

Advanced Programming

Dr. Shirin Baghoolizadeh

Project management system

Such as slack or any other program

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 برنامه‌ای برای مدیریت پروژه طراحی نمایید (میتوانید از slack یا trello ایده‌برداری کنید).

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 ایمیل، و پسورد نیاز خواهید بود.

2. تنظیمات کاربری

در این قسمت منوی پروفایل هرکاربر به او نشان داده می‌شود و کاربر باید بتواند داده‌های مرتبط با خود را تغییر دهد؛ اطلاعاتی نظیر آدرس ایمیل، شماره‌ی تلفن، نام کاربری، عکس پروفایل و درکل هر پارامتر دیگری که از کاربر نگهداری می‌شود ...

3. ایجاد یک پروژه

هرکاربر باید بتواند در نقش یک مدیر پروژه، در برنامه پروژه‌ای را بسازد و از طریق (مثلا لینک افزودن به پروژه) دیگران را به پروژه اضافه نماید. لازم به ذکر است که سطح دسترسی کاربران در این پروژه را مدیر پروژه تعیین می‌کند؛ برای مثال یک سری از کاربران باید فقط بتوانند وضعیت را مشاهده کنند، یک سری دیگر باید بتوانند تسک‌های به پروژه اضافه یا کم بکنند، یک سری حتی باید بتوانند تسک‌ها را به کسانی محول کنند و سری آخر از کاربران همچون مدیر پروژه باید توانایی هرگونه تغییری را در پروژه داشته باشند.

4. تعریف وظایف

هر کاربر در هر پروژه‌ای که سطح دسترسی آن را داشته‌باشد، باید بتواند تسکی را اضافه کند. تسک‌ها سه حالت واگذار نشده (assign نشده)، واگذار شده و به پایان رسیده دارند که همین وضعیت را نیز کاربران دارای دسترسی مناسب مشخص می‌کنند.

5. تغییر وضعیت وظایف

این بخش در قسمت قبل تاحدی توضیح داده شده است؛ به عنوان مثال وقتی تسکی در یک پروژه ایجاد می‌شود باید وضعیت تسک واگذار نشده باشد؛ در ادامه کارمند A قرار است آن وظیفه را انجام دهد پس وضعیت تسک به حالت واگذار شده به کارمند A تغییر می‌کند؛ در نهایت وقتی کارمند A وظیفه‌اش را انجام داد وضعیت تسک به حالت انجام‌شده تغییر می‌کند.

6. پیام‌رسانی

کاربران یک پروژه باید بتوانند از طریق پنبلی با تمام یا افراد خاصی از پروژه از طریق پیام در ارتباط باشند. توجه فرمایید که هر کاربر می‌تواند چندین صفحه چت متفاوت داشته باشد و این اطلاعات باید نگهداری شوند و در دفعات بعد کاربر خودش انتخاب کند به کدام چت ادامه دهد. (طراحی هرچه بهتر این صفحه نمره‌ی اضافی خواهد داشت)

7. 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 اطلاع‌رسانی 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!)

8. نکات پایانی

در طراحی کلاس‌ها به **encapsulation** و **data hiding** بسیار توجه کنید و حتما حتما 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 حتما 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 در نهایت 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 پزل پروژه‌ی شما الزامی است)

• تحویل این پروژه در سه فاز انجام می شود:

o فاز اول

- طراحی کلاس ها و توابع مورد نیاز باتوجه به اصول شی گرای و الگو های طراحی.
- راه اندازی گیت.

o فاز دوم

- پیاده سازی توابع و کلاس های طراحی شده.

o فاز سوم

- پیاده سازی رابط گرافیکی و بخش های مربوط به شبکه.

زمان تحویل پروژه ۵ام تیرماه خواهد بود؛ زمان ارائه ی پروژه توسط تمام اعضای گروه متعاقبا اعلام می گردد.