**به نام آنکه جان را فکرت آموخت**

**فاز دوم پروژه درس طراحی شی‌گرا**

**اعضای تیم ایجاد و توسعه‌ی گلابی:**

**علی عباسی 91110009**

**علی فهیم‌نیا 90109777**

**سپهر تائب 90105431**

بهار 1395

**فهرست**

[لیست به روز شده‌ی ریسک‌ها: 4](#_Toc451509623)

[ریسک‌های تکنیکی: 5](#_Toc451509624)

[لیست اولویتبندی شدهی نیازمندیها: 6](#_Toc451509625)

[Architecturally Significant Requirements: 7](#_Toc451509626)

[توضیحات پیاده‌سازی نیازمندی‌های عمده‌ی معماری: 7](#_Toc451509627)

[توضیحات کاربری: 7](#_Toc451509628)

[توضیحات فنی: 7](#_Toc451509629)

[برنامه 9](#_Toc451509630)

[فاز Elaboration، قسمت اول: 9](#_Toc451509631)

[فاز Elaboration، قسمت دوم: 9](#_Toc451509632)

[فاز Construction، قسمت اول: 9](#_Toc451509633)

[فاز Construction، قسمت دوم: 10](#_Toc451509634)

[فاز Transition: 10](#_Toc451509635)

[کارت‌های CRC: 11](#_Toc451509636)

[نمودارهای فعالیت: 20](#_Toc451509637)

[ورود به سیستم: 20](#_Toc451509638)

[مشاهده‌ی اطلاعات حساب کاربری: 21](#_Toc451509639)

[ویرایش اطلاعات حساب کاربری: 22](#_Toc451509640)

[خروج از سیستم: 23](#_Toc451509641)

[تعلیق حساب کاربری: 23](#_Toc451509642)

[حذف حساب کاربری: 24](#_Toc451509643)

[ایجاد کاربر جدید: 25](#_Toc451509644)

[ثبت منابع مورد نیاز یک واحد سازمان: 26](#_Toc451509645)

[ثبت یک منبع جدید: 27](#_Toc451509646)

[اختصاص منبع به ماژول: 28](#_Toc451509647)

[آزاد کردن یک منبع از ماژول: 29](#_Toc451509648)

[حذف منبع: 30](#_Toc451509649)

[درخواست گزارش منابع موجود: 30](#_Toc451509650)

[دریافت گزارش چرخشی منابع: 31](#_Toc451509651)

[دریافت گزارش منابع مورد نیاز: 32](#_Toc451509652)

[تخمین منابع مورد نیاز سیستم نرم‌افزاری: 33](#_Toc451509653)

[تخمین نیازمندی‌های ضروری براساس منابع مورد نیاز: 34](#_Toc451509654)

[جست‌وجو در میان سیستم‌های نرم‌افزاری: 35](#_Toc451509655)

[ثبت فرآیند ایجاد: 36](#_Toc451509656)

[ویرایش فرآیند ایجاد: 37](#_Toc451509657)

[حذف فرآیند ایجاد: 37](#_Toc451509658)

[ثبت فرآیند نگهداری: 38](#_Toc451509659)

[ویرایش فرآیند نگهداری: 39](#_Toc451509660)

[حذف فرآیند نگهداری: 39](#_Toc451509661)

[ثبت ماژول: 40](#_Toc451509662)

[ویرایش ماژول: 41](#_Toc451509663)

[حذف ماژول: 41](#_Toc451509664)

[چک لیست استخراج شده از اسلاید‌ها: 42](#_Toc451509665)

[موارد کاربرد(use-case): 42](#_Toc451509666)

[لیست ریسک‌ها: 42](#_Toc451509667)

[لیست نیازمندی‌ها: 42](#_Toc451509668)

[واژه‌نامه: 42](#_Toc451509669)

[کارت‌های CRC: 43](#_Toc451509670)

[نمودارهای فعالیت: 43](#_Toc451509671)

[Executable architecturally baseline: 43](#_Toc451509672)

# 

# لیست به روز شده‌ی ریسک‌ها:

|  |  |
| --- | --- |
| اولویت | ریسک |
| بحرانی | 1. عدم تعامل مناسب نماینده­ی مشتری با تیم ایجاد و نگهداری گلابی. |
| خیلی بالا | 1. محدودیت زمان با توجه به اینکه افراد تیم ایجاد و نگهداری گلابی همه عازم هستند، یکی از جدیترین ریسک­ها محسوب می­شود. |
| 1. تجربه­ی ناکافی اعضای تیم در فرآیندهای ایجاد یک نرم­افزار، ممکن است پیشبرد پروژه را با تهدید جدی مواجه کند. |
| 1. مشخص نبودن محدوده‌ی دقیق پروژه و احتمال تغییر این محدوده در زمان انجام پروژه. |
| 1. با توجه به تخمین بسیار نا دقیق از زمان­بندی­ها، برنامه­ی انجام ممکن است با مشکلات جدی مواجه شود. |
| بالا | 1. با توجه به ابهام در تعیین تکنولوژی­های قابل استفاده در پروژه از سوی نماینده­ی مشتری، پروژه با تهدید مواجه است. |
| 1. ابهام در نیازمندی مربوط به پیش­بینی |
| متوسط | 1. عدم آشنایی کافی اعضای تیم با برنامه­نویسی به صورت گرافیکی |
| کم | 1. عدم هماهنگی کلاس­های اعضای تیم در طول ترم |

* ریسک‌های با الویت بحرانی به شدت پروژه را در معرض خطر شکست قرار می‌دهند.
* ریسک­های با اولویت خیلی بالا، پیشبرد پروژه را با تهدید جدی مواجه می­کنند و نیاز است که اقدام به رفع آن­ها شود.
* ریسک­های با اولویت بالا، پیشبرد پروژه را مختل کرده و باید جدی گرفته شوند.
* ریسک­های با اولویت متوسط، به راحتی قابل رفع بوده مشکل چندانی ایجاد نخواهند کرد.
* ریسک­های با اولویت کم، در پیشبرد پروژه خللی وارد نکرده و قابل چشم­پوشی هستند

متاسفانه از زمان انجام فاز اول پروژه تا کنون هیچ یک از ریسک‌ها حذف نشده‌اند و صرفا عدم تعامل مشتری به وضعیت بحرانی درآمده است، زیرا در این مدت فیدبک فاز اول به دست ما نرسید به علاوه به دلیل مشغله‌ی اعضای گروه نتوانستیم که بعضی ریسک‌ها که ناشی از عدم مهارت کافی بوده را رفع کنیم که امیدواریم به مرور زمان بتوانیم آن‌ها را حل کنیم.

# ریسک‌های تکنیکی:

1. از این نظر که اعضا تجربه‌ی کمی در پیاده‌سازی سیستم‌های نرم‌افزاری با زبان جاوا دارند و آشنایی کافی با چارچوب‌های موجود برای این کار با زبان جاوا را ندارند این ریسک عمده‌ترین ریسک تکنیکی شناسایی شده است.

2. اعضا در پیاده‌سازی پایگاه داده‌ها نیز تجربه‌ی اجرایی کافی ندارند و مهارت آن‌ها در این حوزه صرفا به پروژه‌ی درس پایگاه‌داده‌ برمی‌گردد که مشخصا با بسیاری از چالش‌های موجود در این حوزه را در آن پروژه روبه‌رو نشده‌اند و از این نظر ممکن است در تولید این سیستم به این مشکلات برخورند.

برای رفع این ریسک‌ها لازم است تا اعضا در طول انجام پروژه به مطالعه در این دو مورد بپردازند تا به تدریج ریسک ناشی از این موارد را به حداقل برسانند

# لیست اولویت­بندی شده­ی نیازمندی­ها:

|  |  |  |
| --- | --- | --- |
| اولویت | نیازمندی | ریسک |
| خیلی بالا | 1. امکان افزودن فرآیند ایجاد | کم |
| 1. امکان حذف فرآیند ایجاد | کم |
| 1. امکان افزودن فرآیند نگهداری | کم |
| 1. امکان حذف فرآیند نگهداری | کم |
| 1. امکان افزودن منبع | کم |
| 1. امکان حذف منبع | کم |
| 1. امکان افزودن نیازمندی­های سازمان | کم |
| 1. امکان حذف نیازمندی­های سازمان | کم |
| بالا | 1. امکان ویرایش فرآیند ایجاد | کم |
| 1. امکان ویرایش فرآیند نگهداری | کم |
| 1. امکان ویرایش منبع | کم |
| 1. امکان ویرایش نیازمندی­های سازمان | کم |
| 1. امکان ثبت اندازه­ی سیستم نرم­افزاری | کم |
| 1. امکان دریافت گزارش منابع موجود | کم |
| 1. امکان دریافت گزارش جریان چرخشی مصرف منابع موجود | کم |
| 1. امکان دریافت گزارش منابع مورد نیاز | کم |
| 1. داشتن UI گرافیکی | کم |
| متوسط | 1. امکان جستجو در سیستم­های مشابه برای تخمین منابع | کم |
| 1. امکان جستجو در سیستم­های مشابه برای یافتن نیازمندی­های سازمان | کم |
| کم | نداریم |  |

* نیازمندی­های پروژه که با اولویت خیلی بالا شناخته می­شوند، در صورتی که پیاده­سازی نشوند، باعث شکست پروژه خواهند شد.
* نیازمندی­های پروژه که با الویت بالا شناخته می­شوند، باید پیاده­سازی شوند.
* نیازمندی­های پروژه که با اولویت متوسط شناخته می­شوند، خوب است که پیاده­سازی شوند.
* نیازمندی­های پروژه که با اولویت کم شناخته می­شوند، در صورت وجود زمان و منابع پیاده­سازی خواهند شد.

# Architecturally Significant Requirements:

این نیازمندی‌ها شامل نیازمندهایی هستند که روی معماری سیستم تاثیر گذارند که شامل شماهای پایگاه‌داده و عملیات درج و حذف در این شماها می‌باشد که در این فاز پیاده‌سازی شده‌اند:

|  |
| --- |
| نیازمندی‌های عمده معماری |
| 1. امکان افزودن فرآیند ایجاد |
| 1. امکان حذف فرآیند ایجاد |
| 1. امکان افزودن فرآیند نگهداری |
| 1. امکان حذف فرآیند نگهداری |
| 1. امکان افزودن منبع |
| 1. امکان حذف منبع |
| 1. امکان افزودن نیازمندی­های سازمان |
| 1. امکان حذف نیازمندی­های سازمان |

# توضیحات پیاده‌سازی نیازمندی‌های عمده‌ی معماری:

در این فاز از پروژه ما به پیاده­سازی قسمتی از پروژه پرداختیم که از لحاظ معماری مهم محسوب می­شد و نیاز بود تا حدی پیاده­سازی انجام شود که از برطرف شدن ریسک­هایی که ممکن بود اتمام پروژه با موفقیت را با مشکل مواجه کند، کاملا مرتفع شوند. بنابراین به پیاده­سازی قسمتی پرداختیم که کل زمینه­ی اصلی معماری پروژه را شامل می­شد.

## توضیحات کاربری:

در ابتدای اجرای پروژه، از شما صفحه­ی اصلی بر روی کنسول نمایان می­شود که با وارد کردن عدد مورد نظرتان می­توانید گزینه­ی مورد نظرتان را انتخاب کنید. با انتخاب هر گزینه وارد یک منوی جدید می­شوید که در نهایت به گزینه­ی مورد نظر شما ختم می­شود. سپس طبق الگوها و راهنمایی­های برنامه موارد مورد نظر خود را حذف یا درج می­کنید. در مجموع با توجه به اهمیت بالای عملیات حذف و درج نیاز بود که ابتدا این 2 خاصیت پیاده شود و در این فاز کاربر فقط از این گزینه­ها برخوردار خواهد بود.

## توضیحات فنی:

* این برنامه با استفاده از زبان جاوا زده شده است.
* پایگاه داده­ی این برنامه با استفاده از MySQL پیاده­سازی شده است.
* در این فاز برای برنامه از 3 پکیج استفاده شده است:
  + UI Package: که در آن یک Interface با نام View تعریف شده است که تمام کلاس­های موجود در این پکیج این interface را implement می­کنند.
  + Items Package: در این پکیج، تمامی مواردی که نیاز است در پایگاه داده ثبت شوند قرار دارد. در فوق کلاس این کلاس­ها، 2 متد به صورت abstract تعریف شده است که با استفاده از این 2 متد می­توان آن­ها را در پایگاه داده درج کرد و یا آن­ها را از پایگاه داده حذف کرد.
  + Data Package: تمامی امور مربوط به پایگاه داده از قبیل آماده کردن داده­ها، ایجاد ارتباط با پایگاه داده و موارد تغییر یا به روز رسانی در این پکیج قرار گرفته­اند.

کلاس­های مختلف با توجه به توضیحات بیان­شده در بالا پیاده­سازی شدند. البته باید توجه داشت که این کلاس­ها هنوز در مرجله­ی ابتدایی پیاده­سازی قرار دارند و کامل نیستند. همچنین ممکن است مشکلاتی از نظر OOD و نیز Bad smell ها وجود داشته باشد که تا حد نهایت و دانش تا کنون، سعی در رفع این مشکلات داشته­ایم.

# برنامه­

با توجه به اینکه در طول یک‌ هفته اتفاق خاصی رخ نداده از این رو تغییری در برنامه‌ای که در فاز قبل ارائه داده بودیم اتفاق نیفتاده است.

## فاز Elaboration، قسمت اول:

در این فاز، برای بازنگری و تصحیح لیست نیازمندی­های پروژه و همچنین موارد کاربرد، مجموعا 10 نفر ساعت زمان مورد نیاز تخمین زده می­شود که بر عهده­ی علی عباسی و سپهر تائب است.

برای نمونه اولیه­ی واسط کاربری قابل اجرا به همراه تصاویر آن­ها، 30 نفر ساعت زمان مورد نیاز تخمین زده می­شود که بر عهده­ی تمام اعضای تیم است.

برای کارت­های CRC و نمودارهای فعالیت، 5 نفر ساعت زمان مورد نیاز تخمین زده می­شود که بر عهده­ی علی فهیم­نیا می­باشد.

برای پیاده سازی executable architectural baseline، 60 نفر ساعت تخمین زده می­شود که بر عهده­ی تمام اعضای گروه می­باشد.

برای چک لیست استخراج شده از اسلایدهای درس، 3 نفر ساعت در نظر گرفته شده که بر عهده­ی تمام اعضای گروه می­باشد.

## فاز Elaboration، قسمت دوم:

برای نسخه تکمیل شده ی Use Case Realizations، لیست بازنگری‌شده اولویت‌ نیازمندی‌هاو ریسک‌ها، برنامه زمان‌بندی به‌روزشده، 6 نفر ساعت در نظر گرفته شده است که بر عهده­ی تمام اعضای گروه است.

برای نمودارهای فعالیت با خطوط شنا، نمودار کلاس های تحلیل، ، نمودارهای توالی تحلیل، نمودار بسته، به 15 نفر ساعت تخمین زده شده است که بر عهده­ی تمام اعضای گروه می­باشد.

برای نمودار کلاس‌ طراحی، نمودار مؤلفه، نمودارهای توالی طراحی، 15 نفر ساعت تخمین زده شده که بر عهده­ی تمام اعضای تیم می­باشد.

برای تکمیل executive architectural baseline در تکرار اول، 45 نفر ساعت تخمین زده می­شود که بر عهده­ی تمام اعضای گروه است.

برای چک لیست استخراج‌شده از اسلایدهای درس، 3 نفر ساعت تخمین زده شده که بر عهده­ی تمام اعضای گروه است.

## فاز Construction، قسمت اول:

برای نسخه تکمیل شده ی Use Case Realizations، لیست بازنگری‌شده اولویت‌ها و ریسک‌ها، 6 نفر ساعت تخمین زده شده که بر عهده­ی تمام اعضای تیم است.

برای نمودار تصحیح/تکمیل شده کلاس‌ تحلیل، نمودارهای توالی تحلیل بازنگری‏شده، به 9 نفر ساعت زمان نیاز خواهد بود که بر عهده­ی تمام اعضا است.

برای کلاس‌های طراحیِ بازنگری‌شده ، نمودارهای توالی طراحی بازنگری‏شده، مستند توضیح الگوهای اعمال شده به 15 نفر ساعت زمان نیاز خواهد بود که بر عهده­ی تمام اعضا می­باشد.

برای نمونه اولیه تکامل‌یافته سیستم، به 60 نفر ساعت زمان نیاز خواهد بود که بر عهده­ی تمام اعضا می­باشد.

برای چک لیست استخراج‌شده از اسلایدهای درس، به 3 نفر ساعت زمان نیاز خواهد بود که بر عهده­ی تمام اعضا می­باشد.

## فاز Construction، قسمت دوم:

برای نسخه تکمیل شده ی Use Case Realizations، لیست بازنگری‌شده اولویت‌ها و ریسک‌ها، به 6 نفر ساعت نیاز خواهد بود که بر عهده­ی تمام اعضا می­باشد.

برای نمودار استقرار، به 9 نفر ساعت زمان نیاز خواهد بود که بر عهده­ی تمام اعضا می­باشد.

برای نسخه نهایی سیستم، مستند استفاده، مستند نصب، به 60 نفر ساعت زمان نیاز خواهد بود که بر عهده­ی تمام اعضا می­باشد.

برای چک لیست استخراج‌شده از اسلایدهای درس، درستی‌سنجی، به 6 نفر ساعت نیاز خواهد بود که بر عهده­ی تمام اعضا می­باشد.

## فاز Transition:

برای تکمیل نسخه نهایی بر مبنای نتایج آزمون‌های فاز قبل، به 30 نفر ساعت نیاز خواهد بود، که بر عهده­ی همه­ی اعضا می­باشد.

با توجه به اینکه اعضای گروه تخمین درستی از میزان زمان مورد نظر برای هر فاز نداشتند، برنامه­ی فوق در هر مرحله به روز رسانی شده و تصحیح خواهد شد.

# کارت‌های CRC:

|  |  |
| --- | --- |
| نام کلاس: User | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری اطلاعات مربوط به همه‌ی کاربران سامانه، از قبیل نام و نام خانوادگی، نام کاربری، کلمه‌ی عبور، آدرس پست الکترونیکی، بخش مورد نظر در سازمان * ورود کاربران به سیستم | Manager  Employee |

|  |  |
| --- | --- |
| نام کلاس: Manager | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری سطوح دست‌رسی مخصوص مدیر سیستم * انجام دست‌رسی‌های مخصوص مدیر سیستم | User  Search  Report |

|  |  |
| --- | --- |
| نام کلاس: Employee | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری سطوح دست‌رسی مخصوص کارمندان * نگه‌داری اطلاعات مدیریت بخش‌های سازمان * انجام دست‌رسی‌های مخصوص کارمندان | User |

|  |  |
| --- | --- |
| نام کلاس: Resource | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری، بروزرسانی و استفاده از اطلاعات مربوط به یک منبع * ایجاد و حذف یک منبع * اختصاص منبع به ماژول و آزاد کردن منبع از یک ماژول | Requirement  HumanResourse  FundingResource  FacilityResource |

|  |  |
| --- | --- |
| نام کلاس: HumanResource | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری، بروزرسانی و استفاده از اطلاعات مربوط به یک منبع انسانی | Resource |

|  |  |
| --- | --- |
| نام کلاس: FundingResource | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری، بروزرسانی و استفاده از اطلاعات مربوط به یک منبع مالی | Resource |

|  |  |
| --- | --- |
| نام کلاس: FacilityResource | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری، بروزرسانی و استفاده از اطلاعات مربوط به یک منبع ابزاری | Resource |

|  |  |
| --- | --- |
| نام کلاس: SoftwareSystem | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری اطلاعات مربوط به یک سیستم نرم‌افزاری * بروزرسانی اطلاعات یک سیستم نرم‌افزاری * ایجاد و حذف یک سیستم نرم‌افزاری * مدیریت ماژول‌های تشکیل‌دهنده‌ی یک سیستم نرم‌افزاری | Module |

|  |  |
| --- | --- |
| نام کلاس: Module | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری و بروزرسانی اطلاعات مربوط به یک ماژول نرم‌فزاری * ایجاد، بروزرسانی و حذف یک ماژول نرم‌افزاری * مدیریت منابع اختصاص یافته به یک ماژول | SoftwareSystem  Resource |

|  |  |
| --- | --- |
| نام کلاس: Process | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری اطلاعات مربوط یه یک فرآیند * ایجاد، بروزرسانی و حذف یک فرآیند | SoftwareSystem  DevelopmentProcess  MaintenanceProcess |

|  |  |
| --- | --- |
| نام کلاس: DevelopmentProcess | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری اطلاعات مربوط به یک فرآیند ایجاد | Process |

|  |  |
| --- | --- |
| نام کلاس: MaintenanceProcess | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری اطلاعات مربوط به یک فرآیند نگه‌داری | Process |

|  |  |
| --- | --- |
| نام کلاس: Requirment | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری اطلاعات مربوط به یک نیاز سازمانی * دانستن واحد سازمانی که این نیاز را دارد * بروزرسانی اطلاعات یک نیاز سازمانی | OrganizationUnit  Process  Module |

|  |  |
| --- | --- |
| نام کلاس: OrganiztionUnit | |
| مسئولیت‌ها | **هم‌کاران** |
| * نگه‌داری اطلاعات یک واحد سازمانی * دانستن کارمندان واحد سازمانی * دانستن مدیر واحد سازمانی | Process  Module  User |

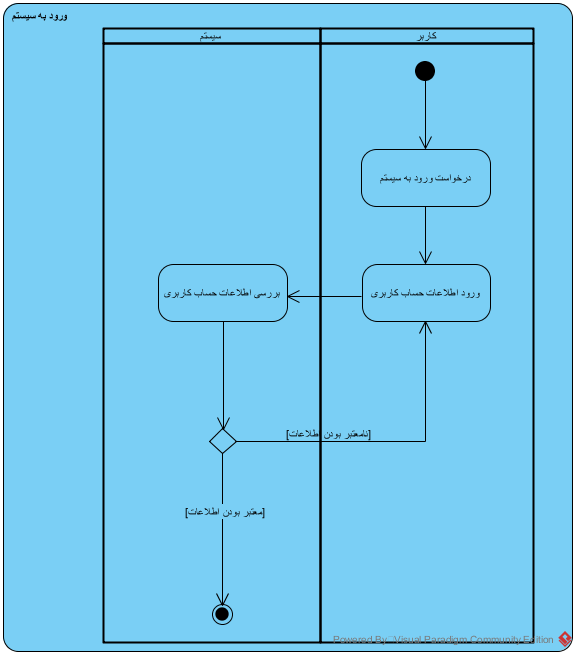
|  |  |
| --- | --- |
| نام کلاس: Report | |
| مسئولیت‌ها | **هم‌کاران** |
| * ایجاد یک گزارش برای یک کاربر سیستم | User  Manager |

|  |  |
| --- | --- |
| نام کلاس: Search | |
| مسئولیت‌ها | **هم‌کاران** |
| * جست‌وجو در بین سیستم‌های نرم‌افزاری برای پیدا کردن سیستم‌هایی با مشخصات تعیین شده توسط کاربر سیستم | Estimator  Manager |

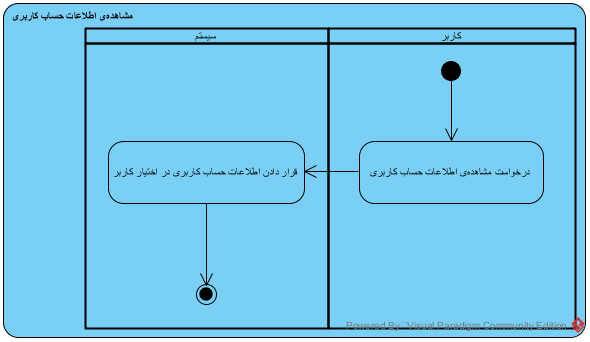
|  |  |
| --- | --- |
| نام کلاس: Estimator | |
| مسئولیت‌ها | **هم‌کاران** |
| * دریافت نتایج یک جست‌وجو و تخمین زمان، بودجه، منابع و ... برای یک سیستم نرم‌افزاری با توجه به خواسته‌ی یک کاربر سیستم | Search  Manager |

# نمودارهای فعالیت:

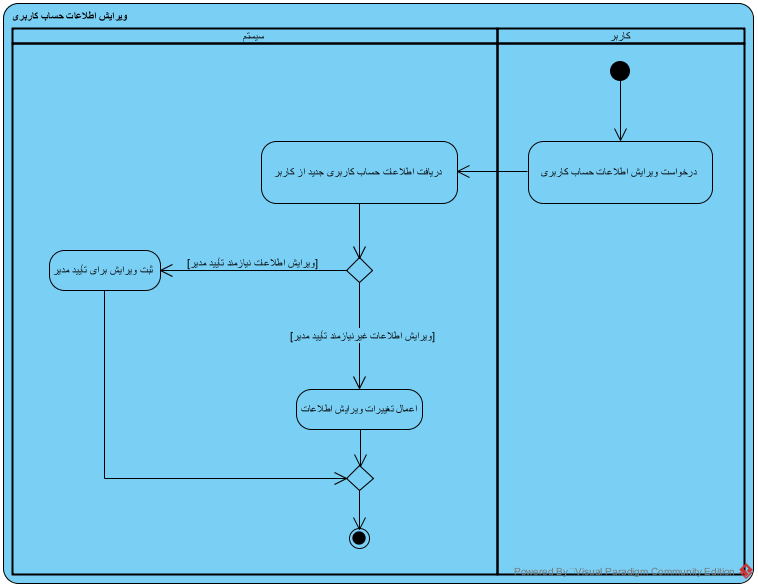
## ورود به سیستم:



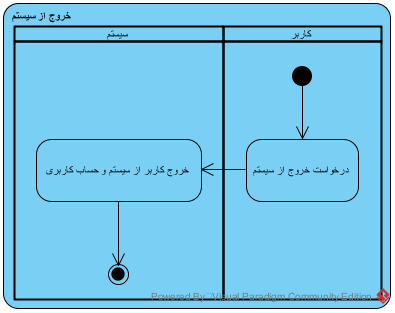
## مشاهده‌ی اطلاعات حساب کاربری:



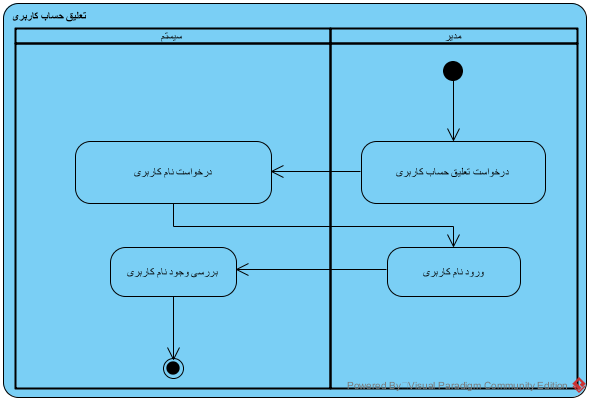
## ویرایش اطلاعات حساب کاربری:



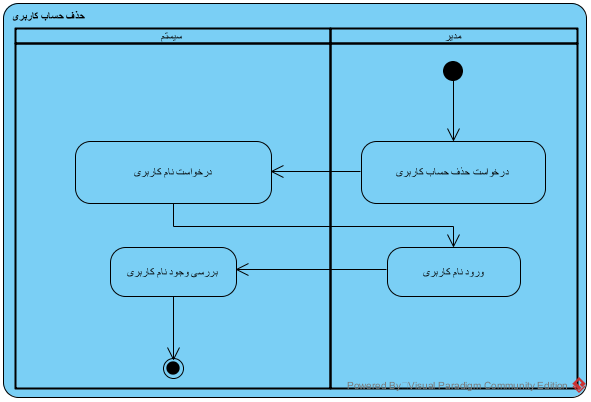
## خروج از سیستم:



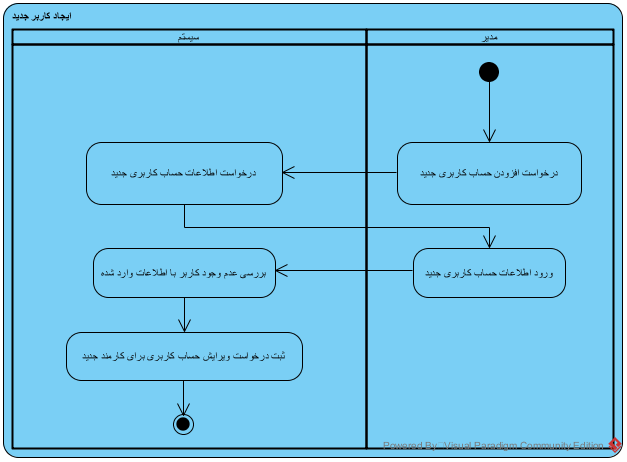
## تعلیق حساب کاربری:



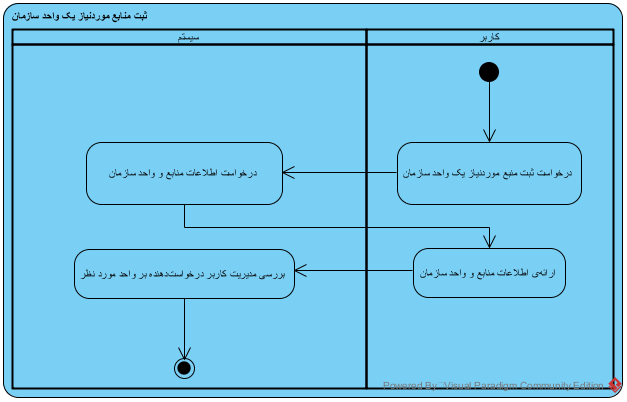
## حذف حساب کاربری:



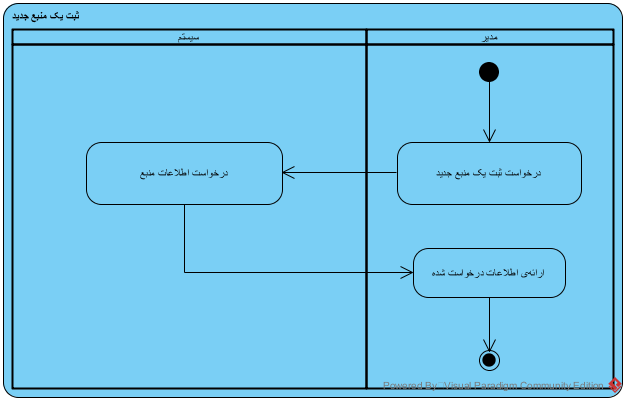
## ایجاد کاربر جدید:



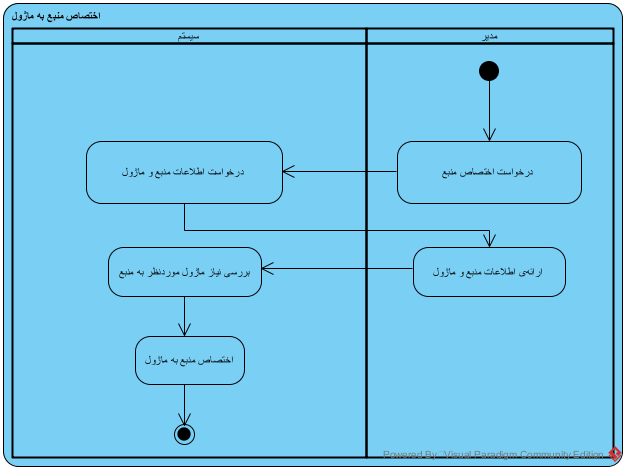
## ثبت منابع مورد نیاز یک واحد سازمان:



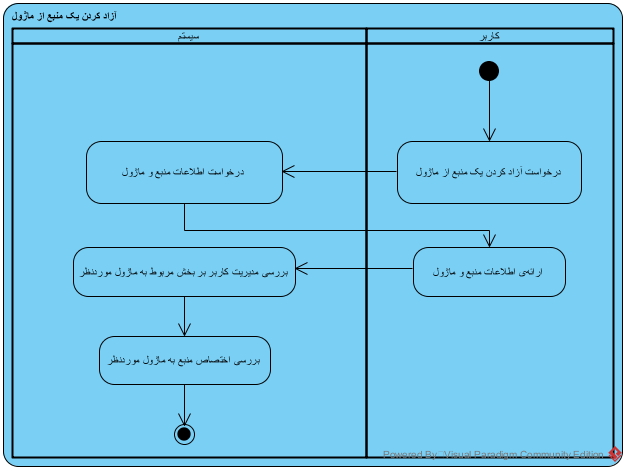
## ثبت یک منبع جدید:



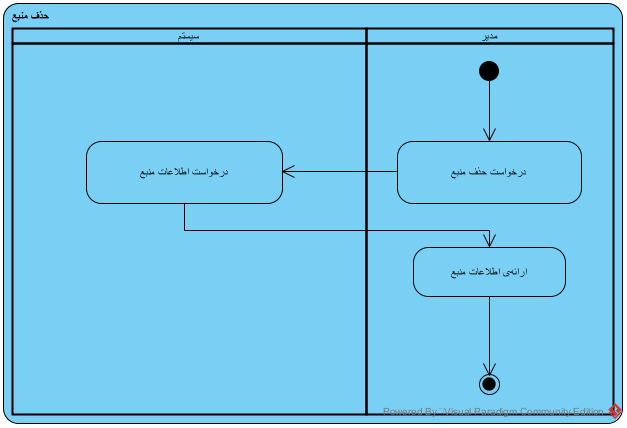
## اختصاص منبع به ماژول:



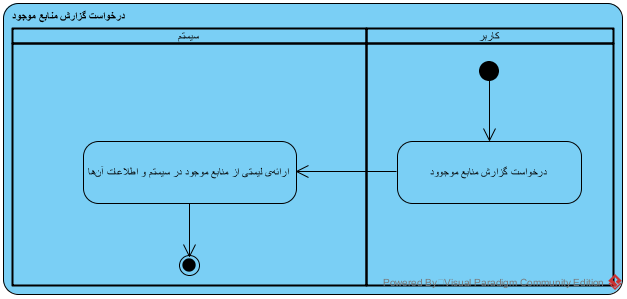
## آزاد کردن یک منبع از ماژول:



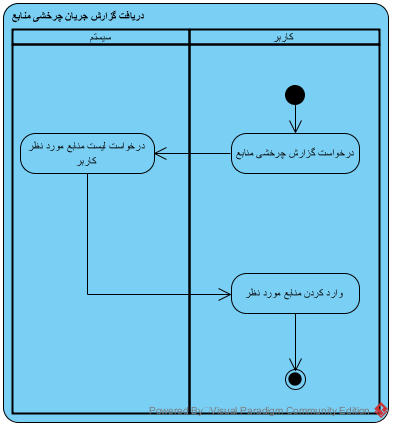
## حذف منبع:



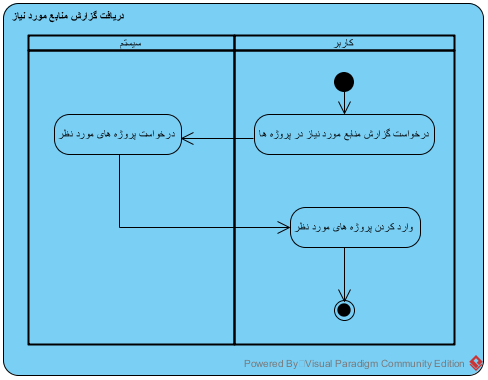
## درخواست گزارش منابع موجود:



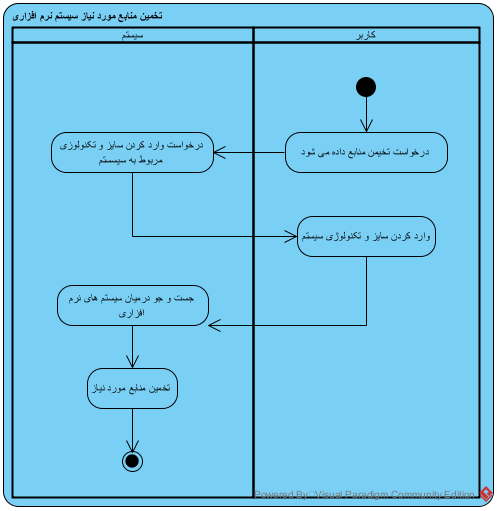
## دریافت گزارش چرخشی منابع:



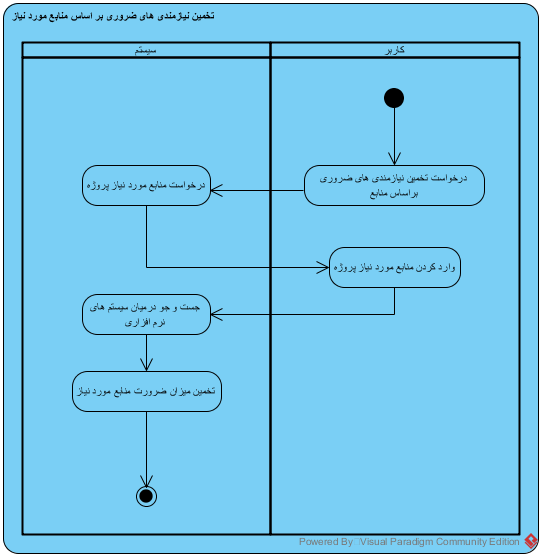
## دریافت گزارش منابع مورد نیاز:



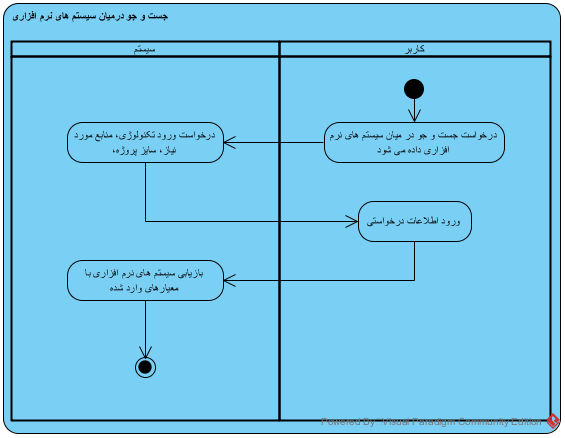
## تخمین منابع مورد نیاز سیستم نرم‌افزاری:



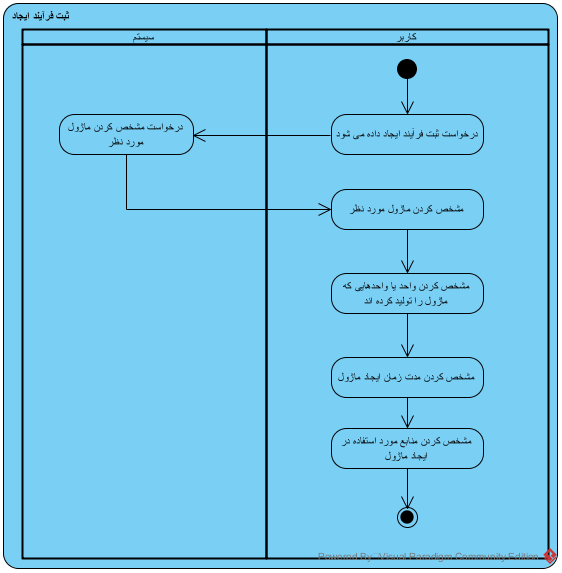
## تخمین نیازمندی‌های ضروری براساس منابع مورد نیاز:



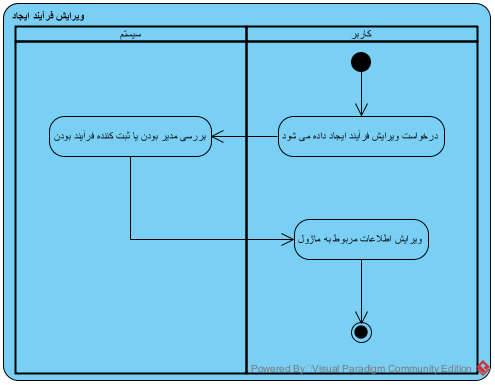
## جست‌وجو در میان سیستم‌های نرم‌افزاری:



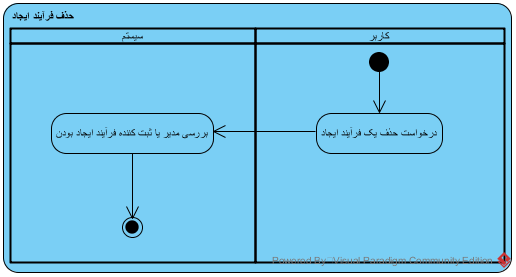
## ثبت فرآیند ایجاد:



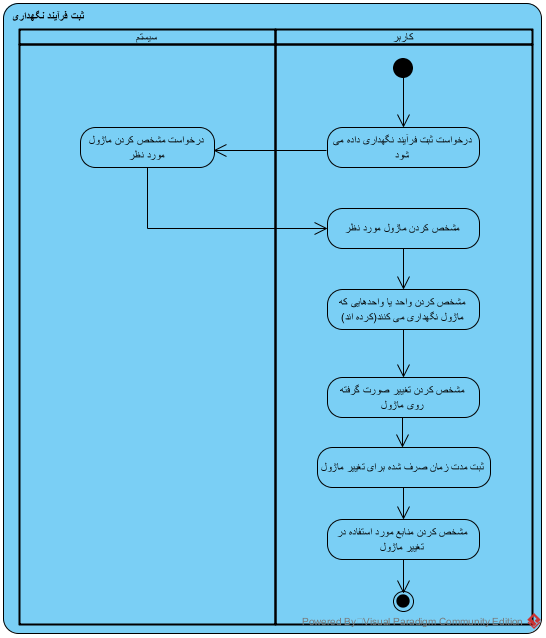
## ویرایش فرآیند ایجاد:



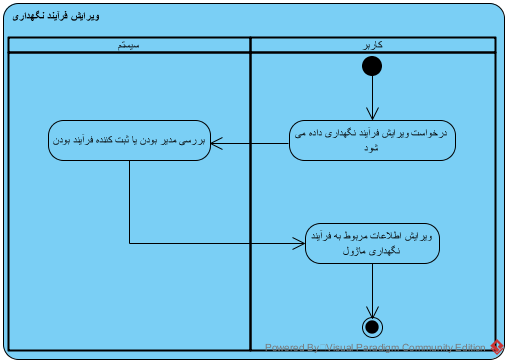
## حذف فرآیند ایجاد:



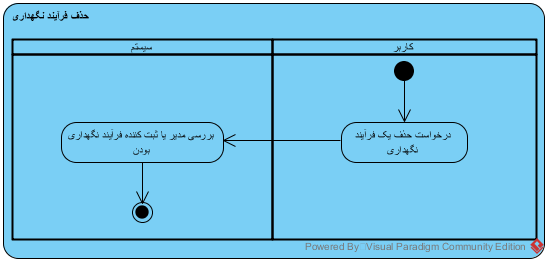
## ثبت فرآیند نگهداری:



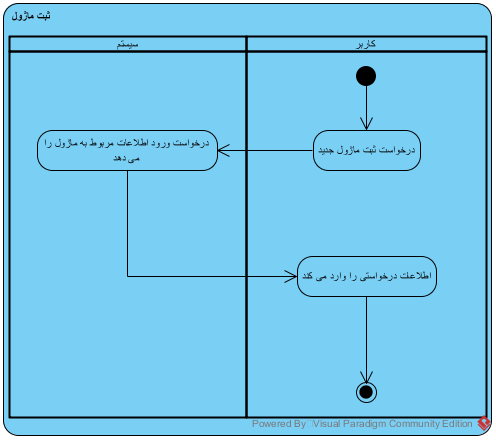
## ویرایش فرآیند نگهداری:



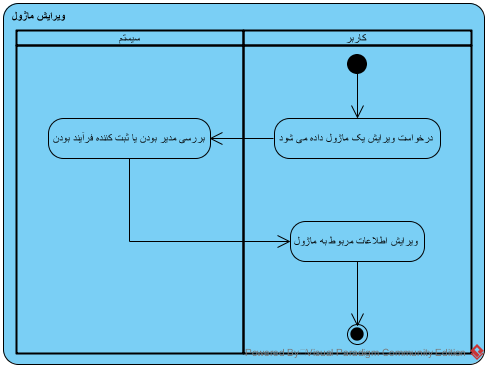
## حذف فرآیند نگهداری:



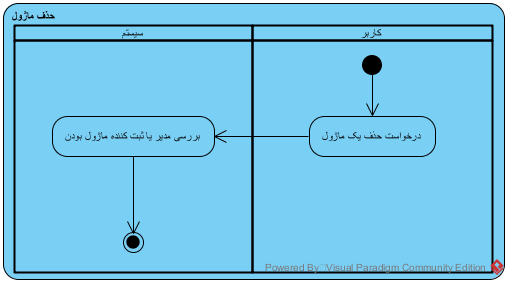
## ثبت ماژول:



## ویرایش ماژول:



## حذف ماژول:



# چک لیست استخراج شده از اسلاید‌ها:

در این قسمت سعی شده که برای هر کدام از تحویل دادنی‌ها در این دو فاز لیستی از ویژگی‌هایی که باید داشته باشند را از اسلایدهای درس استخراج کنیم.

## موارد کاربرد(use-case):

* مستقل از تکنولوژی بودن.
* Atomic بودن هر مورد کاربرد.
* از دید کاربر بودن.
* مشخص بودن اینکه هر کدام از موارد کاربرد مورد استفاده‌ی کدام کنش‌گر است.
* هم‌پوشانی کم بین کنش‌گرهای مختاف.
* نباید هر مورد کاربرد زیادی کوچک یا زیادی بزرگ باشد.
* باید نتیجه‌ای محسوس داشته باشند.
* باید دارای توصیف مرحله به مرحله باشند.
* باید الویت بندی بشوند (که در لیست نیازمندی‌ها این اتفاق افتاده است).
* ساختاردهی باید شوند. (که از include در چند مورد به دلیل نیاز استفاده شده است و gen/spec در تعریف کنش‌گرها).
* مشخص کردن تمام جزئیات شامل پیش‌شرایط، پس‌شرایط، کنش‌گر اصلی و ثانویه، مسیراصلی و مسیرهای فرعی و انجام شدن هر مرحله توسط چه کسی (سیستم یا کاربر).
* مشخص کردن روابط بین موارد کاربرد و کنش‌گرها.
* مشخص کردن مرز سیستم.
* تمرکز بر روی چیستی نه چگونگی.
* عدم تجزیه عملکردی هر مورد کاربرد.

## لیست ریسک‌ها:

* مشخص کردن ریسک‌ها با میزان ریسک ناشی از هر کدام.
* مشخص کردن ریسک‌های بحرانی برای رفع سریع آن‌ها در جهت شکست نخوردن پروژه.
* مشخص کردن ریسک هر کدام از نیازمندی‌ها.

## لیست نیازمندی‌ها:

* لیست نیازمندی‌ها باید اولویت بندی شده باشد.
* لیست نیازمندی‌ها باید وضعیت‌شان به‌روز باشد. (که در این مرحله مشخص شده که الویت‌های بالا پیاده‌سازی شده‌اند)
* ریسک و هزینه نیز باید مشخص شود.(هزینه در این پروژه معنی ندارد)

## واژه‌نامه:

* مشخص کردن کلمات توصیف کننده‌ی سیستم در این واژه نامه به همراه توضیح برای آن‌ها
* شامل شدن لغات با معنی مشابه و انتخاب واژه‌ای که در پروژه استفاده می‌شود.
* شامل شدن لغات چندمعنی و مشخص کردن اینکه در تعریف پروژه هر کلمه دقیقا کدام معنی را می‌دهد.

## کارت‌های CRC:

* اسم کلاس کارش را نشان دهد.
* در حوزه‌ی مساله بودن کلاس‌ها (چون در مرحله‌ی طراحی نیستیم نباید کلاس‌‎ها از حوزه‌ی راه‌حل بیایند).
* باید مجموعه‌ای منسجم از مسئولیت‌ها داشته باشند.
* وابستگی بین کلاس‌ها حداقل باشد.
* شامل یک عملکرد نباشند.
* کلاسی که هیچ همکاری نداشته باشد احتمالا غلط است.
* کلاس همه‌کاره نباید داشته باشیم.
* دارای سلسله‌ی وراثت عمیق نباید باشند.

## نمودارهای فعالیت:

* یک رفتار سیستم را مدل کند.
* هر کدام فقط یک رفتار سیستم را مدل کند.
* هر action node شامل کاری تجزیه نشدنی باشد (به این معنی که نمی‌خواهیم جزئیات آن را مشخص کنیم).
* می‌تواند شامل پیش‌شرایط و پس‌شرایط باشد (که همان پیش‌شرایط و پس‌شرایط هر مورد کاربرد است).
* مسیر اصلی هر مورد کاربرد را نشان دهد (جابجایی پیغام بین بخش‌های مختلف سیستم و کاربر را نشان دهد).

## Executable architecturally baseline:

* پیاده‌سازی نیازمندی‌هایی که برروی معماری سیستم تاثیر می‌گذارند.
* چارچوبی که معماری را مشخص می‌کند.
* در واقع با اجرای این مرحله ریسک‌های مربوط به معماری سیستم تا حد خوبی رفع می‌شوند (و ریسک‌های نیازمندی‌های با اولویت بالا).