

## توضیحات مربوط به جیرا:

ابزار Jira برای برنامه‌ریزی، پیگیری پیشرفت و مدیریت وظایف تیم در این پروژه استفاده می‌شود. هدف اصلی از به‌کارگیری این ابزار، ایجاد شفافیت در روند کار، هماهنگی بین اعضای تیم و اطمینان از پوشش کامل نیازمندی‌های پروژه است.

بورد جیرا نمای کلی از فعالیت‌ها و وضعیت هر بخش را فراهم می‌کند تا اعضا بتوانند در هر لحظه ببینند چه کارهایی در حال انجام، در انتظار یا تکمیل شده هستند. بورد پروژه در قالب Scrum Board پیاده‌سازی می‌شود.

ستون‌های اصلی بورد شامل مراحل زیر هستند:

1. To Do – کارهایی که آماده‌ی شروع هستند و برای اجرای دوره فعلی انتخاب شده‌اند.
2. In Progress – تسک‌هایی که در حال انجام هستند.
3. Review / Testing – کارهایی که نیاز به بازبینی یا تست دارند.
4. Done – تسک‌هایی که به طور کامل تحویل و تأیید شده‌اند.
5. Pending – تسک‌هایی که برای انجام به خاطر شرایطی متوقف می‌شوند.

هر تسک هنگام جابه‌جایی بین مراحل باید وضعیت خود را به‌روزرسانی کند تا جریان کار به‌صورت شفاف قابل مشاهده باشد.

در جیرا برای هر واحد کاری یک issue جداگانه تعریف می‌شود. هر issue بر اساس نوعش دسته‌بندی می‌شود:

- **Story** – وظایف کلان که یک ویژگی قابل مشاهده برای کاربر را توصیف می‌کنند (مثل "نمایش موارد روی نقشه" یا "احراز هویت با ایمیل دانشگاه").
- **Task** – وظایف فنی یا زیرکارهایی که در راستای یک Story انجام می‌شوند (مثل "اتصال API نقشه به بک‌اند").
- **Bug** – خطاها یا مشکلاتی که باید رفع شوند.
- **Epic** – مجموعه‌ای از Storyها که یک بخش بزرگ از پروژه را پوشش می‌دهد (مثلاً Epic "چت‌بات" یا "سیستم احراز هویت").

هر Story باید:

- عنوانی کوتاه و واضح داشته باشد.
- شرح دقیق هدف، ورودی‌ها، خروجی‌ها و معیار پذیرش (Acceptance Criteria) را شامل شود.
- زمان تخمینی انجام و اولویت (Priority) آن تعیین شود.

هر Task باید:

- با برچسب (Label) مشخص شود تا ماژول مربوطه (مثلاً map, auth, chatbot, ui) مشخص گردد.
- به فرد مسئول (Assignee) اختصاص داده شود.

در آغاز هر فاز، تسک‌ها بر اساس خروجی‌های مورد انتظار آن فاز (Deliverables) تعریف و بین اعضا تقسیم می‌شوند.

تسک‌های بزرگ‌تر به Story و زیرتسک‌های قابل انجام در مدت کوتاه‌تر تقسیم می‌شوند تا مدیریت آن‌ها ساده‌تر باشد.

هر عضو تیم موظف است در پایان هر جلسه یا بازه کاری وضعیت تسک‌های خود را به‌روزرسانی کند و در صورت نیاز توضیح یا ضمیمه‌ای (مثل لینک، فایل، یا اسکرین‌شات) اضافه نماید.

در جلسات هماهنگی (Daily Standup یا Retro)، اعضا با استفاده از بورد، کارهای انجام‌شده و موانع موجود را مرور می‌کنند تا به‌صورت گروهی تصمیم‌گیری شود. در هر اسپرینت حداقل ۴ دیلی استندآپ باید برگزار شود. وویس این جلسات را در **این گروه** تلگرام قرار دهید. توجه کنید هر روز حداکثر یک وویس به عنوان دیلی استندآپ پذیرفته می‌شود. بنابراین فرستادن تمام وویس‌ها در روز پایانی نمره‌ای معادل یک دیلی خواهد داشت. همچنین در جلسه رترو روند کلی کارها طی اسپرینت واری و در سه دسته Start / Stop / Continue بحث می‌شوند. نتیجه این جلسه را در یک داک حداکثر ۲-۳ صفحه‌ای آپلود نمایید.

به بورد خود اکانت [sad.fall.2025@gmail.com](mailto:sad.fall.2025@gmail.com) مربوط به دستیاران آموزشی را اضافه نمایید.

در پایان هر اسپرینت نیز burndown chart اسپرینت را در کوئرا آپلود نمایید.

هر دسته تسک (بک‌اند و فرانت) باید **معیار پذیرش (Definition of Done)** مشخصی داشته باشد تا بدانیم چه زمانی کار واقعاً کامل است که در اسپرینت اول مشخص می‌کنید.

در پایان هر فاز (مثلاً فاز طراحی، توسعه اولیه، تست)، بورد به‌عنوان سند اجرایی وضعیت پروژه مورد استفاده قرار می‌گیرد و باید برای مستندسازی تحویل‌داده‌ها به ناظر پروژه نیز ضمیمه شود.

## موارد مربوط به devops

استقرار دائم پروژه

تا جای ممکن باید برنج main شما همواره up باشد و با merge کردن دیگر برنچ‌ها فرایند تست و دیپلوی روی آن انجام شود.

در راستای عدم ایجاد commit خراب و دارای change اضافی، برای git خود تعدادی custom hook تعریف کنید تا عملیات linting و formatting به صورت pre-commit روی فایل‌های stage شده‌ی پروژه انجام شود. این کار از ایجاد change بیجا در commit افراد مختلف گروه جلوگیری می‌کند.

## اسپرینت ۱:

تحويل موارد زیر اجباری است:

### ۱) DoD:

برای هر دو بخش بکند و فرانت به طور جداگانه DoD تحويل داده شود. (برای هر بخش حداکثر دو صفحه)

همچنین در یک فایل text موارد زیر را بنویسید:

- آدرس مخزن گیت‌هاب (به صورت پابلیک)
- آدرس محل نوشتن یا بارگذاری مستندات (گوگل داک، گوگل درایو، وان‌درایو، وان‌نوت و ...) (به صورت پابلیک)
- آدرس بورد اسکرام روی Jira Cloud (اکانت دستیاران را به برد اضافه کنید)

### ۲) فیگما (UI/UX)

در این بخش دیزاین صفحات ثبت نام و ورود با استفاده از فیگما صورت می‌گیرد. نحوه پیاده‌سازی در فیگما به عهده تیم است اما پیشنهاد می‌شود از ui kit‌هایی نظیر chakra یا ant یا material استفاده کنید. همچنین برای راحتی تغییرات ثانویه پیشنهاد می‌شود در صورت استفاده نکردن از ui kit‌ها، طراحی را با تعریف کامپوننت‌ها انجام دهید.

همچنین نیاز است در ابتدای کار یک brand book یا style guide سازمانی مخصوص به خود را داشته باشیم. چرا که هر استارت‌آپی نیاز دارد هویت بصری مختص خودش را داشته باشد. مواردی نظیر فونت و لوگو و پالت‌رنگی و قواعد دیزاین و... باید در این سند موجود باشد و پیش از شروع دیزاین کردن ui/ux محصول باید به توافق برسند.

### اهداف اسپرینت:

پیاده‌سازی سیستم احراز هویت کاربران (به همراه طراحی در فیگما) و امکان ثبت، ویرایش و حذف آیتم‌های گمشده/پیداشده. نمایش ابتدایی نقشه دانشگاه.

## اسپرینت ۲:

تحویل موارد زیر اجباری است:

### نمودارها:

مدلسازی از دیرباز در حوزه‌ی مهندسی نرم‌افزار مورد استفاده قرار گرفته است. ارزش مدلسازی در تسهیل ارتباط و درک بهتر سیستم‌های نرم‌افزاری است. یک نمودار خوب می‌تواند نقش زیادی در انتقال موثر ایده‌ها و مفاهیم طراحی ایفا کند، خصوصا زمانی که هدف، پرهیز از ورود به جزئیات فنی گسترده باشد.

در این پروژه از شما خواسته می‌شود که ۳ نمودار UML زیر را رسم کنید:

#### نمودار Class

نمودار Class انواع اشیاء موجود در سیستم و روابط مختلفی که میان آن‌ها وجود دارد را توصیف می‌کند. این نمودارها همچنین ویژگی‌ها (Properties) و عملیات‌های (Operations) مربوط به هر کلاس را به همراه محدودیت‌هایی که بر نحوه‌ی ارتباط میان اشیاء وجود دارد نمایش می‌دهند.

نمودار Class را برای کامپوننت **ارتباط کاربران با یکدیگر** رسم کنید و کلاس‌ها و روابط بین آنها را نمایش دهید.

#### نمودار Use-Case

نمودار Use-Case نشان می‌دهد چه کارهایی در سیستم انجام می‌شود، بدون آنکه به ترتیب انجام آنها اشاره کند. در این نمودار نشان می‌دهیم چه سرویس‌هایی به actor های بیرونی ارائه می‌دهد.

نمودار Use-Case را **تنها برای کامپوننت authentication/authorization** سیستم خود رسم کنید.

#### نمودار Activity

نمودار Activity جزو نمودارهای رفتاری uml محسوب می‌شود که کارها و ترتیب انجام آنها را نمایش می‌دهد. این نمودارها روشی برای توصیف منطق اجرا، فرایندهای کسب و کار و جریان کار هستند. از بسیاری جهات، این

نمودارها نقشی مشابه فلوجارتها ایفا می‌کنند اما تفاوت اصلی میان آنها و فلوجارتها در این است که نمودارهای Activity از رفتارهای همزمان و parallel نیز پشتیبانی می‌کنند. نمودار Activity را تنها برای کامپوننت چت بات سیستم خود رسم کنید.

تمام نمودارهای uml در [این ویدیو](#) توضیح داده شده است.

این ویدیو به آشنایی بهتر با نمودارهای Class و Activity و Use-Case به شما کمک می‌کند. همچنین اسلایدهای درس در این باره را مطالعه نمایید.

## اهداف اسپرینت:

مدلسازی سیستم، نمایش آیتمها در نقشه، افزودن قابلیت‌های تعامل کاربران از طریق کامنتها و گزارش محتوای نامناسب. پیاده‌سازی سیستم جست‌وجو و فیلتر برای آیتمها.

## اسپرینت ۳:

تحویل موارد زیر اجباری است:

### تست

برای دو سرویس اصلی (مانند مدیریت آیتمها، احراز هویت، ارتباط کاربران با یکدیگر) حداقل ۵ یونیت تست (به ازای هر سرویس) بنویسید. هرکدام از تستها باید معنادار باشند و باعث افزایش test coverage شوند.

## Architecture document

مستند معماری باید شامل بخش‌های زیر باشد:

### - معماری سطح بالا پروژه

در این بخش باید معماری سطح بالا پروژه و ساختار کلی سیستم توضیح داده شود. بهتر است راه حل های مختلف برای معماری پروژه را بررسی کنید و یکی از این راه حل ها را با دلیل انتخاب کنید. به عنوان مثال، ممکن است معماری Backend پروژه به صورت Monolithic یا Microservice طراحی شود. در این صورت لازم است مزایا و معایب هر رویکرد بررسی و دلیل انتخاب یکی از آنها برای پروژه توضیح داده شود.

توجه داشته باشید که این موارد صرفاً نمونه‌هایی از گزینه‌های ممکن هستند. بسته به نیازمندی‌های پروژه می‌توانید سایر الگوهای معماری را نیز ارزیابی کنید.

### - تکنولوژی‌های مورد استفاده در پروژه

فناوری‌ها و ابزارهای مورد استفاده در پروژه را مشخص کنید. به عنوان نمونه مشخص کنید از چه کدام framework ها برای پیاده‌سازی Frontend و Backend استفاده می‌شود.

### - پاسخ به نیازمندی‌های غیروظيفه‌ای

توضیح دهید که چگونه معماری انتخاب شده به نیازمندی‌های غیروظيفه‌ای پروژه پاسخ می‌دهد.

## اهداف اسپرینت:

اضافه کردن چت‌بات برای جست‌وجوی آیتمها - بهبود ویژگی‌های نقشه - معماری سیستم