

Advanced Programming course

Dr. Shirin Baghoolizadeh

Store

Such as digikala or any other online stores

Telegram:

@AmirArsalanYavari

Email:

amirarsalan.yavari@ec.iut.ac.ir

مقدمه

در این پروژه شما ملزم به استفاده از زبان **C++** و تفکر شی‌گرائی + **QT framework** هستید؛ همچنین بایستی موارد ذکر شده در پروژه را رعایت کرده و یک فروشگاه خرید و فروش را طراحی نمایید. قرار است سیستم را بگونه‌ای طراحی کنید که برای هر سه نوع کاربر **client**، **customer**، **admin** به طور متفاوتی کار کند؛ همچنین درگاهی پرداختی باید طراحی شود تا به کمک آن بتوان عملیات انتقال وجه را انجام داد.

شرح پروژه:

1. ورود یا ثبت‌نام

در این قسمت شما باید با کمک **QT framework** یک صفحه طراحی نمایید و همچنین منطق تابع آن را به کمک زبان برنامه‌نویسی **C++** پیاده‌سازی نمایید که توسط آن بتوان به اکانت‌هایی که وجود دارند لاگین کنید یا بتوانید یک حساب کاربری جدید ایجاد نمایید. برای قسمت ورود (**login**)، شما حداقل به دو فیلد نام کاربری و پسورد نیاز خواهید داشت ((همچنین دقت نمایید که باید قسمت فراموشی رمز عبور را داشته باشید که فرد از طریق وارد کردن شماره کابری خود بتواند رمز عبور خود را تغییر دهد؛ نحوه‌ی پیاده‌سازی آن بر عهده‌ی خودتان می‌باشد))؛ برای ایجاد حساب (**register account**) نیز حداقل نیازمند فیلدهای نام، نام کاربری، شماره‌ی تلفن، آدرس، و پسورد نیاز خواهید بود.

2. انواع کاربر و توانمندی‌های آنها

مورد اول کاربر با سطح دسترسی **client** است. این کاربر قرار است هنگامی که با اکانت خود لاگین می‌کند بتواند تمام محصولات فروشگاه را ملاحظه و سپس اقدام به خرید آنها کند. (راحت بگیریم، مثل خودمون وقتی مثلاً میریم تو سایت دیجی‌کالا).
مورد دوم کاربر با سطح دسترسی **customer** است. این کاربر قرار است بتواند فروشنده‌ی یک محصول باشد، در نتیجه باید توان افزودن کالا را داشته باشد؛ بدین منظور باید صفحه‌ای جداگانه طراحی شده باشد. هر فروشنده باید در صفحه‌ای جداگانه لیست کالاها و همچنین لیست خریداران و فاکتور خرید هرکدام را بتواند مشاهده کند.

مورد سوم کاربر با سطح دسترسی **admin** است. این کاربر قرار است به تمام سیستم دسترسی داشته باشد، این اطلاعات شامل تمامی تراکنش‌ها، اطلاعات کاربران سیستم (بجز

پسوردشون) و ... را بتواند مشاهده و در صورت نیاز تغییر دهد. همچنین این کاربر باید بتواند سطح دسترسی افراد را در سیستم تغییر دهد و حتی یک کاربر را از سیستم حذف کند.

((دقت کنید در صورت حذف یک کاربر از سیستم اطلاعات آن تا مدت معینی باید در سیستم در دسترس باشد فقط وضعیت کاربر تغییر کرده و دیگر توان ورود به سیستم را نخواهد داشت))

3. انواع تراکنش

تراکنش‌ها شامل دو مدل پرداخت و دریافت هستند، برای هر یک، یک پنل جدا طراحی کنید (دقت شود که در اینجا abstraction را لحاظ کنید). در صورت نیاز به هر کدام، کاربر به صفحه‌ی مربوطه منتقل گردد.

4. انواع کالا

هر کالا میتواند یک سری فیلدها را نداشته باشد و یک سری فیلدها را نیز داشته باشد؛ اما میدانیم که هر کالایی یکسری ویژگی مشترک نظیر تعداد موجود در انبار، قیمت و ... دارند. در قسمت افزودن کالا شما بایستی به گونه‌ای عمل کنید که فروشنده هر کدام از ویژگی‌هایی که مدنظرش بود را استفاده کند و مابقی نیاز نیست (فیلدها باید تا حد ممکن کامل باشند برای این منظور می‌توانید از صفحه‌ی فیلتر یکی از فروشگاه‌های آنلاین الگو برداری کنید)

5. فیلتر سرچ کردن

دقت فرمایید در پنل خرید باید طبق هر کدام از موجودیت‌های قابل سرچ بتوان جستجو کرد (برای مثال نوع کالا، قیمت، وزن، رنگ، برند و ...)

6. موارد دارای نمره اضافه

استفاده از الگوریتم‌های رمزگذاری RSA و AES256 در ذخیره و ارسال داده‌ها (50 نمره)

در صورتی که بتوانید با API ایمیل کار کنید، 50 نمره‌ی امتیازی خواهید گرفت. (برای مثال در هنگام تراکنش‌ها اطلاع رسانی صورت گیرد یا در هنگام ورود به سیستم هر کاربر با یک کلید رندوم authenticate شود)

اگر از Socket استفاده کنید و برنامه‌ی شما بتواند تحت وب هم کار کند به میزان 100 نمره‌ی اضافی، نمره‌ی بیشتری دریافت خواهید کرد.

اگر پروژه‌ی خود را داکرایز کردید که بتواند روی هر سیستمی که داکر دارد اجرا شود 30 نمره‌ی اضافی خواهید گرفت.

موارد فوق در صورتی شامل نمره خواهند بود که پروژه‌ی شما تقریباً **90 درصد** کامل باشد. (سعی نکنین بخش‌هایی از پروژه رو نزنین که با این قسمت‌های امتیازی نمرش رو جبران کنید!)

7. نکات پایانی

در طراحی کلاس‌ها به **encapsulation** و **data hiding** بسیار توجه کنید و حتماً حتماً آنها را رعایت کنید، همچنین در مواردی که ارث‌بری انجام می‌دهید به مواردی ظیر **abstraction** نیز دقت و در صورت نیاز انجام دهید.

تمیزی و **reusable** بودن کد شما بسیار مهم است و از نمره‌ی اضافی ویژه‌ای برخوردار است؛ بدین منظور می‌توانید به این [لینک](#) مراجعه نمایید.

برنامه‌ی شما تحت هیچ‌شرایطی نباید مموری‌لیک داشته باشد. پس مدیریت منابع را واجب‌تر از نون شب بدونید :)

برنامه‌ی شما باید مولتی‌ترد (**multy thread**) باشد و درست کار کند. بدین صورت که قبل از نمایش صفحه‌ی ورود یک **server app** دارید که به ازای هر اینستنس جدید باید یک اینستنس جدید ایجاد شود. اینترفیس لازم برای کار کردن با **server app** برعهده‌ی خود شماست.

در انجام این پروژه ملزم هستید از **git** استفاده کنید و موارد زیر را حتماً رعایت کرده باشید چرا که در غیر اینصورت میزانی از نمره‌ی پروژه‌ی شما کسر خواهد شد.

1. کامیت‌ها باید روند اجرا و تکمیل پیوسته‌ی پروژه را نشان دهد.
2. با **github** باید کار کنید.
3. بعد از رفع دیپندنسی‌های اولیه اعضای هر گروه ملزم به ایجاد برنچ و درنهایت مرج کردن آن در زمان مناسب با برنچ اصلی می‌باشند.

4. کامیت‌ها باید بامعنی باشند؛ بدین منظور میتوانید به این [لینک](#) مراجعه فرمائید.
5. برای پروژه‌ی خود در github باید یک readmi مناسب (کوتاه و مختصر و در عین حال کامل) بنویسید.
6. افراد گروه پس از اطلاع رسانی گروهشان به یک ریپازیتوری اضافه می‌شوند و بایستی تمام پروژه را در آن repo ادامه دهند. (مشارکت همه‌ی اعضای گروه در پل پروژه‌ی شما الزامی است)

• تحویل این پروژه در سه فاز انجام می‌شود:

o فاز اول

- طراحی کلاس‌ها و توابع مورد نیاز با توجه به اصول شی گرای و الگوهای طراحی.
- راه‌اندازی گیت.

o فاز دوم

- پیاده‌سازی توابع و کلاس‌های طراحی شده.

o فاز سوم

- پیاده‌سازی رابط گرافیکی و بخش‌های مربوط به شبکه.

زمان تحویل پروژه ۵ام تیرماه خواهد بود؛ زمان ارائه‌ی پروژه توسط تمام اعضای گروه متعاقباً اعلام می‌گردد.