



پروژه درس برنامه نویسی پیشرفته



بخش اول

زمستان 1399 - دانشکده علوم ریاضی

دانشگاه صنعتی شریف

تیم طراحی:

مجتبی استواری

سروش تابش

سید محمد ترابی

نیما خداویسی

حسین رحمانی

محمد مهدی زارع

سید امیر محمد سادات شکوهی

محمدعلی علما

نیما علیزاده

غزل فراهانی

سید عرفان موسویان

هلیا یزدان یار

معرفی پروژه

در این پروژه شما باید یک شبکه اجتماعی طراحی کنید. پیشنهاد می‌شود قبل از شروع به کار فایل توضیحات کلی پروژه را که قبلاً در اختیار شما قرار گرفته است را مطالعه کرده باشید. همچنین خوب است با برخی از شبکه‌های اجتماعی نظیر توییتر آشنا باشید.

اهداف بخش اول پروژه

- شناخت پیدا کردن نسبت به توسعه نرم افزار (Software Development)
- آشنایی و طراحی رابط‌های کاربری نوشتاری
- آشنایی با معماری‌های طراحی نرم افزارهای کاربردی، برنامه نویسی شی گرا و تمیز کد زدن
- شناخت ویژگی‌های اختصاصی زبان جاوا
- کار با فایل‌ها
- ذخیره‌سازی درست اطلاعات

رفع اشکال

برای این بخش از پروژه یک جلسه توجیهی برگزار می‌شود که زمان آن متعاقباً اعلام خواهد شد. قبل از جلسه توجیهی، حتماً داک را مطالعه کنید. همچنین شما می‌توانید برای رفع پرسش‌ها، اشکال‌ها و ابهام‌های خود از طریق سایت کوئرا با تیم طراحان پروژه در ارتباط باشید.

ارزیابی و روش تحویل

- این بخش جداگانه از بخش های دیگر ارزیابی و به صورت آنلاین تحویل گرفته می شود.
- هر بخش پروژه باید حداکثر تا تاریخی که به شما اعلام شده است در کوئرا آپلود شود. کدی که در زمان تحویل مورد ارزیابی قرار می گیرد، کد آپلود شده در کوئرا می باشد. دقت کنید که ددلاین های اعلام شده قابل تغییر نمی باشند پس برنامه ریزی لازم را برای رساندن پروژه خود به ددلاین ها داشته باشید.
- دقت کنید که فقط کارکردن کد مدنظر نیست و از شما انتظار می رود که به صورت اصولی، تمیز و پیشرفته کد بننید!
- در هر بخش پروژه شما ملزم به نوشتن یک فایل توضیحی کامل در مورد کد خود هستید. روش نوشتن این فایل توضیحی در صفحه بعدی نوشته شده است.

نوشتن بخش توضیحات

هر بخش (فاز) پروژه شما نیاز به یک فایل توضیحی است که موارد زیر را شامل می شود:

1. منابع استفاده شده برای پیاده سازی کد شامل
 - a. منبع تصاویر و ...
 - b. منبع کدها و ...
 - c. مشورت های انجام شده
 - d. کتابخانه های استفاده شده
 2. روش کارکرد کد شما به همراه نقاط قوت و ضعف آن
 3. ارائه دلیل برای انتخاب هایی که انجام داده اید (مثلاً چرا از یک کتابخانه خاص یا طراحی خاص استفاده کرده اید)
- این نوشته به ارزیابی سریع تر و راحت تر پروژه شما کمک فراوانی می کند. ترجیحاً توضیحات پروژه کوتاه، مختصر و مفید باشد.

موارد غیر مجاز

- عدم تسلط کافی بر کد پروژه
 - شباهت بیش از حد دو یا چند پروژه
 - واگذاری کامل یا بخشی از پروژه به شخصی دیگر
- رخ دادن این اتفاق ها برای هیچ فردی قابل پذیرش نیست و در صورت بروز هر کدام از این اتفاق ها ممکن است هر تصمیمی در رابطه با ارزیابی فرد گرفته شود.
- در صورتی که یکی از این اتفاق ها رخ داده باشد لازم است که افراد حتماً دلیل این مساله را پیش از تشخیص توسط تیم درس اعلام کنند و دلیل این اتفاق را توضیح دهند. در این صورت فقط ارزیابی مربوط به بخش اعلام شده تحت تاثیر قرار خواهد گرفت.
- در صورتی مشاهده یکی از این اتفاق ها توسط تیم درس و پیش از اعلام فرد رخ دهد این تیم از فرد درخواست خواهد کرد که در این رابطه توضیح دهند و در صورت قابل قبول نبودن توضیح، فرد مورد نظر موفق به گذراندن درس نخواهد شد.

نکات مهم :

- ممکن است در آینده اصلاحاتی در این داک صورت بگیرد یا توضیحات بیشتری برای بعضی قسمت‌ها اضافه شود. سعی می‌شود تغییرات ایجاد شده، با متن‌های گذشته تفاوت بصری داشته باشد تا راحت‌تر قابل تشخیص باشند. این اصلاحات، از طریق راه‌های معمول (مثل ایمیل) اطلاع رسانی **نخواهد شد**. بنابراین سعی کنید از داک به صورت آنلاین استفاده کنید یا در صورت استفاده از قالب pdf ، **به طور مرتب** نسبت به دریافت نسخه تازه اقدام کنید.
- در ادامه، توضیحات کلی مواردی که لازم است در برنامه شما وجود داشته باشد داده شده است. در صورتی که درباره جزئیات چیزی توضیح داده نشده، از **خلاقیت** خود استفاده کنید و به هر صورتی که دوست دارید آن قسمت را پیاده سازی کنید.

رابط کاربری نوشتاری CLI

رابط کاربری نوشتاری (Command line interface) یعنی یک محیط درون ترمینال یا cmd است که به وسیله‌ی دستوراتی از پیش تعیین شده به ما اجازه تعامل با یک برنامه را می‌دهد. بسیاری از پروژه‌های برنامه نویسی از چنین سیستمی به جای یک رابط گرافیکی استفاده می‌کنند. در این بخش شما باید چنین محیطی را برای تعامل با کاربر پیاده سازی کنید. دستوراتی که باید پیاده سازی کنید و جزئیات آن در ادامه به شما داده شده است.

توجه داشته باشید که CLI فاز اول پروژه شما، در فاز بعد تبدیل به یک محیط گرافیکی خواهد شد. بنابراین سعی کنید طراحی CLI شما مستقل از سایر منطق برنامه باشد تا در فاز بعد به راحتی بتوانید CLI را با رابط گرافیکی جایگزین کنید.

ذخیره سازی اطلاعات

بدیهی است که یکی از مهم ترین قابلیت های یک شبکه اجتماعی (و کلا هر برنامه دیگری) ذخیره سازی اطلاعات است. در شبکه اجتماعی شما نیز، باید اطلاعات کاربران به نحو صحیح ذخیره شود و در مواقع لزوم از آنها استفاده شود.

معمولا در پیاده سازی دیتا مدل های یک پروژه مشکلات زیر رخ می دهد:

- چگونه باید مدل های ساخته شده با یک زبان خاص را طوری ذخیره کرد که یک برنامه دیگر بتواند از آن استفاده کند؟
- چگونه باید این مدل ها را به صورتی ذخیره کرد که هم برای ماشین خوانایی داشته باشد و هم برای انسان عادی؟

این مشکلات وقتی نمود پیدا می کنند که برنامه نیاز دارد داده های خود را در فایل، دیتابیس و... ذخیره کند که به زبان برنامه نویسی وابسته نیستند.

توجه کنید که در بخش های آینده پروژه، دیتا مدل های جدید (چه قبل از کامپایل و چه در حال اجرا) به برنامه شما اضافه می شود. پس طراحی شما باید Abstraction کافی را داشته باشد تا این موارد را مدیریت کند و در هنگام اجرا به مشکل برخورد.

در کنار این ها، شاید لازم شود که مقادیر پیش فرض دیتا مدل ها را بدون اجرای دوباره ی سورس کد تغییر دهید. بنابراین، از شما می خواهیم که تمامی دیتا مدل ها را به صورت جداگانه در فایل هایی ذخیره کنید.

بعضی از مقادیر باید به صورت پیش فرض وجود داشته باشند و قابل تغییر نیستند اما بعضی دیگر توسط کاربر در آینده ممکن است تغییر کنند.

این فایل ها در هنگام اجرا بارگذاری می شوند و اطلاعات مورد نیاز را به برنامه می دهند.

در نهایت انتخاب روش ذخیره سازی داده ها، به عهده خودتان است.

به عنوان نمونه، یک شی را میتوان به این صورت ذخیره کرد :

```
{
  "username" : "AP",
  "password" : "spring",
  "id" : "1297150266630",
  "emailAddress" : "ap2021spring@gmail.com"
}
```

استفاده از آبجکت استریم ها و آبجکت سریالایزهای جاوا، در کل پروژه و تمرین ها ممنوع است.

لاگ کردن (Logging):

یکی از اصول مهمی که در تولید هر برنامه‌ی با کیفیتی رعایت می‌شود، لاگ کردن رخدادهای برنامه است به طوری که بتوان با خواندن لاگ، به روندهای اجرا شده و تغییرات انجام شده پی‌برد. این قابلیت باعث می‌شود نگهداری و توسعه کد با سرعت بسیار بیشتری انجام شود و همچنین اشخاص دیگری که کد شما را می‌خوانند، زمان کمتری را صرف درک کد کنند که دو نکته کلیدی در تولید نرم‌افزار است.

در این پروژه از شما انتظار می‌رود این اصل را در تمامی فازها رعایت کنید و لاگ‌های ساختارمندی ذخیره کنید. به این منظور پیشنهاد می‌کنیم برای تولید لاگ، از کتابخانه‌های موجود استفاده کنید. این کتابخانه‌ها، مسئولیت ذخیره لاگ، حذف لاگ‌های قدیمی، تولید لاگ به فرمت مشخص (سطح اهمیت لاگ، زمان، کلاس تولید کننده لاگ و...)، ارتباط با دیتابیس و... را بر عهده دارند.

برای مثال می‌توانید از کتابخانه [log4j](#) استفاده کنید. در [این صفحه](#) می‌توانید آموزش مربوط به آن را نیز ببینید.

چگونه لاگ کنیم؟

دقت کنید که هدف ما از لاگ کردن، متوجه شدن روند کلی برنامه و اتفاقات رخ داده است و همچنین نباید لاگ کردن، سربار عملیاتی زیادی داشته باشد. برای مثال به هنگام برخورد کردن به خطای زمان اجرا، ذخیره کردن stack trace کار خوبی نیست. گرچه اطلاعات زیادی در اختیار ما قرار می‌دهد ولی بسیار حجیم است و برای درک آن، نیازمند درک کد هستیم. به علاوه، در صورتی که از اتفاقات پیش از رخداد خطای خبر باشیم، ممکن است درک خطا را دشوار کند. بنابراین به دنبال داشتن یک خلاصه از اجرای روند برنامه هستیم.

اصول خوبی از لاگ کردن را می‌توانید در [این صفحه](#) ببینید. **نمره دهی به طور کلی بر اساس موارد داخل این صفحه خواهد بود. موارد ۱، ۲، ۴، ۶ و ۷ مهم‌ترین مواردی هستند که باید رعایت کنید.**

تعدادی از مواردی که می‌توانید در لاگ رعایت کنید:

- چه کلاسی
- چه تابعی
- چه زمانی
- سطح اهمیت
- روی چه داده‌هایی
- در چه شرایط محیطی
- چه نتیجه‌ای

چه چیز را لاگ کنیم؟

از شما انتظار می‌رود چیزهایی مشابه به موارد زیر را لاگ کنید:

- شروع برنامه
- باز کردن و بستن فایل
- ورود یا ثبت نام کاربر
- تغییر اطلاعات کاربری
- تغییر یا افزودن اطلاعات
- خطاهای مربوط به منطق برنامه
- اتصال و قطع شدن از دیتابیس (و خطاهای مربوطه)
- وضعیت ترنزکشن‌های دیتابیس (باز شدن، کامیت شدن، لغو شدن)
- خطاهای معمول زمان اجرا (برای مثال عدم توانایی ایجاد فایل، اشکال در ارتباط با دیتابیس، اشکال شبکه و...)

● صفحه ورود - Login

در ابتدای اجرای برنامه، ابتدا از کاربر پرسیده می شود که حساب کاربری دارد یا خیر :

- اگر شخص از قبل حساب کاربری نداشته باشد، باید برای او یک حساب کاربری ساخته شود: (موارد ستاره دار را حتما باید تکمیل کند، سایر موارد را میتواند خالی بگذارد)

اطلاعات کاربر دارای سه سطح است :

- i. کاملا خصوصی (فقط سرور برنامه به آن دسترسی دارد). مشخصاتی که در ادامه با رنگ قرمز مشخص شده اند باید کاملا خصوصی باشند.
- ii. نیمه خصوصی (کاربر سطح دسترسی را مشخص کند). مشخصاتی که در ادامه با رنگ آبی مشخص شده است باید نیمه خصوصی باشند.
- iii. عمومی (همه افراد دسترسی دارند). مشخصاتی که در ادامه با رنگ سبز مشخص شده است باید عمومی باشند.

● نام و نام خانوادگی *

● نام کاربری *

● رمز عبور *

● تاریخ تولد

● آدرس ایمیل *

● شماره تلفن

● بیوگرافی

می توانید به سلیقه خودتان به موارد بالا اضافه کنید. اما طراحی موارد بالا الزامی است.

برای هر حساب کاربری باید موارد زیر نیز در نظر گرفته شود:

- I. هر کاربر، یک ID دارد. این ID باید برای هر حساب کاربری، منحصر بفرد باشد.
- II. وضعیت فعال بودن یا نبودن حساب کاربری (توضیحات در ادامه)
- III. آخرین زمانی که کاربر در برنامه بوده است. (آخرین زمانی که آنلاین بوده است.)

- برای ورود به برنامه اگر از قبل حساب کاربری داشته باشد، کافی است نام کاربری و رمز عبور خود را وارد کند.

- توجه داشته باشید که خطاهای مرتبط با ورود و ثبت نام باید مدیریت شوند. برای مثال:

- هنگام ثبت نام، نام کاربری از قبل موجود باشد.
- هنگام ثبت نام، برای این آدرس ایمیل (یا شماره تلفن) قبلا حساب کاربری ساخته شده است.
- هنگام ورود، حساسی با آن نام کاربری وجود نداشته باشد.
- هنگام ورود، رمز عبور را اشتباه وارد کند.
- ...

- ارتباط برنامه با کاربر:

سعی کنید بعد از گرفتن ورودی از کاربر، پیام مناسبی را به کاربر نشان دهید.

به مثال زیر توجه کنید. (این صرفاً یک نمونه است و شما می توانید رابط نوشتاری را طبق سلیقه خود طراحی کنید. در صورت تمایل به استفاده از رنگ های متفاوت در کنسول، میتوانید از کلاس موجود در این [لینک](#)، استفاده کنید.)

```
Already have an account ? (y/n)
n
Enter your User :
ap
User already Exists .
Enter your User :
AP
Enter your Password
spring
Your account has been created .
Already have an account ? (y/n)
y
Enter your User :
APP
Account Not Found . Try Again .Enter your User :
AP
account found .
Enter your Password
SPRING
Wrong password . Try Again
Enter your Password
spring

Login Successfull .
```

صفحه اصلی (منو برنامه)

کاربر بعد از ورود به برنامه، به صفحه اصلی برنامه منتقل خواهد شد. از این صفحه می‌توان به بخش‌های مختلف برنامه رفت که در ادامه هر کدام را توضیح می‌دهیم.

● صفحه شخصی

○ در این صفحه، کاربر بعضی از کارهای شخصی خود را انجام می‌دهد:

■ **توییت جدید بگذارد.**

■ **توییت‌های کاربر نمایش داده شوند.**

■ **صفحه شخصی خود را ویرایش کند.**

● زمانی که وارد این قسمت می‌شود، می‌تواند مشخصات خود را تغییر دهد.

■ **لیست ها (Followers, Followings, Black List)**

● هر کاربر می‌تواند کاربران دیگر را دنبال کند. (توضیحات این مورد در ادامه داده می‌شود.)

● کاربر در این قسمت می‌تواند لیست افرادی که او را دنبال کرده‌اند، یا او آنها را دنبال کرده است ببیند.

○ اگر نام کاربر خاصی از این لیست تایپ شود، پروفایل آن کاربر به نمایش در می‌آید. (توضیحات در ادامه)

● افراد مسدود شده توسط کاربر، در لیست سیاه قرار می‌گیرند.

○ کاربر می‌تواند افراد موجود در این لیست را در زمان دلخواه، رفع مسدودی کند.

■ **بخش info**

● پس از ورود به بخش info، مشخصات کاربر به نمایش در می‌آید.

■ **بخش اعلانات**

● **بخش درخواست‌ها**

○ کاربر باید بتواند درخواست‌های سایر کاربران برای دنبال کردن خود را ببیند و پاسخ دهد.

i. در این حالت پس از ورود به این قسمت، درخواست‌ها به کاربر نمایش داده می‌شوند.

ii. در پاسخ به هر کدام از این درخواست‌ها، سه گزینه پیش روی کاربر است:

● قبول کند.

● رد کند و برنامه به شخص مقابل این موضوع را اطلاع دهد.

● رد کند، بدون اینکه برنامه به شخص مقابل اطلاع دهد.

iii. در هر سه حالت، آن درخواست از لیست درخواست‌ها حذف می‌شود.

○ همچنین بتواند وضعیت درخواست‌های دنبال کردن خود از سایر کاربران را ببیند.

○ نحوه نمایش و پیمایش بین درخواست‌ها به عهده خودتان است.

● **بخش پیام‌های سیستم**

○ اگر کسی شروع به دنبال کردن کاربر کند، باید به او اعلام شود.

○ اگر کسی دنبال کردن کاربر را متوقف کند، به کاربر اطلاع داده می‌شود.

● صفحه تایم لاین

○ در این بخش کاربر می‌تواند توییت‌هایی که کاربرانی که دنبال می‌کند به اشتراک گذاشته‌اند و یا پسندیده‌اند و یا باز ارسال کرده‌اند را ببیند.

همچنین می‌توانید به سلیقه‌ی خود الگوریتمی را به کار ببرید که توییت‌هایی که احتمالا مورد پسند کاربر باشد، داخل تایم لاین نمایش داده شوند.

در این بخش باید لیستی از توییت‌های قابل نمایش موجود باشد که کاربر پس از ورود به تایم لاین از منوی اصلی برنامه، به صورت مستقیم و یا غیرمستقیم به صفحه مربوط به اولین توییت در این لیست هدایت شود. در صفحه مربوط به هر توییت حتما باید نام نویسنده و محتوای آن نمایش داده شود. همچنین می‌توانید موارد بیشتری نیز به انتخاب خودتان نمایش دهید. (برای مثال تاریخ انتشار و یا افرادی که توییت را پسندیده‌اند و...)

در صفحه مربوط به هر توییت با وارد کردن دستوری در cli باید بتوان عملیات‌های مشخصی را انجام داد :

- به توییت بعدی یا قبلی در لیست فعلی رفت.
- توییت را لایک کرد.
- توییت را به پیام‌های مورد علاقه (Saved messages) اضافه کرد.
- توییت را باز ارسال کرد.
- توییت را برای سایر کاربران، فرورارد کرد. (قوانین بخش پیام رسانی را ببینید.)
- نویسنده را مسدود کرد.
- نویسنده را ساکت کرد. (یعنی دیگر توییت‌های آن فرد به کاربر نشان داده نشود.)
- گزارش هرزنامه داد.
- کاربر به صفحه شخصی نویسنده هدایت شود.
- کاربر بتواند نظر خودش را در مورد توییت بنویسد که به لیست کامنت‌های آن اضافه شود.
- کاربر به لیست کامنت‌های توییت هدایت شود.

توضیحات مربوط به کامنت‌ها:

- هر توییت یک لیست از کامنت‌ها دارد.
- ماهیت هر کامنت کاملاً مشابه با یک توییت است. پس باید همه‌ی مواردی که برای توییت گفتیم برای هر کامنت نیز طراحی شود.
- در صفحه مربوط به هر کامنت با وارد کردن دستور بازگشت، کاربر به صفحه مربوط به کامنت و یا توییت بالاتر هدایت می‌شود. برای توییت‌هایی که کامنت نیستند با وارد کردن این دستور کاربر از تایم لاین خارج می‌شود.

نحوه حرکت بین توییت‌ها و کامنت‌ها را یک بار دیگر مختصر توضیح می‌دهیم. با ورود به تایم لاین ابتدا کاربر بر روی لیست اصلی حرکت می‌کند. سپس با ورود به کامنت‌های هر توییت حرکت بر روی لیست مربوطه انجام می‌شود. و با وارد کردن دستور بازگشت به لیست قبلی بر می‌گردد.

● صفحه اکسپلورر

این بخش دارای دو قسمت می‌باشد:

- بخش جست‌وجو: در فاز اول پروژه تنها کافی است با جست‌وجوی حساب کاربری به صفحه شخصی مربوط به آن حساب کاربری رفت.
- در بخش دوم توییت‌های تصادفی و یا پیام‌های پرتعداد شبکه اجتماعی به کاربر نمایش داده می‌شود. الگوریتم مربوط به انتخاب این توییت‌ها را به سلیقه خودتان طراحی کنید. سایر جزئیات این بخش مشابه تایم لاین می‌باشد.

● صفحه پیام رسانی

- با ورود به این قسمت، کاربر می تواند با سایر کاربران گفتگو کند.
- در بخش فعلی، برنامه شما تنها قابلیت گفتگوهای دوفره را دارد.
- **پیام های ذخیره شده (Saved Messages) را مشاهده کنید.**
 - کاربر می تواند پیام ها و یا توییت هایی که دوست دارد را در جایی برای خود ذخیره کند.
 - همچنین در این قسمت کاربر میتواند پیام هایی نیز برای خود بنویسد و ذخیره کنید.
- **پیام هایی که دیگران فرستاده اند را مشاهده کند.**
 - هر کاربر می تواند با کاربران دیگر صحبت کند و با آنها گپ بزند. (جزییات و منطق پیام رسانی در قسمت های بعدی توضیح داده می شود).
 - هنگامی که کاربر وارد این قسمت از برنامه می شود، تمام گفتگو هایی که انجام داده است را مشاهده می کند.
- در این قسمت، پیام های هر گپ به نمایش در نمی آید. صرفا نام کاربری اشخاصی که با آنها به گفتگو پرداخته است را مشاهده می کند.
- با نوشتن نام یکی از مخاطبان، به صفحه شخصی او منتقل شود. (جزییات پیام رسانی در ادامه آمده است)
- نیازی به اعمال ترتیب خاصی برای نمایش کاربران نیست. اما اگر مخاطبی برای کاربر پیامی ارسال کرده که هنوز ندیده است، آن را در بالا قرار دهید. (اگر چند مخاطب دارای این شرایط هستند، ترتیب بین آنها نیز مهم نیست، صرفا بالای سایر گپ ها قرار بگیرند کافیست).
- در کنار نام اشخاصی که کاربر هنوز پیام های آنها را نخوانده است، تعداد پیام های خوانده نشده را نیز نشان دهید. برای سایر اشخاص نیاز نیست.

● صفحه تنظیمات

○ تنظیمات حریم خصوصی

توضیحات سه مورد بعدی، در بخش منطق برنامه داده شده است. در این بخش کاربر باید بتواند آنها را تغییر دهد.

- تنظیمات public , private
- تنظیمات last seen & online
- باز و بسته کردن حساب کاربری (فعال یا غیرفعال بودن)
- تغییر رمز عبور

○ قابلیت حذف حساب کاربری

- کاربر می تواند حساب کاربری خود را حذف کند.
- پس از حذف کردن حساب کاربری، به طبع از تمام لیست های تمام کاربران حذف می شود.
- نام کاربری او آزاد می شود و شخص دیگری می تواند با آن ثبت نام کند.
- سایر جزئیات را می توانید به دلخواه خود طراحی کنید.

○ قابلیت Log Out

- کاربر باید بتواند از حساب کاربری خود خارج شود.

همچنین برنامه شما باید قابلیت های زیر را داشته باشد:

- سایر منطق برنامه شما که باید آنها را پیاده سازی کنید:

○ هر حساب کاربری ، باید یک سری تنظیمات حریم خصوصی داشته باشد.

■ تنظیمات Last Seen & Online

- شامل سه مدل : "همه"، "هیچکس" و "کسانی که کاربر آنها را دنبال کرده" است.
- هنگامی که کاربر وارد صفحه شخصی یک کاربر می شود باید در کنار سایر مشخصات، وضعیت او را نیز ببیند.
- کاربر می تواند این مورد را در بخش تنظیمات تغییر دهد.
- در این فاز صرفاً ۲ وضعیت داریم: last seen recently, last seen date, منظور از date آخرین زمانی است که کاربر آنلاین بوده است.
- به طور پیش فرض، وضعیت تمام افرادی که هنوز کاربر را دنبال نکرده اند، last seen recently نمایش داده می شود. (مستقل از وضعیت اصلی آنها)

■ تنظیمات privacy

- هر حساب کاربری، می تواند public یا private باشد.
- در صورتی که صفحه کاربر خصوصی باشد، توییت های او در محیط اکسپلورر نمایش داده نمی شود.
- اگر حساب کاربر خصوصی باشد، سایر افراد برای دنبال کردن او، باید ابتدا درخواست کنند.
- این مورد را در بخش تنظیمات می تواند تغییر دهد.

○ مشاهده صفحه شخصی یک کاربر

- هنگامی که کاربر، صفحه شخصی یک شخص دیگر را باز می کند، با موارد زیر مواجه می شود:

- ابتدا یک سری مشخصات آن کاربر نمایش داده می شود:
- نام و نام خانوادگی
- نام کاربری
- وضعیت last seen & online
- آن کاربر را دنبال کرده اید یا نه.
- اگر آن کاربر را دنبال کرده باشد، می تواند به او پیام دهد.
- بتواند آن کاربر را مسدود کند.
- بتواند آن کاربر را گزارش دهد.

○ دنبال کردن اشخاص

- کاربر باید بتواند سایر افرادی که در برنامه حساب کاربری دارند را دنبال کند.
- ابتدا باید وارد صفحه شخصی شخص دیگر شود. در صفحه شخصی او، می تواند گزینه دنبال کردن را انتخاب کند.
- اگر حساب کاربری شخص مقابل خصوصی باشد، باید حتماً به او درخواست دهد و منتظر نتیجه بماند.
- دنبال کردن و درخواست دادن یک شخص دیگر، منوط به آن است که توسط او مسدود نشده باشد.

○ دسته بندی و مدیریت آن

- کاربر باید بتواند افرادی را که دنبال کرده است را در دسته های دلخواه دسته بندی کند. (برای مثال دوستان، همکاران، فامیل، افراد مشهور و...)
- باید بتواند دسته های جدیدی را بسازد و یا حذف کند.
- باید بتواند افراد را به هر کدام از لیست ها اضافه یا از آن حذف کند.
- هر دنبال شونده می تواند در چند دسته قرار بگیرد.

- کاربر می‌تواند پیام ارسال کند به طوری که مخاطبین آن با توجه به این دسته‌ها مشخص شده باشند.
- ارسال پیام به افراد یک دسته، مشابه آن است که برای هر کدام از آنها یک پیام شخصی فرستاده باشید.
- در نتیجه اگر هر کدام از اشخاص مقابل نیز جوابی بدهد، فقط برای همین کاربر فرستاده می‌شود.
- توضیحات پیام‌رسانی در ادامه داده می‌شود.

○ قابلیت مسدود کردن و رفع مسدودی

- هر کاربر، می‌تواند افراد دیگر را مسدود کند.
- پس از مسدود شدن، فرد مسدود شده به لیست سیاه اضافه می‌شود.

○ باز و بسته بودن حساب کاربری

- حساب کاربری هر فرد، هنگامی که ایجاد می‌شود فعال است.
- هر کاربری می‌تواند حساب کاربری خود را برای مدتی غیر فعال کند. (این امر با حذف کردن حساب کاربری متفاوت است.)
- هنگامی که حساب کاربری یک کاربر، غیر فعال باشد، حساب کاربری او به طور موقت، در لیست‌های سایر کاربران نمایش داده نمی‌شود. دقت کنید که از آن لیست‌ها حذف نشده است و صرفاً نمایش داده نمی‌شود.
- تا زمانی که یک حساب کاربری غیر فعال باشد، توییت‌های او در اکسپلورر نمایش داده نمی‌شوند (تحت هر شرایطی).

○ پیام‌رسانی

- همانطور که قبلاً گفته شد، هر کاربر می‌تواند تحت شرایطی با دیگر کاربران گفتگو کند. پیام‌رسانی به دو صورت انجام می‌شود :

■ پیام‌رسانی شخصی :

- دو کاربر زمانی مجاز به گفتگو هستند که حداقل یکی از آنها، دیگری را دنبال کرده باشد.
- ارتباط بین دو کاربر تنها زمانی برقرار است، که هیچ یک در لیست سیاه دیگری نباشند.
- هنگامی که کاربر وارد صفحه گپ یک کاربر دیگر می‌شود:
 - باید بتواند به تمام پیام‌های گذشته دسترسی داشته باشد.
 - این پیام‌ها باید به ترتیب تاریخ ارسال باشند و ابتدا جدیدترین پیام‌ها نمایش داده می‌شوند.
 - باید بتواند به کاربر دوم، پیام دهد.
 - پیام‌های ارسالی کاربر اول و دوم، از هم قابل تفکیک باشند.
- توجه داشته باشید که نیازی به نمایش دادن یکباره تمام پیام‌ها نیست و می‌توانید در چند مرحله پیام‌ها را نشان دهید.
- نحوه نمایش پیام‌ها به سلیقه شماست. مثلاً می‌توانید در بین چند پیام، جدیدترین پیام (از لحاظ زمانی) را در بالا و قدیمی‌ترین پیام را در پایین نمایش دهید یا برعکس. اما این ترتیب را همیشه رعایت کنید .

■ پیام‌رسانی جمعی

- توجه داشته باشید که این مدل به معنای تشکیل گروه و ایجاد ارتباط بین تمام اشخاص گروه نیست. بلکه بدین معناست که کاربر به طور همزمان، یک پیام را برای چند کاربر دیگر ارسال می‌کند.
- در این حالت کاربر با نوشتن پیام مورد نظر و انتخاب افراد مورد نظر، پیام خود را ارسال می‌کند.
 - می‌تواند به صورت دستی، افراد مورد نظر را انتخاب کند.
 - می‌تواند از بین دسته‌هایی که ایجاد کرده است، دسته(ها)ی را انتخاب کند.
 - می‌تواند برای تمام کاربران ارسال کند.

- توجه داشته باشید که شرایط ارسال پیام در این قسمت نیز دقیقاً مطابق با پیام رسانی فردی هست.

- بنابراین، توجه داشته باشید که همیشه برای انتقال پیام به یک کاربر دیگر، لزومی به مراجعه به صفحه شخصی او نیست. برای مثال زمانی که کاربر یک توییت را برای یک فرد فوروارد می‌کند، یا به یک دسته از دوستان خود، پیامی را ارسال می‌کند.

○ ذخیره کردن اطلاعات

- تمام اطلاعات و اتفاقات برنامه باید ذخیره شوند. یعنی اگر کاربر در هر لحظه اراده به خروج از برنامه کند و در زمان دیگری به برنامه برگردد، اطلاعات و کارهایی که انجام داده است، نباید از دست رفته باشد.

مثالی کوتاه از CLI

در زیر، نمونه‌ای از دستورات آمده است. توجه داشته باشید که هیچ لزومی به استفاده از ادبیات دستورات زیر نیست (شما میتوانید به هر شیوه‌ای و با هر ادبیاتی، ارتباط کاربر و برنامه را برقرار کنید).

دستورات مربوط به مدیریت کاربران:

در ابتدا پس از شروع و اجرای برنامه از کاربر می‌پرسید که می‌خواهد ثبت نام کند یا قبلاً حساب داشته است:

already have an account?(y/n)

• ثبت نام کاربر جدید در برنامه

اگر کاربر جدید بود یک نام کاربری و گذرواژه برای ایجاد حساب می‌گیرید:

(if Y)

Username:

Password:

• ورود به برنامه

و در صورتی که از قبل حساب کاربری داشت این بار هم برای ورود به حساب نام کاربری و گذرواژه را از کاربر دریافت می‌کنید:

(if N)

Username:

Password:

• خروج از برنامه

زمانی که وارد برنامه شده‌اید و تصمیم بر خروج از برنامه می‌گیرید دستور زیر را زده و از برنامه خارج می‌شوید (LOG OUT کنید):

quit

همچنین برای خروج از برنامه و توقف اجرای برنامه دستور زیر را بزنید: (این دستور در هر بخش از برنامه قابل اجراست).

exit

• حذف کاربر از برنامه

برای حذف کاربر ابتدا وارد برنامه شده و لاگین می‌کنید و سپس دستور:

delete user

را زده و پس از آن گذرواژه حساب کاربری را می‌خواهید:

Password:

یعنی در اینجا حسابی که در آن قرار دارید را با وارد کردن گذرواژه‌اش حذف می‌کنید. بعد از این عملیات کاربر باید از برنامه به طور اتوماتیک خارج شود.

پیشاپیش عید شما مبارک =)

تیم درس برنامه نویسی پیشرفته، زمستان 1399