی آن ها چون ثابت بود از enum استفاده کردم.

فروشنندگان با اجازه ی مدیر میتوانند محصولات خود را بفروشند برای این کار یک arraylist در کلاس ادمین ایجاد کردم که اطلاعات فروشنندگان بعد از درخواست مجوز در آنجا ذخیره میشد و مدیران با متد ها و قابلیت های تعریف شده میتوانند درخواست ها را ببینند و قبول و رد کنند . بعد از قبول شدن اطلاعات فروشنندگان به arraylist دیگری به نام sellerscancel در شاپ منتقل میشود . اگر فروشنندگان بخواهند محصولی را به لیست محصولات شاپ اضافه کنند اول چک میشود با contain(object) که آیا اطلاعات فروشنده در آن لیست نبود فروشنندگان اجازه فروش و اضافه کردن محصول را ندارند .

برای مشاهده ی محصولات از <> Iterator استفاده کردم . در حین حلقه خریدار میتواند به سبد خرید خود محصولاتی را اضافه کنند .

برای سفارشات و محصولات خریداری شده و سبد خرید من از کلاسی به نام cart استفاده کردم که شامل ویژگی هایی مانند خریدار ، فروشنده ، تعداد خریداری شده و.... است.

خریداران که محصولات را به سبد خرید اضافه میکنند میتوانند آن را ببینند و اگر خواستند محصولاتی را به آن اضافه کنند . اگر میخواستند خرید خود را نهایی کنند هر کدام از این محصولات موجود در سبد خرید را به لیست سفارشات اضافه میکنند(اگر پول کافی در کیف پول خود نداشته باشد این سفارش ثبت نمیشود) اگر پول کافی بود با این کار این موجود از کلاس Cart در لیستی در کلاس ادمین به نام request for buying product ذخیره میشود و ادمین ها میتوانند این معامله را تایید یا رد کنند با این کار 90 درصد هزینه ی معامله از کیف پول خریدار کم و به کیف پول فروشنده

اضافه میشود. و ده درصد هم سود فروشگاه از معامله است (در ویژگی سود در کلاس فروشگاه ذخیره میشود)

فقط مدیران میتوانند به کیف پول کاربران وجه اضافه کنند برای این کار فروشندگان و خریداران ویژگی به نام کیف پول دارند ولی برای اضافه کردن پول باید به مدیر درخواست دهند برای همین در منوی اصلی میتوانند درخواست وجه را به مدیر دهند با این کار در مشخصات آن ها در `arraylist` ای به نام `request for money` ذخیره میشود و مدیران میتوانند این درخواست را رد یا قبول کنند. با قبول کردن درخواست با توجه به متد های نوشته شده پول به کیف پول شخص درخواست دهنده ارسال میشود.

مشکل : گاهی وقتا وقتی برنامه ران میشه از برنامه میاد بیرون و هنوز راه حلی پیدا نکردم.