

به نام خدا



## درس برنامه سازی پیشرفته

فاز سوم پروژه

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال اول ۰۱ - ۰۰

---

استاد:

دکتر وحید سلمانی

مهلت ارسال:

جمعه ۱۵ بهمن ۱۴۰۰ ساعت ۱۴:۵۹:۲۳

مسئول پروژه:

حسام اثنا عشری

مسئول فاز سوم:

علی ثالثی

طراحان فاز سوم:

مهراد میلانلو، فاطمه علیمزادی، علی رحیمی زاد، علی ثالثی، هیربد بهنام و پوریا عارفی

مسئولین تنظیم مستند:

علی ثالثی و هیربد بهنام

# فهرست

۲	نکات قابل توجه
۳	مقدمه
۳	معماری Client-Server
۳	احراز هویت
۴	اتاق گفت‌وگو (Chat Room)
۴	سرویس نوتیفیکیشن
۴	اعمال تغییرات به صورت لحظه‌ای
۵	قابلیت کیک کردن یا تغییر مشخصات کاربران توسط sysadmin و تیم لیدر
۶	دسترسی‌های این قسمت
۶	امکانات مورد نیاز
۷	بخش‌های امتیازی
۷	اتاق گفت‌وگو (Chat Room)
۷	سرویس نوتیفیکیشن
۸	کار با پایگاه داده



## نکات قابل توجه

- پس از اتمام این فاز، در گیت خود یک تگ با ورژن "v3.0.0" بزنید. در روز تحویل حضوری این tag بررسی خواهد شد و کدهای پس از آن نمره‌ای نخواهد گرفت. برای اطلاعات بیشتر در مورد شیوه ورژن‌گذاری، می‌توانید به [این لینک](#) مراجعه کنید. البته برای این پروژه صرفاً رعایت کردن همان ورژن گفته شده کافیست، اما خوب است که با منطق ورژن‌بندی هم آشنا بشوید.
- در روز تحویل حضوری مشارکت تمام اعضای تیم در پروژه بررسی خواهد شد و در صورت عدم مشارکت بعضی از اعضا، نمره‌ی ایشان برای آن فاز پروژه "صفر" لحاظ می‌گردد. مشارکت، با توجه به commit های افراد تیم در مخزن گیت‌هاب پروژه بررسی می‌شود.
- توجه کنید که به دلیل نزدیک بودن به مهلت ارسال نمرات، امکان تاخیر برای فاز سوم وجود ندارد.
- در صورت کشف تقلب از هریک از تیم‌ها، برای بار اول منفی نمره آن فاز برای آن تیم ثبت می‌شود و برای بار دوم، نمره منفی کل پروژه برای تیم لحاظ خواهد شد که معادل مردود شدن در درس است.



## مقدمه

همانطور که می‌دانید، در دو فاز قبلی بخش‌های مربوط به منطق و گرافیک پروژه را پیاده‌سازی کردیم. در این فاز قابلیت‌های تحت شبکه را به پروژه اضافه می‌کنیم تا کاربران مختلف بتوانند به صورت همزمان و از راه دور از آن استفاده کنند و پروژه به یک برنامه کاربردی و عملی تبدیل بشود.

## معماری Client-Server

معمولا برنامه‌های تحت شبکه به صورت peer-to-peer یا client-server نوشته می‌شوند. در معماری peer-to-peer کامپیوترها به طول مستقیم به همدیگر وصل می‌شوند و داده‌ها را جا به جا می‌کنند. از مثال‌های این نوع معماری می‌توان به تورنت، یا بعضی از بازی‌ها مثل GTA Online اشاره کرد. در نوع کلاینت سرور، همیشه یک یا چند سرور مرکزی وجود دارد که کلاینت‌ها درخواست‌هایشان را به آنها می‌فرستند و سرور با توجه به درخواست آنها جواب می‌دهد. به عنوان سایت‌هایی که روزمره از آنها بازدید می‌کنیم، تلگرام و دیسکورد از جمله این دسته هستند.

در این فاز از پروژه شما باید از معماری کلاینت-سرور استفاده بکنید. بدین صورت که در هنگام لاگ‌این، چت و ... هر کلاینت یک درخواست به سرور می‌فرستد که سپس توسط آن پردازش می‌شود و جوابی به کلاینت فرستاده می‌شود. می‌توانید برنامه‌ی سرور را به صورت کاملاً جدا از کلاینت طراحی کنید یا اینکه کاری کنید که یک برنامه در صورت نیاز بتواند به عنوان سرور نیز شروع به کار کند.

## احراز هویت

زمانی که کلاینت شما می‌خواهد با سرور ارتباط برقرار کند، سرور باید به روشی بفهمد که دارد با چه کلاینت یا کاربری اطلاعات رد و بدل می‌کند. فرض کنید که در ابتدا شما در سایتی لاگ‌این می‌کند و سپس بر روی تنظیمات کاربری کلیک می‌کنید. سرور باید از کجا بفهمد که باید تنظیمات چه کسی را نشان دهد؟ در اینجا دو راه معروف برای احراز هویت کاربران وجود دارد:

- احراز هویت بر اساس Token: در این نوع از احراز هویت، زمانی که کاربر وارد سایت می‌شود سرور به آن کاربر یک Token - که معمولا یک رشته‌ی رندوم و یکتا است - را می‌فرستد. سپس کاربران باید به ازای هر درخواست از سرور این Token را نیز به سرور ارسال کنند که سرور متوجه بشود که با چه کسی در حال برقراری ارتباط است. معمولا Token ها بعد از مدتی و یا بعد از logout منقضی می‌شوند.



- احراز هویت بر اساس Session: در نوع از احراز هویت زمانی کلاینت یک socket برای برقراری ارتباط با سرور باز می‌کند و در آن اطلاعات کاربری خود را می‌فرستد. بعد از فرستادن آن، socket بسته نمی‌شود و هر درخواستی که کلاینت از سرور داشت باید در آن socket فرستاده شود.

ما به شما پیشنهاد می‌کنیم که از احراز هویت بر پایه‌ی توکن (token-based authorization) استفاده کنید.

از لینک‌های زیر می‌توانید برای مطالعه بیشتر استفاده کنید:

- [تفاوت‌های این دو روش احراز هویت](#)

- [بررسی این دو روش در برابر هم](#)

- [احراز هویت بر پایه‌ی توکن](#)

## اتاق گفت‌وگو (Chat Room)

همانطور که در فاز یک اشاره شده بود، در این فاز قصد داریم ویژگی اتاق در منوی چت‌روم همانند همه چت‌روم‌هایی که در هر برنامه دیگری دیده‌اید، هر کاربر قابلیت ارسال پیام دارد و پیام‌های ارسال شده به ترتیب زمان ارسال برای هر کس نمایش داده می‌شوند.

## سرویس نوتیفیکیشن

سرویس نوتیفیکیشن باید شامل ویژگی‌های زیر باشد:

- اگر ادمین در چت‌روم پیامی را ارسال یا پین (در صورت انجام دادن این بخش) کرد، برای اعضای تیم اعلان با پیام مربوطه ارسال شود.
- با تغییر دادن وضعیت یک تسک توسط هر کدام از اعضا به دیگر اعضای تیم و ادمین اعلان با پیام مربوطه ارسال شود.
- در صورت اضافه شدن عضوی از تیم به انجام‌دهندگان یک تسک، برای آن عضو جدید و دیگر اعضا اعلان با پیام مربوطه ارسال شود.

## اعمال تغییرات به صورت لحظه‌ای

- برای همه منوها در همان لحظه انجام تغییرات، تغییرات انجام شده توسط کلاینت برای سرور ارسال شود و در آن جا ذخیره شود.



- برای همه منوها دکمه refresh قرار داده شود تا در هر لحظه در صورت درخواست کاربر اطلاعات جدید ذخیره شده در سرور (اطلاعات ذخیره شده پس از آخرین به روز رسانی یا آخرین ورود به منو) به کلاینت منتقل شده و در آن جا نمایش داده شود.
- **امتیازی:** در صورتی که منوهای چت و یا نوتیفیکیشن باز باشند هر چند ثانیه یک بار به صورت اتوماتیک درخواست به روزرسانی منو برای سرور ارسال شود تا در صورتی که نوتیفیکیشن یا پیام جدیدی بود نمایش داده شوند.
- **امتیازی:** اگر قابلیت امتیازی نمایش دادن چت باکس و منوی نوتیفیکیشن در گوشه منوهای دیگر پیاده سازی شده است به روزرسانی اتوماتیک برای آن ها هم پیاده سازی شود.

## قابلیت کیک کردن یا تغییر مشخصات کاربران توسط sysadmin و تیم لیدر

کاربران در برنامه‌های Jira را می‌توان به صورت دستی یا از طریق مدیریت کاربر توسط پنل مخصوص مدیر پروژه و یا leader team که به آن مدیریت خارجی می‌گویند مدیریت کرد. از طریق صفحه مدیریت تیم شما می‌توانید تا کاربران را به صورت دستی مدیریت کنید و در صورت لزوم مشخصات کاربران را تغییر داده، آن‌ها را از گروه اخراج کرده و یا دسترسی‌های آن‌ها را مدیریت کنید. برای اینکه کاربر بتواند وارد شده و از امکانات برنامه استفاده کند، باید دسترسی لازم را داشته باشد. دسترسی برنامه با عضویت در گروهی که به یک برنامه اختصاص داده شده است به دست می‌آید. عضویت در این گروه‌ها را می‌توان در هر زمان بر اساس هر کاربر تغییر داد. و این تغییرات توسط مدیر پروژه و team leader میسر است. برای آن که از امکانات این بخش استفاده کنید باید ابتدا احراز هویت شده و متناسب با دسترسی‌های خود عمل کنید. زیرا همان‌طور که در دوفاز گذشته نیز مشاهده کردین، ادمین و تیم لیدر قابلیت‌های به خصوصی داشتند. یکی از بخش‌های فاز ۳ این است که بتوانید آن قابلیت‌ها را با استفاده از معماری کلاینت-سرور به گونه‌ای پیاده‌سازی که نتایج تغییراتی که ادمین و تیم لیدر اعمال می‌کنند به صورت آنی برای کاربران در دسترس باشند. از جمله این تغییرات می‌توان به حذف کاربر، تغییر رتبه، ایجاد تسک و ... اشاره کرد. می‌توانید با استفاده از فایل این کار را انجام دهید ولی بهتر است که از DBMSها استفاده کنید. برای کسب اطلاعات بیشتر به بخش DBMSهای داک مراجعه کنید.



## دسترسی های این قسمت<sup>۱</sup>

- **مدیر کل:** شما باید مجوز جهانی Jira Administrator یا Jira System Administrator داشته باشید تا بتوانید کاربران را در برنامه‌های Jira مدیریت کنید همچنین قاعدتا چون سطح دسترسی بیشتری از مدیر پروژه و team leader دارید پس اعمال آن‌ها را نیز می‌توانید انجام دهید.
- **مدیر پروژه و team leader:** شما با استفاده از این مجوز می‌توانید کاربر را به یک پروژه یا تیم دعوت کنید و به طور مشابه آن‌ها را از تیم حذف کنید ( به اصطلاح کیک کردن کاربران)

## امکانات مورد نیاز

۱. **تغییر نام کاربری، رمز عبور، ایمیل و پسورد**  
مواردی از جمله نام کاربری، نام کامل، ایمیل و پسورد کاربران توسط مدیر کل می‌تواند به صورت مستقیم تغییر کند و برای این منظور پس از آنکه کاربر وارد شده به عنوان مدیر کل احراز هویت شد می‌تواند موارد فوق را تغییر دهد.

۲. **امکان remove کردن یک کاربر**  
همچنین مدیر می‌تواند در صورت نیاز کاربر را remove کند. برای پاک کردن کاربر دو شیوه می‌توان استفاده کرد:

(آ) **آن اکانت را به طور کامل پاک کنید**  
در صورت پاک کردن کامل یک اکانت دیگر امکان استفاده از آن و همچنین مشاهده تاریخچه فعالیت های آن اکانت وجود ندارد و به طور کامل از دیتابیس (که به یک شیوه برای ذخیره فعالیت ها پیاده سازی شده است) پاک شده و در دسترس نیست.

(ب) **آن اکانت را غیر فعال کنید**  
در این صورت کاربر دیگر نمی‌تواند از آن اکانت استفاده کند ولی همچنان تاریخچه فعالیت‌های وی و مواردی که قبلا برای آن حساب کاربری ذخیره شده است موجود خواهد بود و امکان استفاده در آینده در صورت فعال شدن یوزر وجود دارد

۳. **امکان ناشناس کردن یک عضو**  
ناشناس کردن یک کاربر، هرگونه داده ای را که می‌تواند او را به عنوان یک شخص واقعی شناسایی کند، پنهان یا حذف می‌کند. هرگونه داده های شخصی آنها، مانند نام

<sup>۱</sup> به داک فاز اول مراجعه شود



کاربری یا نام کامل، به یک نام مستعار غیرقابل تشخیص تغییر می‌کند. این فرایند باید قابل بازگشت باشد و بتوان یک کاربر ناشناس را به کاربر عادی تبدیل کرد.

#### ۴. دعوت کردن یک کاربر که عضو jira نیست به پروژه

این امکان نیز بدین صورت خواهد بود که می‌توان با وارد کردن ایمیل کاربر توسط مدیر پروژه، کاربر مورد نظر به پروژه دعوت گردیده و پس از ثبت نام به پروژه مربوطه به صورت خودکار اضافه شود.

**پیشنهاد:** برای ایجاد دسترسی برای کاربری که وجود ندارد می‌توانید یک کاربر بدون مشخصات ایجاد کنید و بعد از ورود کاربر آن را تکمیل کنید، مثلاً می‌توانید با ساختن یک UUID یک توکن برای کاربر ایمیل کنید (یا به یک شیوه دیگر در اختیار کاربر قرار دهید) و با وارد کردن آن UUID در یک قسمت خاص امکان اضافه شدن وی به پروژه و ثبت نام و تکمیل اطلاعات را فراهم آورید.

## بخش‌های امتیازی

### اتاق گفت‌وگو (Chat Room)

علاوه بر اتاق گفت‌وگوی عمومی، می‌توانید بخش‌های زیر را هم به صورت امتیازی پیاده‌سازی کنید.

- برای چت کردن هر یک از اعضای تیم با ادمین، صفحه چت جدا ایجاد شود که در منوی چت‌روم قابلیت انتخاب و نمایش داده‌شدن به جای چت‌روم اصلی را داشته‌باشد.
- ادمین تیم در چت‌روم قابلیت‌های حذف هر پیام، پین کردن هر پیام را داشته‌باشد. همچنین اگر خواست ارسال پیام برای اعضای تیم را ممنوع یا آزاد کند.
- همه اعضای تیم امکان ادیت یا حذف پیام ارسالی خودشان را داشته‌باشند.
- در بقیه منوهای برنامه هم در گوشه صفحه یک باکس باز باشد که بتوان پیام‌ها را در آن هم خواند.
- قبل از ورود به چت‌روم تعداد پیام‌های خوانده‌نشده آن نمایش داده‌شوند.

### سرویس نوتیفیکیشن

در سرویس نوتیفیکیشن می‌توانید علاوه بر موارد گفته شده این موارد را نیز پیاده کنید:

- کسی که تغییری در وضعیت تسک ایجاد کرده تصمیم بگیرد اعلان فقط برای ادمین ارسال شود یا همه اعضای تیم آن را دریافت کنند.





- هر عضو تیم بتواند دریافت اعلان را فقط برای اعلان های ارسالی توسط ادمین باز بگذارد.
- همانند قسمت چت روم برای اعلان ها هم یک باکس در گوشه صفحه هر منو ایجاد شود.

## کار با پایگاه داده

پایگاه داده مجموعه‌ای از داده‌هاست که به صورت منظم و با ساختار خاصی درون رایانه نگهداری می‌شوند. پایگاه های داده معمولاً توسط سامانه‌های مدیریت پایگاه داده (Database Management Systems یا DBMS ها) کنترل می‌شوند. در این فاز، شما می‌توانید پایگاه داده‌ی خود را درون یک فایل به صورت ساده نگهداری کنید اما توصیه می‌شود تا بر روی DBMS ها کار کنید؛ چرا که با توجه به شرایط کامپیوترهای موجود، DBMS های توسعه یافته‌ای وجود دارند که عملیات های معمول مانند جستجو، دریافت داده‌ها با شرایط خاص و تغییر در پایگاه داده را با سرعت بالا انجام می‌دهند. در اینجا دو نمونه از انواع آن‌ها را ذکر کرده‌ایم که با توجه به موقعیت می‌توانید از آن‌ها استفاده کنید.

### • پایگاه‌داده‌های رابطه‌ای (RDBMS)

این خانواده از DBMS ها بسیار وسیع و پرکاربرد هستند و آن‌ها از زبان درخواست ساخت یافته (Structured Query Language یا SQL) استفاده می‌کنند. در RDBMS ها، داده‌ها در جدول هایی با ویژگی‌های مشخص ذخیره می‌شوند که با استفاده از آن‌ها می‌توان عملیات‌های معمول مانند انتخاب، درج، به روز رسانی، حذف و سایر ابزارهای مورد نیاز برای مدیریت پایگاه داده‌ها در کنار هم داشته باشیم. SQLite، PostgreSQL و MySQL از جمله معروف ترین RDBMS ها هستند.

[در این لینک](#) می‌توانید استفاده SQLite را در همراهی با جاوا مشاهده کنید.

[در این لینک](#) می‌توانید استفاده MySQL را در همراهی با جاوا مشاهده کنید.

### • پایگاه‌داده‌های غیررابطه‌ای (NoSQL)

با گسترش تکنولوژی و در نتیجه نیازهای بشر، دیگر RDBMS ها پاسخگوی همه‌ی نیازها نبودند و برای برخی مفاهیم، نیاز به نوع دیگری از DBMS ها حس می‌شد. در نتیجه‌ی این نیازها DBMS های دیگری مانند NoSQL ها ساخته شدند. این نوع از پایگاه‌های داده بر پایه‌ی جدول‌ها و روابط نیستند و از زبان SQL پشتیبانی نمی‌کنند. در عوض، قابلیت های دیگری مانند نگهداری پایگاه داده در چند سرور مجزا، جستجو در گراف و ... را می‌توان در این پایگاه‌های داده مشاهده کرد. مواردی مانند Redis و Cassandra و MongoDB و ElasticSearch از دیتابیس های شناخته شده و پرکاربرد NoSQL هستند.



## در این لینک میتوانید استفاده MongoDB را در همراهی با جاوا مشاهده کنید.

در این فاز شما محدودیتی در استفاده از پایگاه‌های داده ندارید جز آنکه اطلاعات در دیسک ذخیره شود و با بازگشت در برنامه بتوان به آن‌ها دسترسی پیدا کرد. می‌توانید با توجه به نیاز و علاقه‌تان، از هر نوعی از پایگاه‌های داده که می‌خواهید استفاده کنید.

موفق باشید!