

به نام آنکه جان را فکرت آموخت
فاز پنجم پروژه درس طراحی شی گرا

اعضای تیم ایجاد و توسعه‌ی گلابی:

علی عباسی ۹۱۱۱۰۰۰۹

علی فهیم‌نیا ۹۰۱۰۹۷۷۷

سپهر تائب ۹۰۱۰۵۴۳۱

بهار ۱۳۹۵

فهرست

لیست به روز شده ی ریسک‌ها:	۵
ریسک‌های تکنیکی:	۶
لیست اولویت‌بندی شده ی نیازمندیها:	۷
برنامه	۸
فاز Construction، قسمت دوم:	۸
فاز Transition:	۸
نمودار کلاس تحلیل:	۹
نمودارهای توالی تحلیل:	۱۰
ورود به سیستم:	۱۰
مشاهده اطلاعات حساب کاربری:	۱۰
درخواست ویرایش اطلاعات حساب کاربری:	۱۱
خروج از سیستم:	۱۱
تعليق حساب کاربری:	۱۲
حذف حساب کاربری:	۱۲
ایجاد کاربر جدید:	۱۳
ثبت منابع مورد نیاز یک واحد سازمان:	۱۳
ثبت یک منبع جدید:	۱۴
اختصاص منبع به ماژول:	۱۴
حذف منبع:	۱۵
دریافت گزارش منابع موجود:	۱۵
دریافت گزارش جریان چرخشی منابع:	۱۵
دریافت گزارش منابع مورد نیاز:	۱۶
تخمین منابع مورد نیاز سیستم نرم‌افزاری:	۱۶
تخمین نیازمندی‌های ضروری براساس منابع مورد نیاز:	۱۶
ثبت فرآیند ایجاد:	۱۷
ویرایش فرآیند ایجاد:	۱۷
حذف فرآیند ایجاد:	۱۸

۱۸.....	ثبت فرآیند نگهداری:.....
۱۹.....	ویرایش فرآیند نگهداری:.....
۱۹.....	حذف فرآیند نگهداری:.....
۲۰.....	ثبت ماژول:.....
۲۰.....	ویرایش ماژول:.....
۲۱.....	حذف ماژول:.....
۲۲.....	نمودار کلاس طراحی (ویرایش شده):.....
۲۳.....	نمودارهای توالی طراحی (ویرایش شده):.....
۲۳.....	ورود به سیستم:.....
۲۳.....	مشاهده اطلاعات حساب کاربری:.....
۲۴.....	درخواست ویرایش اطلاعات حساب کاربری (ویرایش شده):.....
۲۴.....	خروج از سیستم (ویرایش شده):.....
۲۵.....	تعلیق حساب کاربری:.....
۲۵.....	حذف حساب کاربری:.....
۲۶.....	ایجاد کاربر جدید (ویرایش شده):.....
۲۶.....	ثبت منابع مورد نیاز یک واحد سازمان:.....
۲۶.....	ثبت یک منبع جدید:.....
۲۷.....	اختصاص منبع به ماژول (ویرایش شده):.....
۲۷.....	حذف منبع (ویرایش شده):.....
۲۷.....	دریافت گزارش منابع موجود:.....
۲۸.....	دریافت گزارش جریان چرخشی منابع:.....
۲۸.....	دریافت گزارش منابع مورد نیاز:.....
۲۹.....	تخمین منابع مورد نیاز سیستم نرم افزاری:.....
۲۹.....	تخمین نیازمندی های ضروری براساس منابع مورد نیاز:.....
۳۰.....	ثبت فرآیند ایجاد:.....
۳۰.....	ویرایش فرآیند ایجاد (ویرایش شده):.....
۳۱.....	حذف فرآیند ایجاد (ویرایش شده):.....
۳۱.....	ثبت فرآیند نگهداری:.....

۳۲.....	ویرایش فرآیند نگهداری:
۳۲.....	حذف فرآیند نگهداری:
۳۳.....	نمودارهای بسته
۳۶.....	نمودار مؤلفه
۳۶.....	شمای پایگاه داده
۳۷.....	مستند الگوهای استفاده شده
۳۸.....	چک لیست استخراج شده

لیست به روز شده‌ی ریسک‌ها:

اولویت	ریسک
بحرانی	۱. عدم تعامل مناسب نماینده‌ی مشتری با تیم ایجاد و نگهداری گلابی.
خیلی بالا	۲. محدودیت زمان با توجه به اینکه افراد تیم ایجاد و نگهداری گلابی همه عازم هستند، یکی از جدی‌ترین ریسک‌ها محسوب می‌شود.
	۳. تجربه‌ی ناکافی اعضای تیم در فرآیندهای ایجاد یک نرم‌افزار، ممکن است پیش‌برد پروژه را با تهدید جدی مواجه کند.
	۴. مشخص نبودن محدوده‌ی دقیق پروژه و احتمال تغییر این محدوده در زمان انجام پروژه.
	۵. با توجه به تخمین بسیار نادقیق از زمان‌بندی‌ها، برنامه‌ی انجام ممکن است با مشکلات جدی مواجه شود.
بالا	۶. با توجه به ابهام در تعیین تکنولوژی‌های قابل استفاده در پروژه از سوی نماینده‌ی مشتری، پروژه با تهدید مواجه است.
	۷. ابهام در نیازمندی مربوط به پیش‌بینی
متوسط	
کم	۸. عدم آشنایی کافی اعضای تیم با برنامه‌نویسی به صورت گرافیکی
	۹. عدم هماهنگی کلاس‌های اعضای تیم در طول ترم

- ریسک‌های با اولویت 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

متأسفانه از زمان انجام فاز اول پروژه تا کنون هیچ یک از ریسک‌ها حذف نشده‌اند و صرفاً 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.

به روز رسانی:

- مشخص شدن برنامه‌ی دقیق تحویل‌های فازهای پروژه باعث شد که برنامه‌ریزی دقیق‌تری به دست آید.
- اعضا با تعامل با یکدیگر به حل مشکل کمبود تجربه در زمینه‌ی برنامه‌نویسی به زبان جاوا پرداختند.
- به ریسک‌های بالا، ریسک خیلی بالای عدم همخوانی محصولات ارائه شده اضافه شده است.
- مشکل کمبود زمان تشدید شده و به ریسک بحرانی تبدیل شده است.
- به منظور رفع ابهامات مهم و هماهنگی بیشتر، جلسه‌ای با مشتریان در حال برنامه‌ریزی است.

ریسک‌های تکنیکی:

۱. 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.

به روز رسانی:

آشنایی لازم به میزان مورد نیاز برای انجام فازهای اولیه به صورت کامل انجام گرفته است. برنامه‌ریزی‌ها برای آشنایی با مبانی پیشرفته‌تر نیز ادامه دارد.

۲. اعضا در پیاده‌سازی پایگاه داده‌ها نیز تجربه‌ی اجرایی کافی ندارند و مهارت آن‌ها در این حوزه صرفاً به پروژه‌ی درس پایگاه‌داده برمی‌گردد که مشخصاً 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 برخورد کنند.

به‌روزرسانی:

تجربه‌ی اعضا در زمینه‌ی کار با پایگاه داده‌ها بسیار افزایش یافته است و این ریسک اولویت خود را از دست داده است.

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	۱. امکان افزودن فرآیند ایجاد	رفع شده
	۲. امکان حذف فرآیند ایجاد	رفع شده
	۳. امکان افزودن فرآیند نگهداری	رفع شده
	۴. امکان حذف فرآیند نگهداری	رفع شده
	۵. امکان افزودن منبع	رفع شده
	۶. امکان حذف منبع	رفع شده
	۷. امکان افزودن نیازمندی‌های سازمان	رفع شده
	۸. امکان حذف نیازمندی‌های سازمان	رفع شده
بالا	۹. امکان ویرایش فرآیند ایجاد	رفع شده
	۱۰. امکان ویرایش فرآیند نگهداری	رفع شده
	۱۱. امکان ویرایش منبع	رفع شده
	۱۲. امکان ویرایش نیازمندی‌های سازمان	رفع شده
	۱۳. امکان ثبت اندازه‌ی سیستم نرم‌افزاری	رفع شده
	۱۴. امکان دریافت گزارش منابع موجود	رفع شده
	۱۵. امکان دریافت گزارش جریان چرخشی مصرف منابع موجود	رفع شده
	۱۶. امکان دریافت گزارش منابع مورد نیاز	رفع شده
متوسط	۱۷. داشتن UI گرافیکی	کم
	۱۸. امکان جستجو در سیستم‌های مشابه برای تخمین منابع	متوسط
	۱۹. امکان جستجو در سیستم‌های مشابه برای یافتن نیازمندی‌های سازمان	متوسط
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 اولویت 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.

به‌روزرسانی: با پیاده‌سازی بخش عمده‌ی نیازمندی‌ها و ایجاد واسط کاربری، ریسک نیازمندی‌های مربوطه کاهش یافته یا مرتفع شدند.

برنامه

فاز Construction، قسمت دوم:

برای نسخه تکمیل شده ی Use Case Realizations، لیست بازنگری شده اولویت‌ها و ریسک‌ها، به ۶ نفر ساعت نیاز خواهد بود که بر عهده‌ی تمام اعضا می‌باشد.

برای نمودار استقرار، به ۹ نفر ساعت زمان نیاز خواهد بود که بر عهده‌ی تمام اعضا می‌باشد.

برای نسخه نهایی سیستم، مستند استفاده، مستند نصب، به ۶۰ نفر ساعت زمان نیاز خواهد بود که بر عهده‌ی تمام اعضا می‌باشد.

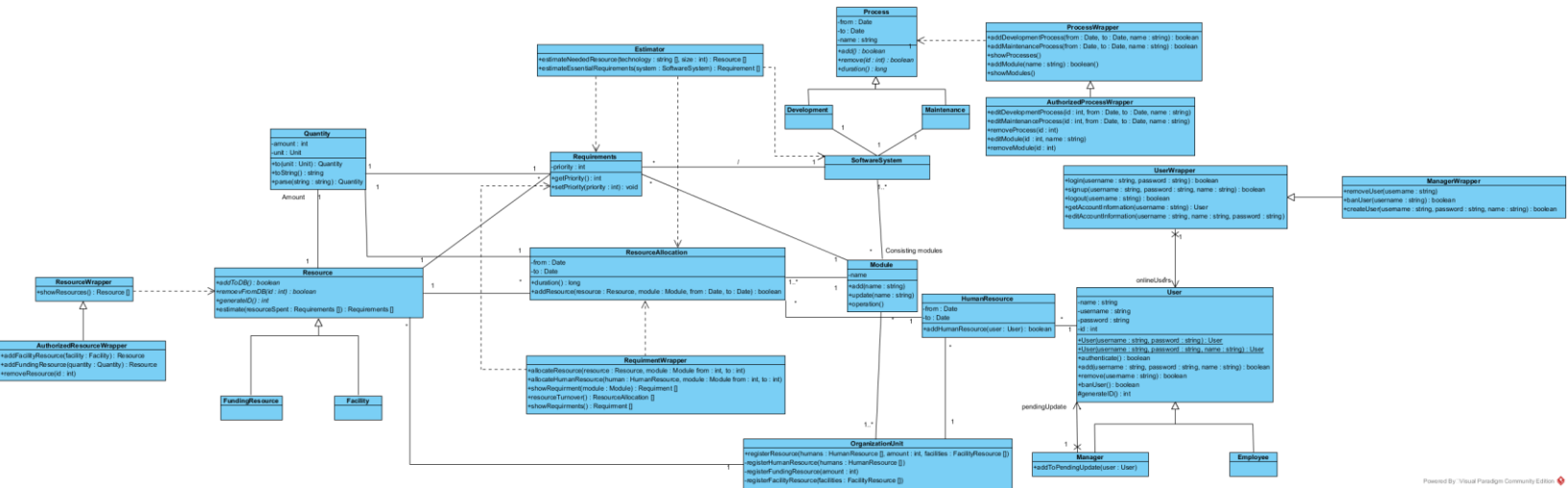
برای چک لیست استخراج شده از اسلایدهای درس، درستی سنجی، به ۶ نفر ساعت نیاز خواهد بود که بر عهده‌ی تمام اعضا می‌باشد.

فاز Transition:

برای تکمیل نسخه نهایی بر مبنای نتایج آزمون‌های فاز قبل، به ۳۰ نفر ساعت نیاز خواهد بود، که بر عهده‌ی همه‌ی اعضا می‌باشد.

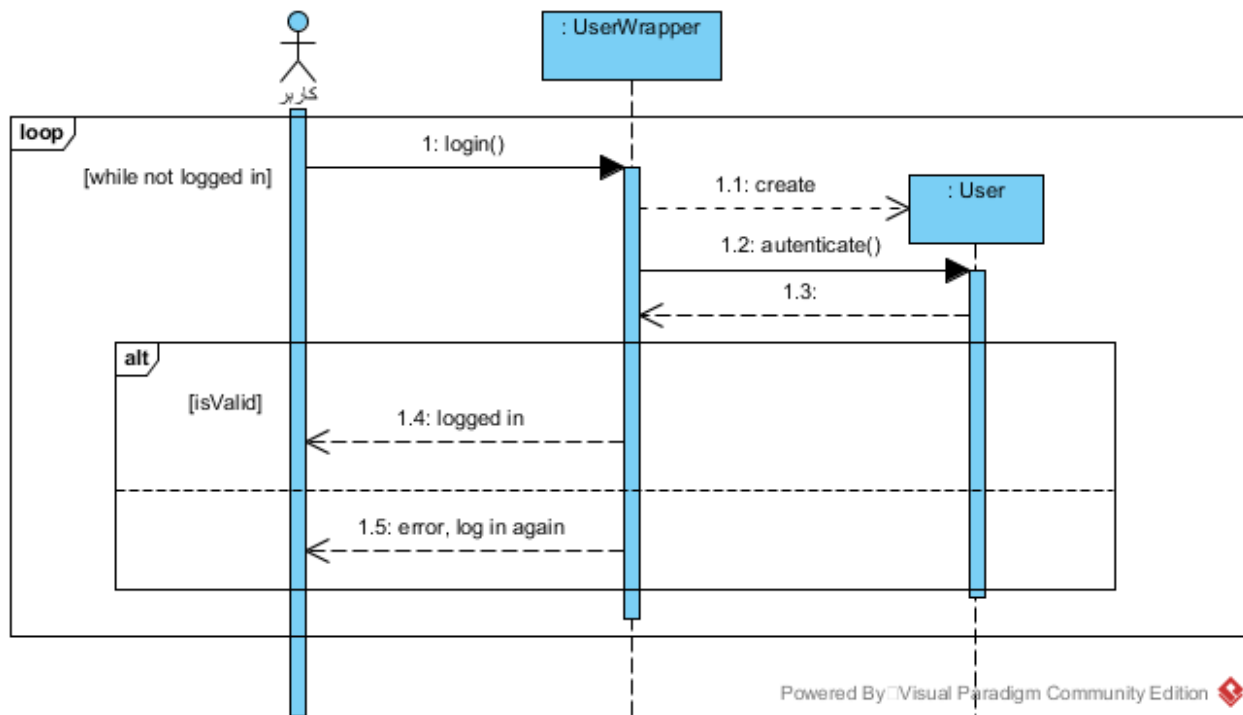
با توجه به اینکه اعضای گروه تخمین درستی از میزان زمان مورد نظر برای هر فاز نداشتند، برنامه‌ی فوق در هر مرحله به روز رسانی شده و تصحیح خواهد شد.

نمودار کلاس تحلیل:

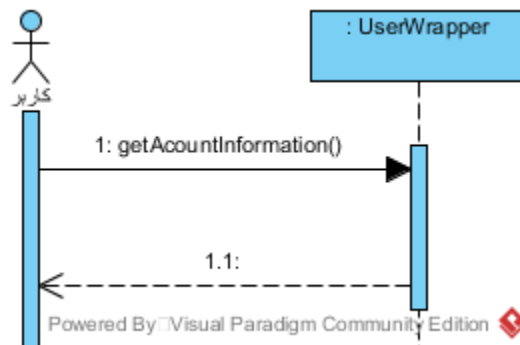


نمودارهای توالی تحلیل:

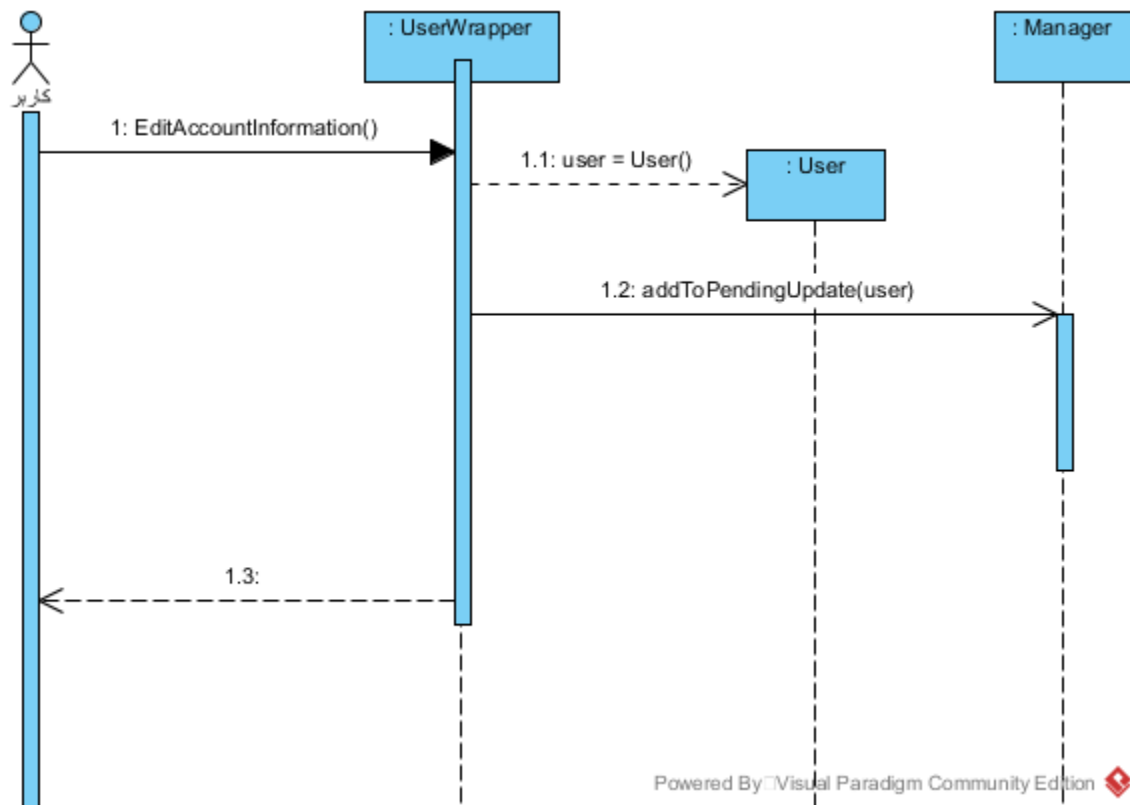
ورود به سیستم:



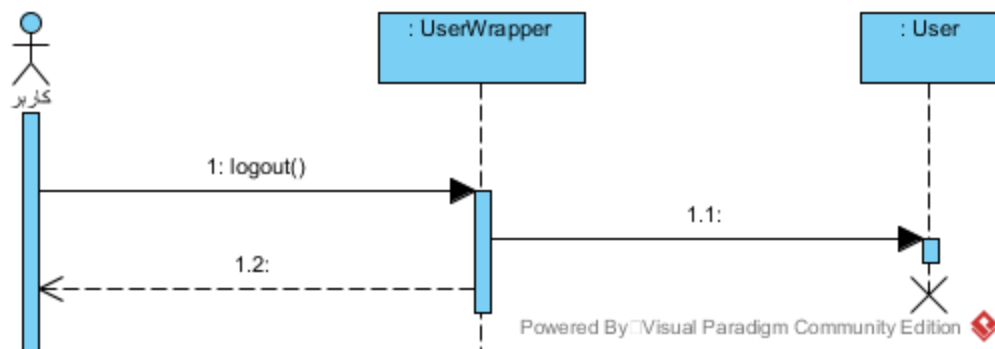
مشاهده اطلاعات حساب کاربری:



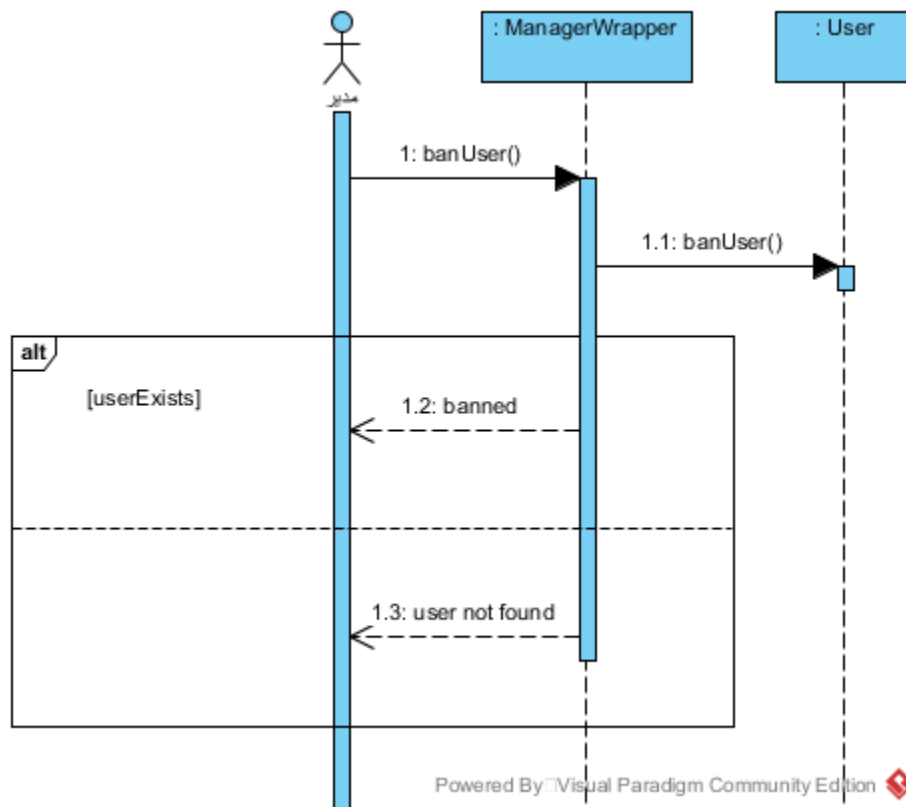
درخواست ویرایش اطلاعات حساب کاربری:



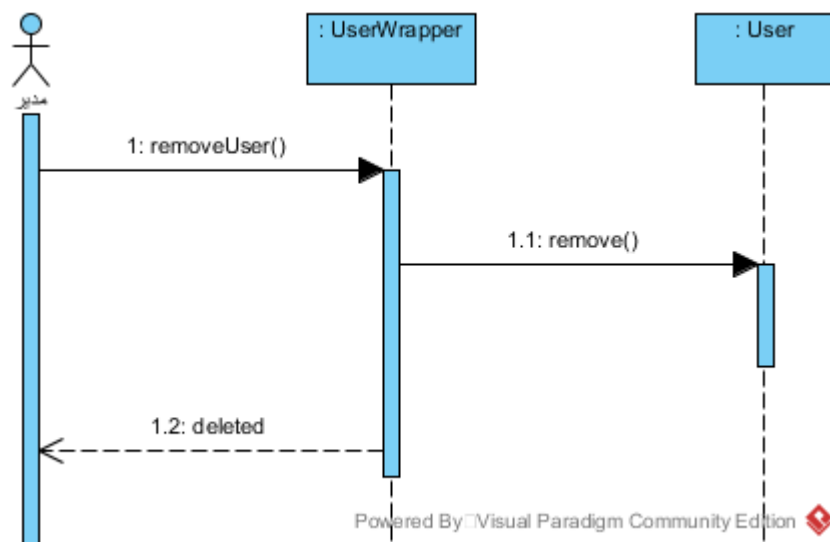
خروج از سیستم:



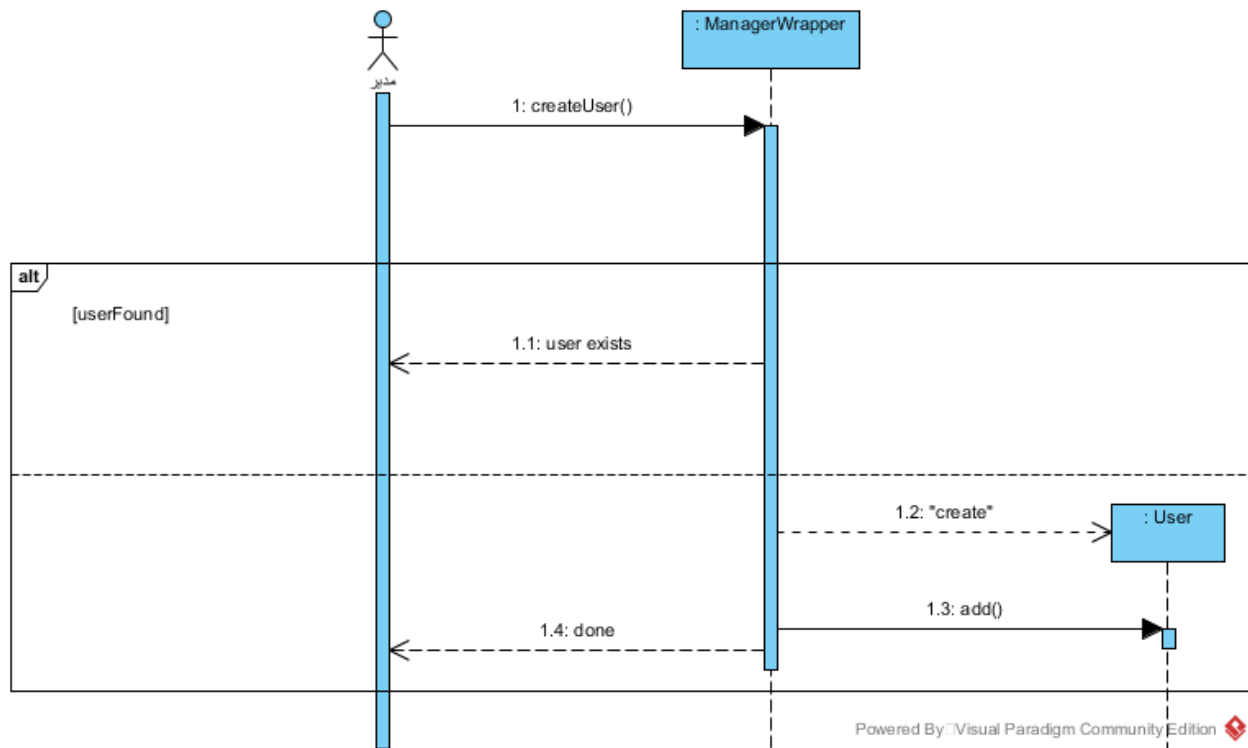
تعليق حساب کاربری:



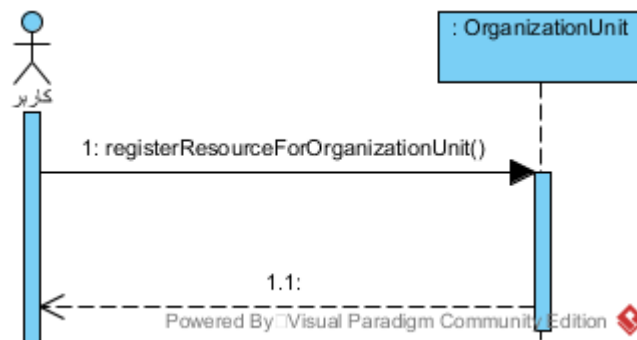
حذف حساب کاربری:



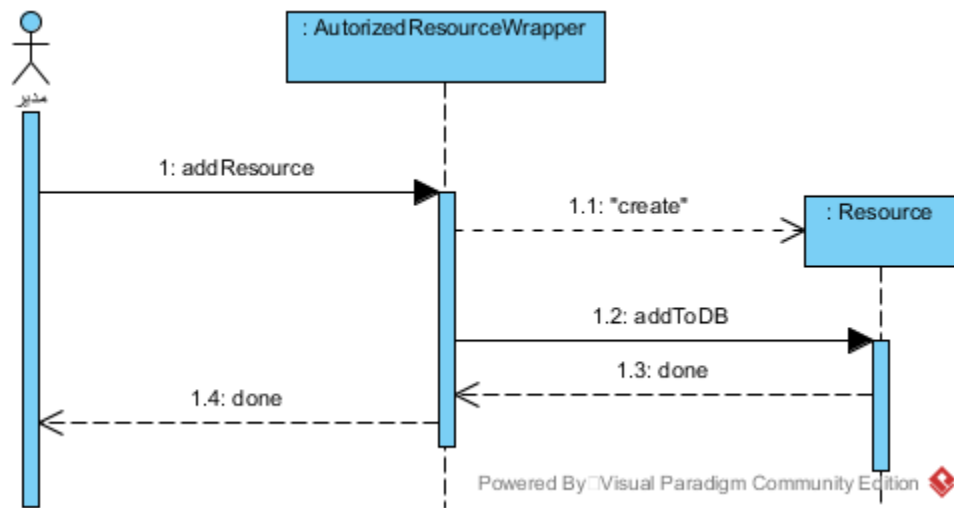
ایجاد کاربر جدید:



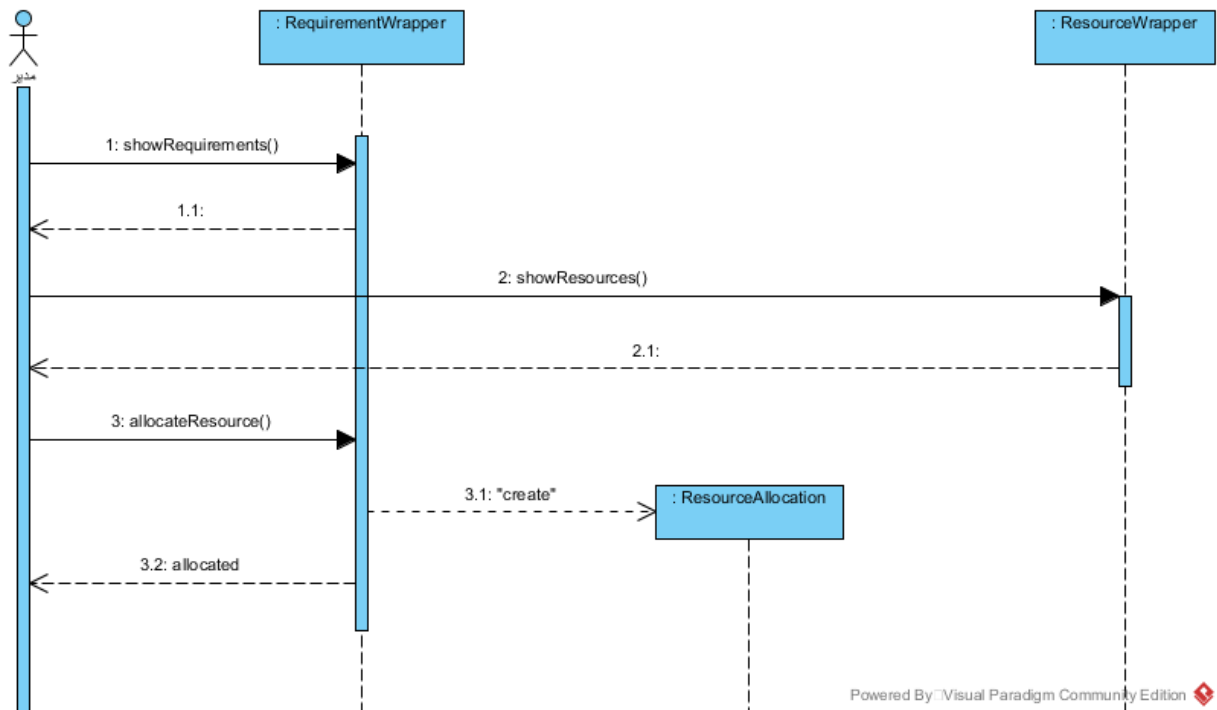
ثبت منابع مورد نیاز یک واحد سازمان:



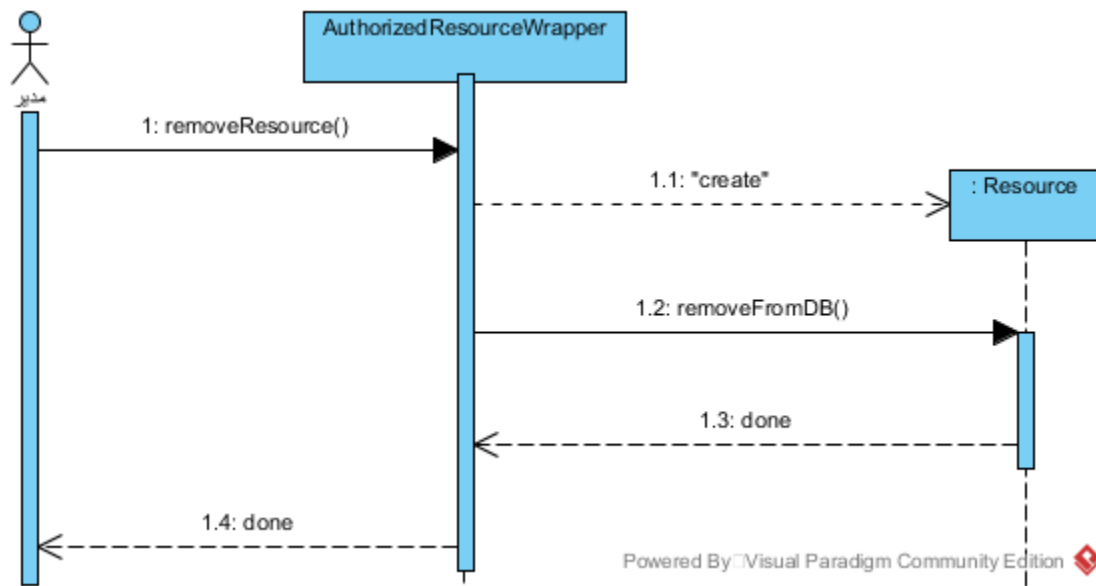
ثبت یک منبع جدید:



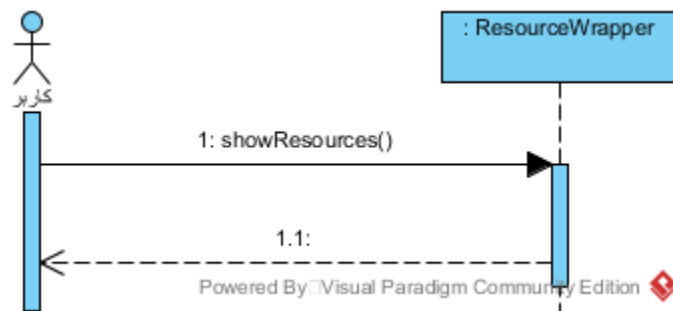
اختصاص منبع به ماژول:



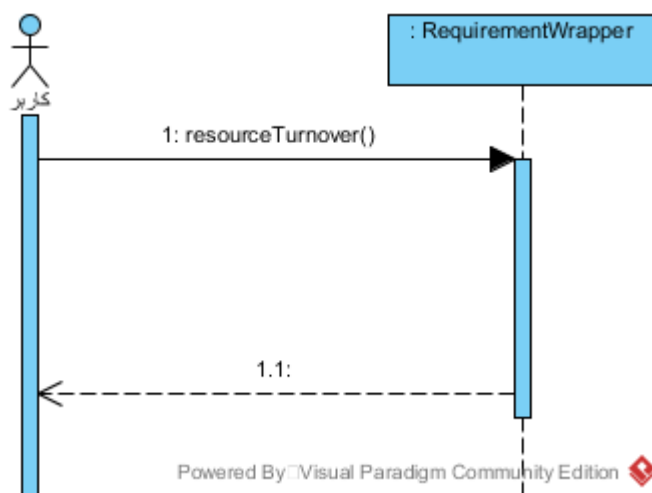
حذف منبع



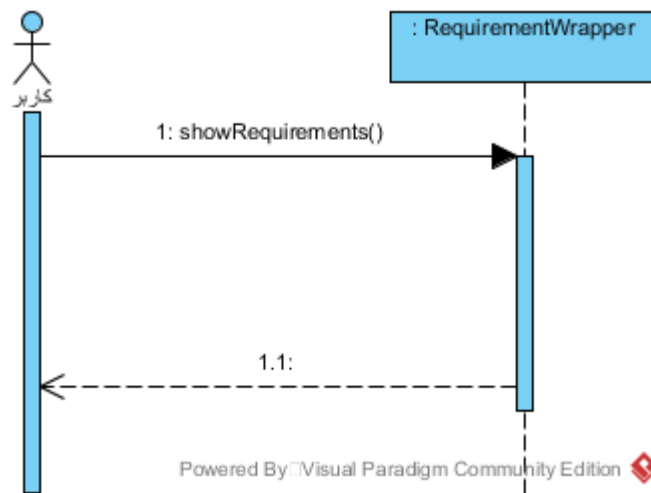
دریافت گزارش منابع موجود



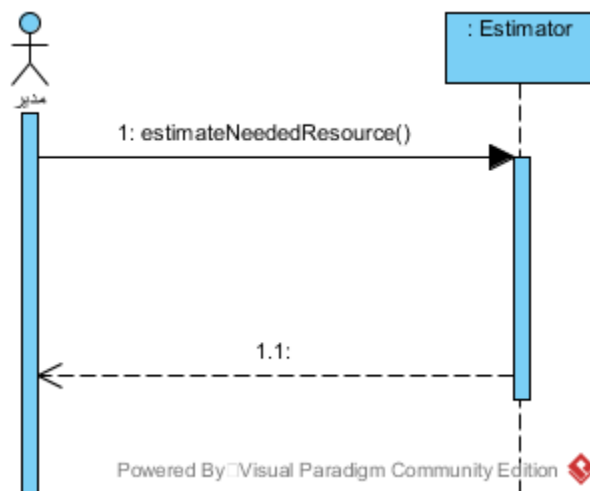
دریافت گزارش جریان چرخشی منابع:



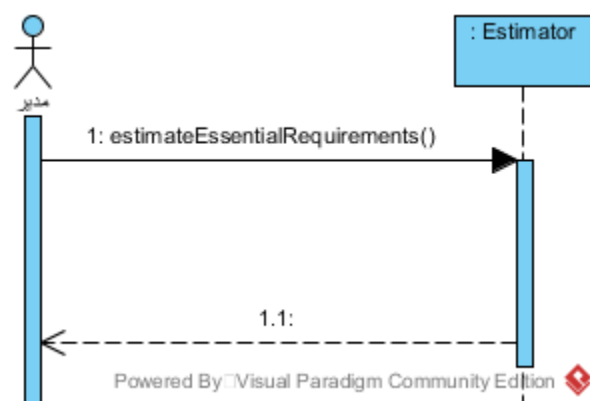
دریافت گزارش منابع مورد نیاز:



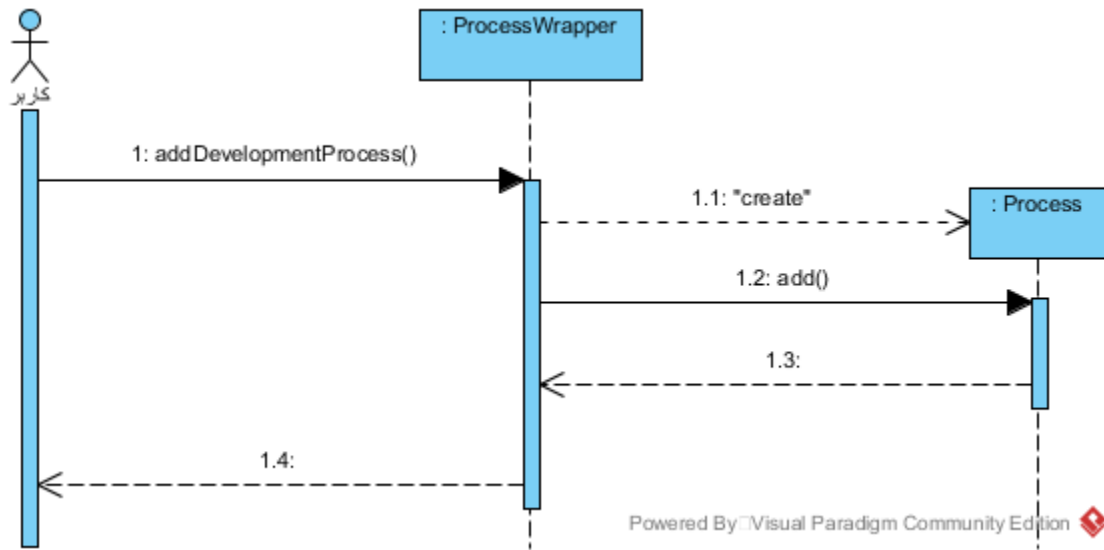
تخمین منابع مورد نیاز سیستم نرم‌افزاری:



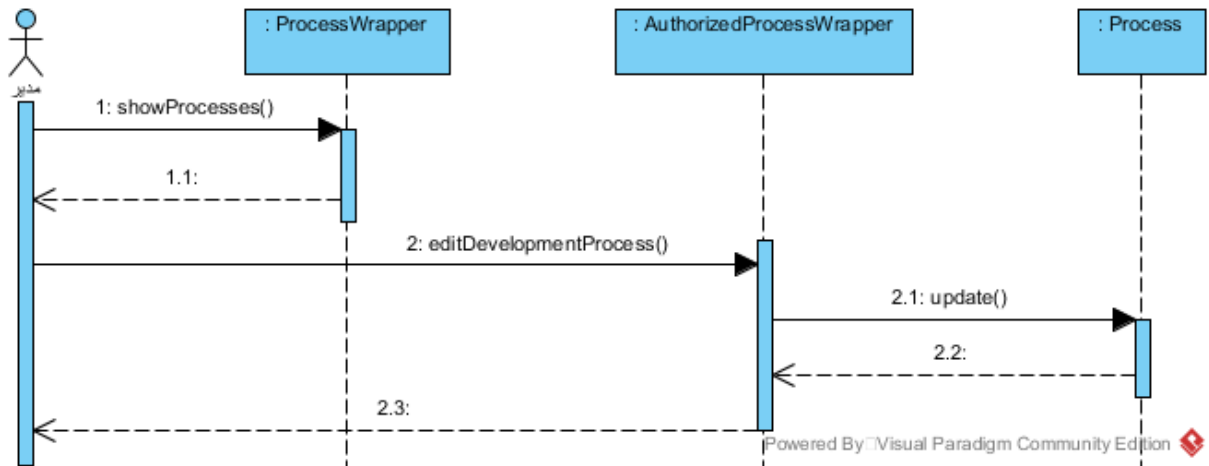
تخمین نیازمندی‌های ضروری براساس منابع مورد نیاز:



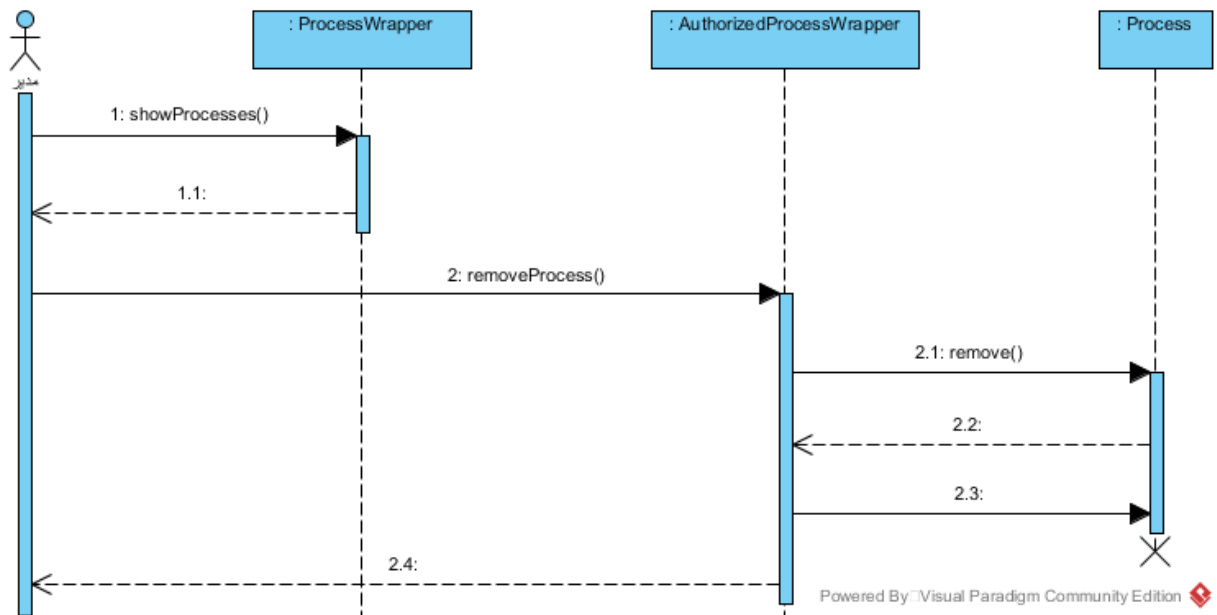
ثبت فرآیند ایجاد:



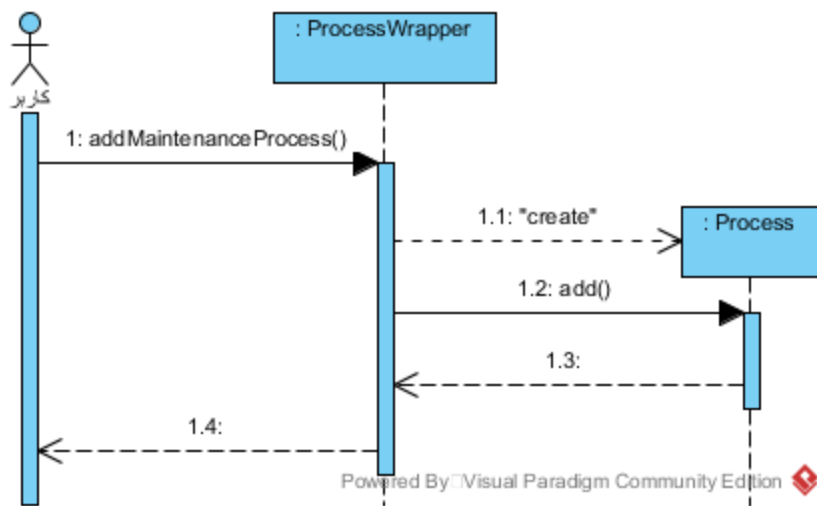
ویرایش فرآیند ایجاد:



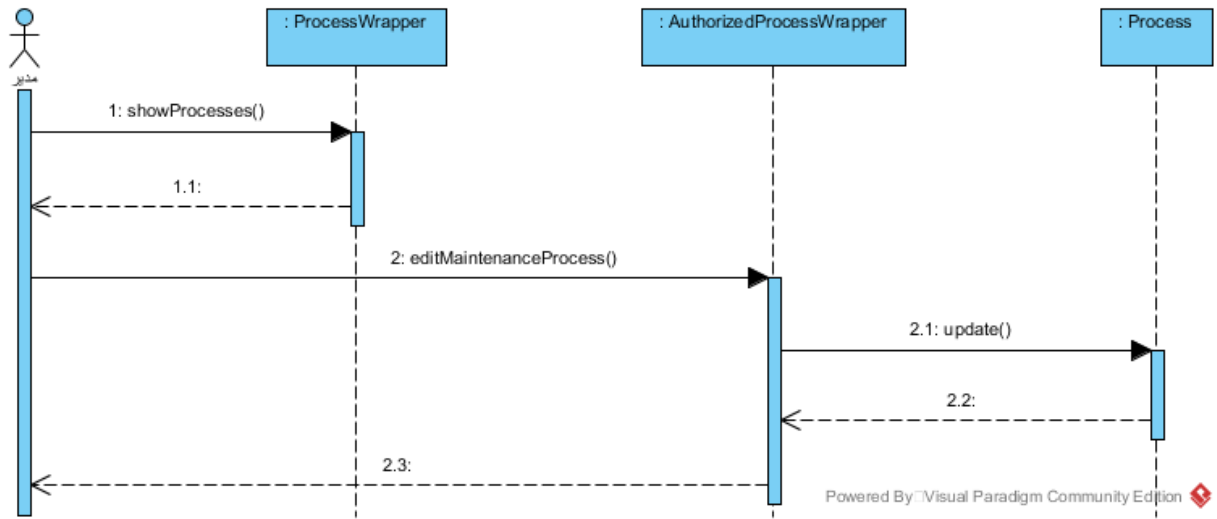
حذف فرآیند ایجاد:



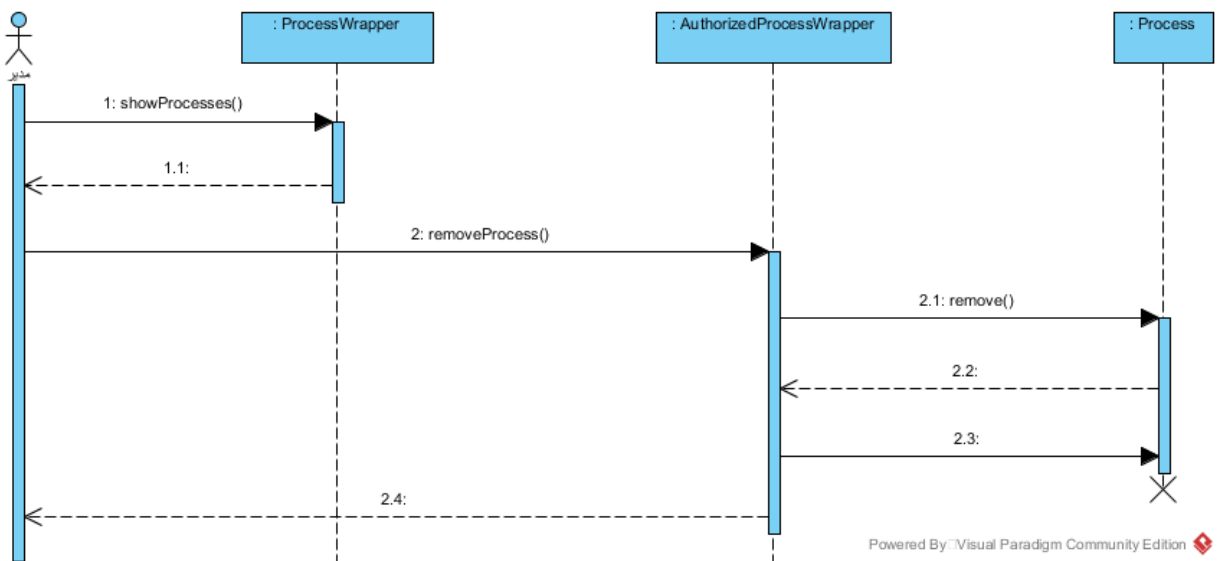
ثبت فرآیند نگهداری:



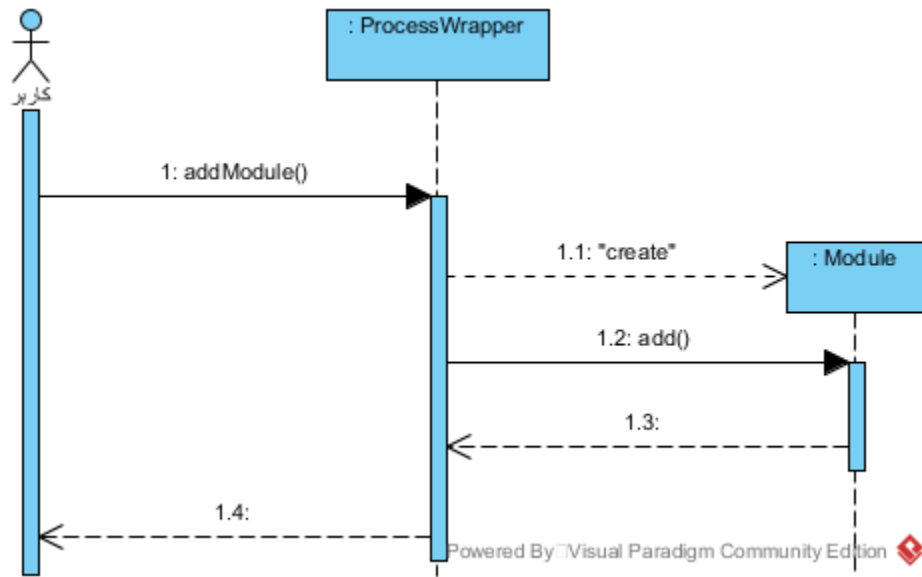
ویرایش فرآیند نگهداری:



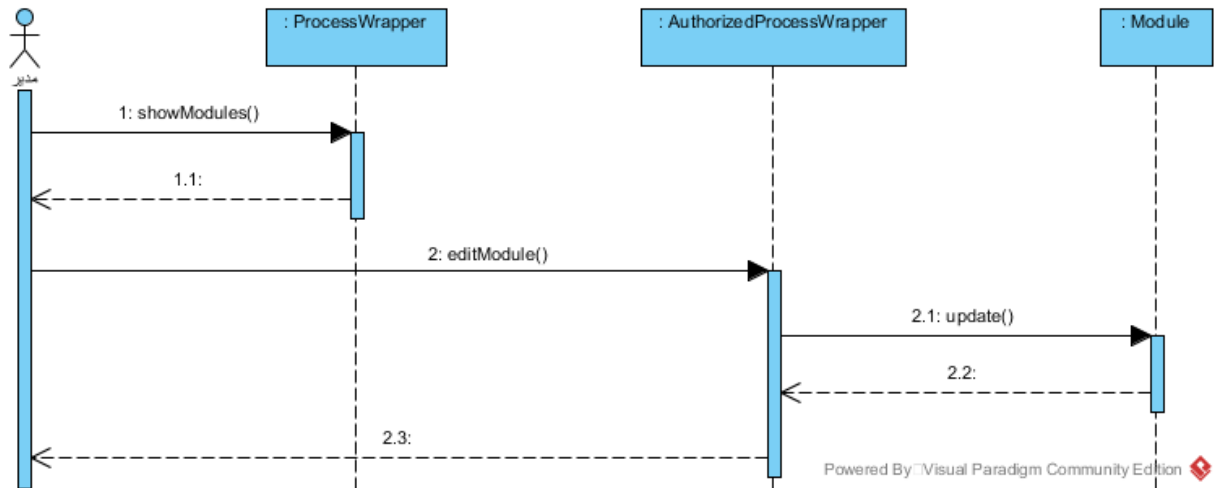
حذف فرآیند نگهداری:



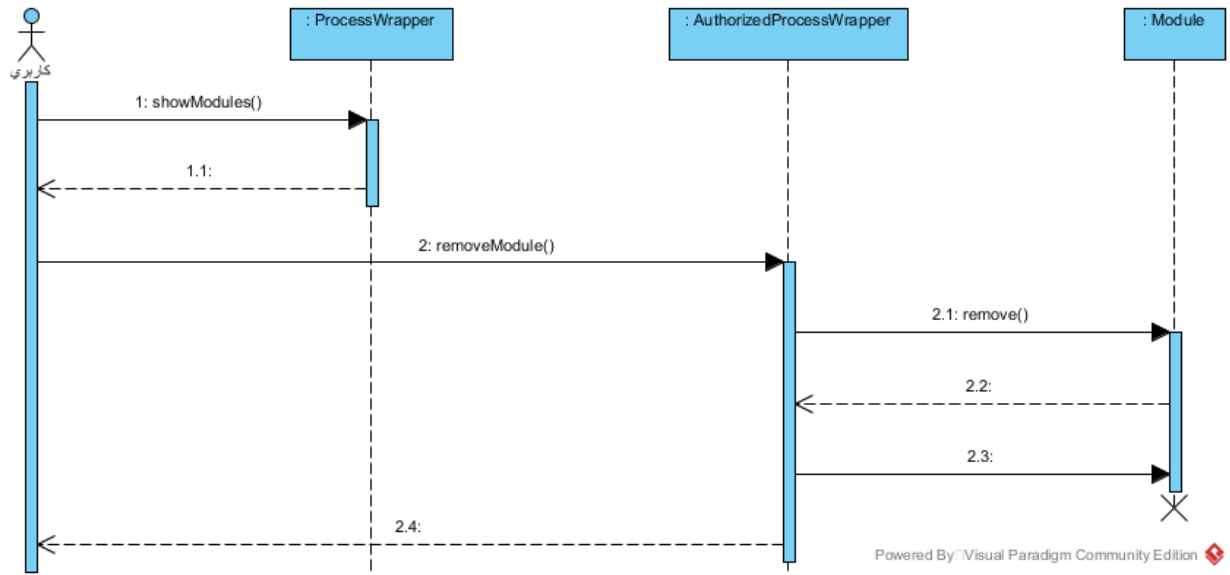
ثبت ماژول:



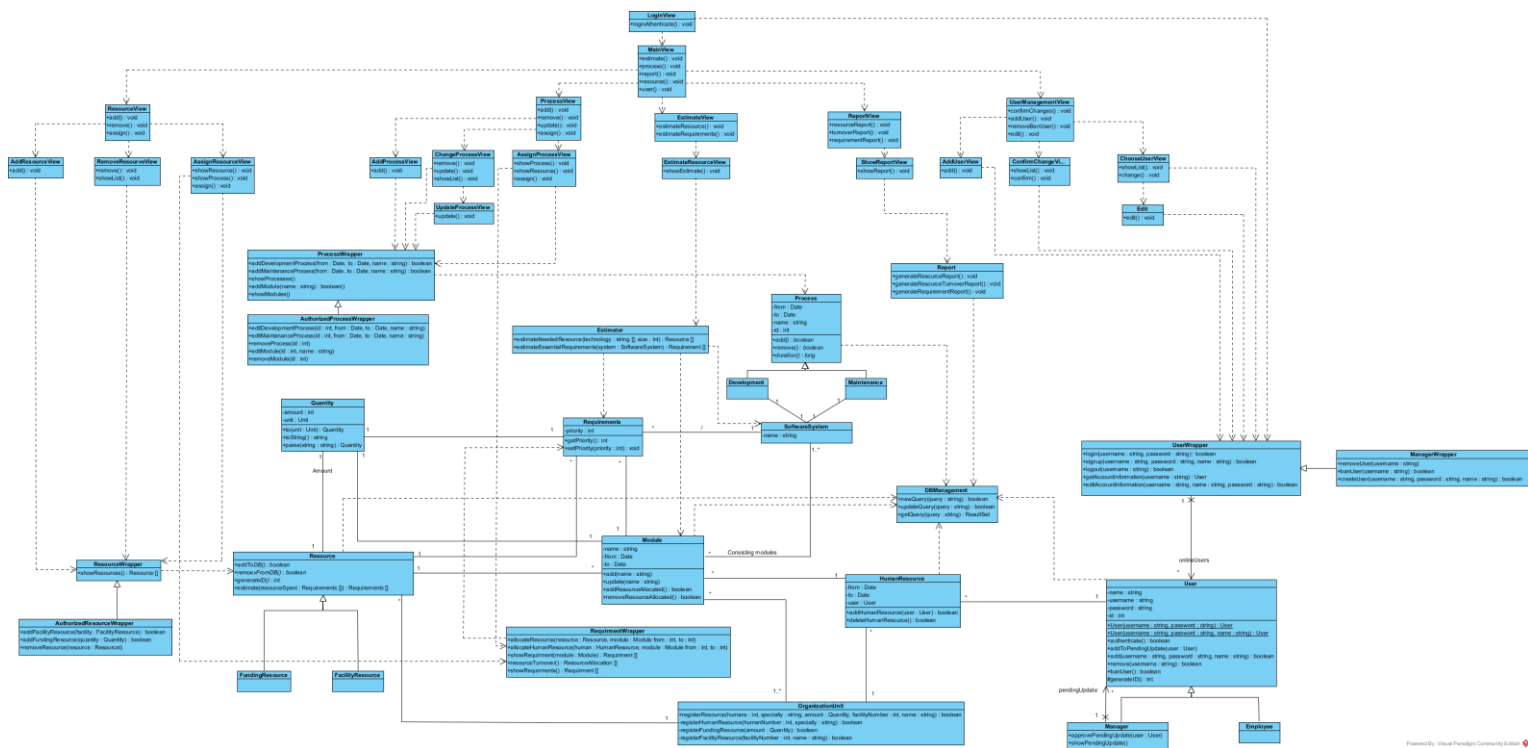
ویرایش ماژول:



حذف ماژول:

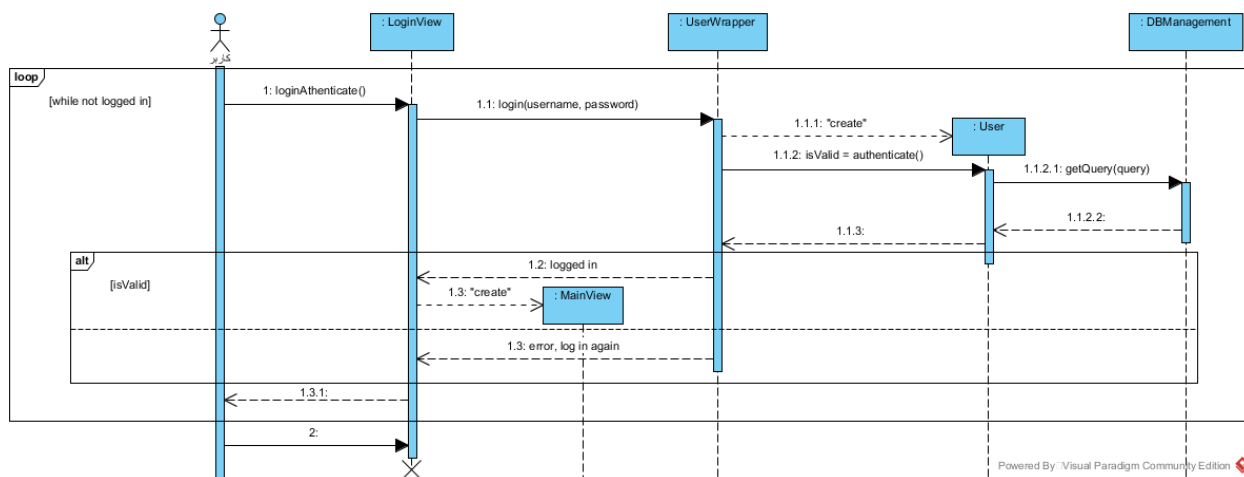


نمودار کلاس طراحی (ویرایش شده):

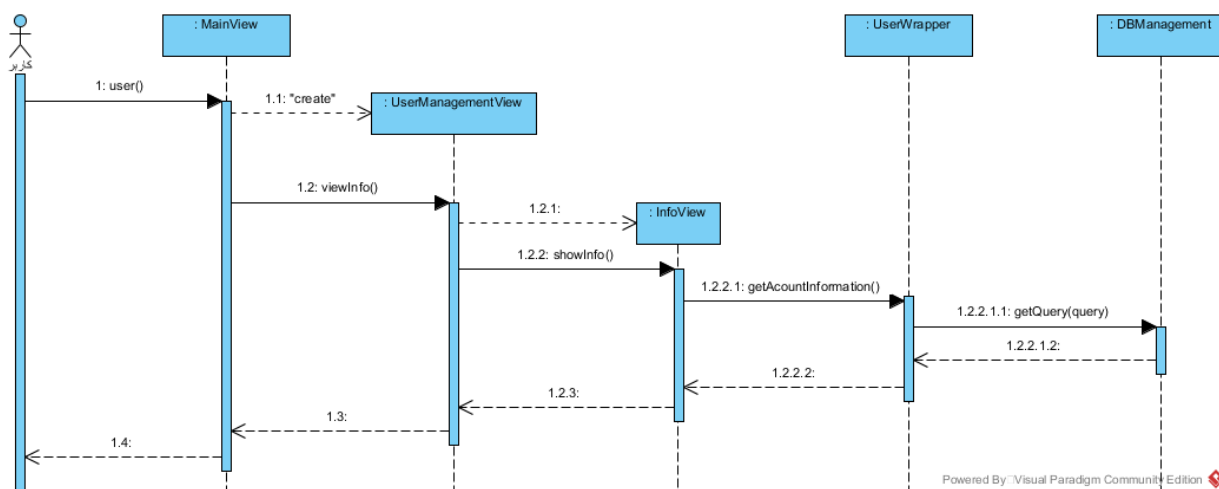


نمودارهای توالی طراحی (ویرایش شده):

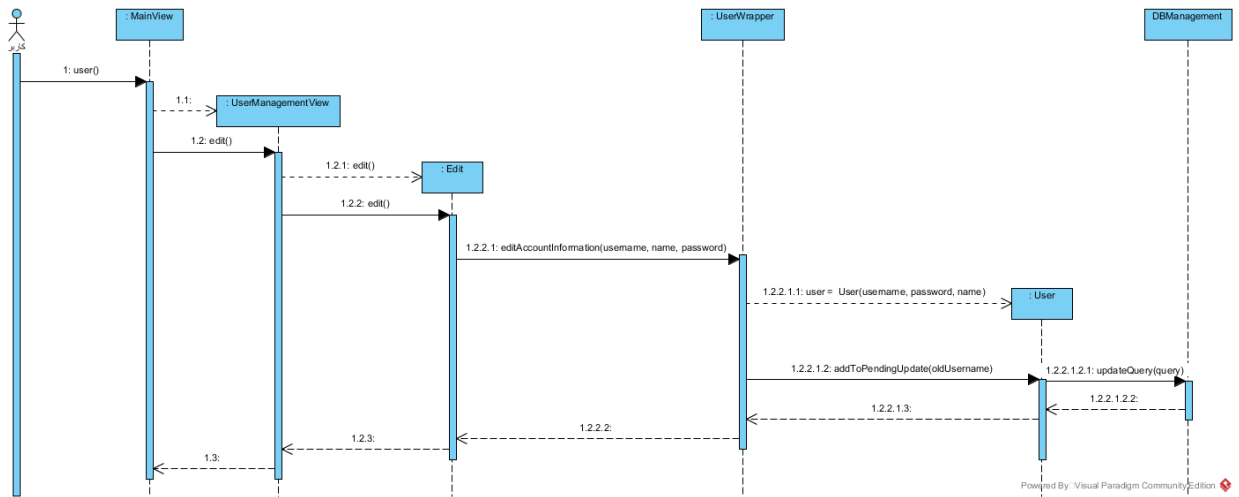
ورود به سیستم:



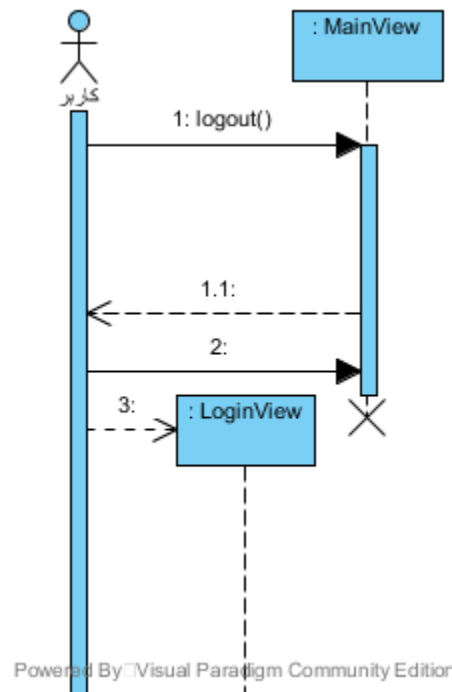
مشاهده اطلاعات حساب کاربری:



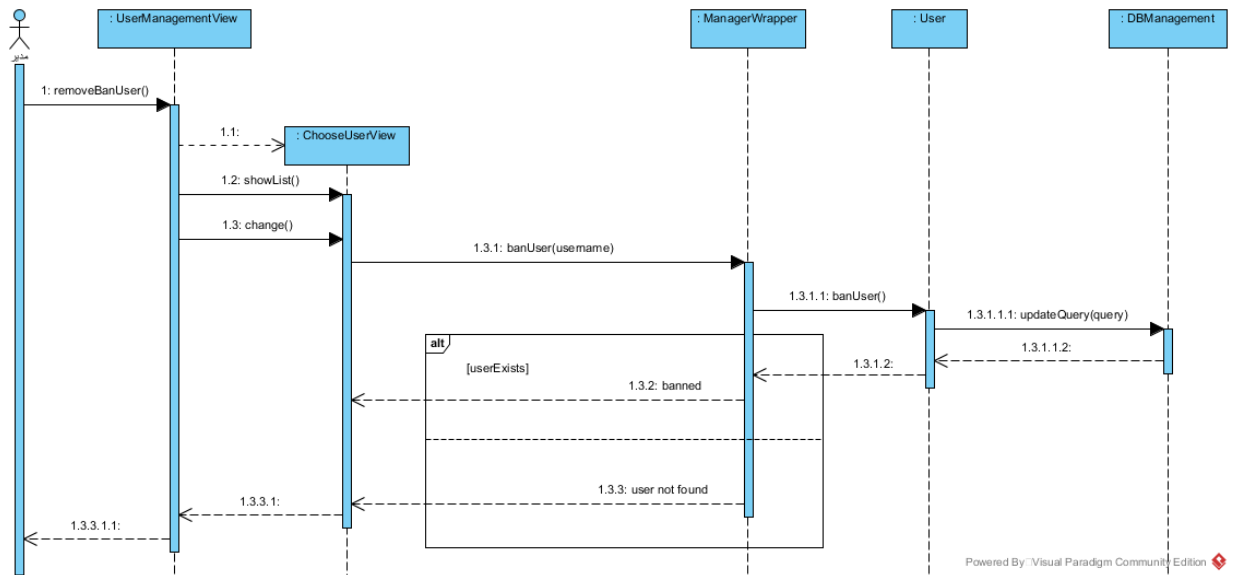
درخواست ویرایش اطلاعات حساب کاربری (ویرایش شده):



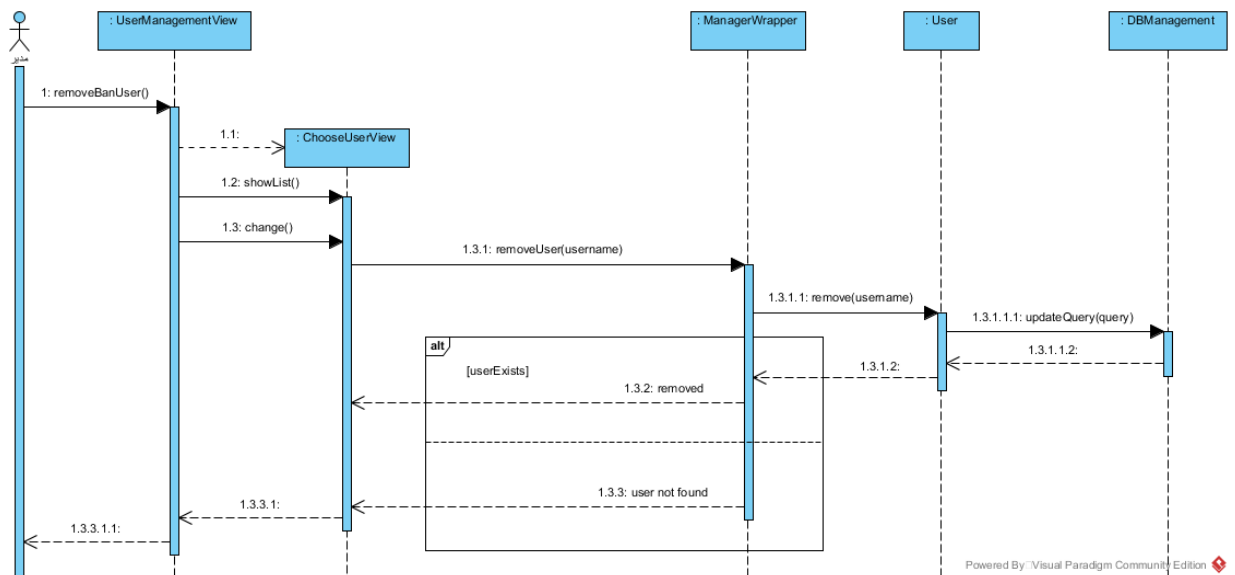
خروج از سیستم (ویرایش شده):



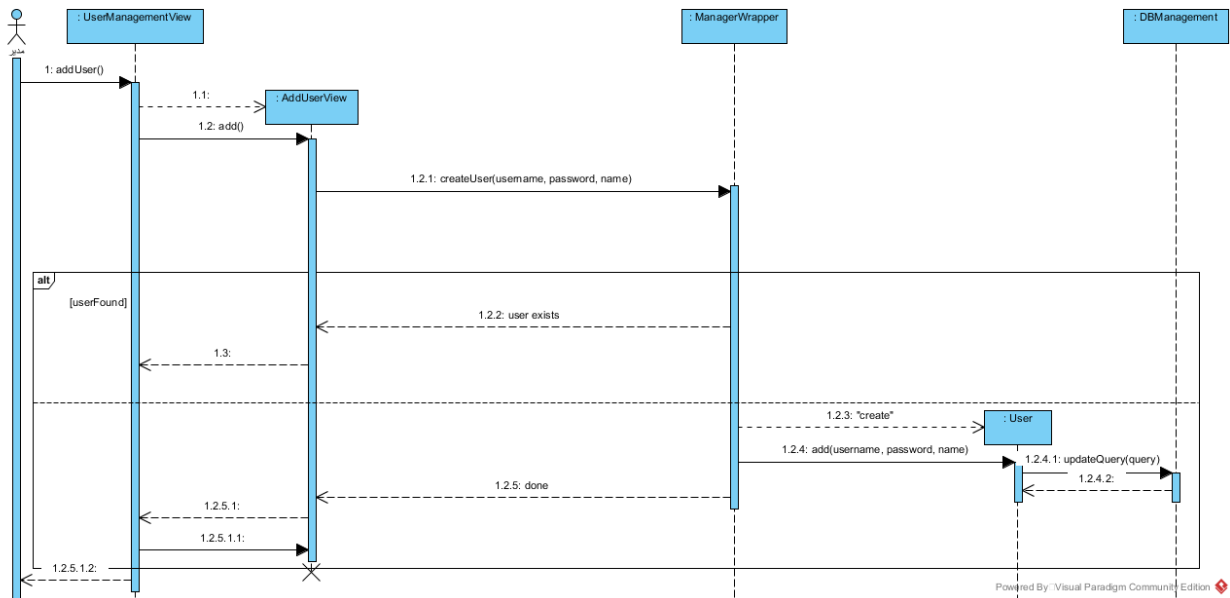
تعليق حساب کاربری:



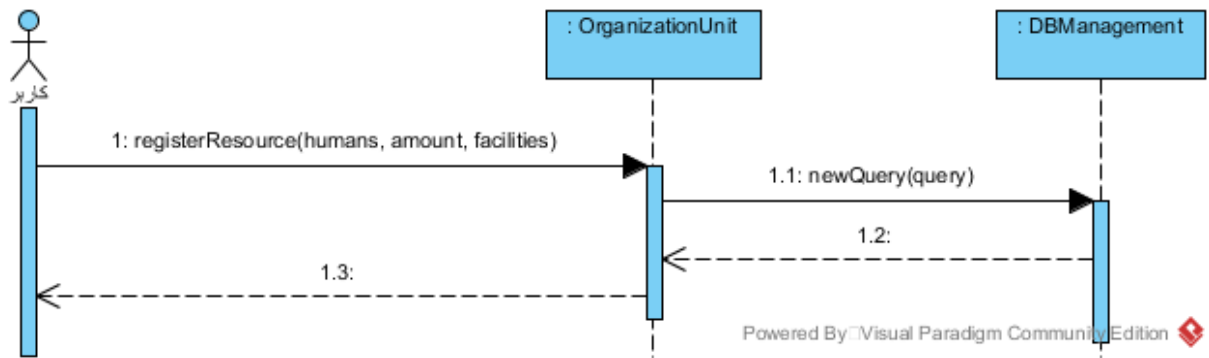
حذف حساب کاربری:



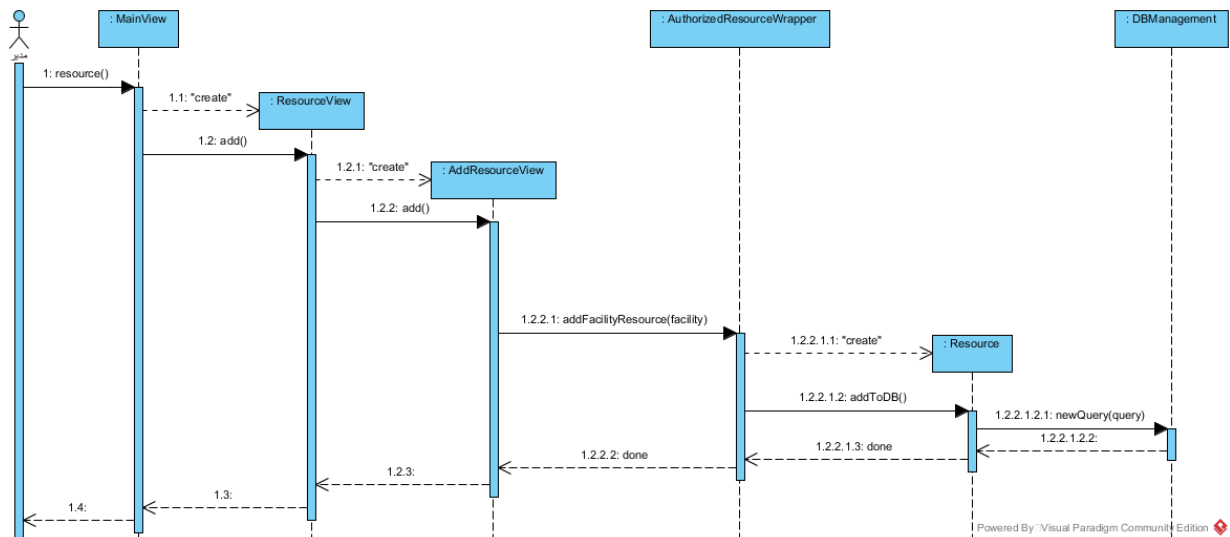
ایجاد کاربر جدید (ویرایش شده):



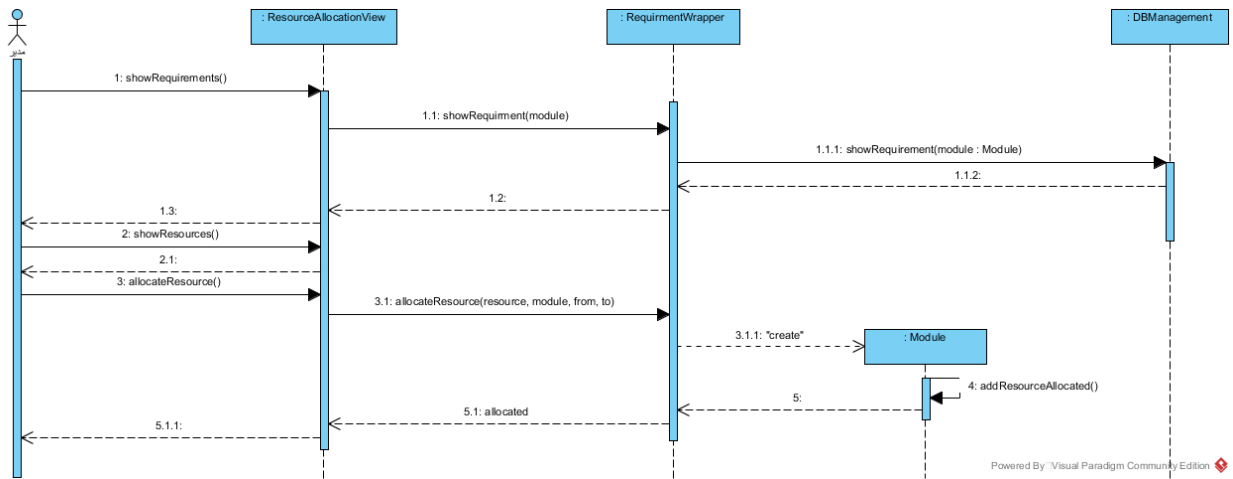
ثبت منابع مورد نیاز یک واحد سازمان:



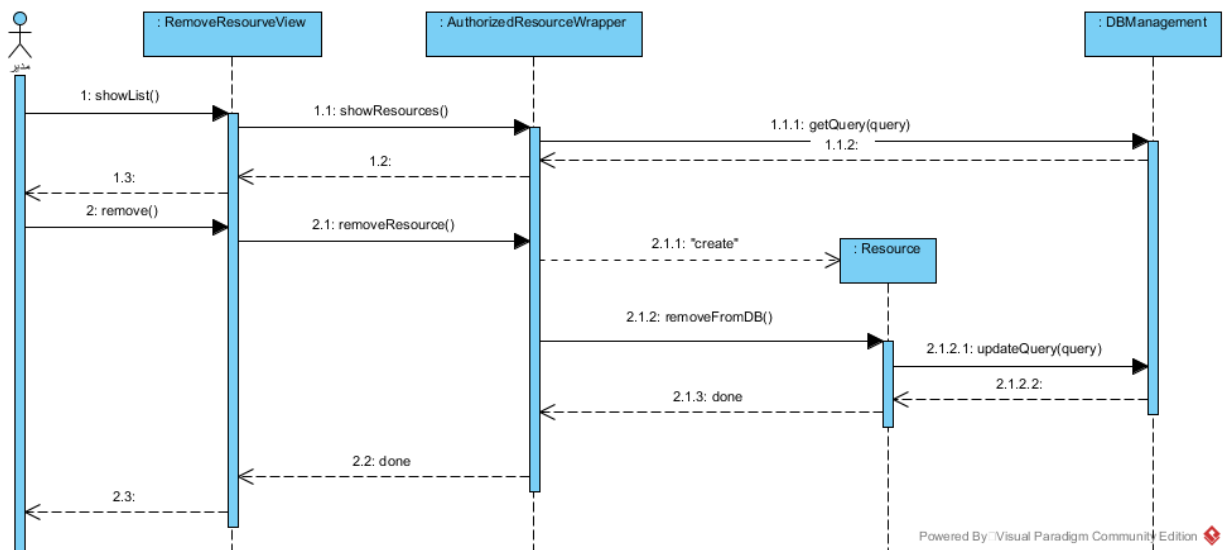
ثبت یک منبع جدید:



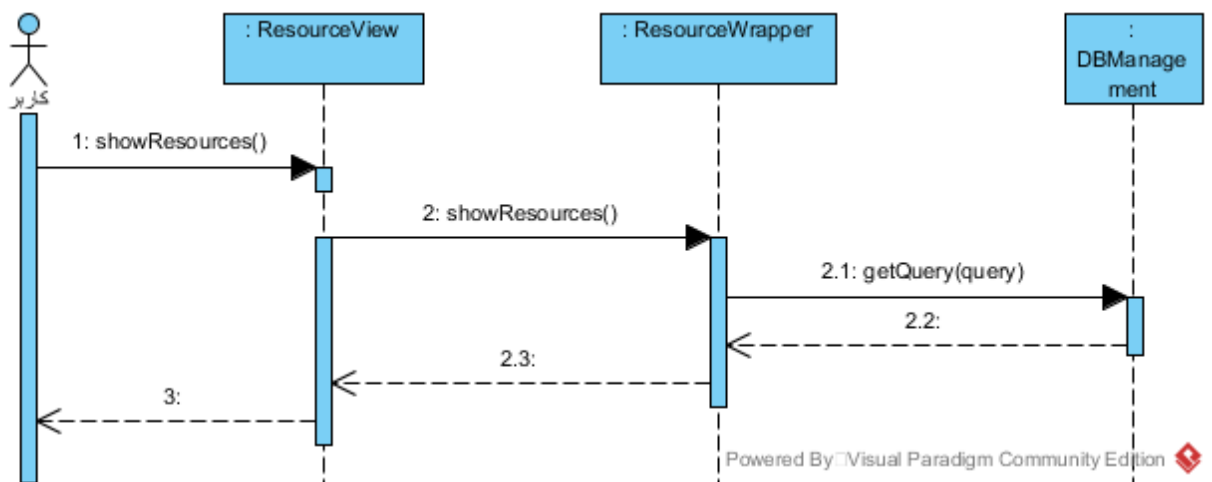
اختصاص منبع به ماژول (ویرایش شده):



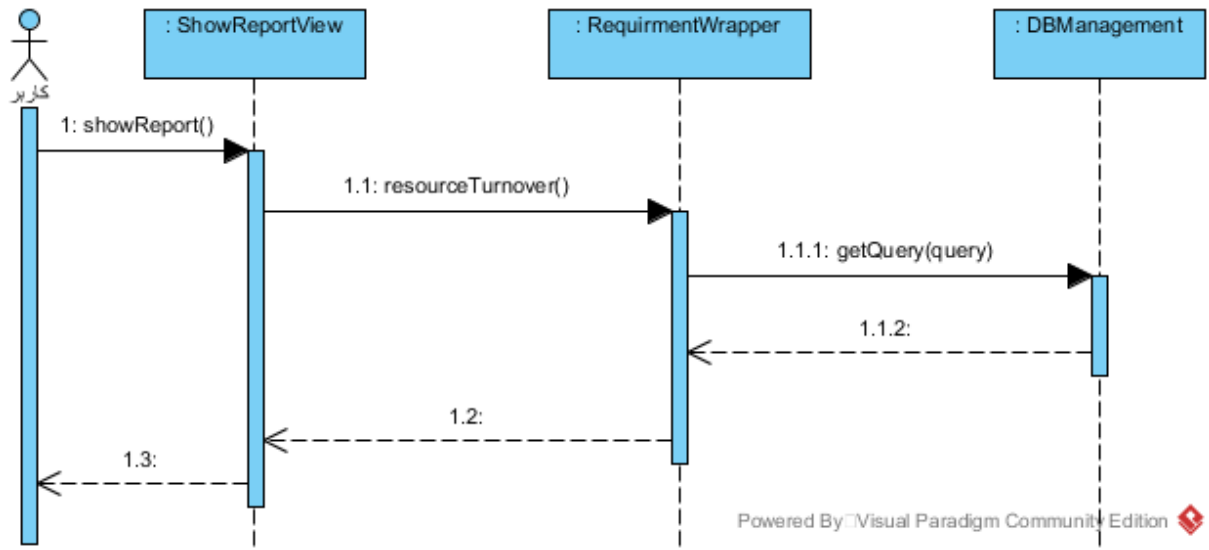
حذف منبع (ویرایش شده):



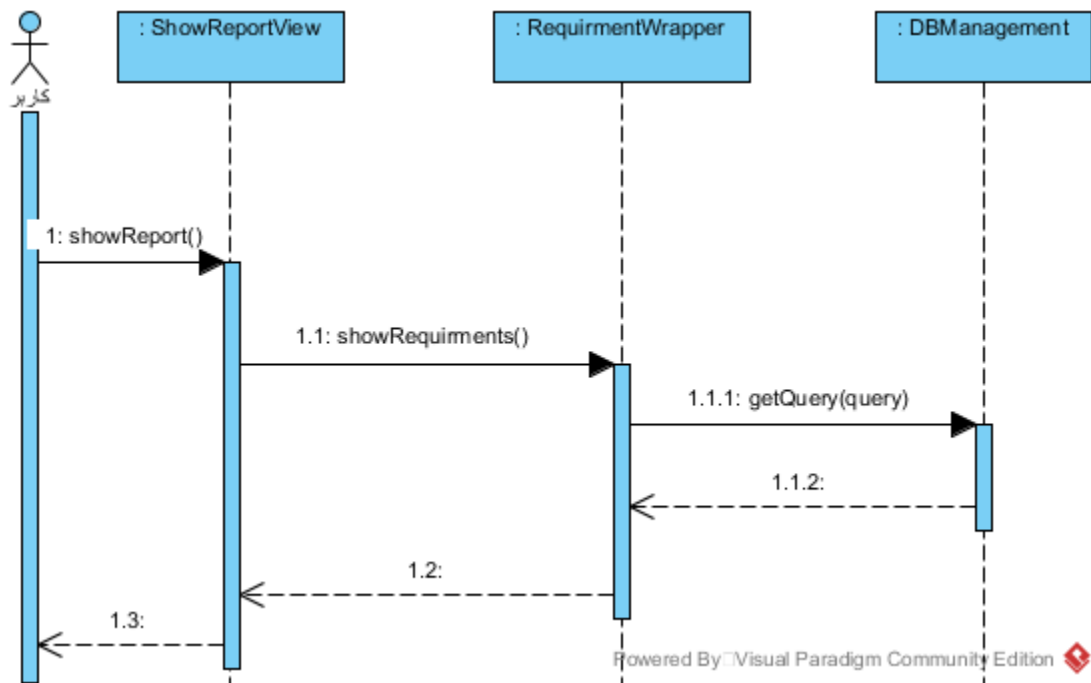
دریافت گزارش منابع موجود:



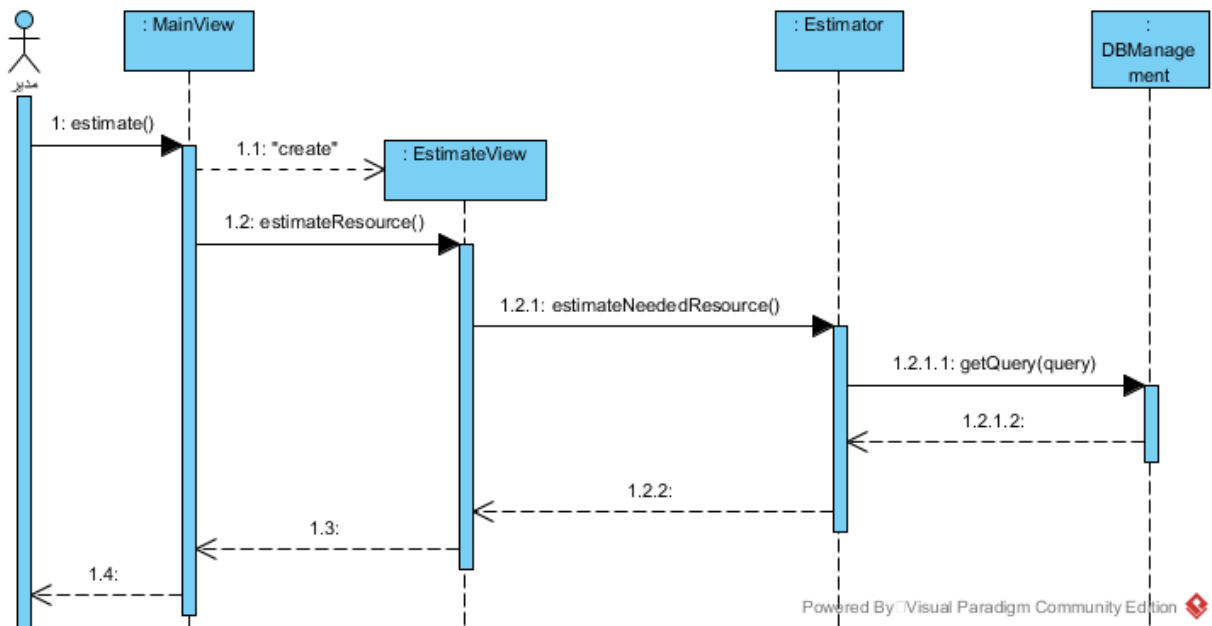
دریافت گزارش جریان چرخشی منابع:



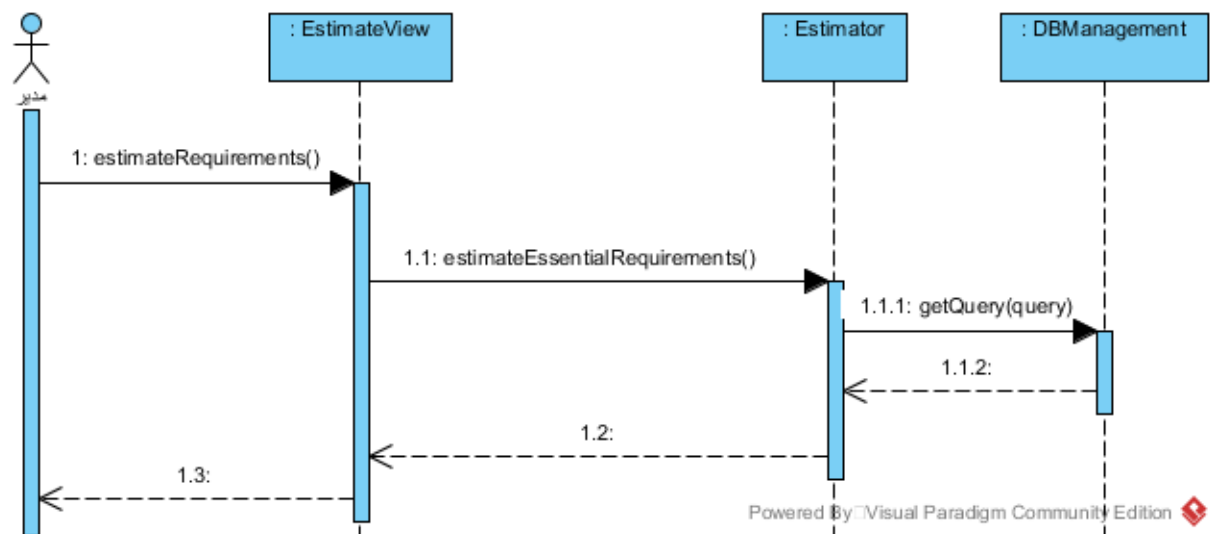
دریافت گزارش منابع مورد نیاز:



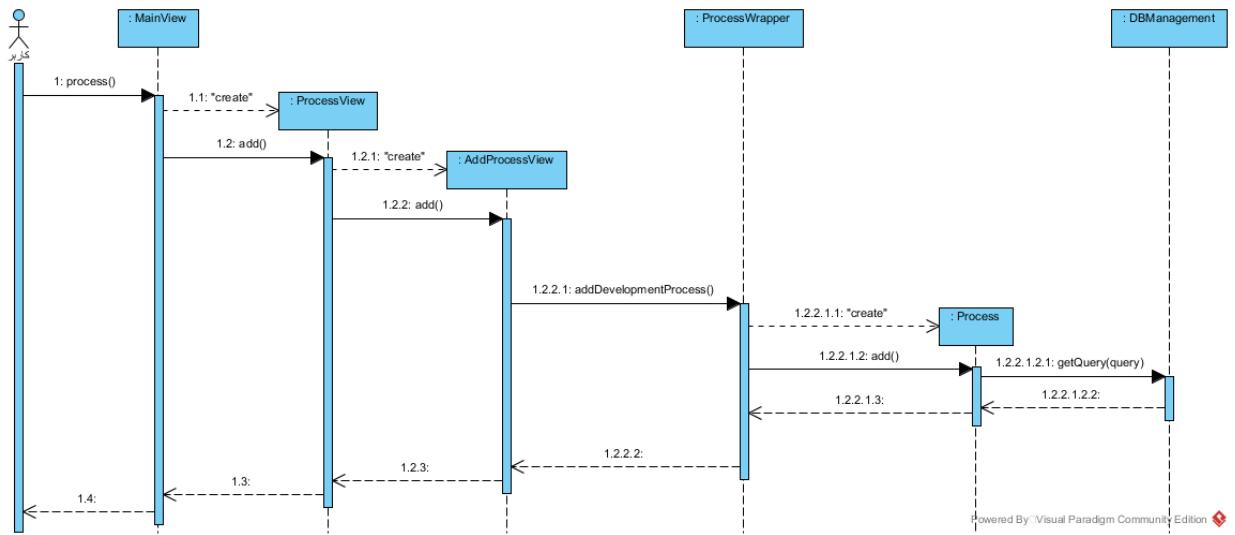
تخمین منابع مورد نیاز سیستم نرم‌افزاری:



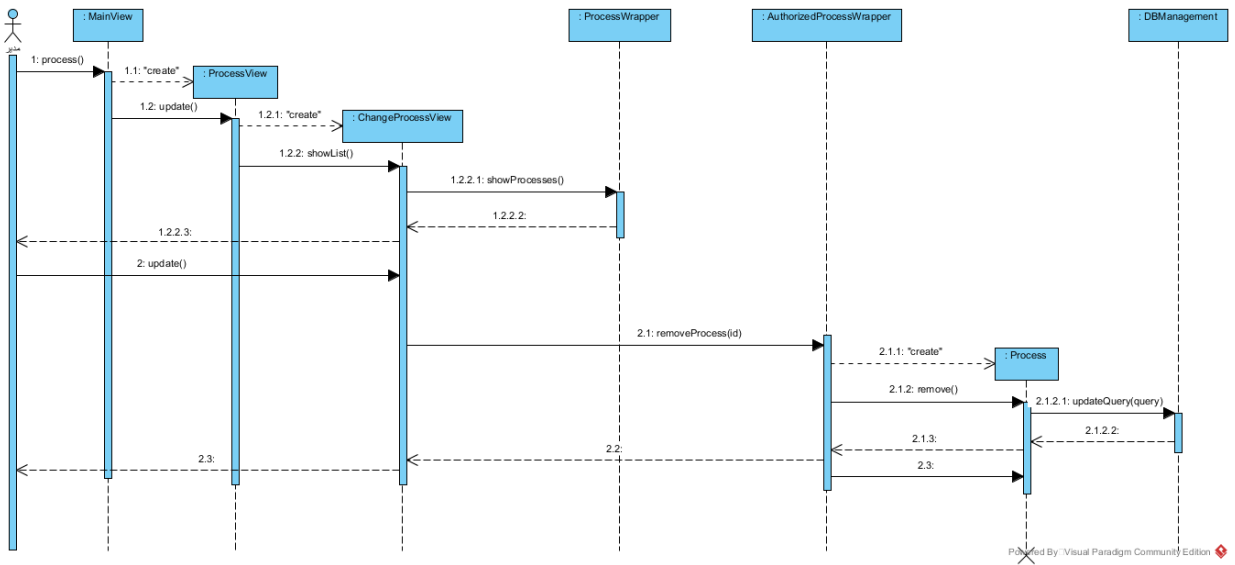
تخمین نیازمندی‌های ضروری براساس منابع مورد نیاز:



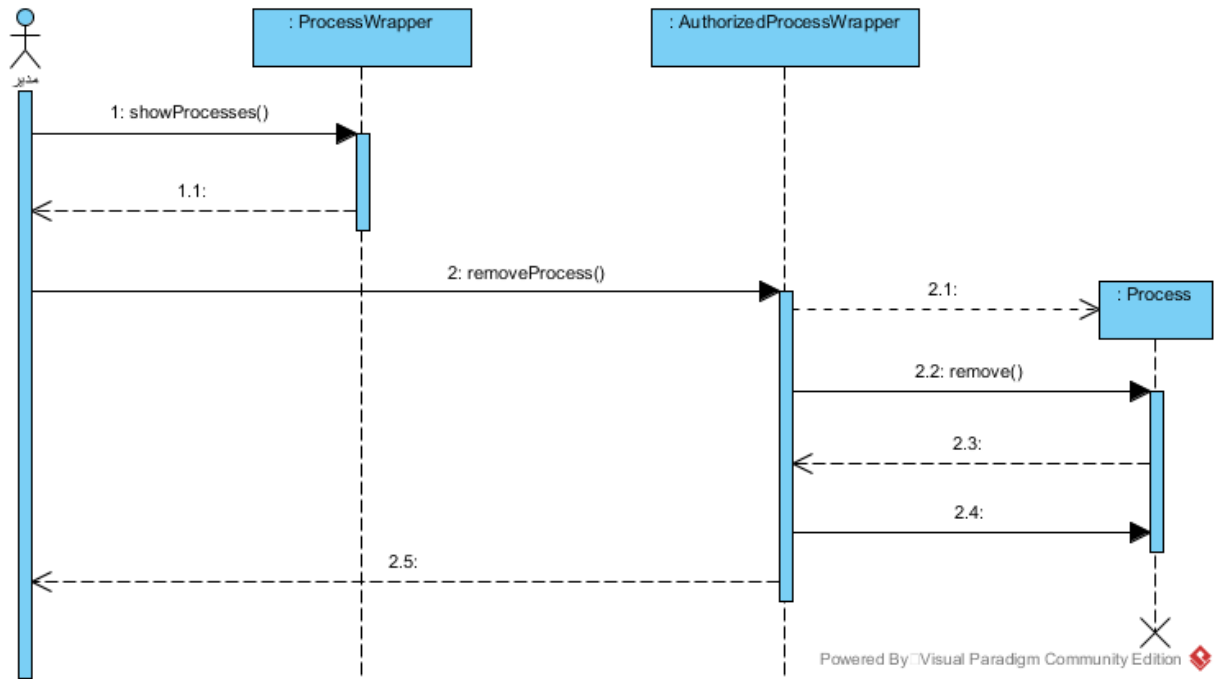
ثبت فرآیند ایجاد:



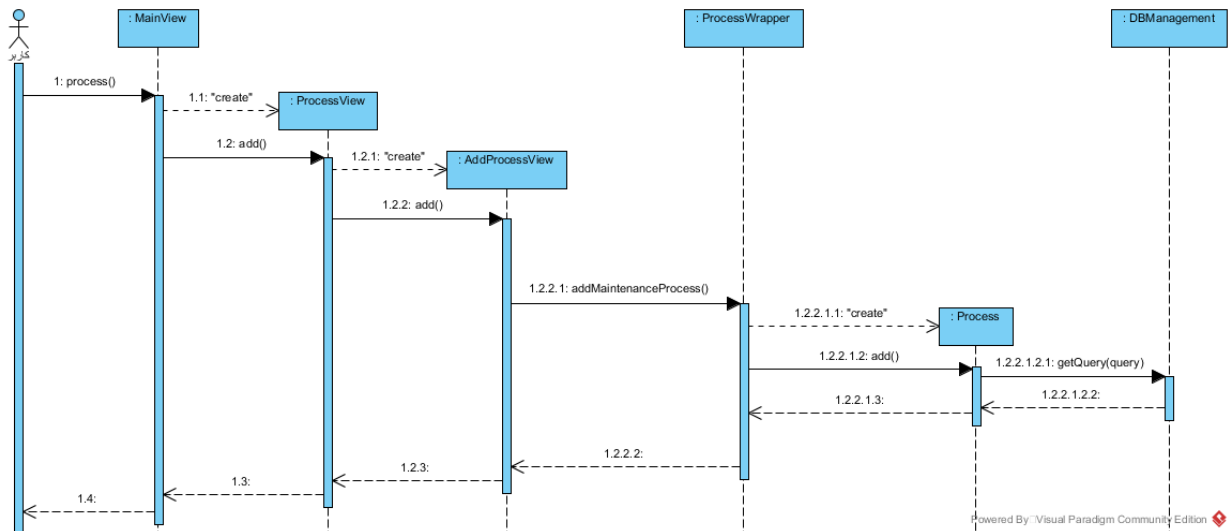
ویرایش فرآیند ایجاد (ویرایش شده):



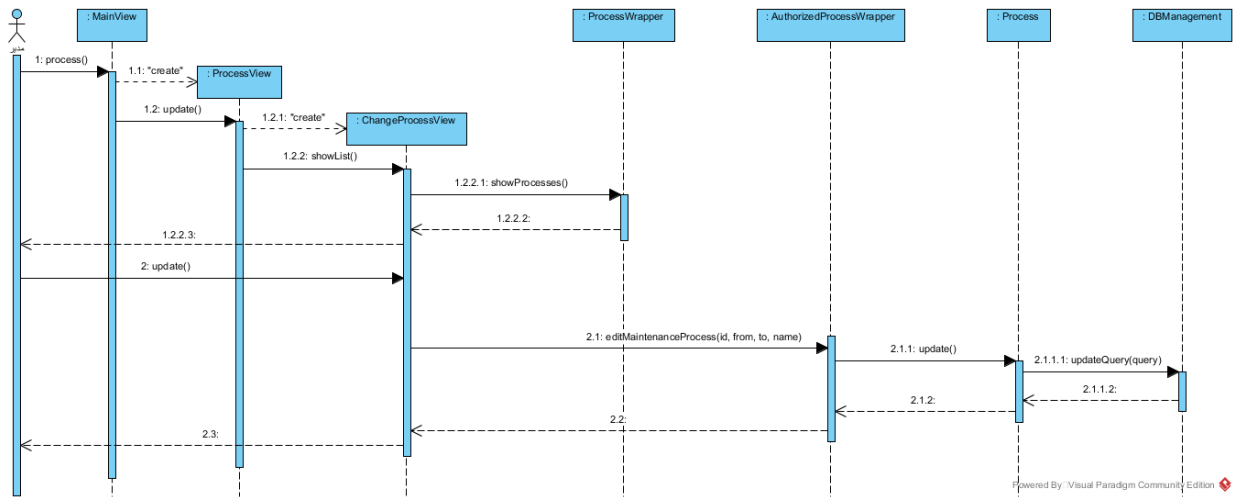
حذف فرآیند ایجاد (ویرایش شده):



ثبت فرآیند نگهداری:

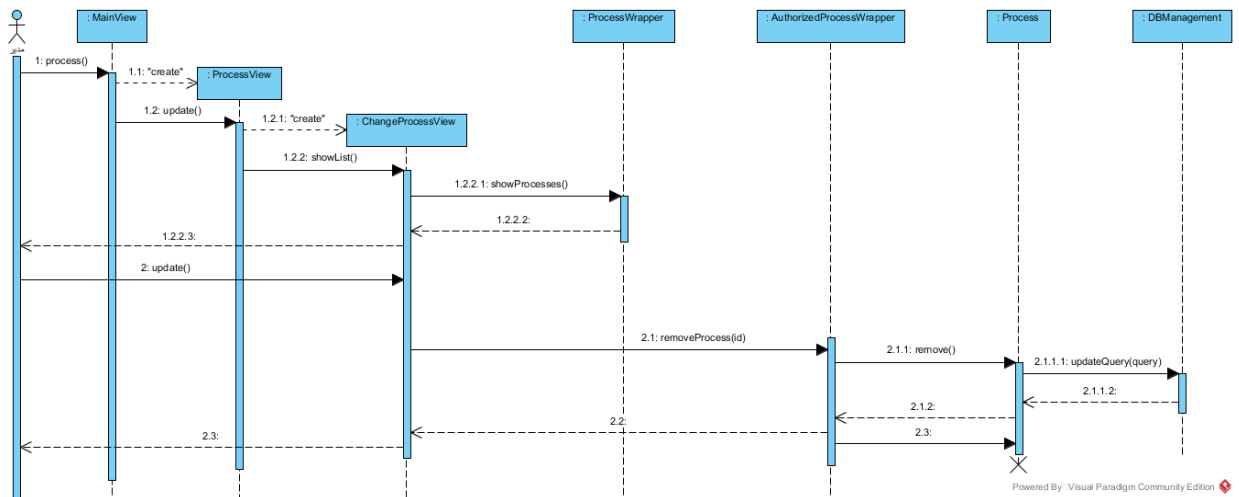


ویرایش فرآیند نگهداری:



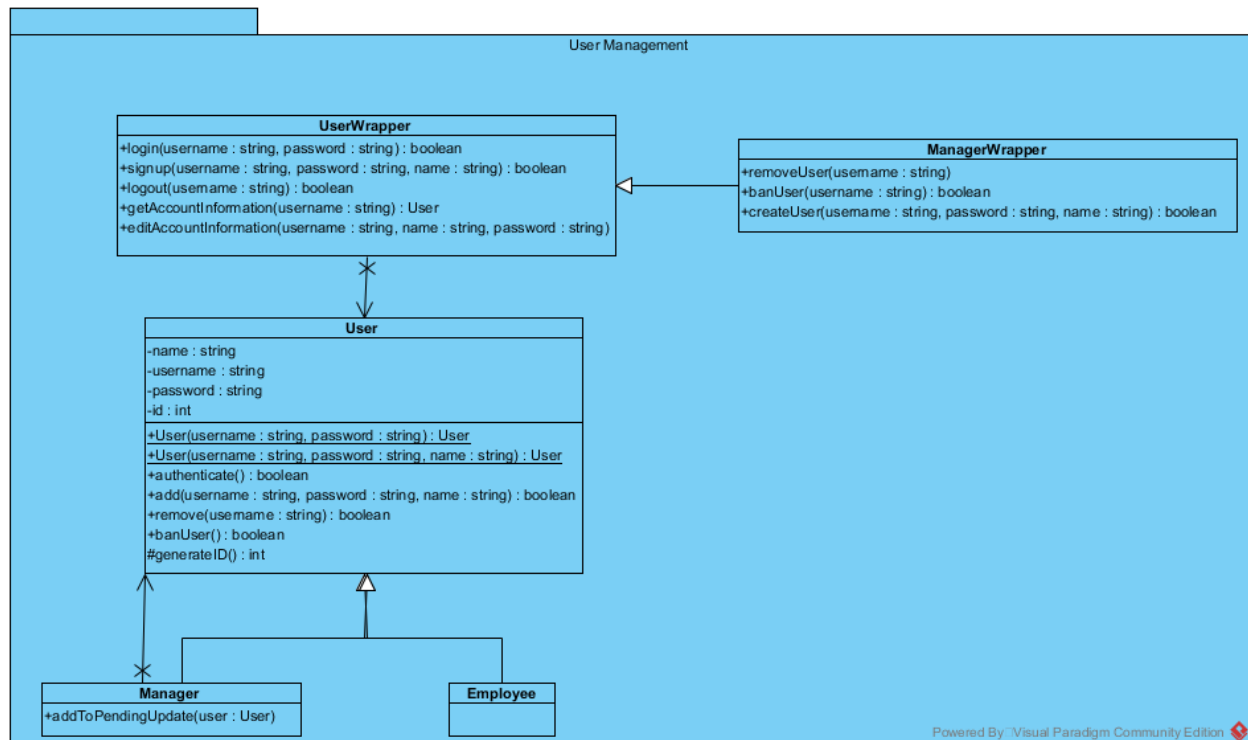
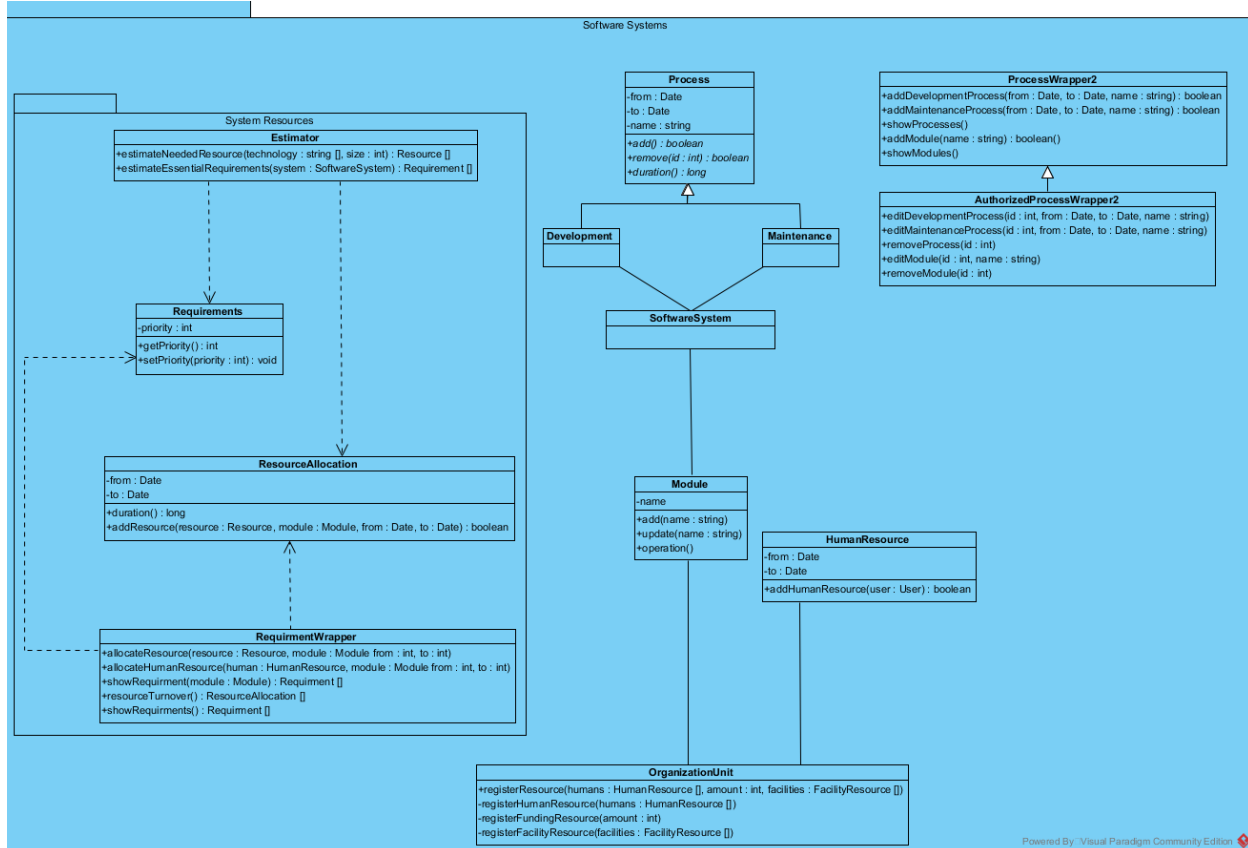
Powered By Visual Paradigm Community Edition

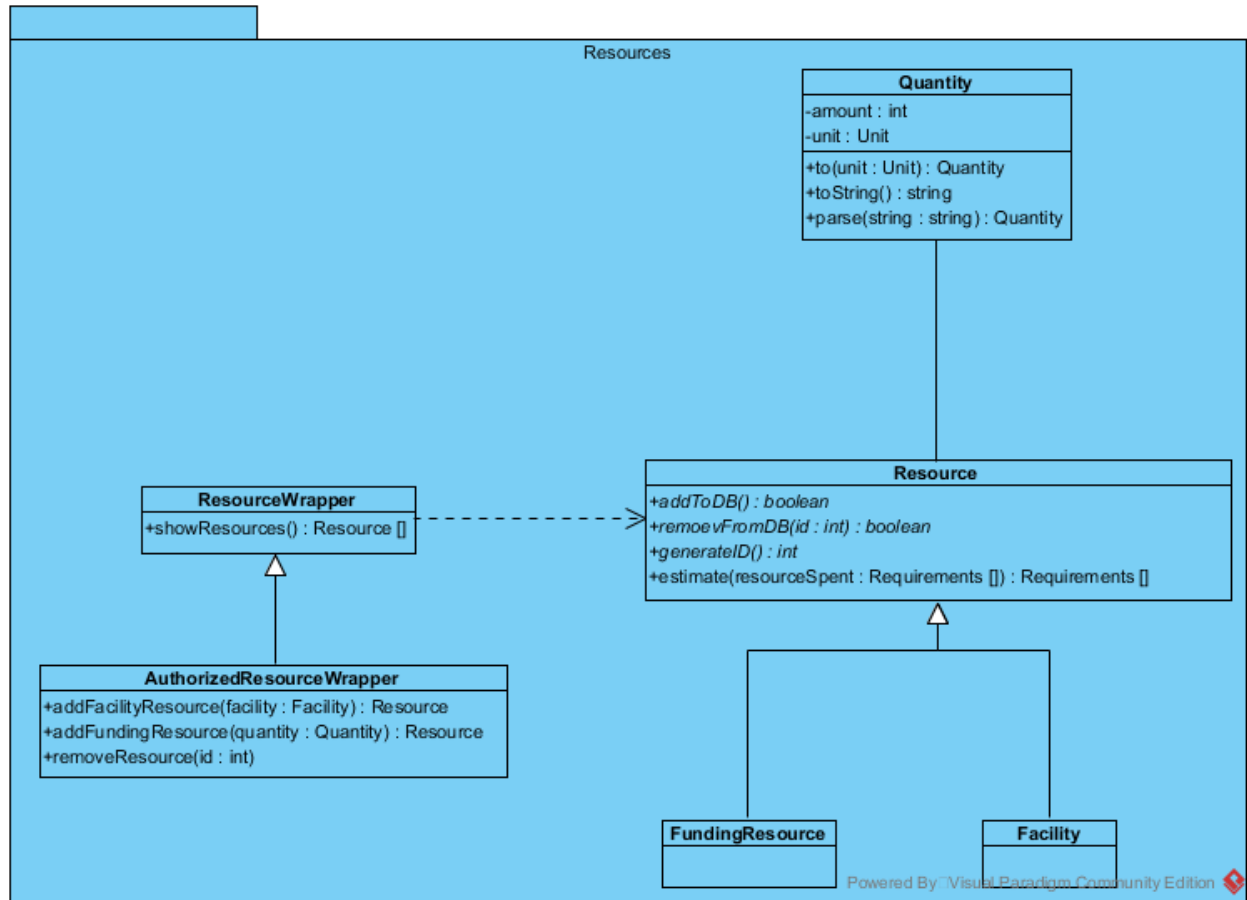
حذف فرآیند نگهداری:



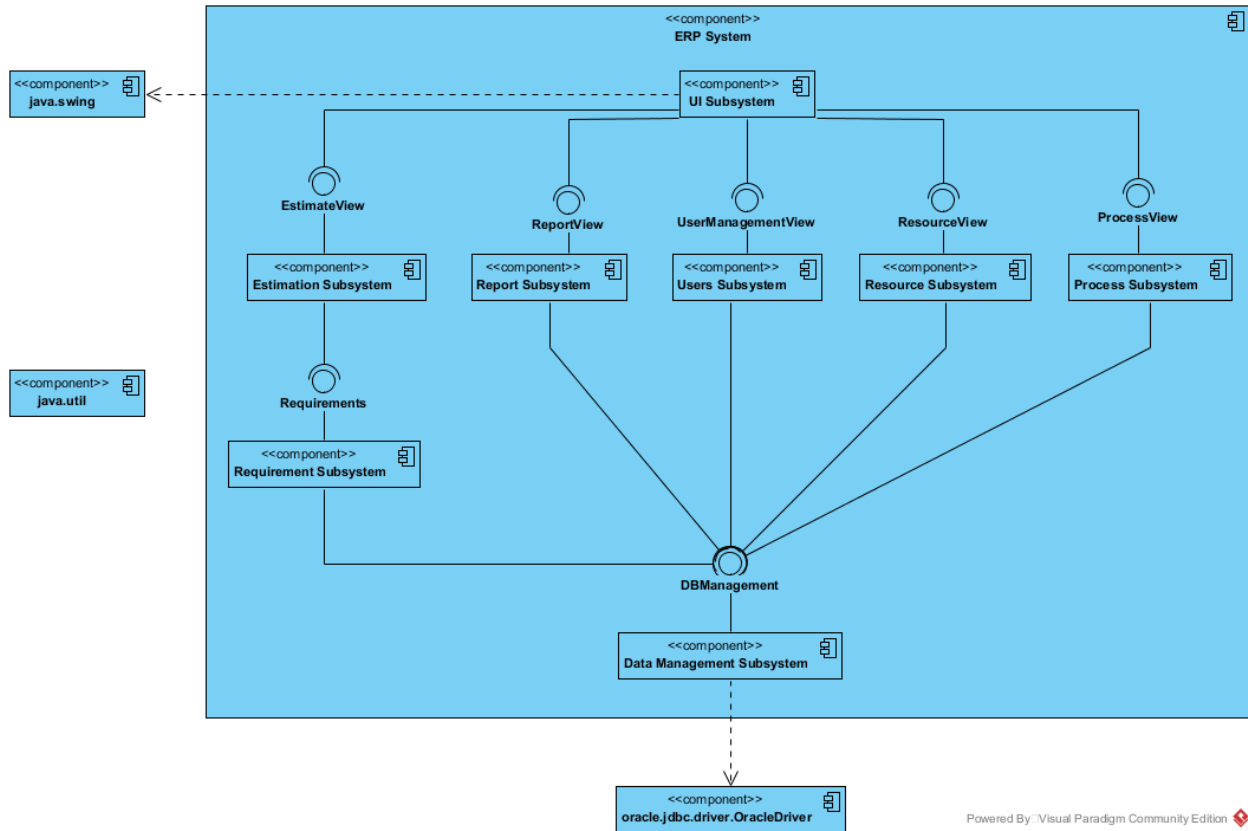
Powered By Visual Paradigm Community Edition

نمودارهای بسته



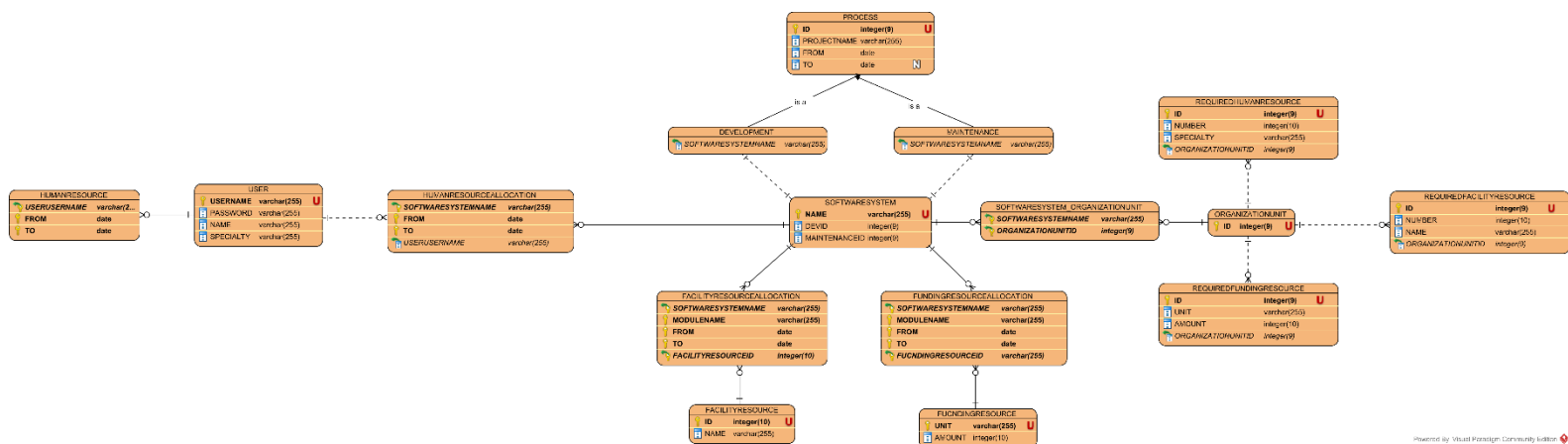


نمودار مؤلفه



Powered By: Visual Paradigm Community Edition

شمای پایگاه داده



Powered By: Visual Paradigm Community Edition

مستند الگوهای استفاده شده

در مورد Wrapper ها از الگوی Singleton استفاده شده است. زیرا این کلاس ها تنها به تعداد یک عدد در کل سامانه نیاز هستند. به طور مثال در مورد UserWrapper به شکل زیر است:

```
public class UserWrapper {  
    private static UserWrapper uniqueInstance;  
  
    private UserWrapper() {};  
  
    public static UserWrapper getInstance() {  
        if (uniqueInstance == null)  
            return (uniqueInstance = new UserWrapper());  
        else  
            return uniqueInstance;  
    }  
}
```

هم چنین Wrapper ها کاربرد الگوی Facade هستند. این کلاس ها اینترفیس واحدی برای زیرسیستم های مرتبط خود ارائه می دهند.

چک لیست استخراج شده

۱. در انتهای فاز construction قرار داریم که با توجه به اسلایدهای درس باید در حدود ۸۰ تا ۹۰ درصد از کد کامل باشد که خوشبختانه این هدف ارضا شده است.
۲. تحلیل به طور کامل پایان یافته و تغییری در نیازمندی‌ها و یا موارد کاربرد مشاهده نمی‌شود و محصول در آستانه‌ی تحویل اولیه است.
۳. تمام کدهای تولید شده بر اساس نمودار توالی بوده و تمام نمودارهای توالی بر اساس محصولات فازهای قبل قابل ردگیری هستند.
۴. در موارد مورد نیاز، از الگوهای طراحی مناسب استفاده شده است، مانند singleton.