

گزارش پروژه چهارم

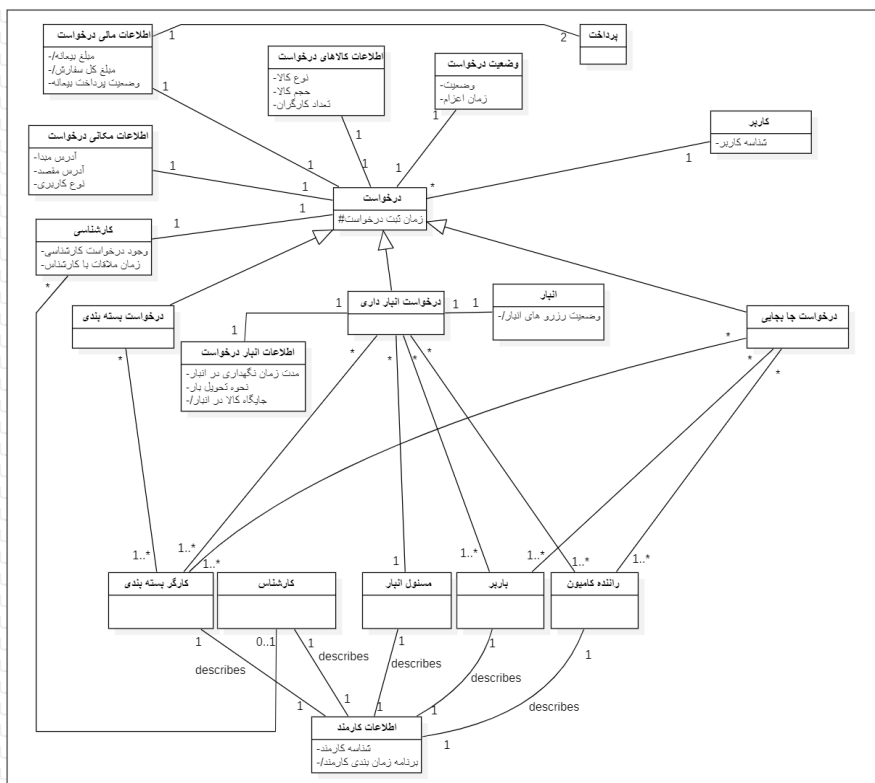
- سید احسان حسن بیگی-810197619
- علی بابابیگ-810197463
- علی بهاری-810196688

مدل دامنه:

در ابتدا مدل دامنه خود را که برای فاز قبل تهیه کرده بودیم، به صورت زیر تغییر دادیم.

تغییرات اعمال شده به دلیل وجود تعداد زیاد attribute ها در کلاس «درخواست» می باشد که با اصل های **high cohesion** و **low coupling** در تناقض بود. بنابراین attribute هایی را که با یکدیگر در یک خانواده قرار می گرفتند، تحت عنوان کلاس های جدا قرار دادیم که لیست آن ها به شرح زیر است:

- اطلاعات کالاهای درخواست
- اطلاعات مکانی درخواست
- اطلاعات مالی درخواست
- وضعیت درخواست
- کارشناسی



مدل دامنه جدید

نمودار تعامل:

در این قسمت به تخصیص method ها به هر کلاس پرداختیم.

مهم ترین اصلی که در این قسمت بر طراحی ما حاکم است، اصل **expert doer** است که هر تسک را به مرتبط ترین کلاس می سپاریم. به عنوان مثال کلاس «کاربر» دارای دو متد اصلی `init_transit_order` و `init_stash_order` است که دو مورد کاربرد اصلی و سطح بالایی هستند که کاربر با هدف انجام آن ها به سیستم ما درخواست می دهد. استفاده از این اصل در طراحی ما گسترده بوده و بر تمامی method ها اعمال شده است. استفاده از اصل **creator** نیز در تمامی موارد بر این اساس بوده است که کلاس `container` آبجکت از کلاس مورد نیاز را ساخته و `association` را برقرار می کند و با آبجکت ساخته شده در ارتباط می باشد. نمونه های استفاده از این اصل را می توان با بررسی متد `create` در نمودار تعامل یا نمودار کلاس مشاهده کرد.

در رابطه با اصل **controller** نیز می توان به کلاس «تخصیص منابع» اشاره کرد که وظیفه آن اختصاص دادن منابعی چون راننده، باربر، مسئول انبار و یا فضای مشخص شده در انبار می باشد. در واقع همان طور که انتظار می رود، کلاس کنترلر به عنوان یک واسطه عمل کرده و بین کلاس های دیگر تقسیم وظیفه کرده و متد های آن ها را صدا می کند.

تخصیص متد به هر کلاس:

- کلاس «کاربر»
 - این کلاس دارای دو متد اصلی برای درخواست جا به جایی و درخواست انبار داری است.
- کلاس «درخواست انبارداری» و «درخواست جا به جایی»
 - متد های موجود در این کلاس در واقع همان تابع های موجود در قرارداد سیستمی است که در فاز قبل به شرح آن ها پرداختیم.
- کلاس «اطلاعات مکانی درخواست»
 - متد های موجود در این کلاس همگی `setter` هستند و `attribute` های مربوطه را `set` می کنند.
- کلاس «اطلاعات کالاهای درخواست»
 - متد های موجود در این کلاس همگی `setter` هستند و `attribute` های مربوطه را `set` می کنند.
 - * لازم به ذکر است که تعداد کارگران را نیز در این کلاس نگهداری و `set` می کنیم زیرا مربوط به حجم و نوع کالا ها می باشد.
- کلاس «اطلاعات انبار درخواست»
 - متد های موجود در این کلاس همگی `setter` هستند و `attribute` های مربوطه را `set` می کنند.
 - * لازم به ذکر است که نحوه ترخیص کالا را نیز در این کلاس نگهداری و `set` می کنیم.
- کلاس «وضعیت درخواست»
 - متد های موجود در این کلاس همگی `setter` هستند و `attribute` های مربوطه را `set` می کنند.
 - * لازم به ذکر است که زمان اعزام را نیز در این کلاس نگهداری و `set` می کنیم.
- کلاس «اطلاعات مالی درخواست»
 - این کلاس با دریافت اطلاعات مکانی و اطلاعات کالا ها مبلغ بیعانه و مبلغ نهایی را محاسبه کرده و ذخیره می کند. سپس درخواستی برای پرداخت به سامانه پرداخت آنلاین می فرستد. نتیجه ی پرداخت را نیز تحت عنوان یک `attribute` ذخیره کرده و وضعیت پرداخت توسط متد `check_payment` قابل استعلام است.
- کلاس «تخصیص منابع» **static class**
 - این کلاس تنها یک متد `assign_resources` دارد که با صدا کردن آن متد های مورد نیاز بر روی آبجکت های دیگر صدا شده و `association` ها برقرار می شود.
- کلاس «وضعیت کارمندان» **static class**

- این کلاس دارای متد `find_free_workers` است که با جست و جو بین کارمندان و با در دسترس داشتن برنامه های زمانی هر یک، کارگران آزاد در دسترس را برمی گرداند.
- کلاس «انبار» `static class`
 - این کلاس دارای دو متد `is_available` و `reserve` است که برای چک کردن وجود فضای خالی و رزرو کردن نهایی فضای انبار استفاده می شوند.
 - کلاس های مربوط به کارمندان
 - این دسته از کلاس ها دارای یک متد `assign_task` هستند که برای محول کردن کار ها استفاده می شود.