

به نام خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر و فناوری اطلاعات



Enigma



نام گروه: Enigma

سرپرست تیم: حسن کاظمی طهرانی

نماینده مشتری: مهیار سلامت

سایر اعضای گروه:

سروش مهربان برآغوش

محمد مهدی منتظر

استاد درس: علی کمالی

نیم سال اول ۹۹-۱۳۹۸

صفحه	فهرست مطالب
۱	فصل اول فرضیات پروژه
۱-۱	۱-۱- توضیحات
۲	۱-۲- نمودار محتوا (Context Diagram)
۳	فصل دوم انتخاب متدلوژی و نحوه انجام کار
۳	۲-۱- نیازمندی های کلیدی
۳	۲-۲- سناریو تست پذیرش (Acceptance Test)
۴	۲-۳- انتخاب متدلوژی (یا فریمورک) و Process Model
۴	۲-۳-۱- انتخاب متدلوژی (یا فریمورک)
۶	۲-۳-۲- انتخاب Process Model
۱۵	فصل سوم آنالیز نیازمندی های مبتنی بر سناریو
۱۵	۳-۱- نمودار Use Case
۲۲	۳-۲- نمودار فعالیت (Activity Diagram) و Swim lane
۳۳	۳-۳- خروجی Technical Review
۳۴	فصل چهارم آنالیز نیازمندی ها جنبه ساختاری
۳۴	۴-۱- شناسایی کلاس های کاندید، دسته بندی و انتخاب نهایی
۳۵	۴-۲- رسم CRC Cards
۳۸	۴-۳- Class Diagram در سطح نیازمندی ها
۴۱	۴-۴- Sequence Diagram در سطح نیازمندی ها
۵۶	۴-۵- خروجی Technical Review
۵۷	فصل پنجم طراحی معماری
۵۷	۵-۱- انتخاب و استدلال استایل های کاندید برای معماری سیستم
۵۹	۵-۲- انتخاب استدلال الگوهای کاندید برای معماری سیستم
۶۰	۵-۳- ارائه حداقل دو طرح برای معماری سیستم در قالب
۷۰	۵-۴- خروجی Technical Review
۷۱	فصل ششم مدیریت پروژه
۷۱	۶-۱- انتخاب ابزار CD/CI
۷۲	۶-۲- تکمیل موارد متدلوژی چابک
۷۲	۶-۲-۱- Sprint Map
۷۳	۶-۲-۲- User Stories

۹۴.....	DoD-۶-۲-۳
۹۵.....	DoR-۶-۲-۴
۹۶.....	Sprint Review از یک جلسه

فصل اول

فرضیات پروژه

۱-۱- توضیحات

مختصر توضیحی در مورد پروژه:

این سیستم قرار است با به کارگیری روشی به کامپیوتری در نقطه ای دیگر به کمک شبکه (اینترنت یا شبکه داخلی) متصل شود و توانایی مانیتور کردن کامپیوتر هدف و کنترل کردن آن را داشته باشد که مناسب برای افرادی که نیاز به دسترسی به کامپیوتر خاصی از راه دور دارند می باشد مانند مدیران، پدر و مادر ها، مسئولین امنیت شبکه و ...

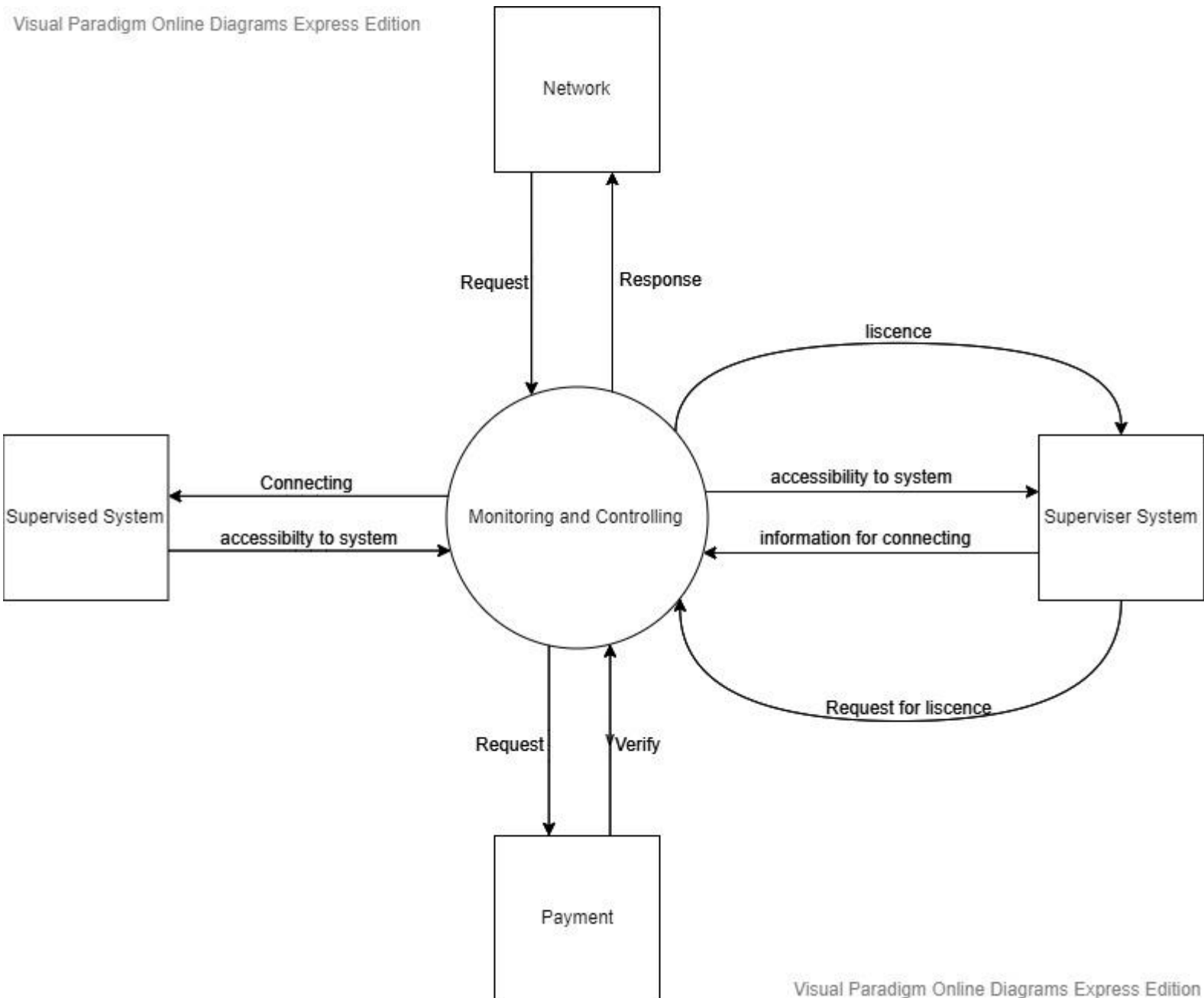
عنوان ذی نفعان اصلی:

۱. تیم enigma
۲. مدیران
۳. مسئولین امنیت شبکه شرکت ها

عنوان زیر سیستم های اصلی:

۱. اتصال به شبکه
۲. مانیتورینگ صفحه و قابلیت ها
۳. کنترل کننده دسکتاپ
۴. UI
۵. هماهنگ کننده بخش های مختلف سیستم

۱-۲ - نمودار محتوا (Context Diagram)



فصل دوم

انتخاب متدلوژی و نحوه انجام کار

۲-۱- نیازمندی های کلیدی

۱. اتصال به شبکه
۲. دریافت استریم ویدئویی از خروجی صفحه نمایش کامپیوتر هدف
۳. کنترل ورودی کامپیوتر هدف (ماوس و کیبورد)

۲-۲- سناریو تست پذیرش (Acceptance Test)

۱. **Scope**: کاربر وارد سیستم می شود و پس از وارد کردن IP Address کاربر بتواند به کاربر مقصد متصل شود و عبارت "Connected" چاپ شود.
Not scope: در صورت نداشتن permission متصل نشود و عبارت "Not Connected" چاپ شود.
۲. **Scope**: کاربر پس از اتصال به کاربر مقصد استریم صفحه نمایش کاربر مقصد را مشاهده می کند و با صفحه نمایش خروجی کاربر مقایسه صورت می پذیرد.
Not scope: فرمت استریم ارسالی در کامپیوتر مقصد قابل شناسایی باشد.
۳. **Scope**: کاربر درخواست کنترل ورودی را می کند و پس از فعال شدن می تواند ورودی کیبورد و ماوس کاربر مقصد را کنترل کند و آن را از صفحه نمایش استریم مشاهده کند.
Not scope: اگر permission نداشته باشیم، نتوان ورودی را کنترل کرد.

۲-۳- انتخاب متدلوژی (یا فریمورک) و Process Model

۲-۳-۱- انتخاب متدلوژی (یا فریمورک)

Methodology/ Framework Name	Scrum	XP	ASD	DSDM	Crystal	FDD
Team Size	5-9	2-10	Small to large teams	2-10	All Ranges	4-20
Project Size	All	Small	Small	All	All	All
Requirement gathering	Product Backlog	User Story	JAD	Functional Model Iteration	Use Case	Feature
Iteration length(week)	4	2	4-8	Not specified	Project specific	2
Team meetings	Informal daily stand up meeting	Informal daily stand up meeting	Face to face meetings	Info sharing through doc	Face to face meetings	Info sharing through doc
Documentation	Basic	Basic	Basic	Average Value	Varied	Important
End-user involvement	Project owners represent that	Actively involved	Participate in all of the incremental releases	Participate in all of the incremental releases	Participate in all of the incremental releases	Participate through reports
Usage	52%	2%	2%			

نتیجه گیری:

تیم مورد نظر ما تیم کوچکی است که تمام اعضا به طور full stack کار میکنند و از کار یکدیگر باخبرند همچنین ممکن است در اواسط پروژه به یک نیازمندی و تغییر جدید در پروژه برخوردیم لذا متدلوژی های ساخت یافته مناسب پروژه نیست.

با توجه به این که پروژه قابلیت توسعه و پیشرفت زیادی دارد و نرم افزار حاصل اندازه کوچکی ندارد و بایستی با توجه به شبکه محور بودن از امنیت بالایی برخوردار باشد، متدلوژی های ASD و XP با توجه به این که برای پروژه هایی با سایز کم مناسب اند، مناسب پروژه تعریف شده نیستند.

با توجه به این که تیم مورد نظر تیم کوچکی است، تلاش بر قابل فهم بودن کد و با خبر بودن اعضا از تمام مراحل می باشد و مستندات اهمیت چندانی ندارند (هر چند در سطح basic مورد نیازند). لذا متدلوژی FDD مناسب پروژه نمی باشد.

تلاش تیم پروژه بر ارتباطات رو در رو می باشد و با توجه به این که DSDM بر اساس مستندات ارتباطات را برقرار میکند، مناسب پروژه نمی باشد.

در آخر با توجه به محبوبیت بیشتر **فریم ورک Scrum** و جهت آشنایی بیشتر با این فریمورک و داشتن شرایط لازم برای توسعه پروژه، از این فریمورک جهت توسعه استفاده می شود.

۲-۳-۲- انتخاب Process Model

Model Name	Time Control	Change Acceptance	Security Management	Risk Management	Available Component Involvement	Limitation of cost	SUM
Code and Fix	0	2	0	0	0	0	2
The Waterfall	3	1	2	2	0	2	10
V-Model	2	0	3	3	0	3	11
Incremental	3	3	3	2	0	3	14
Prototyping	1	1	1	2	0	1	6
The Spiral	2	3	3	3	0	0	11
Concurrent	0	3	0	0	0	0	3
RAD	3	1	0	2	3	3	10
Component based	3	0	0	3	3	2	9
Reuse Oriented	2	1	0	2	2	2	9
Formal	0	0	1	0	0	0	1
Aspect Oriented	2	2	3	2	0	2	11
The Unified Process	1	1	3	3	0	2	10

معرفی معیار های جدول:

در این جدول تمامی Process Model های پایه در جنبه هایی که در مورد پروژه حائز اهمیت اند، مورد ارزیابی قرار میگیرند. برای هر خانه جدول عددی از ۰ تا ۳ اختصاص داده شده است که هر کدام نشان می دهند Process model مورد نظر تا چه حد آن جنبه مورد نظر را پوشش میدهد:

۰: پوشش نمیدهد

۱: ضعیف

۲: متوسط

۳: خوب

در ادامه هر کدام از جنبه هارا بررسی میکنیم.

Time Control:

با توجه به محدود بودن زمان ترم، نیاز به مدلی داریم که بتواند به خوبی زمان را کنترل کند و در زمان مطلوب تعیین شده بتواند پروژه را به اتمام برساند

- **Code and Fix**: از کنترل این مدل خارج است و میتواند در یک حلقه مدام تکرار شود و به کیفیت مطلوب نرسد

- **The Waterfall**: در این مدل همه چیز از ابتدا به طور کامل مشخص است و زمان به بهترین شکل کنترل میگردد.

- **V-model**: مطابق با Waterfall خطی میرود اما اگر تعداد تست ها زیاد شود میتواند مدیریت زمان به هم بخورد، لذا به طور متوسط کنترل میکند

- **Incremental**: با برنامه ریزی دقیق میتوان increment هارا مشخص کرد تا در زمان مطلوب همگی به پایان برسند

- **Prototyping**: با توجه به ایجاد مکرر prototype باعث میشود روند کندی داشته باشیم و پوشش این جنبه به شدت ضعیف است

- **The Spiral**: schedule کردن در مرحله planning تا حد قابل قبولی مناسب است اما مشکل تخمین زمان دارد. چرا که جز Evolutionary ها می باشد و این چرخش میتواند مدام تکرار شود
- **Concurrent**: مشخص نشده است
- **RAD**: به دلیل استفاده از component های آماده سرعت بسیار بالایی دارد
- **Component based**: به دلیل استفاده از component های آماده سرعت بسیار بالایی دارد
- **Reuse Oriented**: بخشی از کار که با component آماده ساخته میشود سرعت بالا، اما بخش دیگر سرعت کمی دارد.
- **Formal**: چون تمام مراحل باید به زبان ریاضی بیان شود بسیار سرعت پایین است
- **Aspect oriented**: به دلیل دسته بندی کردن concern ها، سرعت جواب دادن به آنها هم بالا میرود
- **The Unified Process**: در این مدل به دلیل داشتن مستندات کامل کمی سرعت پایین می آید

Change Acceptance

- ممکن است در مراحل توسعه نرم افزار دچار تغییر شویم، این که هرکدام از مدل های فرایند چه واکنشی نشان میدهند و آیا میتوانند با هزینه کم قبول کنند یا خیر، مورد اهمیت است
- **Code and Fix**: اهمیتی به تغییرات نمیدهد و به طور متوسط به کنترل آن می پردازد.
 - **The Waterfall**: عدم انعطاف پذیری جز معایب این مدل است و سبب هزینه زیاد می شود. تنها در مدل iterative آن پس از اتمام میتوان بازگشت
 - **V-model**: با توجه به خطی بودن به تغییرات پاسخگو نیست
 - **Incremental**: انعطاف پذیری از ویژگی های این مدل است و می توان increment هارا موازی انجام داد و به تغییرات نیز به خوبی پاسخ داد

- **Prototyping**: کاربر با دیدن prototype مدام ممکن است از تیم توسعه تغییرات بخواهد و این امر ریتم تیم توسعه را مختل می سازد
- **The Spiral**: انعطاف پذیری در مورد نیازمندی و تغییرات جز ویژگی این مدل است و به خوبی پوشش میدهد
- **Concurrent**: تغییرات با به دیگران اطلاع می دهد و به خوبی پوشش می دهد
- **RAD**: به دلیل سرعت بالا در نمایش محصول به مشتری باعث افزایش تغییرات خواهد شد و این امر ریتم تیم توسعه را مختل می سازد
- **Component based**: به دلیل باز نبودن دست برنامه نویس در استفاده از component ها، تغییرات کمی را میتوان پاسخ داد.
- **Reuse Oriented**: بخش component آن تغییرات کم، و بخش دیگر متوسطی را پوشش میدهد.
- **Formal**: هر گونه تغییری باید به زبان ریاضی در بیاید و سپس تمام پروژه را عوض کند که بسیار هزینه بر است
- **Aspect oriented**: مانند incremental، به دلیل دسته بندی مناسب تغییرات هم به خوبی پاسخ داده میشوند
- **The Unified Process**: به دلیل حضور دائم مشتری در مراحل پیشرفت، تغییرات زیاد خواهند بود و این باعث کند شدن پیشرفت هم میشود

Security Management

- با توجه به موضوع پروژه، بحث امنیت بسیار مورد اهمیت است، لذا این که مدل فرایند ما بتواند آن را به خوبی کنترل کند را بایستی بررسی کنیم
- **Code and Fix**: از هیچ نوع اصولی به طور کلی جهت امنیت استفاده نمیکند و پوشش نمیدهد

- **The Waterfall** : با توجه به معلوم بودن نیازمندی ها از ابتدا، اصول امنیتی برای توسعه نرم افزار انتظار می رود که رعایت شود.
- **V-model**: مطابق با Waterfall اما با ارزیابی و تست کردن بهتر میتواند امنیت را کنترل کند
- **Incremental**: در هر increment مرحله V and V موجود است و با تست آن ها میتوان امنیت را به خوبی تامین کرد
- **Prototyping**: با شناسایی نقاط گنگ تا حد کمی پوشش میدهد اما هیچ نکته امنیتی برای نرم افزار تحت شبکه را نمیتواند گوش زد کند
- **The Spiral**: با تحلیل شدید ریسک ها و تعیین قواعد سبب میشود به خوبی این جنبه را پوشش دهد
- **Concurrent**: مشخص نشده است
- **RAD**: به دلیل استفاده از component های آماده امنیت بسیار پایین است
- **Component based**: به دلیل استفاده از component های آماده امنیت بسیار پایین است
- **Reuse Oriented**: به دلیل استفاده از component های آماده امنیت بسیار پایین است. و راجع به بخش دیگر آن اطلاعاتی نمی دهد
- **Formal**: چون همه چیز کاملاً مشخص است تا حدی امنیت را بالا میبرد
- **Aspect oriented**: اگر یکی از concern های ما در پروژه امنیت باشد، به خوبی این روش آن را مهیا میکند
- **The Unified Process**: به دلیل داشتن تست های زیاد، از امنیت نسبتاً بالایی برخوردار است

Risk Management

انتخاب راه حل مناسب در میان راه حل ها به طوری که هزینه را کاهش دهد و به کمبود بودجه در پروژه اهمیت بدهد حائز اهمیت است

- **Code and Fix**: توجهی به این اصل ندارد و میتواند به قدری طول بکشد که هزینه به شدت افزایش یابد

- **The Waterfall**: این مدل تمامی خطر ها را در ابتدا مشخص میکند و انتظار میرود دچار ریسک در اواسط پروژه نشویم (مدیریت متوسط)

- **V-model**: همانند Waterfall ولی با تست های مکرر سبب میشود از ریسک های پروژه بهتر آگاه شویم

- **Incremental**: خطاها اگر در اواسط کار پیش بیاید یک increment را تحت نظر قرار میدهد که از نکات مثبت است. اما اگر معماری در ابتدای کار اشتباه انجام گیرد کل increment ها و لذا کل پروژه بر باد میرود.

- **Prototyping**: کاهش زود هنگام خطاها در مرحله design و requirement gathering از ویژگی های این مدل است اما خطا های اواسط توسعه نرم افزار را پاسخگو نیست

- **The Spiral**: مدیریت ریسک از ویژگی های اصلی این مدل است

- **Concurrent**: مشخص نشده است

- **RAD**: چون سرعت رسیدن به محصول نهایی بالا است باعث کاهش ریسک میشود

- **Component based**: به دلیل مشخص بودن مستندات از قبل، ریسک پذیری بالایی دارد

- **Reuse Oriented**: به دلیل استفاده از component ها در بخشی از آن، ریسک پذیری متوسطی دارد.

- **Formal**: همه چیز مشخص و به زبان ریاضی نوشته شده است. پس ریسک بسیار کمی دارد

- **Aspect oriented**: اگر این خطرات به core concern ها وارد شود ضعیف، اما در cross-cutting concerns خوب عمل میکند

- **The Unified Process**: به دلیل تست ها و مستندات زیاد ریسک پذیری زیادی دارد

Available Component Management

با توجه به محدودیت زمان نیاز داریم برخی از کامپوننت ها مثل Login کردن و یا stream کردن ویدیو را از Component های آماده استفاده کنیم تا هم مشکل امنیتی نداشته باشد (چون افراد زیادی هستند ازش استفاده کردند و اگر داشت تا الان معلوم میشد) و هم به روند توسعه کمک کند

- **Code and Fix**: مشخص نشده است

- **The Waterfall**: مشخص نشده است

- **V-model**: مشخص نشده است

- **Incremental**: مشخص نشده است

- **Prototyping**: مشخص نشده است

- **The Spiral**: مشخص نشده است

- **Concurrent**: مشخص نشده است

- **RAD**: به دلیل ماهیت این روش، استفاده از component ها بسیار بالاست

- **Component based**: : به دلیل ماهیت این روش، استفاده از component ها بسیار بالاست

- **Reuse Oriented**: چون بخشی از آن با component های آماده اجرا میشود، متوسط است

- **Formal**: مشخص نشده است

- **Aspect oriented**: مشخص نشده است

- **The Unified Process**: مشخص نشده است

Limitation of Cost

با توجه به محدودیت بودجه، انتخاب مدلی که هزینه زیادی نداشته باشد حائز اهمیت است

- **Code and Fix**: میتواند هزینه زیاد و کنترل نشده ای را به خرج بیاورد و کنترلی ندارد
- **The Waterfall**: برنامه ریزی در همان ابتدا، هزینه ها را تا حد مقبولی کنترل میکند اما تسلط کامل ندارد و اتفاقات ناخواسته می تواند بیافتد.
- **V-model**: با انجام تست ها موجب می شود خطاها را در مرحله بعد از release متوجه نشویم که هزینه زیادی را برایمان به همراه بیاورد.
- **Incremental**: همانطور که اشاره شد با کاهش خطا هزینه کمتری به خرج میدهم و با برنامه ریزی دقیق میتوان به خوبی این جنبه را پوشش داد
- **Prototyping**: ساخت مکرر prototype تنها برای نشان دادن به کاربر سبب افزایش هزینه می شود و تا حد کمی میتوان کنترل کرد
- **The Spiral**: با توجه به تحلیل ریسک در هر مرحله هزینه زیادی را میطلبد و این جنبه را مورد پوشش قرار نمیدهد
- **Concurrent**: مشخص نشده است
- **RAD**: به دلیل وجود component های رایگان (حداقل در ایران) از هزینه های زیادی چشم پوشی میشود
- **Component based**: به دلیل وجود component های رایگان (حداقل در ایران) از هزینه های زیادی چشم پوشی میشود
- **Reuse Oriented**: component های آماده تا حدی از هزینه های پروژه را کم میکند.
- **Formal**: چون سرعت و ریسک پذیری این روش کم است، هزینه بالاست
- **Aspect oriented**: چون سرعت و ریسک پذیری این روش نسبتا خوب است، تا حدی هزینه ها کم میشود

• **The Unified Process**: کم بودن خطاها باعث کم شدن هزینه میشود اما سرعت

نسبتاً پایین میتواند هزینه را زیاد کند

نتیجه گیری:

با توجه به بررسی های انجام شده، بهترین Process model ای که سبب پوشش بخش های مهم پروژه میباشد، **Incremental** می باشد. اما یکی از جنبه های

مهم پروژه، بخش استفاده از component های آماده نظیر login, video stream و ... می باشد که Incremental روندی برای آن مشخص نکرده است. لذا

توصیه می شود از **Component based** در کنار آن برای این بخش ها استفاده گردد.

فصل سوم

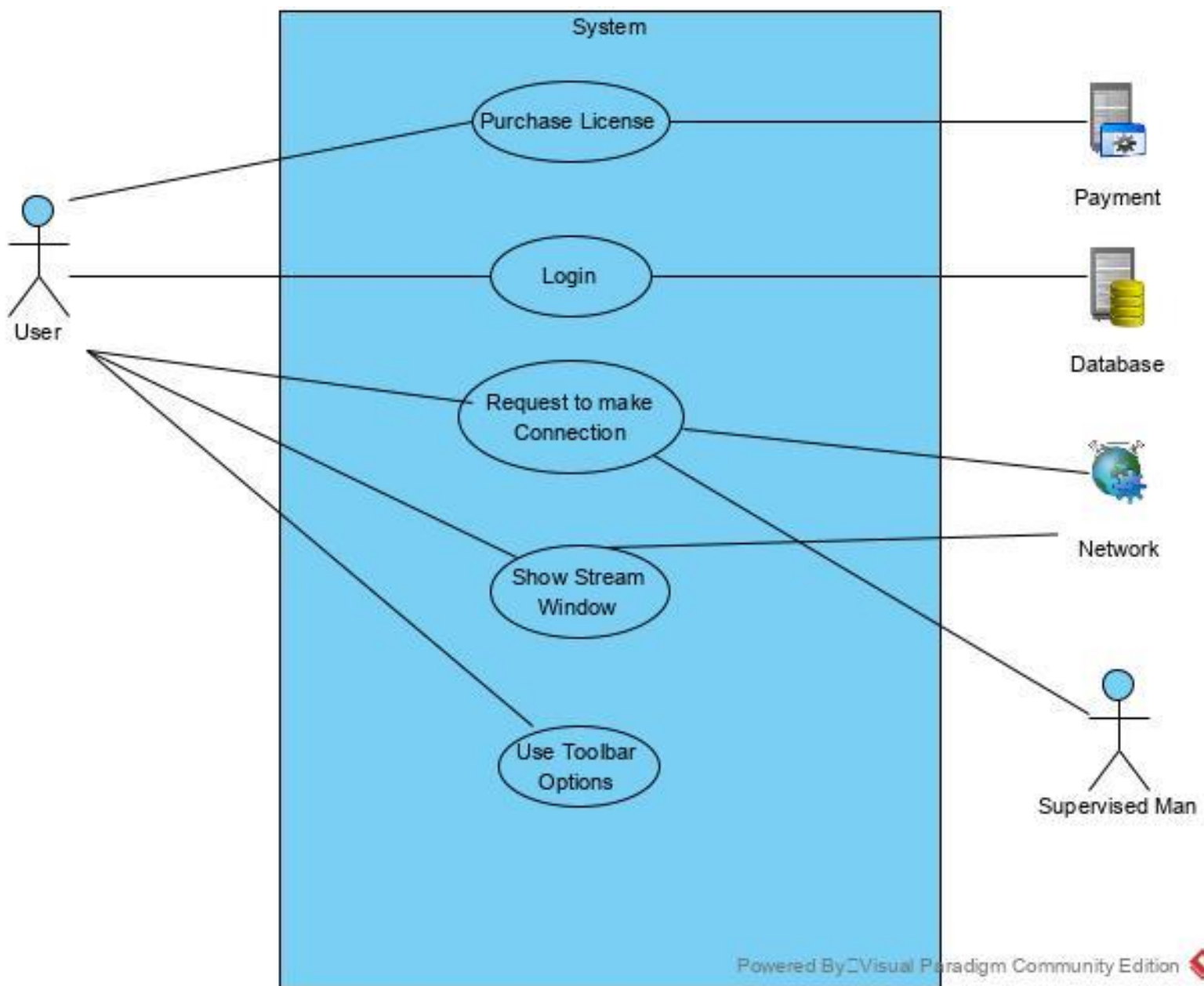
آنالیز نیازمندی های مبتنی بر سناریو

۱-۳- نمودار Use Case

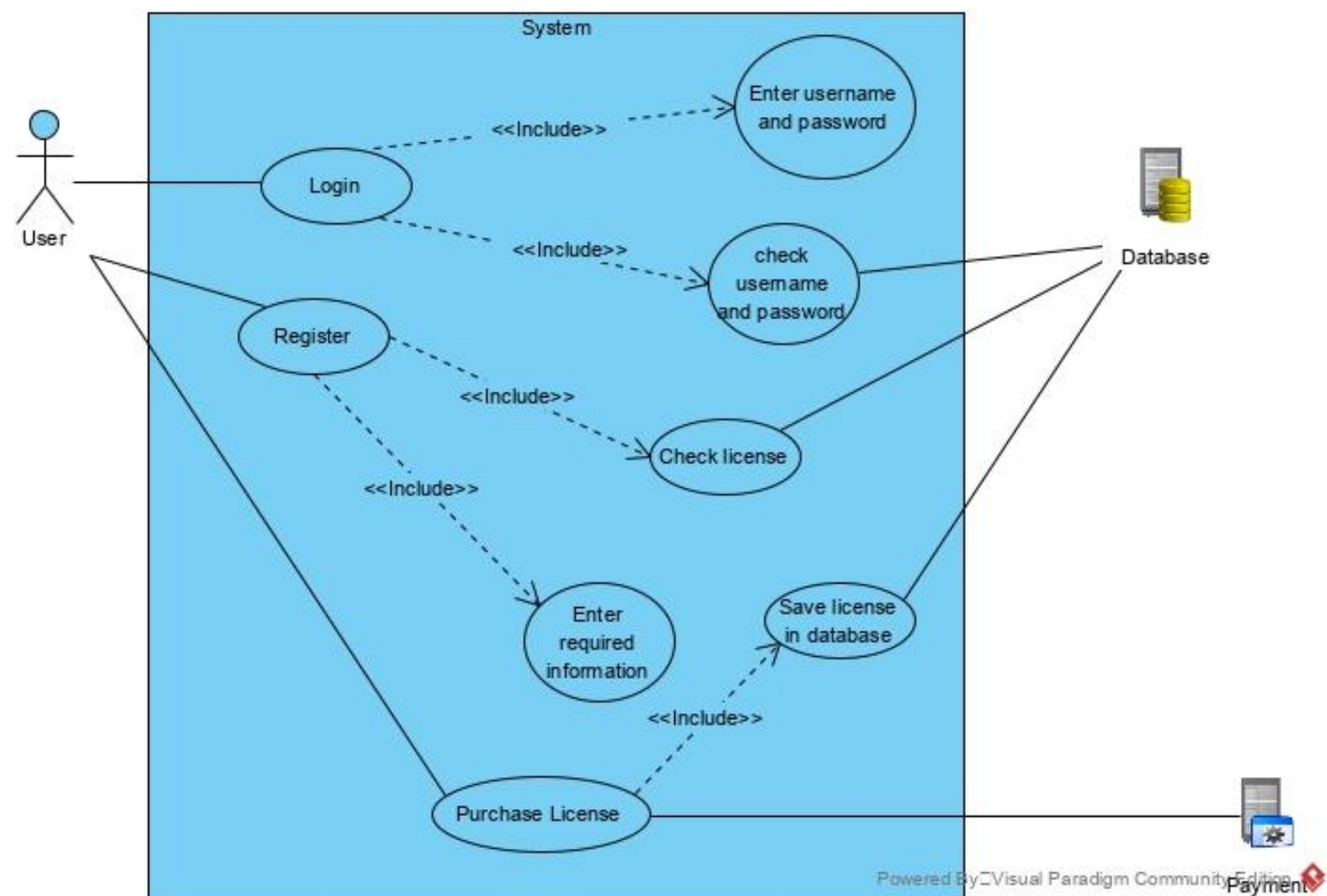
برای رسم Use Case Diagram، پروژه مورد نظر را در سه سطح در نظر گرفته شده و نمودارهای متناسب با آن ها ترسیم شده است:

- **سطح کلی:** تنها Use Case های کلی اشاره شده است و به جزئیات اشاره ای نشده است
- **سطح متوسط:** یک سطح پایین تر به بررسی Use Case های معرفی شده در سطح کلی می پردازیم
- **سطح پایین:** به Use Case هایی که نیاز به توضیح بیشتر دارند اشاره میکنیم

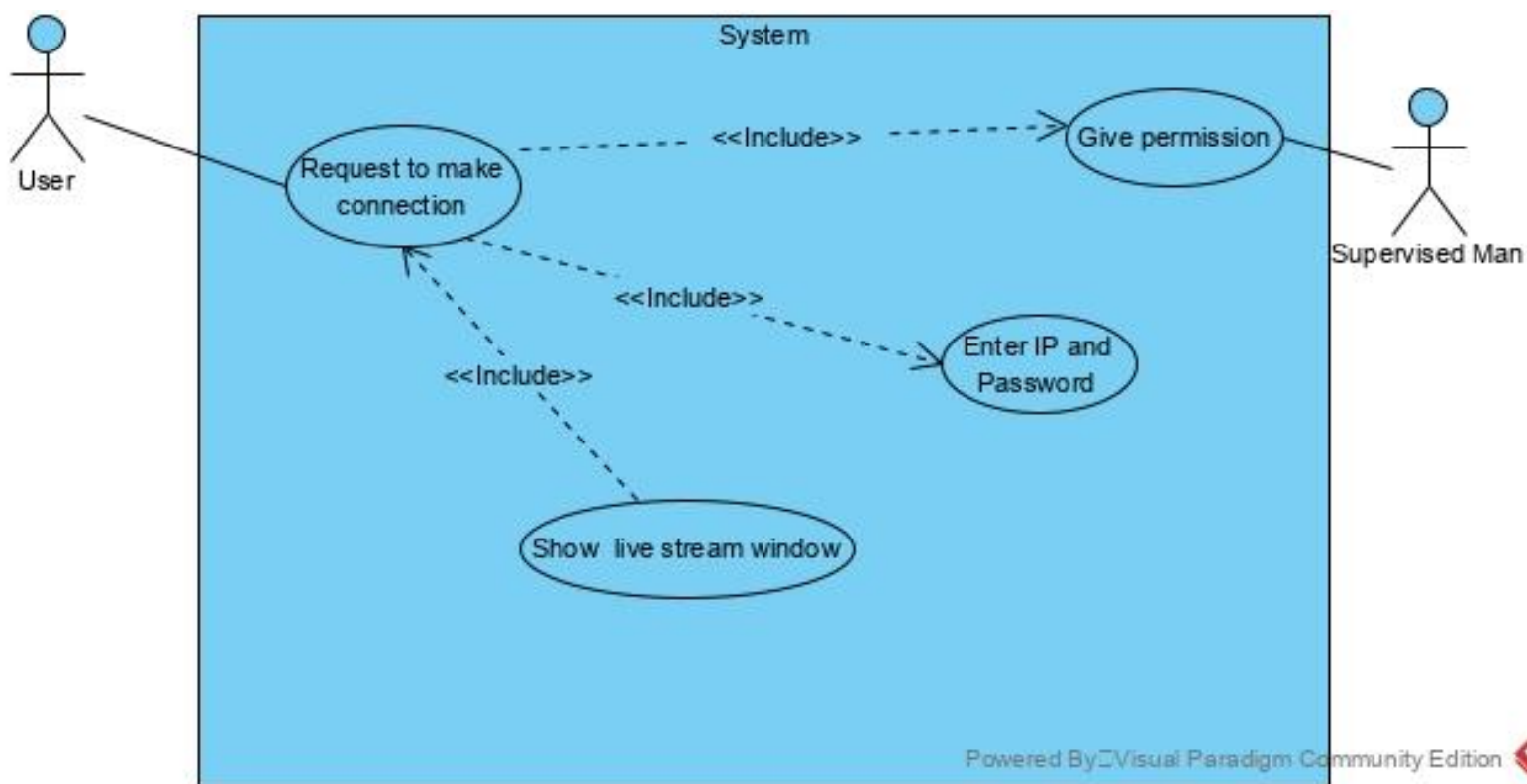
نمودار در سطح کلی:



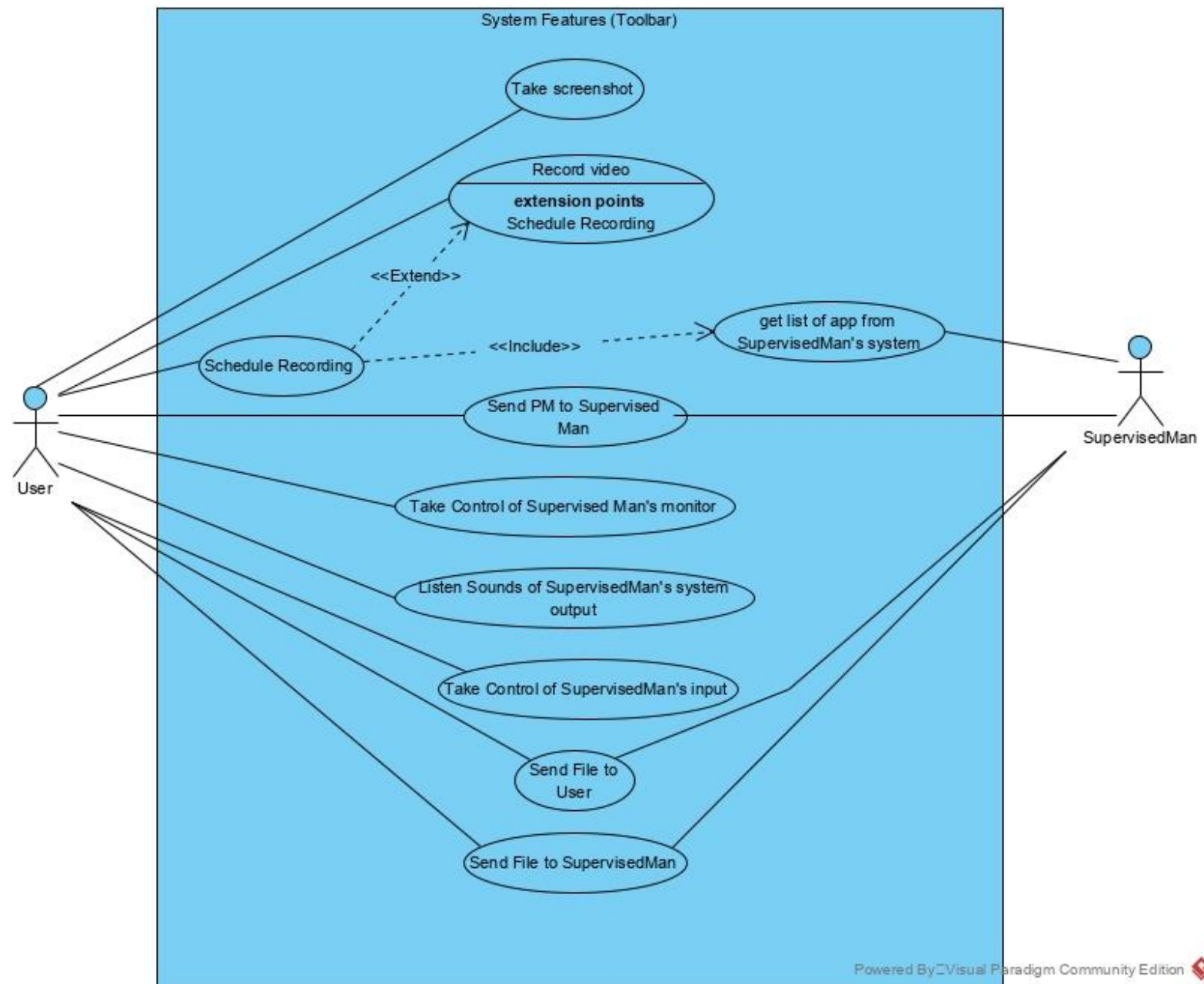
نمودار ورود و عضویت با خرید لایسنس - سطح متوسط (us 19,us 20,us 21):



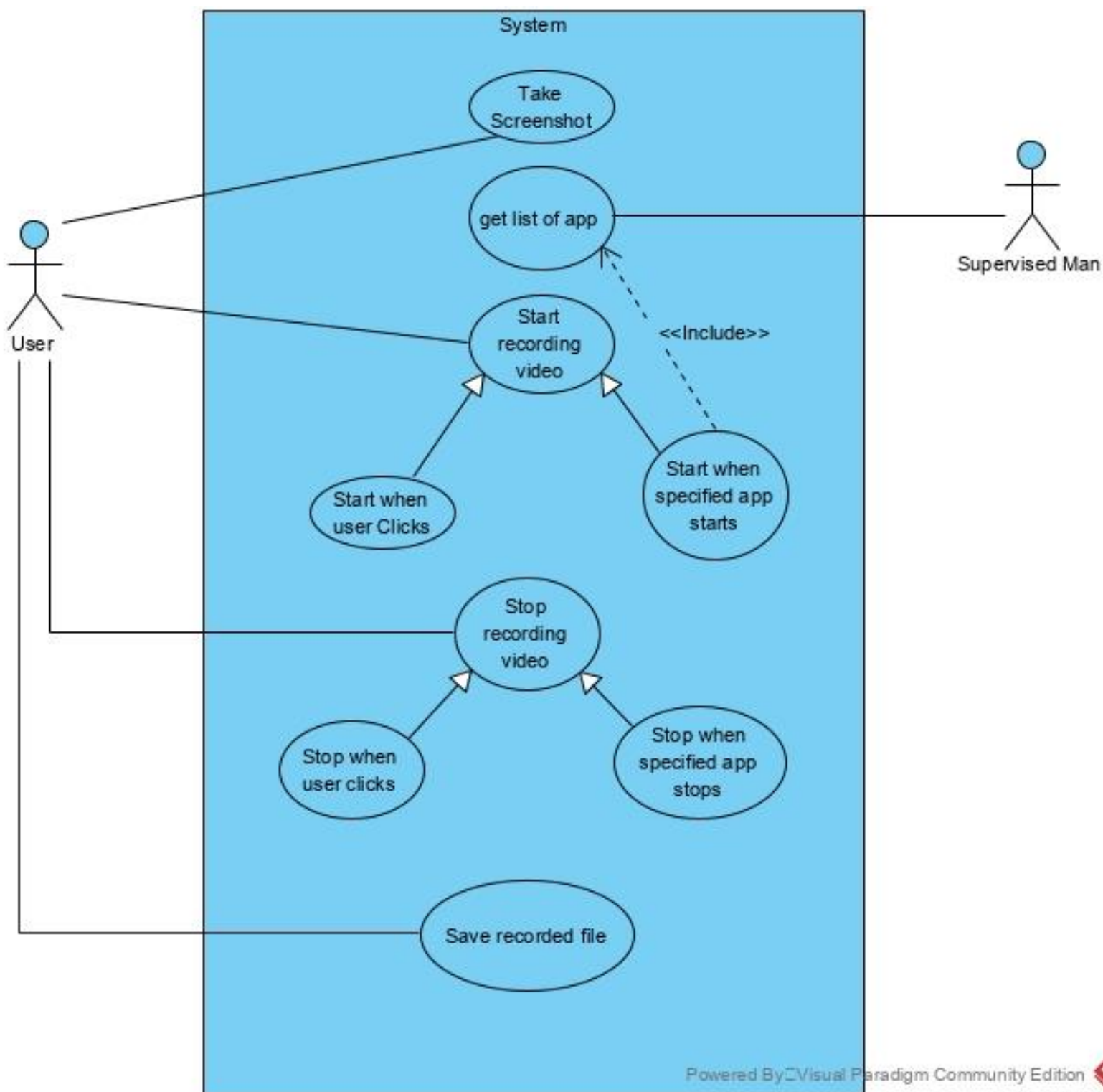
نمودار درخواست اتصال و نمایش Stream - سطح متوسط (us 4,us 5,us 6) :



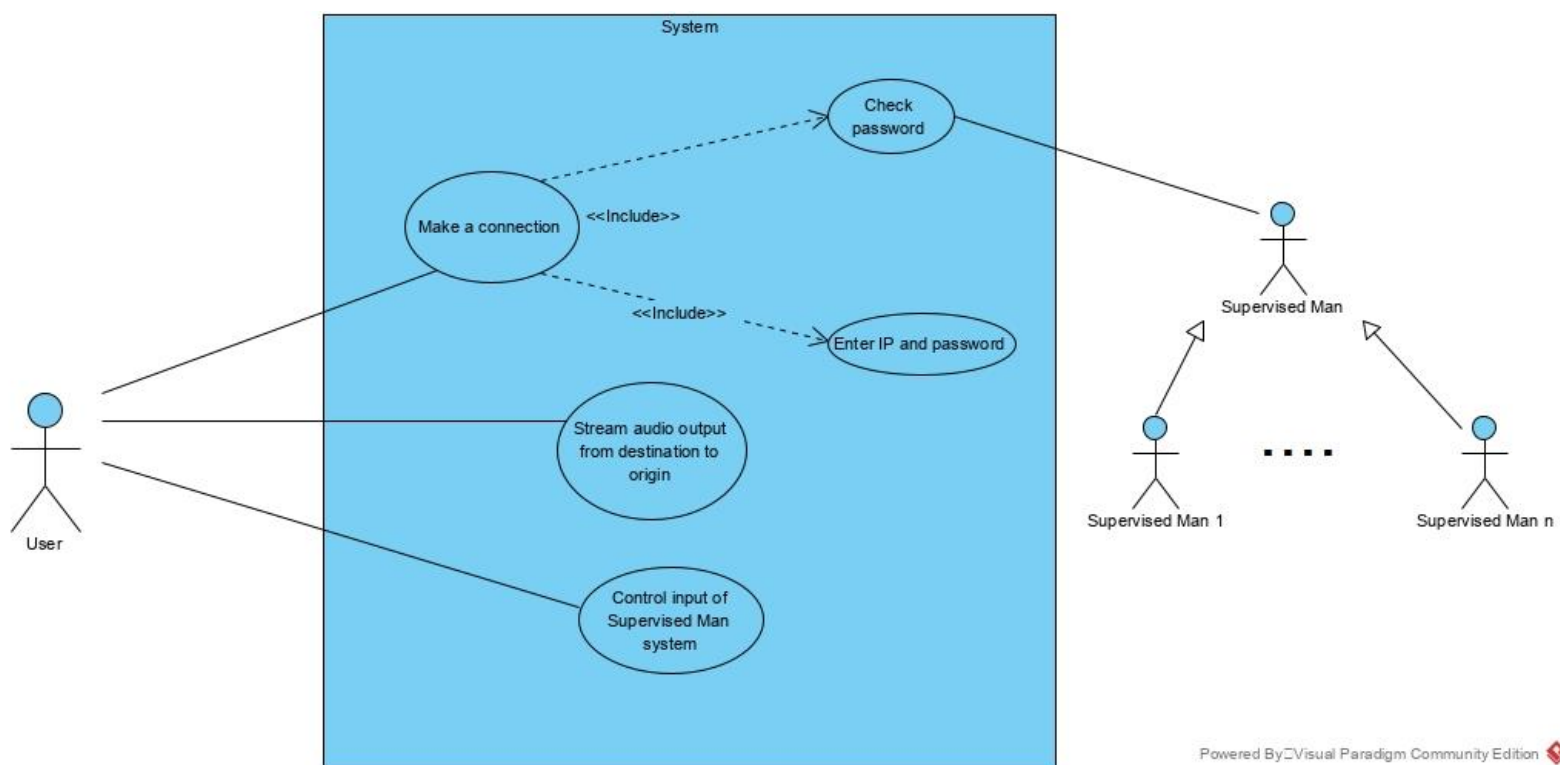
نمودار استفاده از ابزار ها (Toolbar Options) - سطح متوسط (از [us 7](#) تا [us 17](#)):



نمودار استفاده از Screenshot و Video Recording - سطح پایین (us 7, us 8, us 16, us 17) :



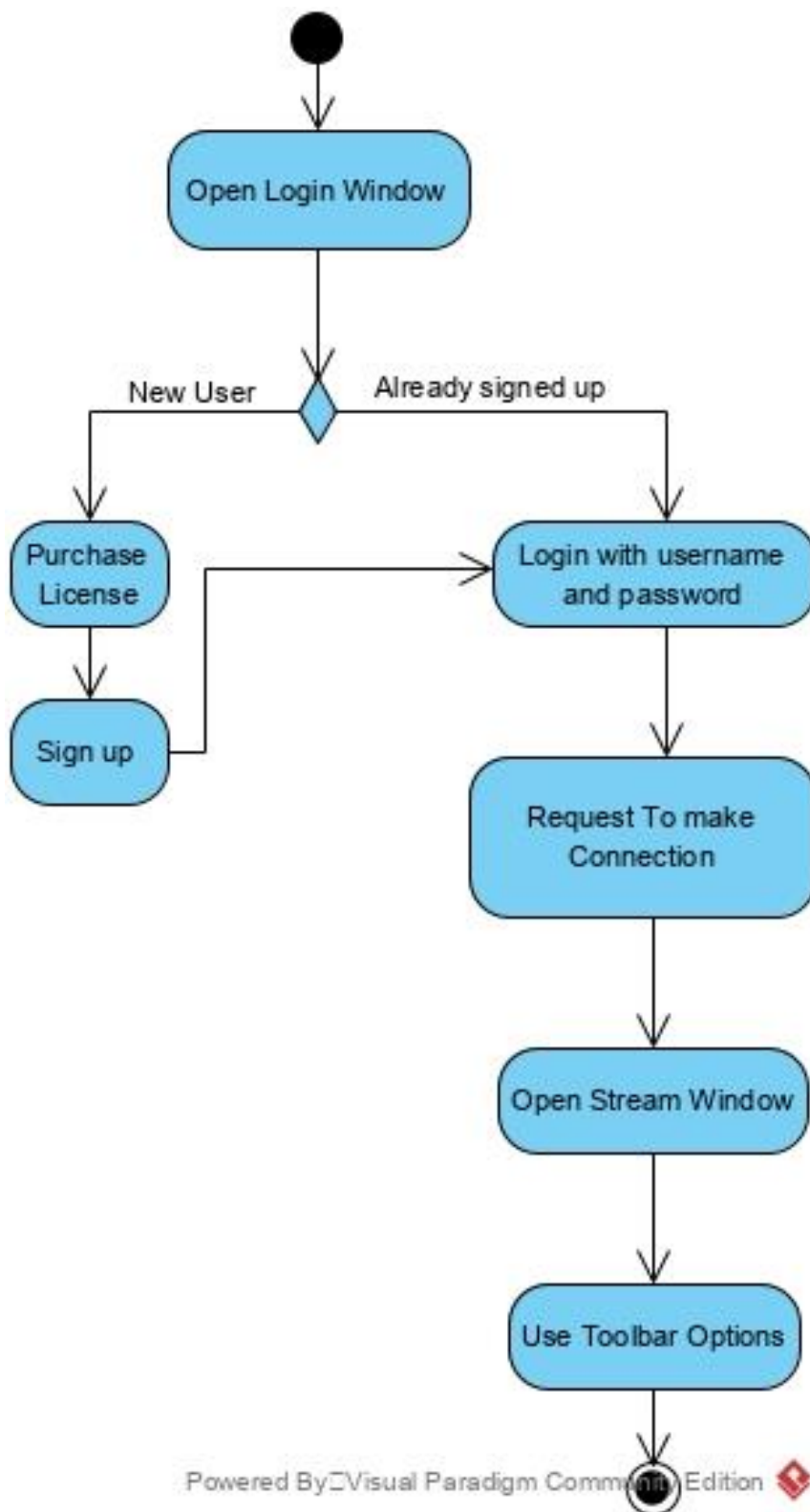
نمودار استفاده از **Audio output** و استفاده از کاربر مقصد-سطح پایین (هنگام اتصال به چندکاربر) ([us 12](#),[us 13](#),[us 14](#)):



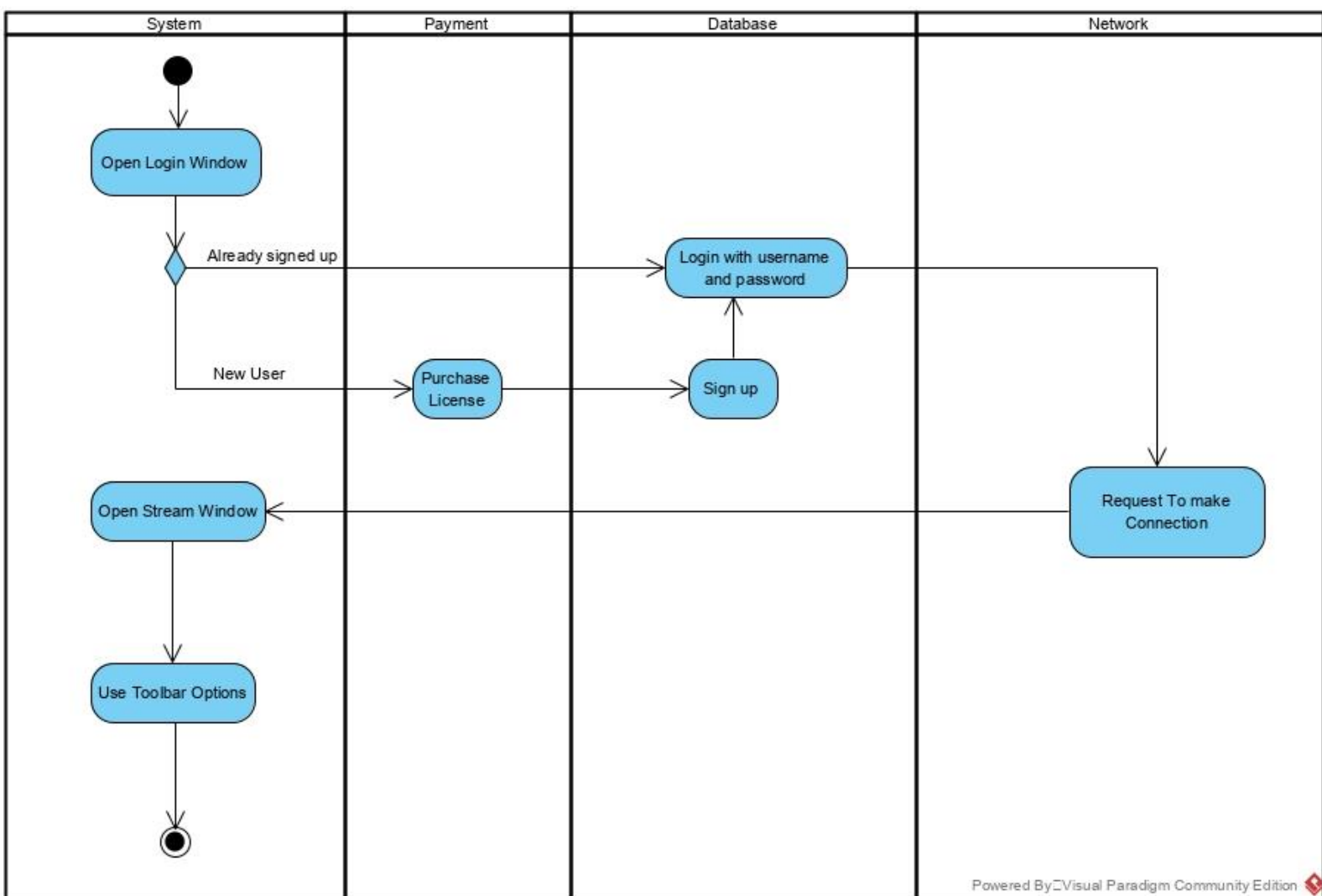
۳-۲- نمودار فعالیت (Activity Diagram) و Swim lane

نمودار در سطح کلی:

Activity Diagram:

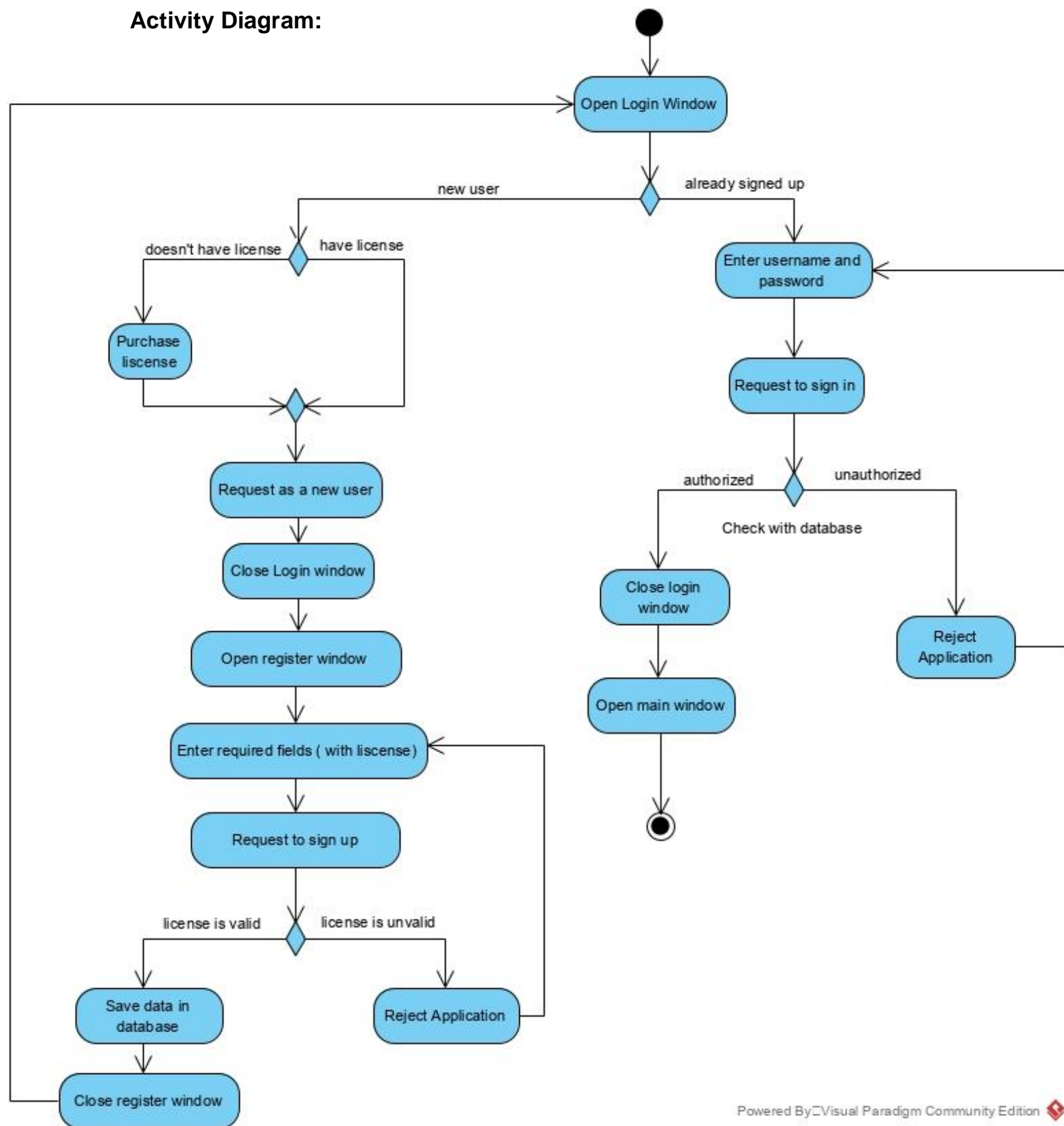


Swimlane Diagram:

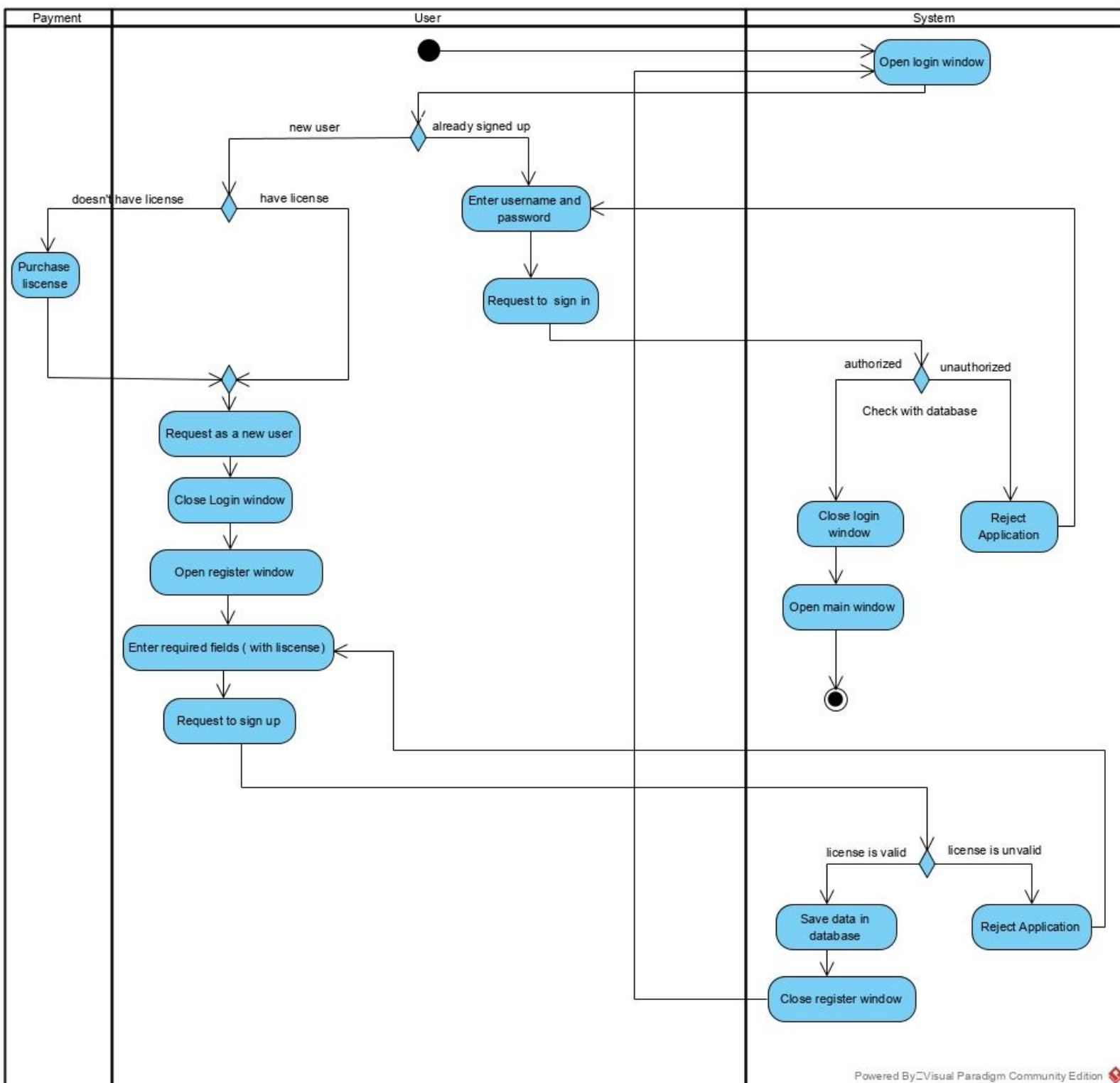


نمودار ورود و عضویت با خرید لایسنس (us 19,us 20,us 21):

Activity Diagram:

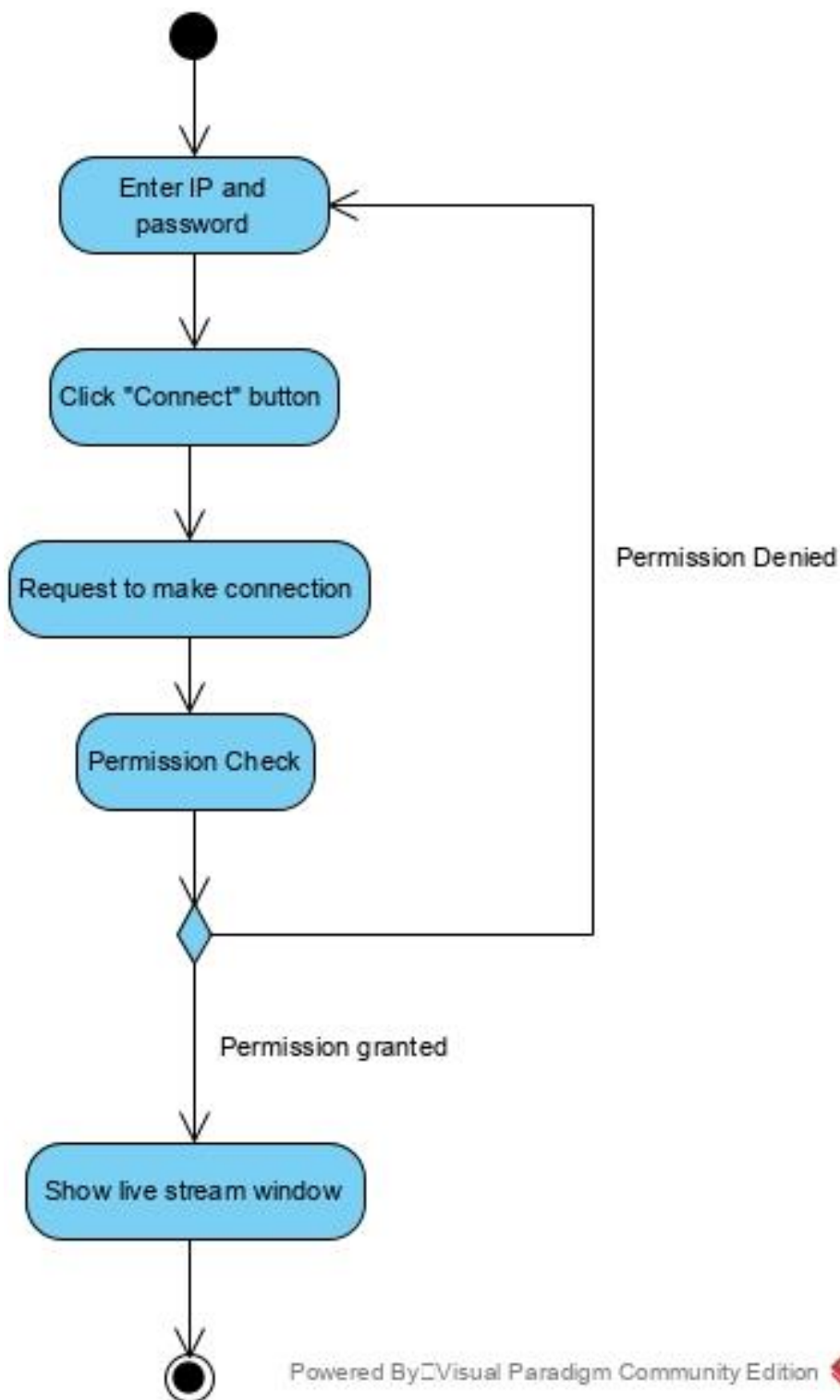


Swimlane Diagram:

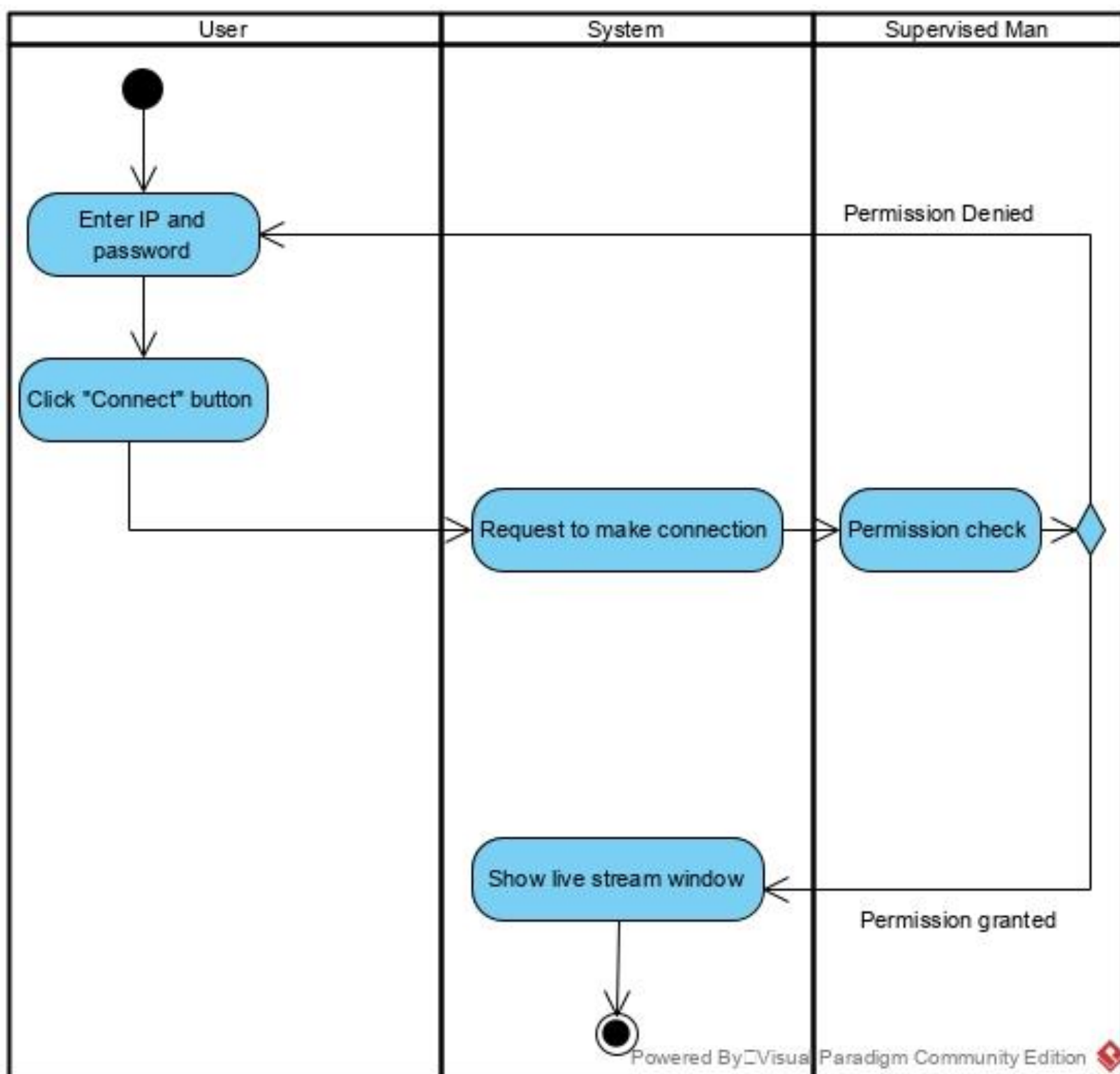


نمودار درخواست اتصال و نمایش Stream (us 4,us 5,us 6) :

Activity Diagram:

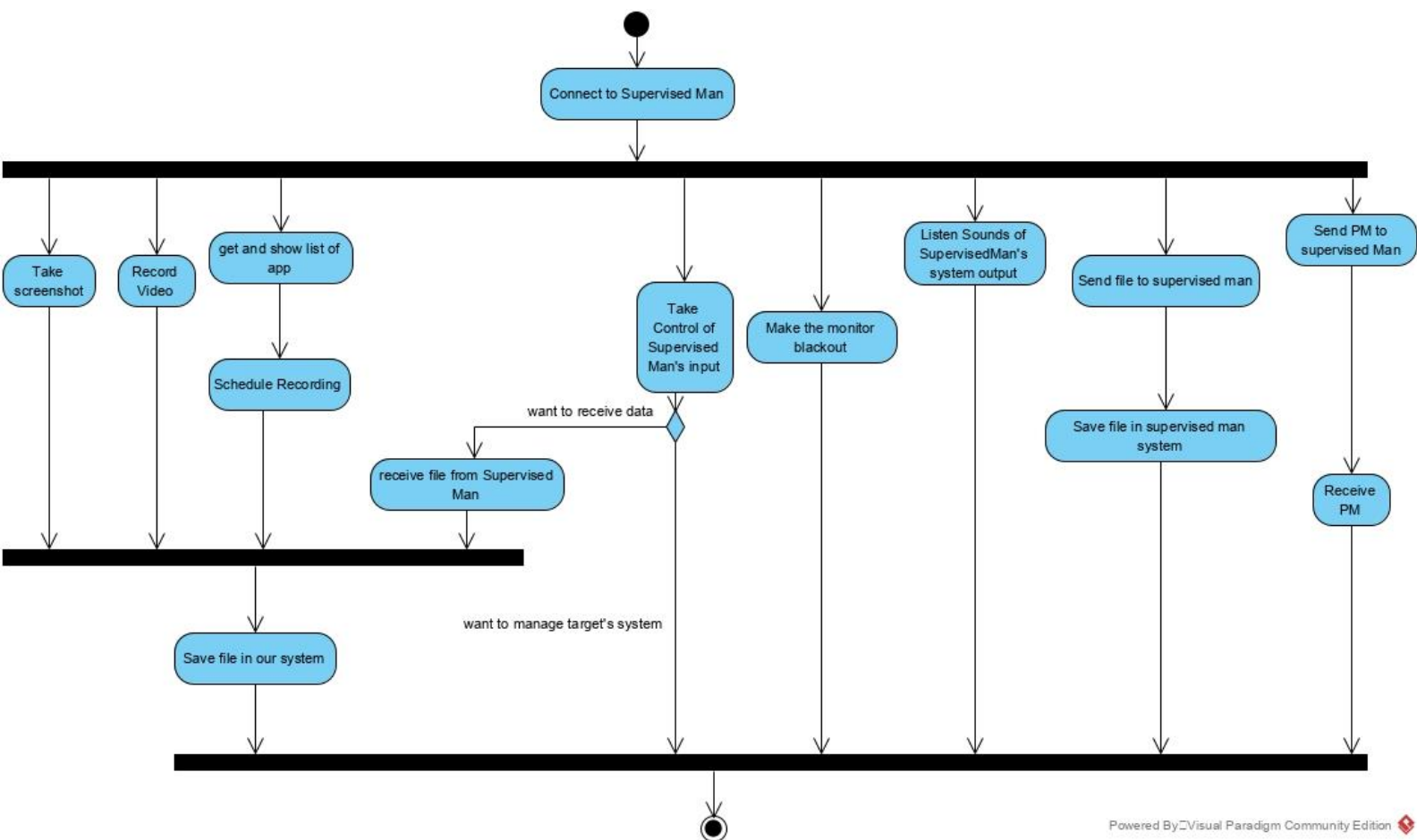


Swimlane Diagram:

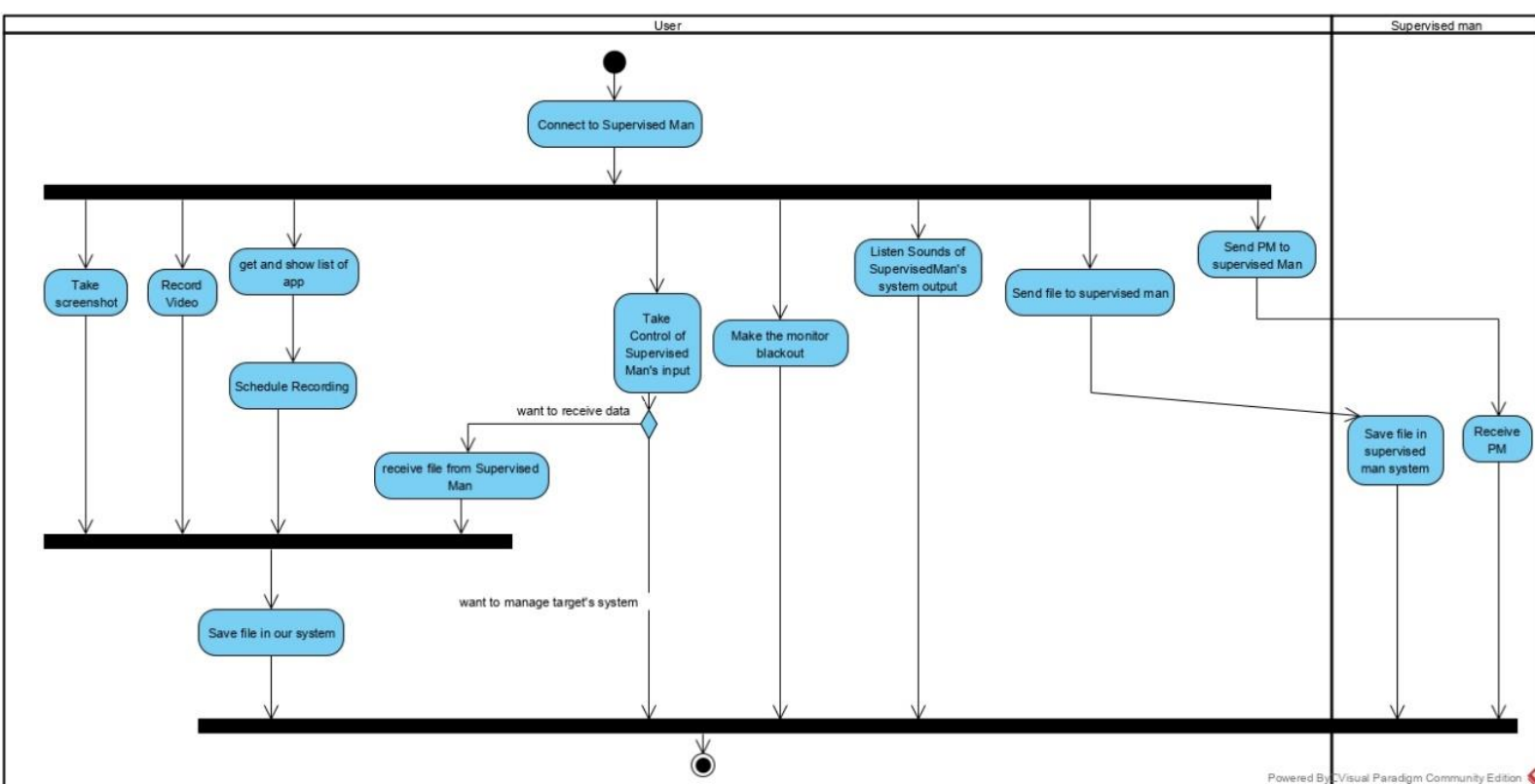


نمودار استفاده از ابزار ها (Toolbar Options) (از [us 7](#) تا [us 17](#)):

Activity Diagram:

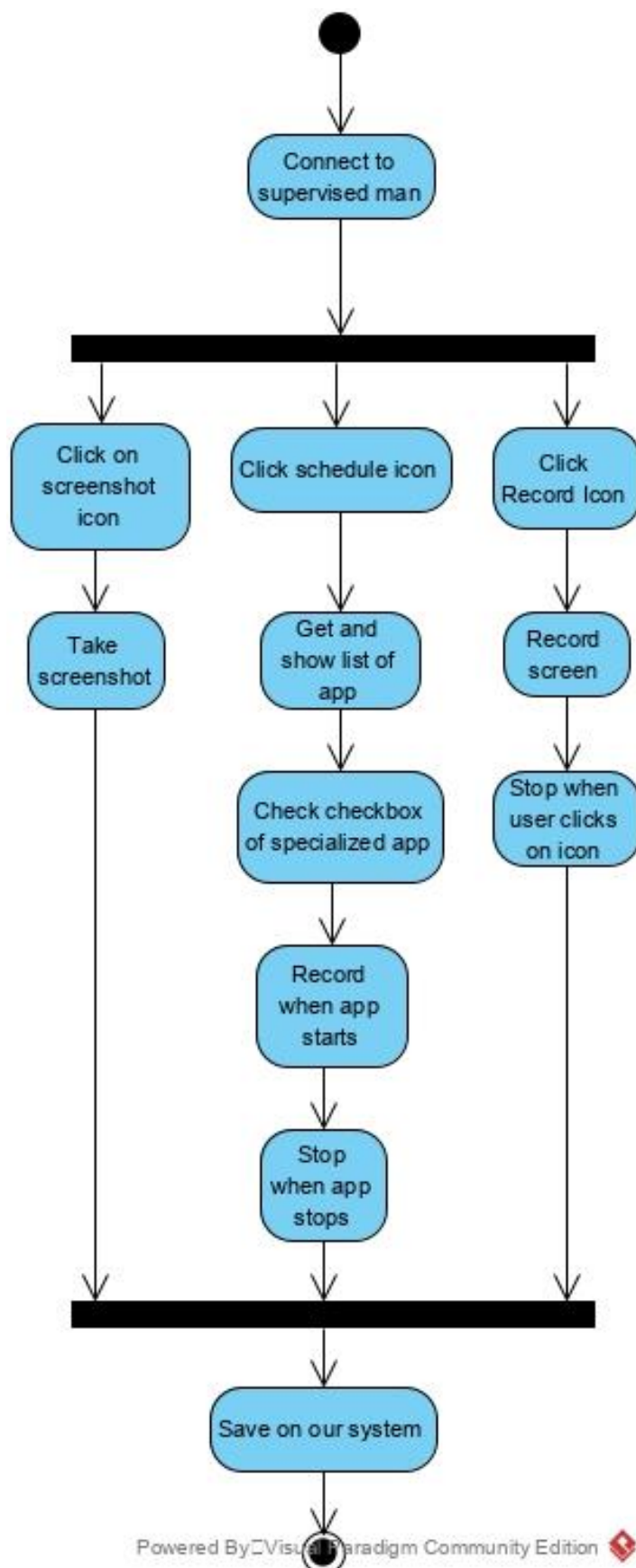


Swimlane Diagram:

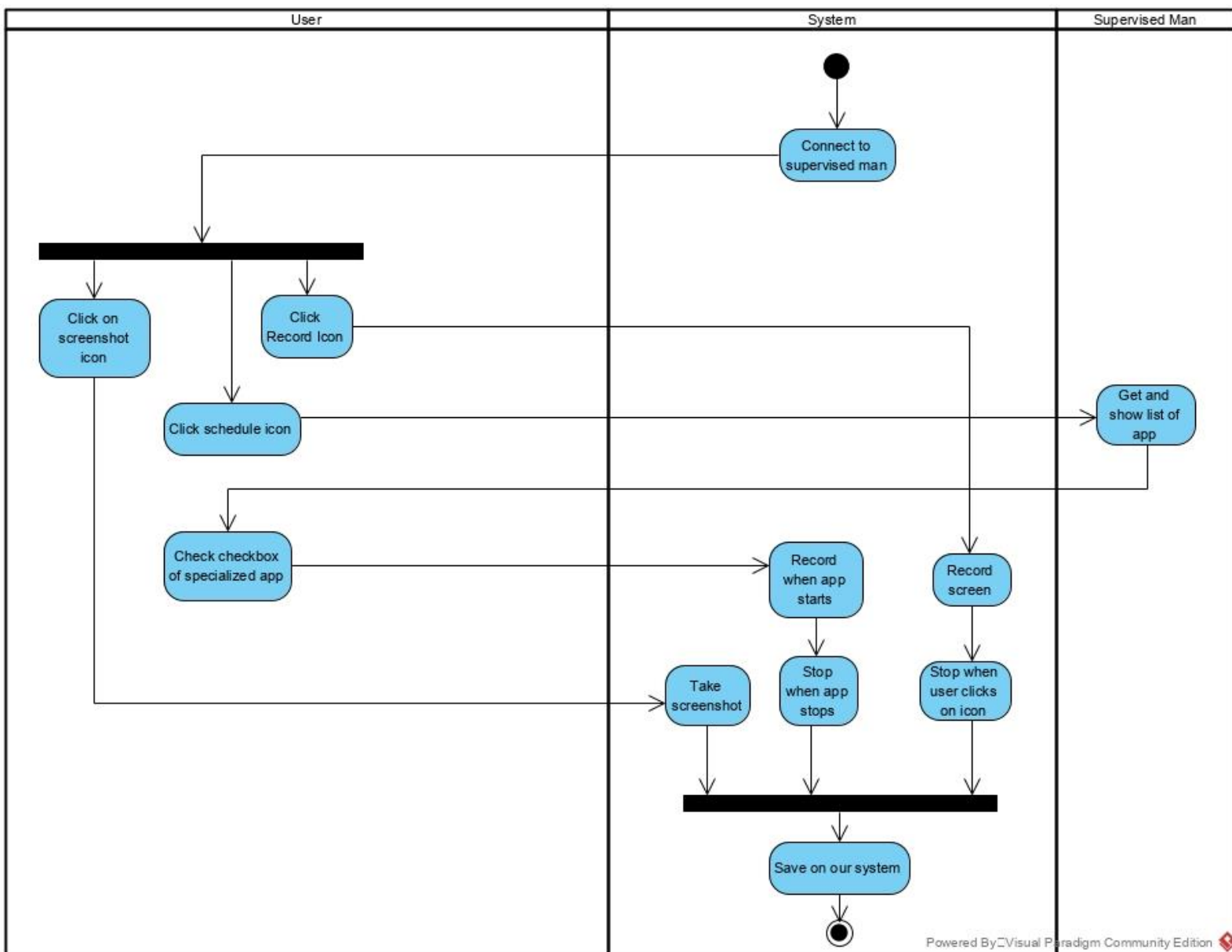


نمودار استفاده از Video Recording و Screenshot (us 7, us 8, us 16, us 17):

Activity Diagram:

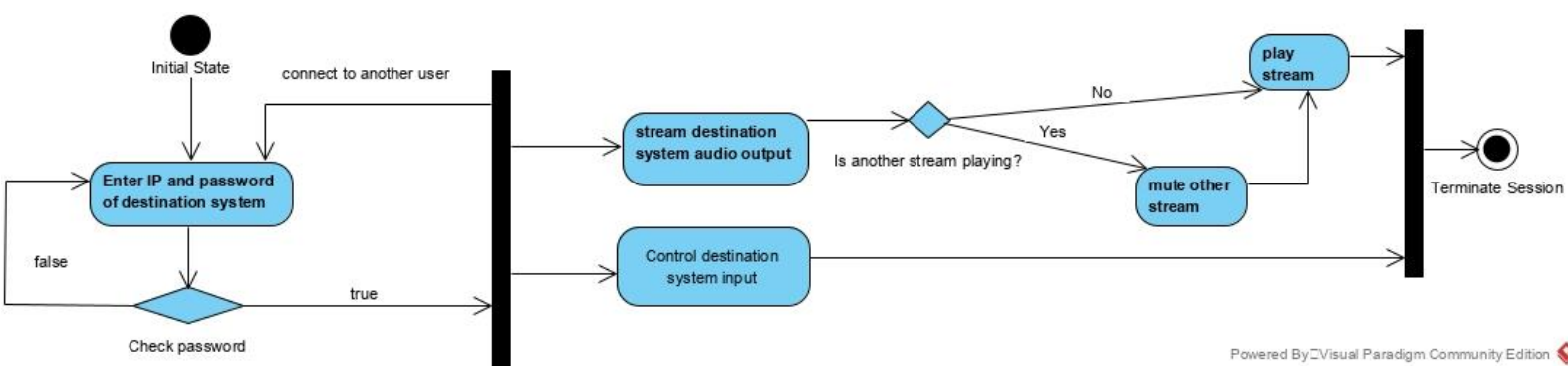


Swimlane Diagram:

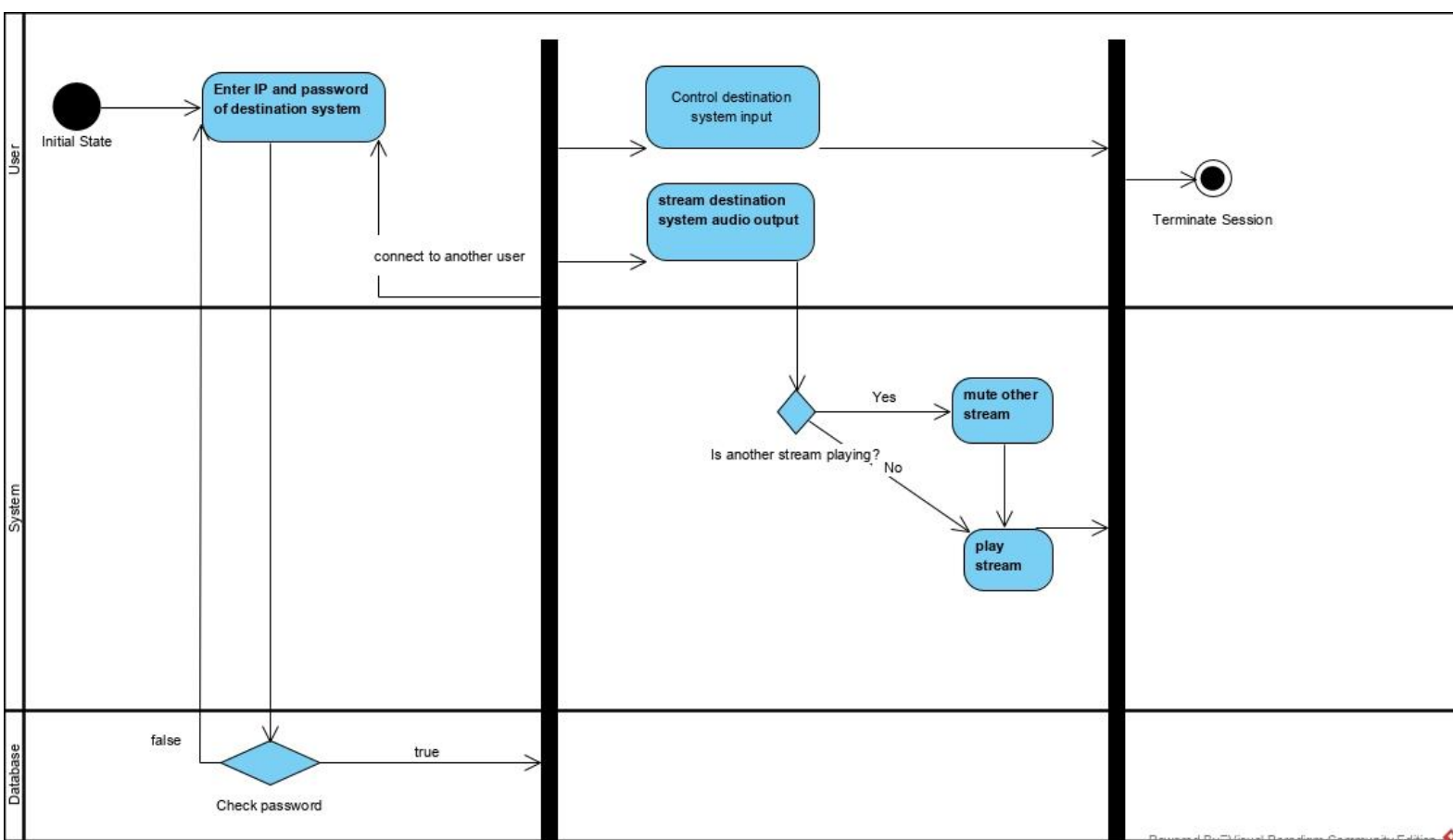


نمودار استفاده از Audio output و استفاده از کاربر مقصد (هنگام اتصال به چند کاربر)
: (us 12,us 13,us 14)

Activity Diagram:



Swimlane Diagram:



۳-۳- خروجی Technical Review

مسئول جلسه : مهیار سلامت

نظرات:

- حسن کاظمی تهرانی: در use case مربوط به درخواست اتصال و نمایش Stream باید Show Stream Window نباید به User متصل باشد زیرا کاربر با نحوه نمایش Stream Window در ارتباط نیست و تنها درخواست اتصال را میدهد
- مهیار سلامت: در use case سطح کلی باید Network با Stream Window در ارتباط باشد
- سروش مهربان براغوش:
- در use case سطح کلی نباید خیلی وارد جزئیات بشیم و ارتباط بین use case ها باید حداقل باشد پس use case های جزئی بایستی حذف بشه.
- در نمودار Activity و Swim lane نباید عبارت "Click login button" تحت عنوان Action قرار گیرد زیرا دست طراح را میگیرد و مجبور میکند که دکمه ای قرار دهد.
- محمد مهدی منتظر: در Login کردن بایستی کاربر password و user name را وارد کند پس include باید بکند

جمع بندی نظرات و تصمیم نهایی:

پس از بررسی تمامی نمودار ها، نتیجه گرفته شد که باقی نمودار های Activity و Swim lane صحیح می باشند و تغییرات مربوط به نظرات در نمودار ها اعمال شد.

فصل چهارم

آنالیز نیازمندی ها جنبه ساختاری

۱-۴- شناسایی کلاس های کاندید، دسته بندی و انتخاب نهایی

مدیریت سوکت ها:

- Network

کاربر ها:

- User

- SupervisedMan

پنجره ها:

- LiveStreamWindow

- WindowToolbar

ابزار ها:

- ScreenshotTaker

- VideoRecorder

- ScheduleRecorder

- ScreenController

- OutputListeningManager

- MessageManager

- InputController

- TransferFileManager

CRC Cards ۲-۴-رسم

نمودار مربوط به سناریو عضویت و ورود ([us 19](#), [us 20](#), [us 21](#))

Login	
Responsibilities:	
Name	Collaborator
Open Main Window if authorized	
Show warning if unauthorized	User
Lock inputs for 2 minutes after 5 try	user
Check username password	

Payment	
Responsibilities:	
Name	Collaborator
Connect to payment gateway	
Show created license to user	User
have option to copy created license	

New User	
Responsibilities:	
Name	Collaborator
Set Required fields to register	Register
purchase license	Payment

Existing User	
Responsibilities:	
Name	Collaborator
Set username and password	Login

Register	
Responsibilities:	
Name	Collaborator
Show Warning if license is invalid	User
Lock inputs for 2 minutes after 5 try	User
Check license	

Powered By  Visual Paradigm Community Edition

نمودار مربوط به سناریو اتصال و مشاهده stream window (us 4,us 5,us 6)

Network	
Responsibilities:	
Name	Collaborator
Make Connection	User, Supervised Man
End Connection	User
Show Warning	
Lock system	
Check password	Supervised Man, User

Supervised Man	
Responsibilities:	
Name	Collaborator
Give Permission	Network
Give password	Network

User	
Responsibilities:	
Name	Collaborator
Request to make connection	Network
Request to end connection	Network

Live Stream Window	
Responsibilities:	
Name	Collaborator
Open Live Stream Window	
Close Live Stream Window	
Receive Data	Network

نمودار مربوط به سناریو استفاده از ابزار ها (از [us 7](#) تا [us 17](#))

ScreenshotTaker	
Responsibilities:	
Name	Collaborator
Take screenshot	Supervised Man, User
Save photo in our system	
Show warning and choose path to save	User

TransferFileManager	
Responsibilities:	
Name	Collaborator
Send file to Supervised Man	User, Supervised Man
Receive file from Supervised Man	User, Supervised Man

OutputListeningManager	
Responsibilities:	
Name	Collaborator
Activate output listening	User, Supervised Man
Deactivate output listening	User, Supervised Man

VideoRecorder	
Sub Classes: ScheduleRecorder	
Responsibilities:	
Name	Collaborator
Record Video	Supervised Man, User
Stop Recording	
Save video in our system	
Show warning and choose path to save	User

ScheduleRecorder	
Super Classes: VideoRecorder	
Responsibilities:	
Name	Collaborator
get list of app	User, Supervised Man
set list of specilized app	User

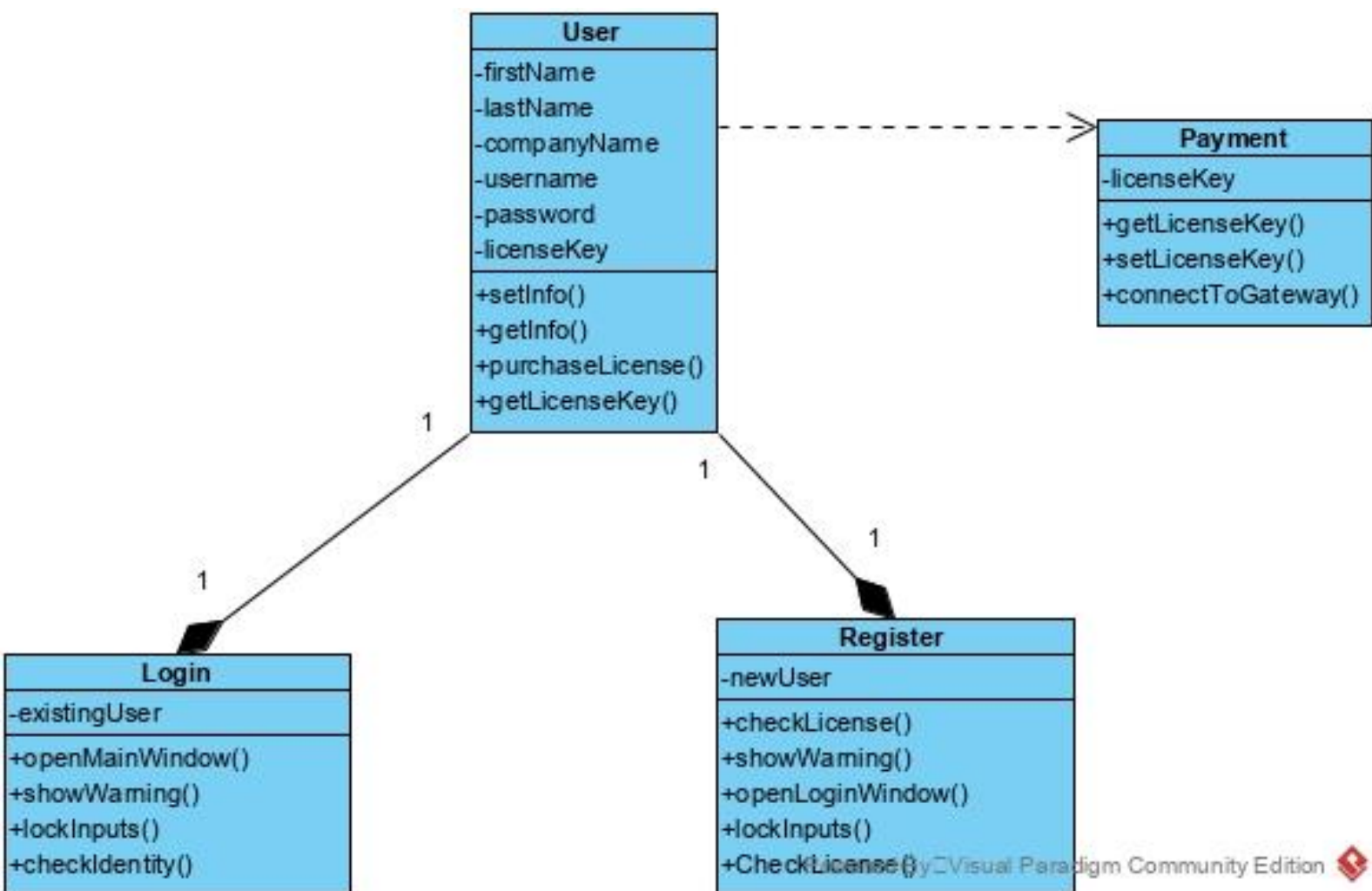
MessageManager	
Responsibilities:	
Name	Collaborator
Send PM to Supervised Man' system	User, Supervised Man

ScreenController	
Responsibilities:	
Name	Collaborator
Make screen black	User, Supervised Man
return to default mode	User, Supervised Man

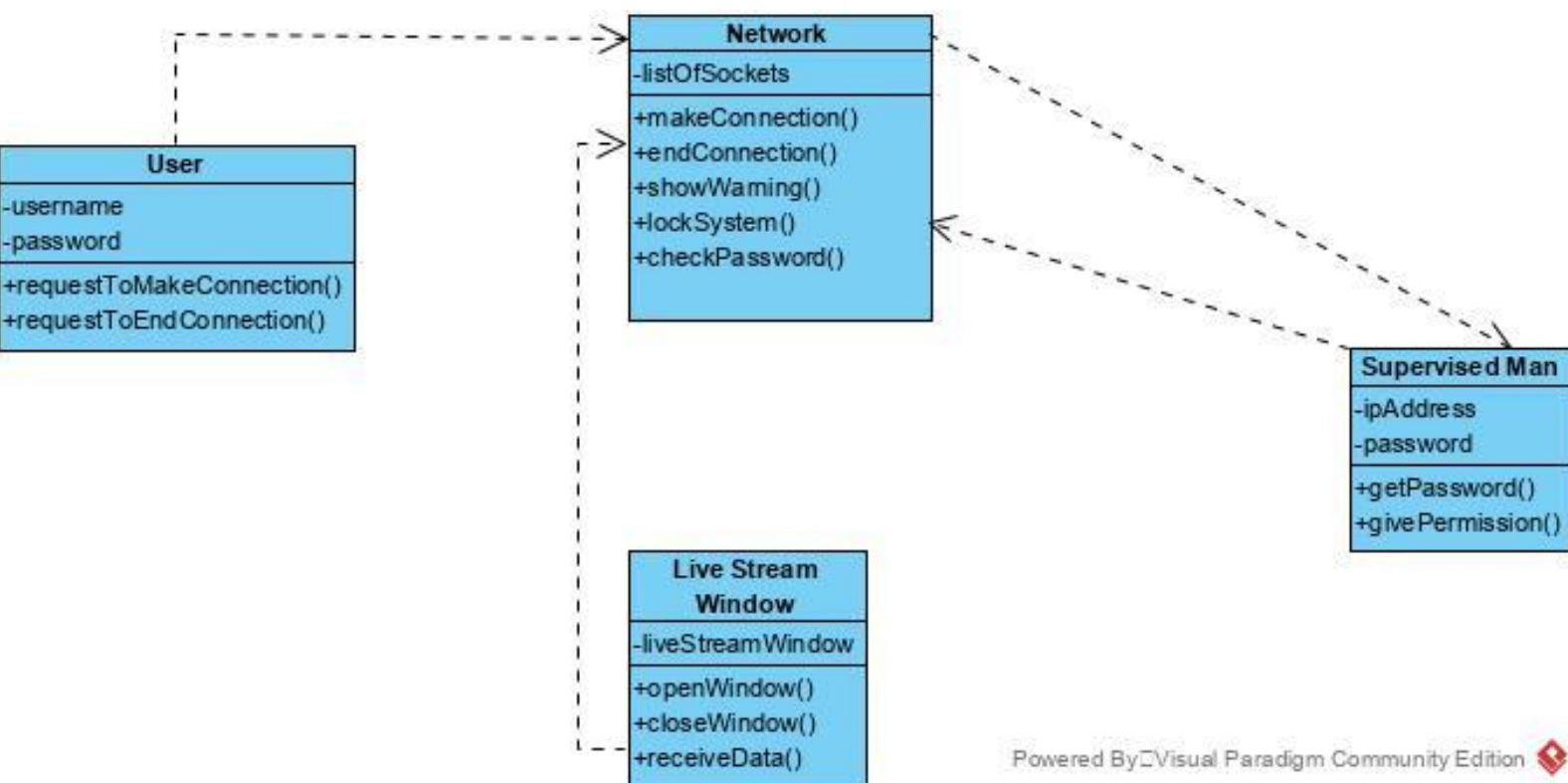
InputController	
Responsibilities:	
Name	Collaborator
Activate input control	User, Supervised Man
Deactivate input control	User, Supervised Man

۳-۴ - Class Diagram در سطح نیازمندی ها

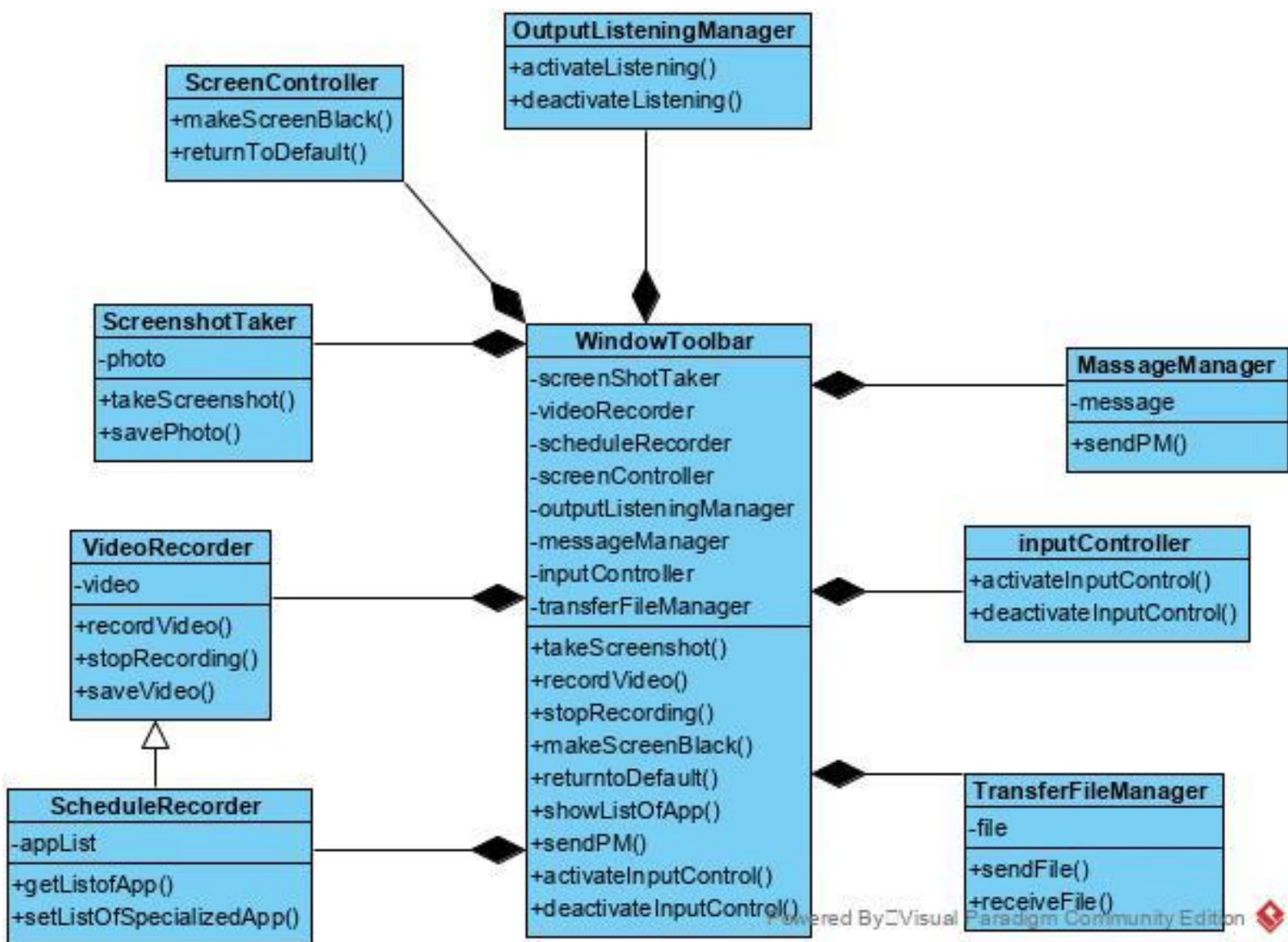
نمودار مربوط به سناریو عضویت و ورود ([us 19](#), [us 20](#), [us 21](#))



نمودار مربوط به سناریو اتصال و مشاهده stream window (us 4,us 5,us 6)

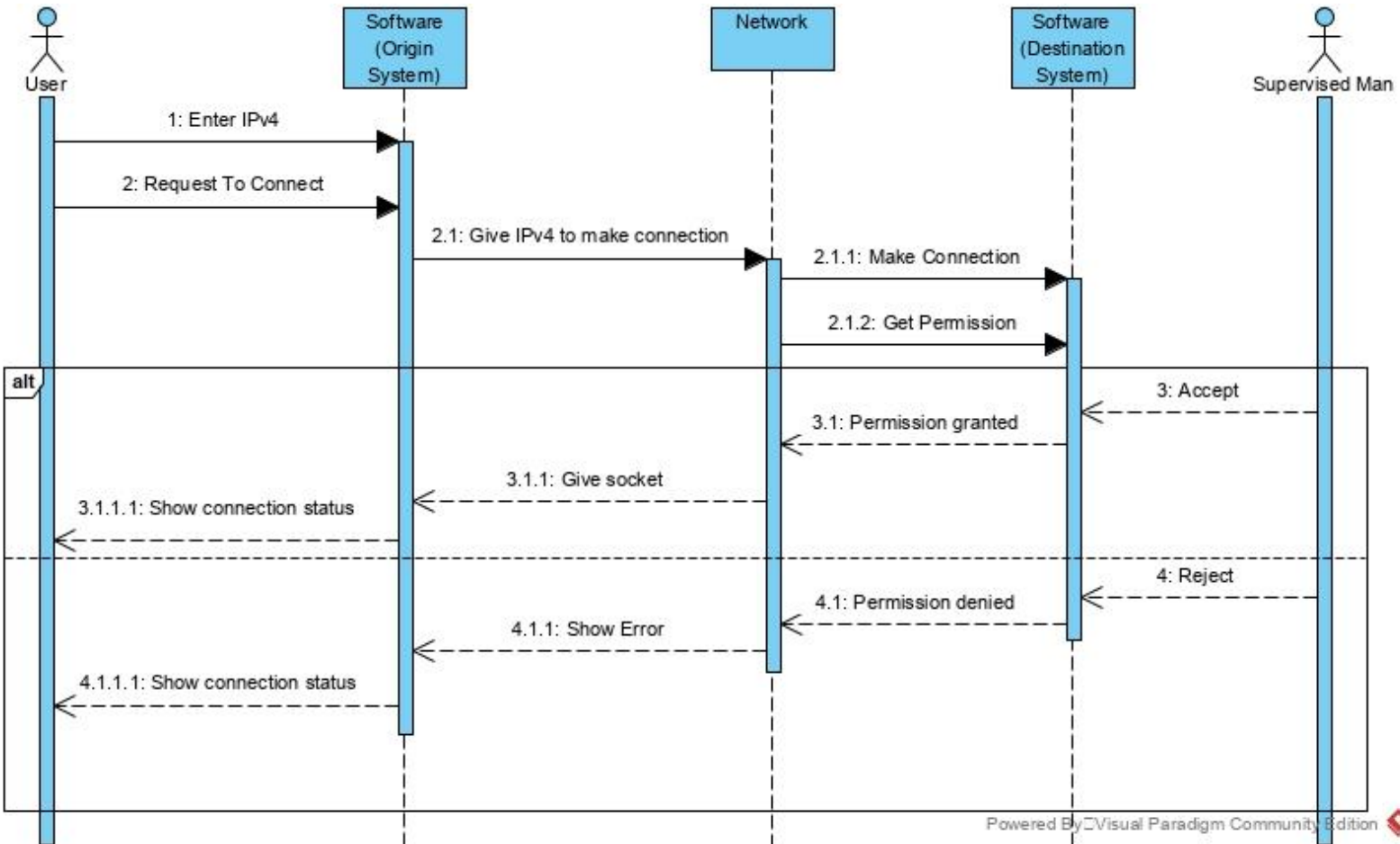


نمودار مربوط به سناریو استفاده از ابزار ها (از [us 7](#) تا [us 17](#))

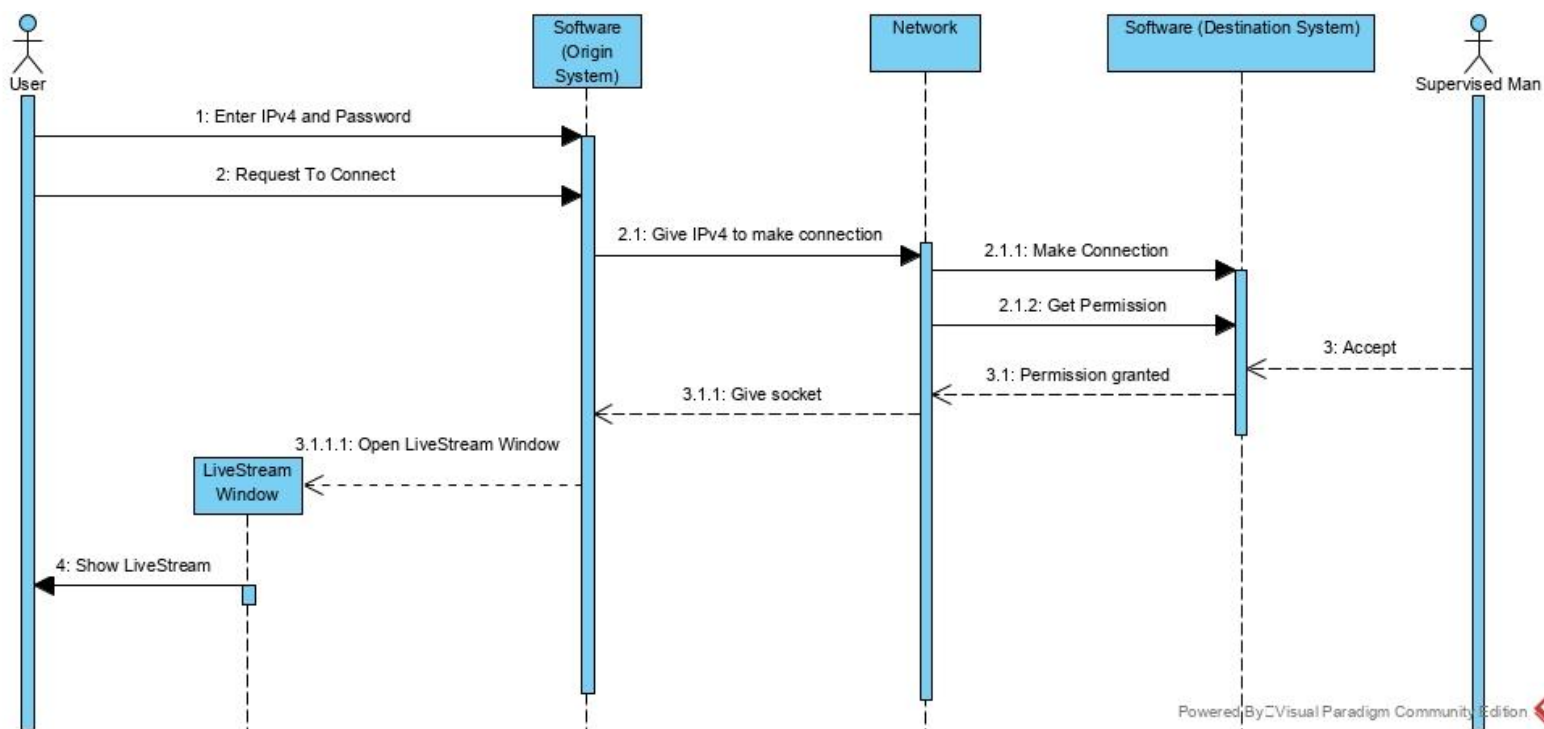


۴-۴ Sequence Diagram در سطح نیازمندی ها

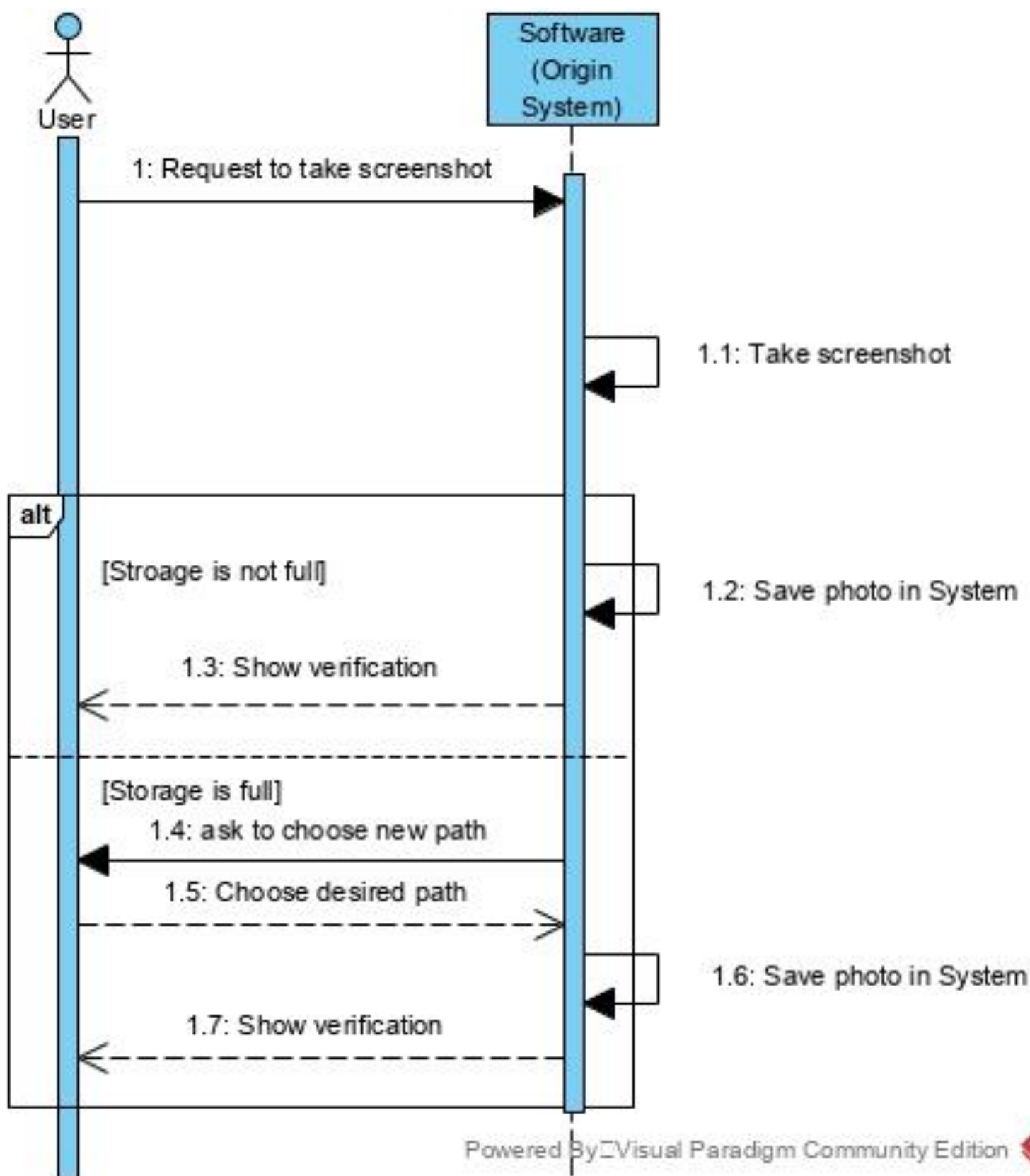
نمودار مربوط به سناریو وصل شدن به سیستم مقصد دلخواه در شبکه داخلی (us_4)



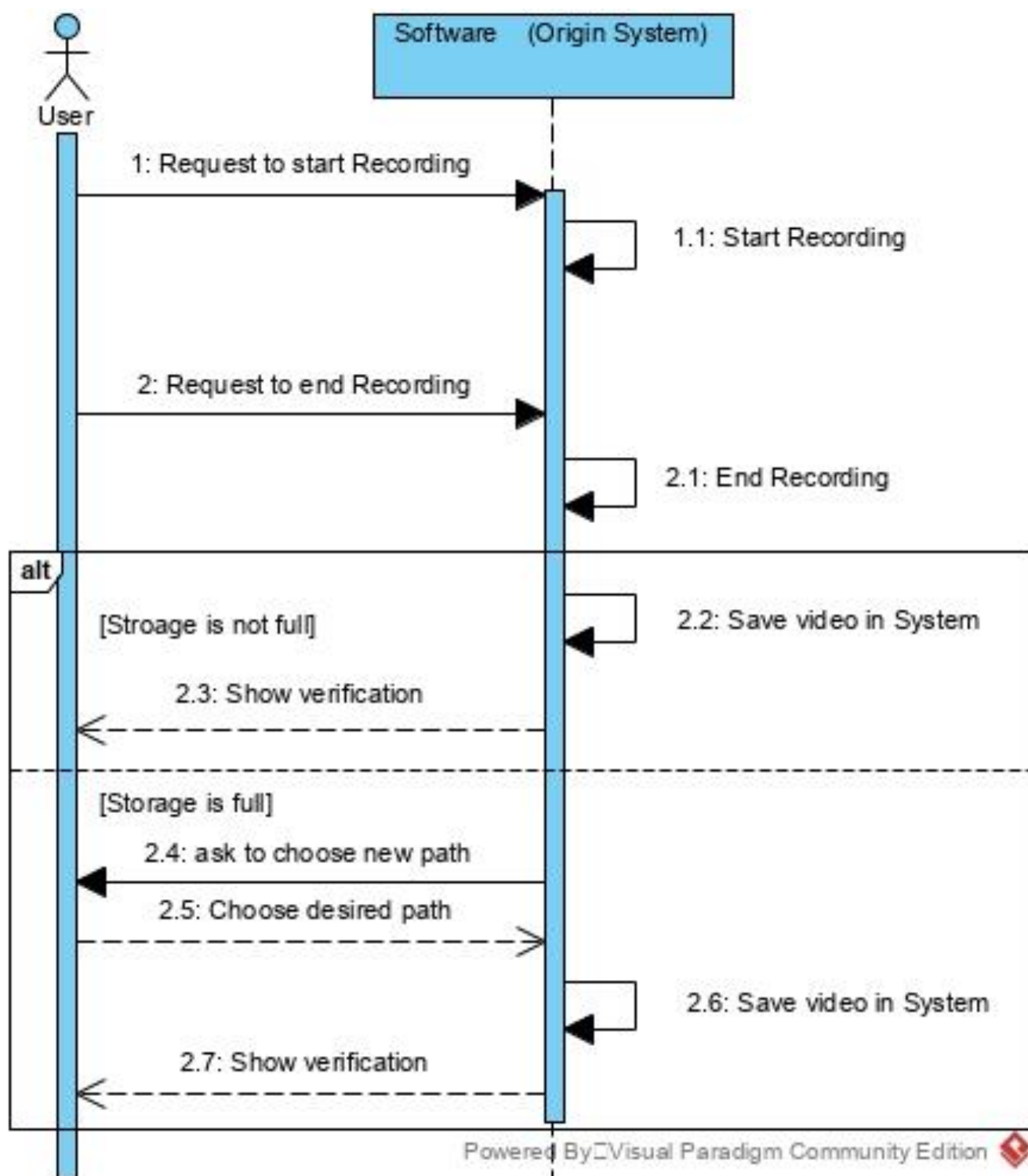
نمودار مربوط به سناریو مشاهده سیستم مقصد به صورت Live stream در شبکه داخلی (us 6)



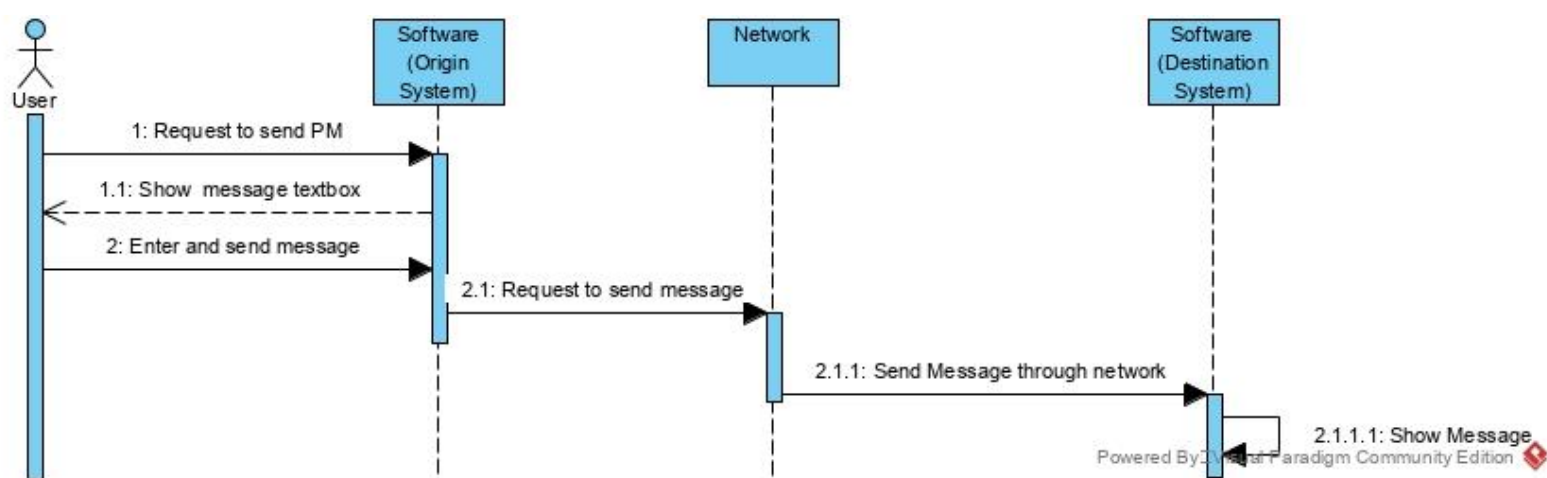
نمودار مربوط به سناریو گرفتن اسکرین شات از کامپیوتر مقصد (us 7)



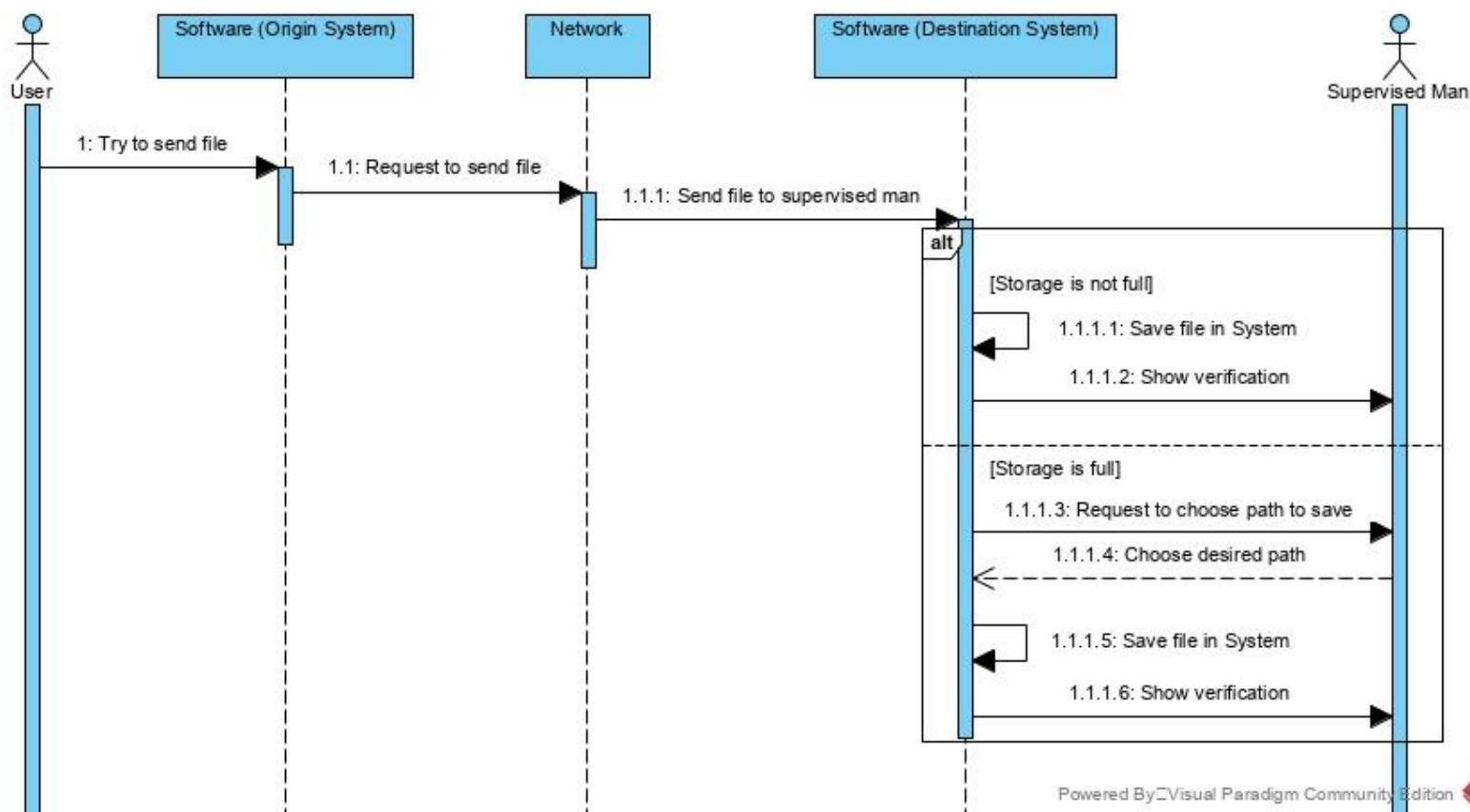
نمودار مربوط به سناریو ضبط ویدیو (us 8)



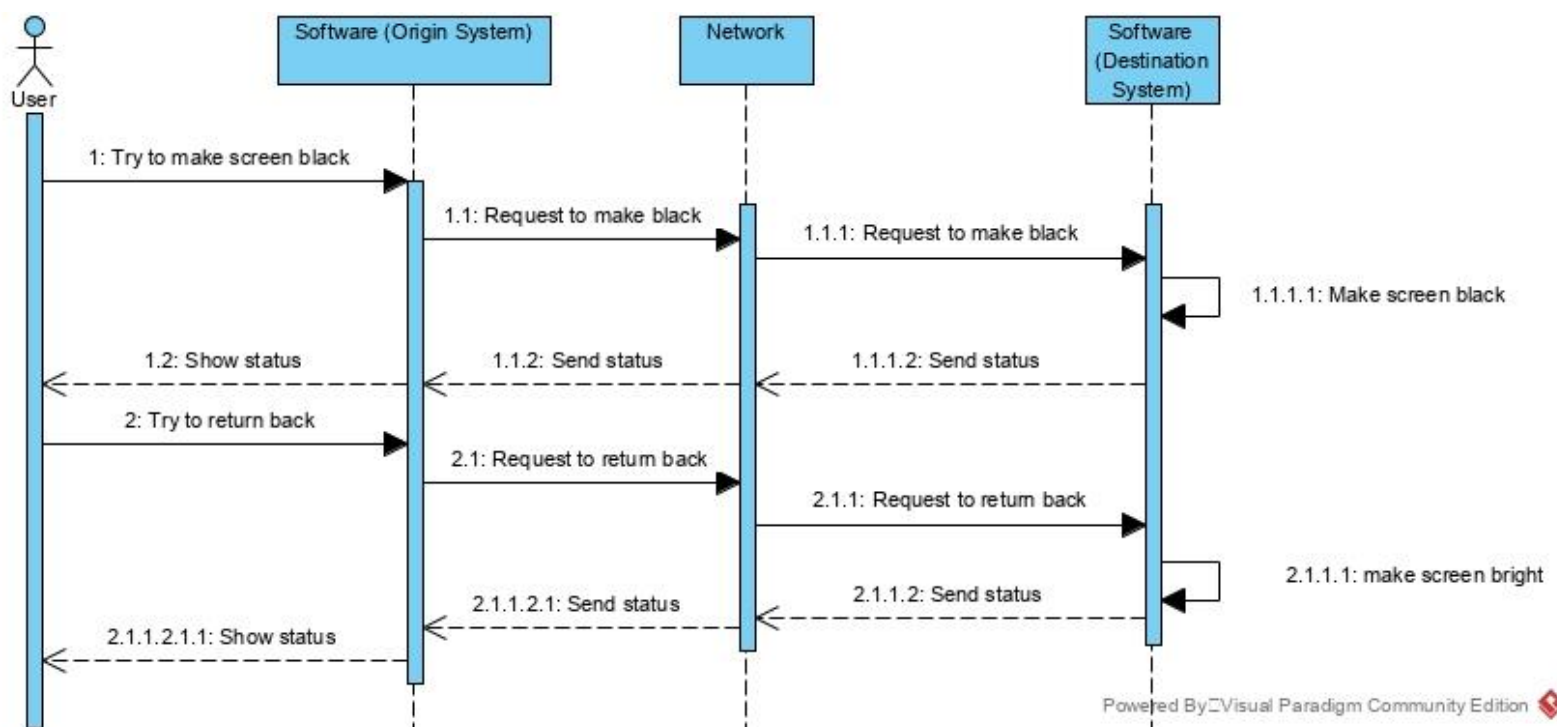
نمودار مربوط به سناریو ارسال پیام به سیستم مقصد (us 9)



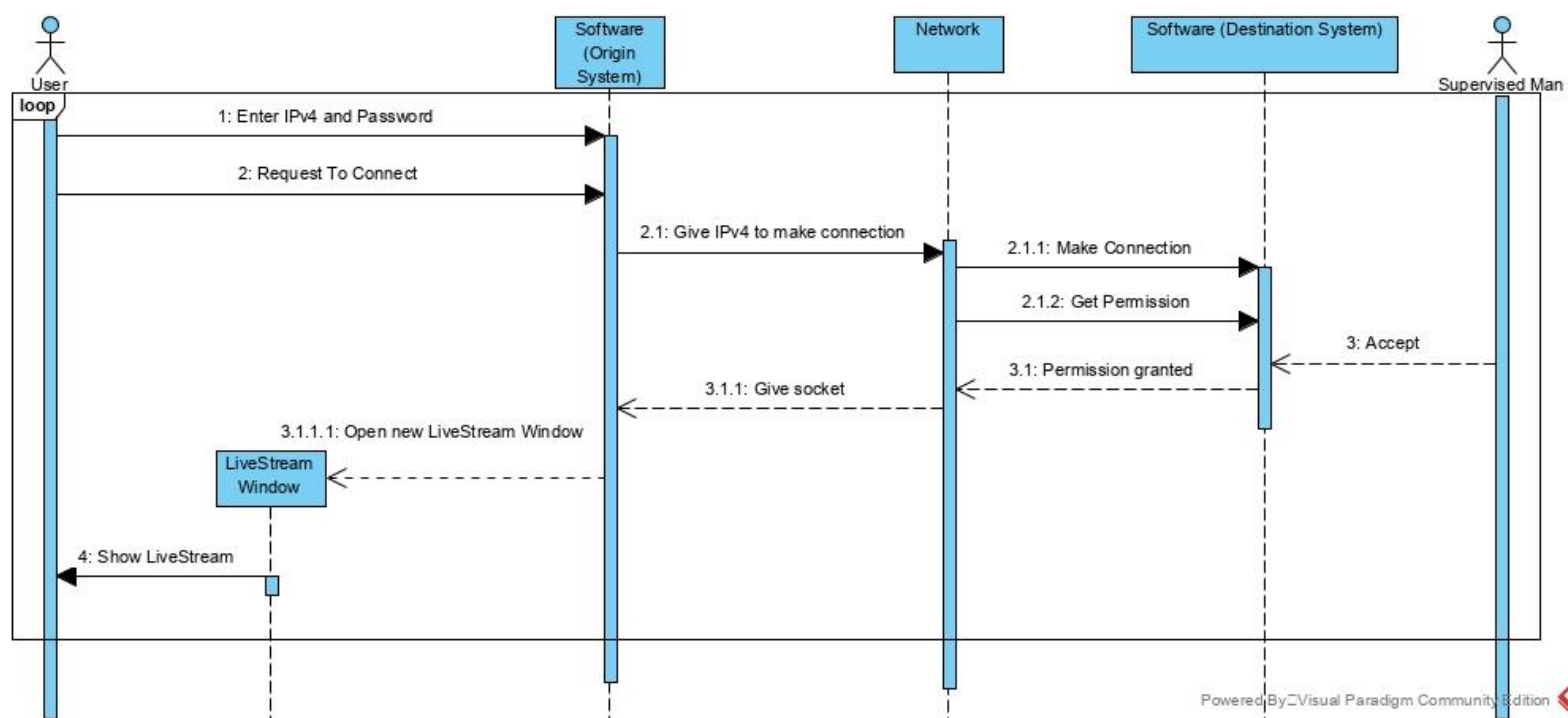
نمودار مربوط به سناریو انتقال فایل از سیستم مبدأ به سیستم مقصد (us 10)



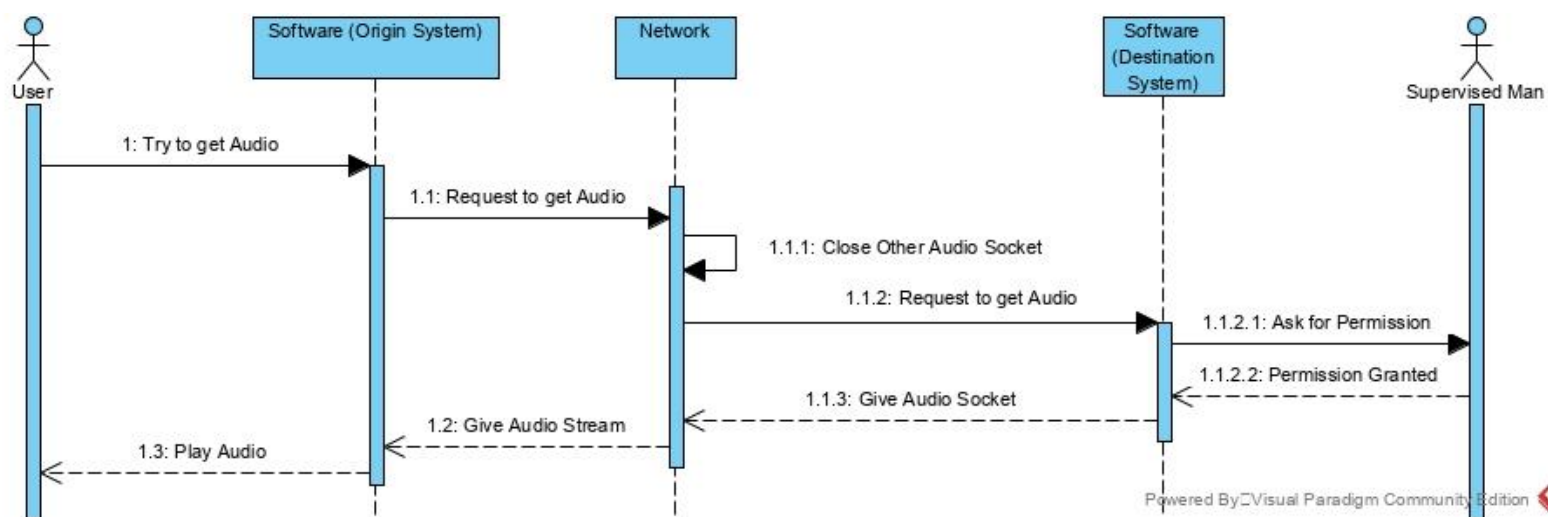
نمودار مربوط به سناریو کنترل مانیتور سیستم هدف (us 11)



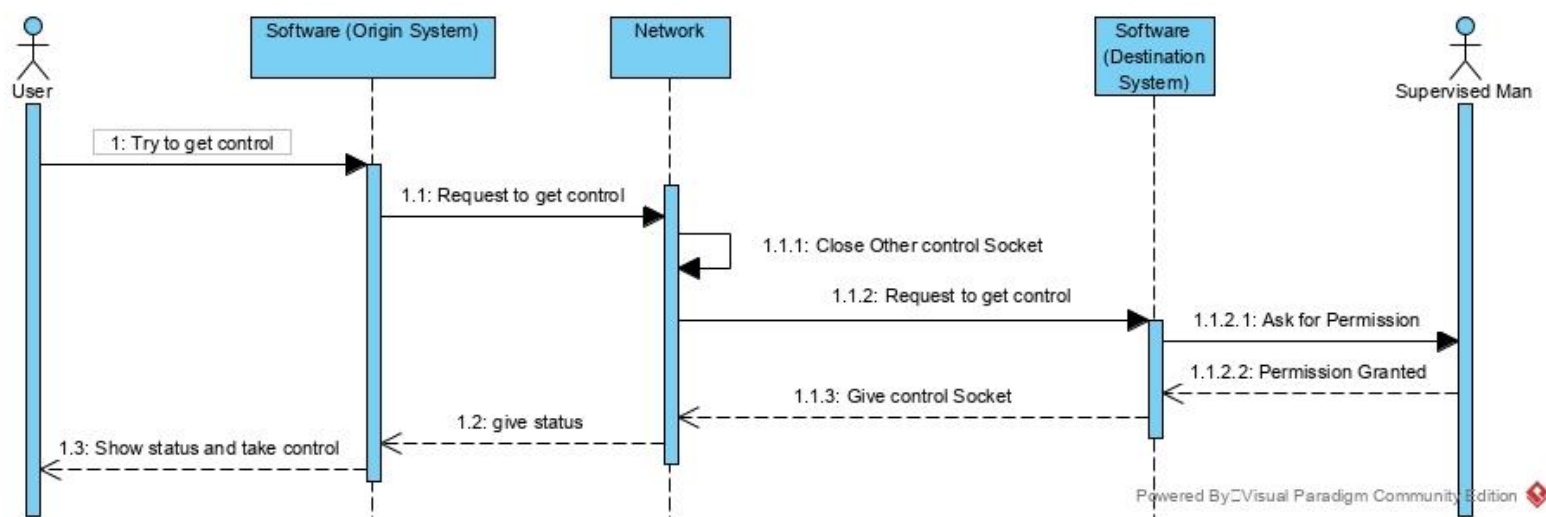
نمودار مربوط به سناریو وصل شدن به چند سیستم دلخواه در شبکه داخلی (us 12)



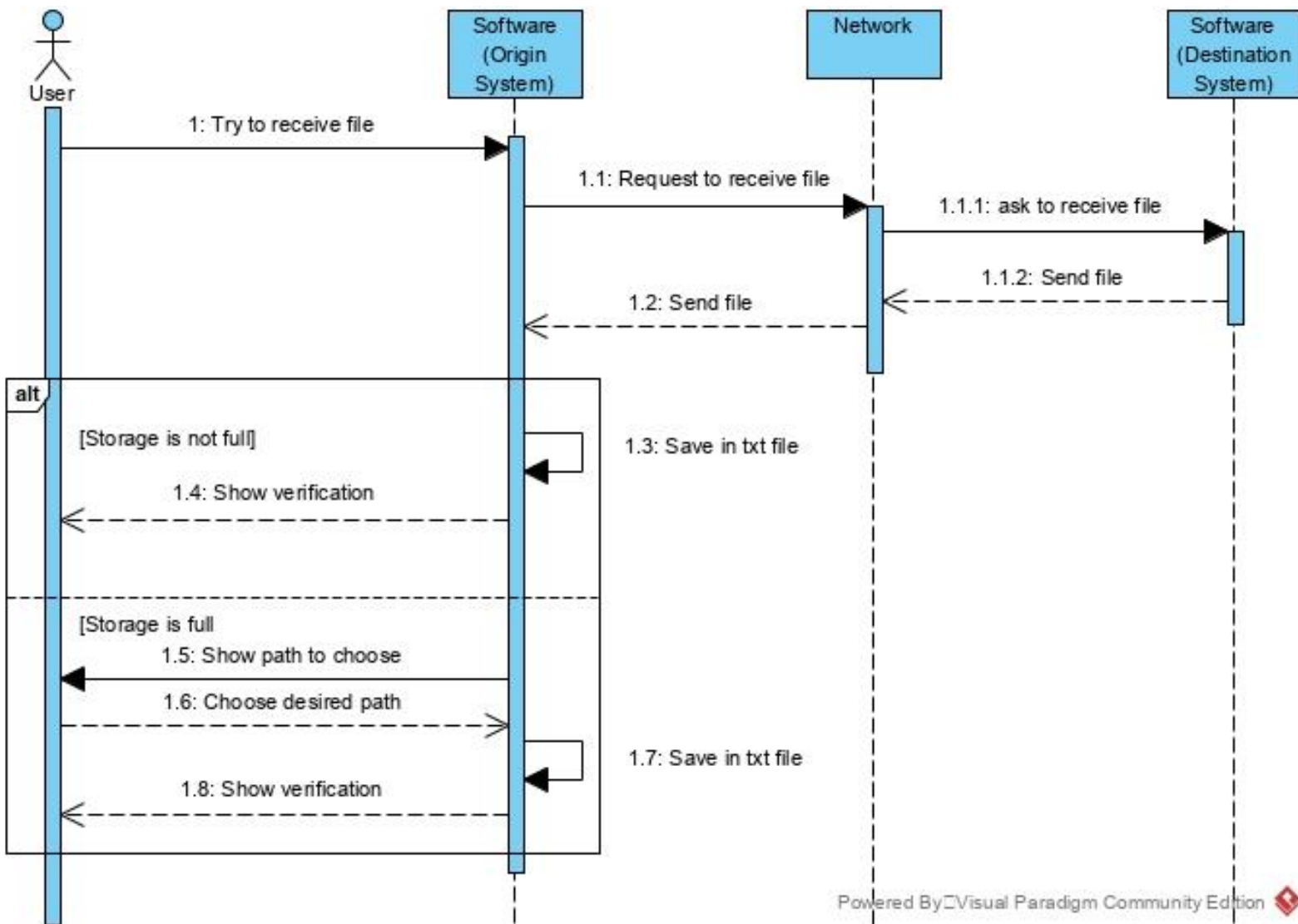
نمودار مربوط به سناریو شنیدن صدای خروجی از کامپیوتر مقصد در شبکه داخلی (us 13)



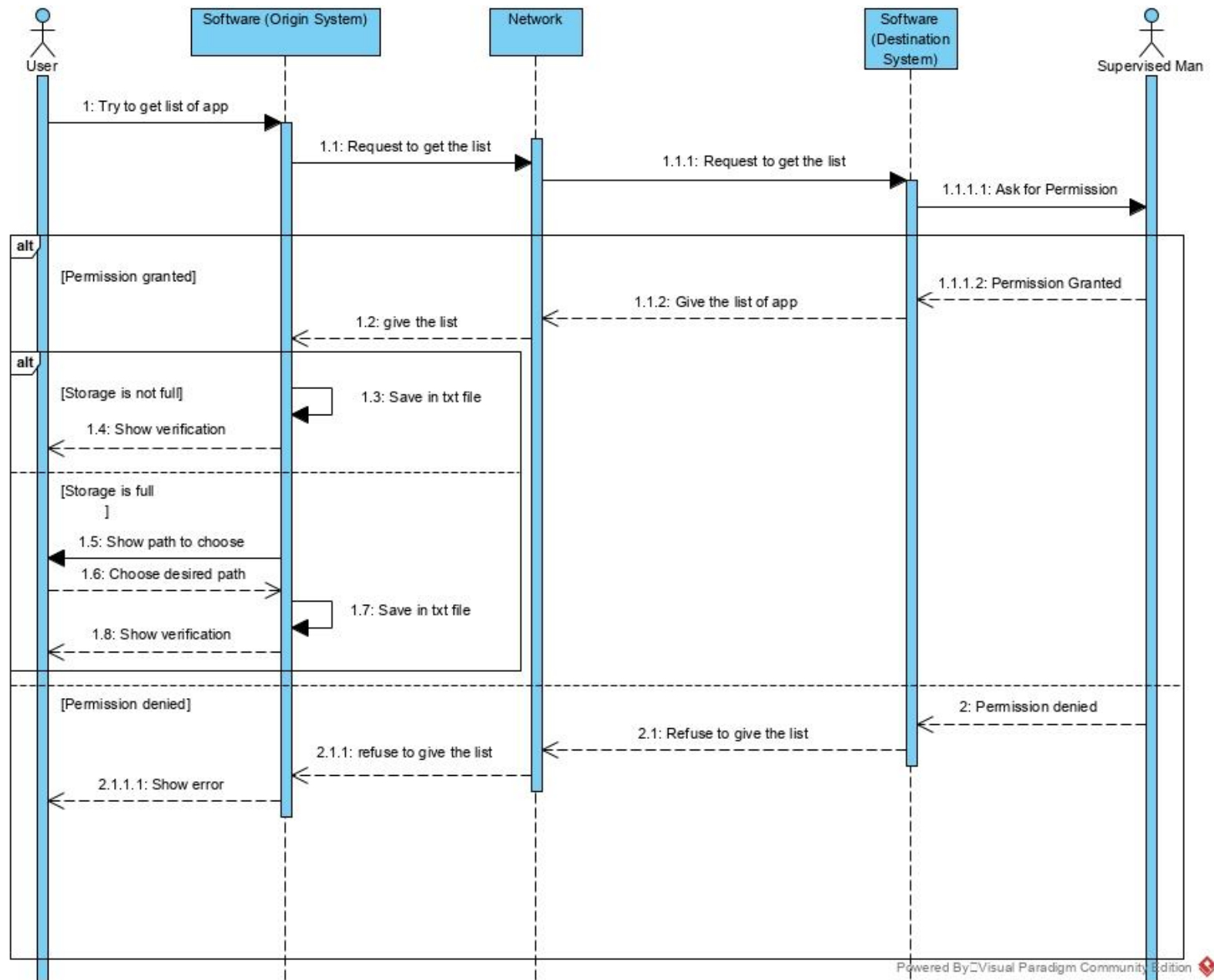
نمودار مربوط به سناریو کنترل ورودی کامپیوتر هدف (us 14)



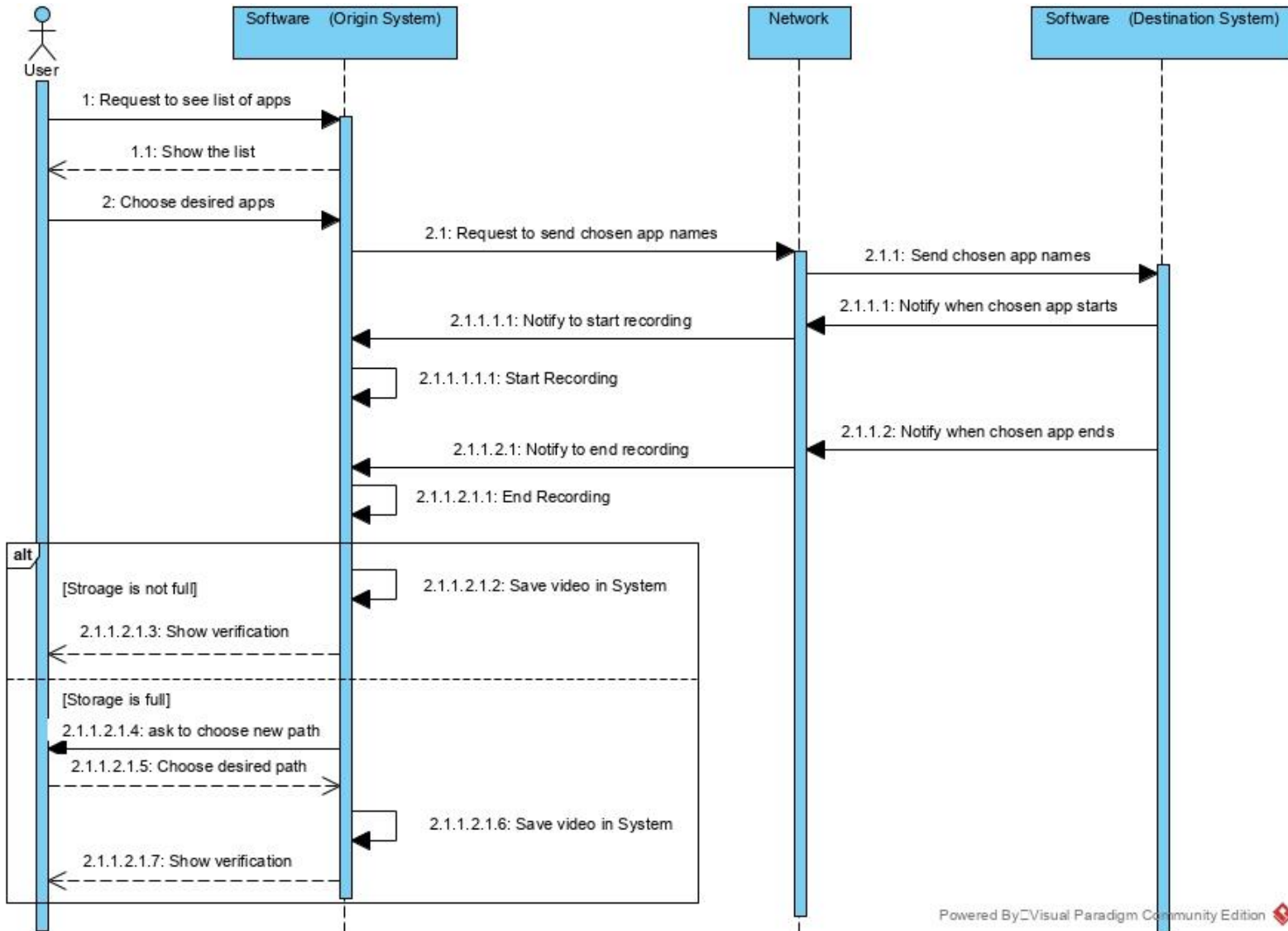
نمودار مربوط به سناریو انتقال فایل از کامپیوتر مقصد به کامپیوتر مبدا (us 15)



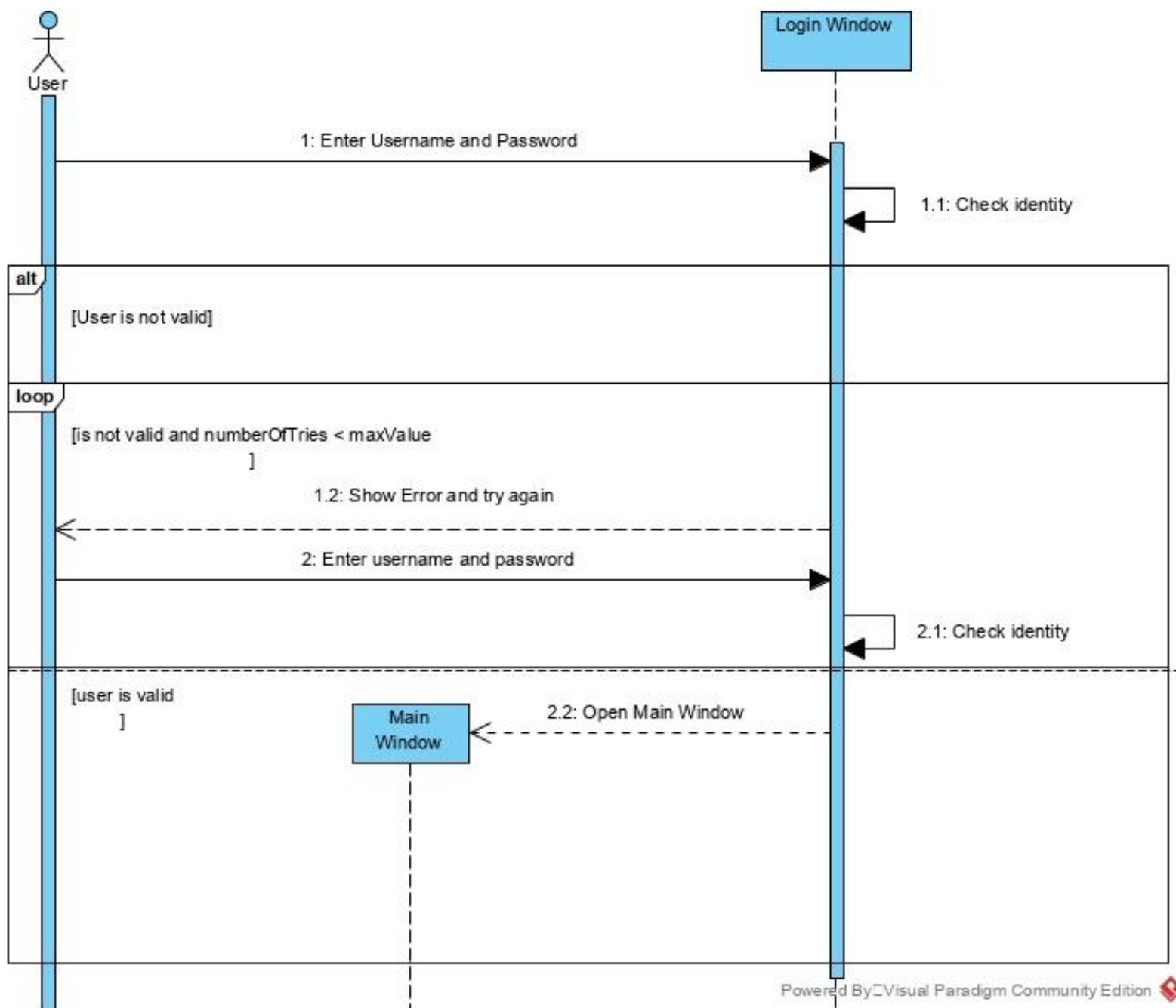
نمودار مربوط به سناریو گرفتن لیست app در یک سیستم (us 16)



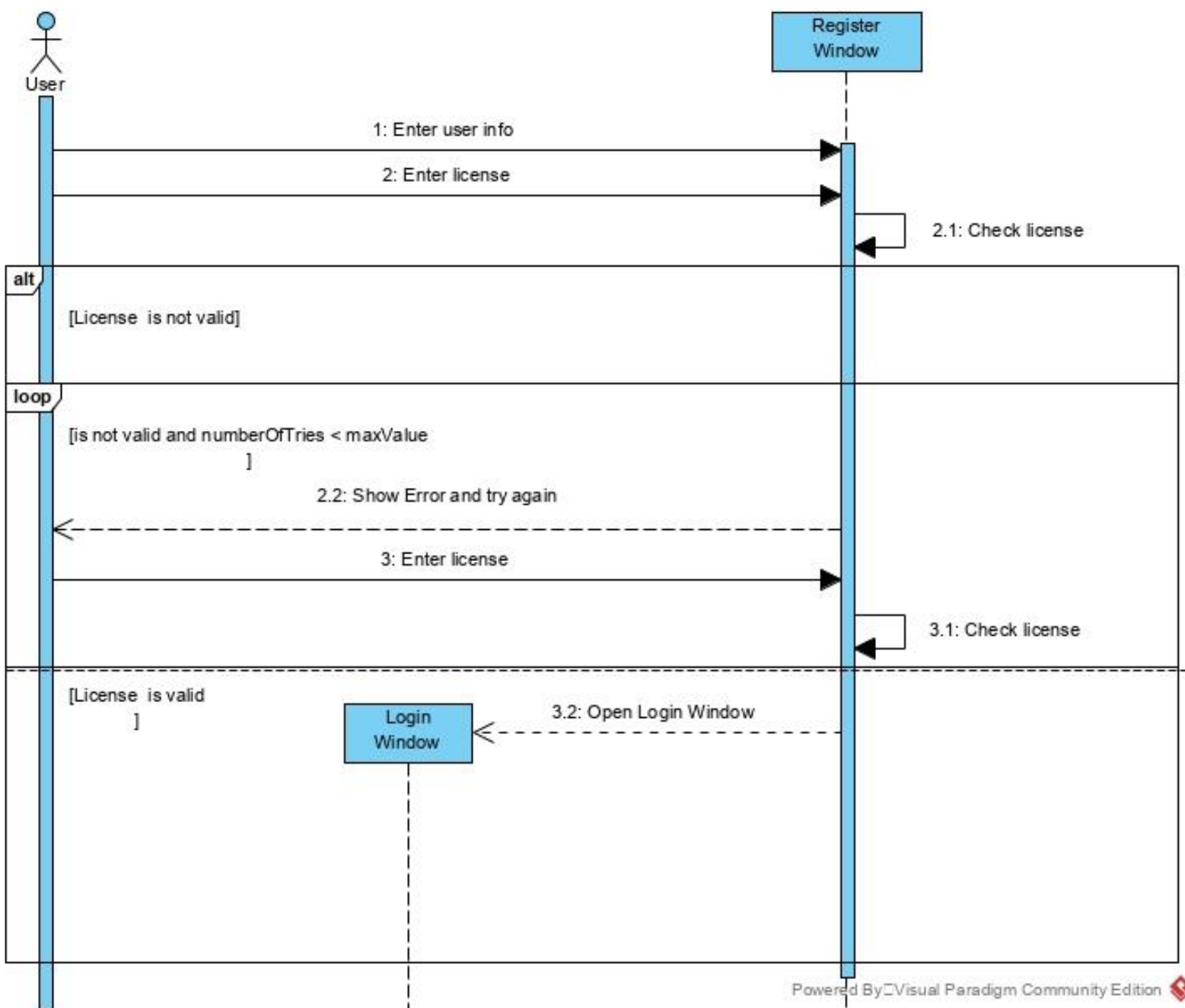
نمودار مربوط به سناریو ضبط ویدیو به محض انجام کار مشخص (us 17)



نمودار مربوط به سناریو Login کردن در نرم افزار (us 19)



نمودار مربوط به سناریو ثبت نام در نرم افزار (us 21)



۴-۵- خروجی Technical Review

مسئول جلسه : محمد مهدی منتظر

نظرات:

- حسن کاظمی تهرانی: در نمودار Class Diagram، بایستی SupervisedMan به Network نیاز داشته باشد نه بالعکس
- مهیار سلامت: به نظرم ارتباط Dependency بین User و Payment نباید باشد
- سروش مهربان براغوش: به نظرم نباید FileManager و InputController ارتباط Composition داشته باشند
- محمد مهدی منتظر: به نظرم بایستی بین SupervisedMan و User ارتباط Association باشد

جمع بندی نظرات و تصمیم نهایی:

نتیجه نظرات:

- پاسخ به نظر حسن کاظمی تهرانی: با توجه به این که Network برای Permission گرفتن از SupervisedMan به آن نیاز دارد، بایستی نسبت به آن Dependency داشته باشد و SupervisedMan برای ارسال داده به شبکه نیاز دارد، لذا Dependency دو طرفه دارد
- پاسخ به نظر مهیار سلامت: User برای خرید License، به کلاس Payment وابسته است
- پاسخ به نظر سروش مهربان براغوش: ارتباطی بین این دو کلاس موجود نیست و بایستی حذف شود
- پاسخ به نظر مهدی منتظر: این دو مستقیم به هم ربطی ندارند و به واسطه Network به یکدیگر وابستن.

فصل پنجم

طراحی معماری

۱-۵- انتخاب و استدلال استایل های کاندید برای معماری سیستم

Layered Architecture: با توجه به این که می توان نرم افزار را به چند لایه مستقل به طور مناسب تقسیم کرد، به طوری که هر لایه تنها با لایه بالایی و پایینی در ارتباط باشد، این استایل، استایل مناسب برای معماری کلی نرم افزار می باشد.

نحوه توزیع لایه ها به صورت زیر می باشد:

- **UI Layer:** در این لایه رابطه کاربری سیستم قرار میگیرد، به طوری که کاربر سیستم به طور مستقیم با این لایه در ارتباط می باشد و باقی لایه ها از نظر کاربر پنهان می باشد
 - **Application Layer:** در این لایه component هایی که کاربر از طریق UI با آن ها در ارتباط است پیاده شده است، کامپوننت هایی نظیر **Toolbar** و **Streamer**
 - **Utility Layer:** این لایه دارای ابزار هایی می باشد که component های لایه فوق آن (Application Layer) برای اجرا به آن ها نیازمند هستند، ابزار هایی نظیر **Payment**. مدیریت لاینس ها و مدیریت سوکت ها
 - **Core Layer:** لایه اصلی نرم افزار و در آن ابزار های اصلی که نرم افزار به آن نیاز دارد، وجود دارند. ابزار هایی نظیر **Database**، شبکه و مدیریت حافظه
- لازم به ذکر است که با توجه به این که نرم افزار مورد نظر در حوزه شبکه میباشد و امنیت در آن نقش بسیار مهمی دارد، از **Cross-cutting concern** ها نیز بایستی استفاده شود که به عنوان مثال امنیت تمام لایه ها را مدیریت کند.

Data-centered Architecture: این استایل به نحوی که دیتابیس و شبکه داخلی Network در وسط قرار بگیرند و باقی ابزار نرم افزار تحت Client Software با آن ها در ارتباط باشند مناسب می باشد. از مزایای استفاده از این استایل قابل Scale بودن آن است، زیرا هر feature ای بخواهد به نرم افزار اضافه شود تنها کافیست با Data-center که در وسط قرار دارد در ارتباط باشد. اما از عیب آن به ارتباط ضمنی بین Client Software ها میتوان اشاره کرد.

برای مشخص کردن flow این استایل از Repository Pattern استفاده میکنیم که در بخش بعدی به توضیح آن می پردازیم.

۲-۵- انتخاب استدلال الگوهای کاندید برای معماری سیستم

MVC Pattern: این الگو، الگوی بسیار مناسبی جهت بخش UI برنامه می باشد. به کمک این الگو میتوان منطق و ظاهر UI را از هم تفکیک کرد و بطور مستقل توسعه داد و با این کار maintenance را تا حدی بالا برد.

این الگو از ۳ بخش Model, View, Controller تشکیل شده است که در نرم افزار ما:

- **Model:** بخش منطق GUI برنامه، این که با انتخاب هر گزینه چه کارهایی بایستی در Back-End برنامه صورت گیرد را تشکیل میدهد.
- **View:** بخش GUI برنامه که کاربر به طور مستقیم با آن کار میکند، گزینه ای را انتخاب می کند و عبارتی را می نویسد را تشکیل میدهد.
- **Controller:** رابط بین GUI و منطق GUI

Repository Pattern: این الگو برای استایل Data-centered می باشد و مشخص میکند

یک Repository به صورت passive در وسط قرار دارد و باقی اجزا با آن کار میکنند. در مثال نرم افزاری ما Network به صورت Passive قرار دارد و باقی اجزای سیستم با Socket های درون Network تعامل دارند.

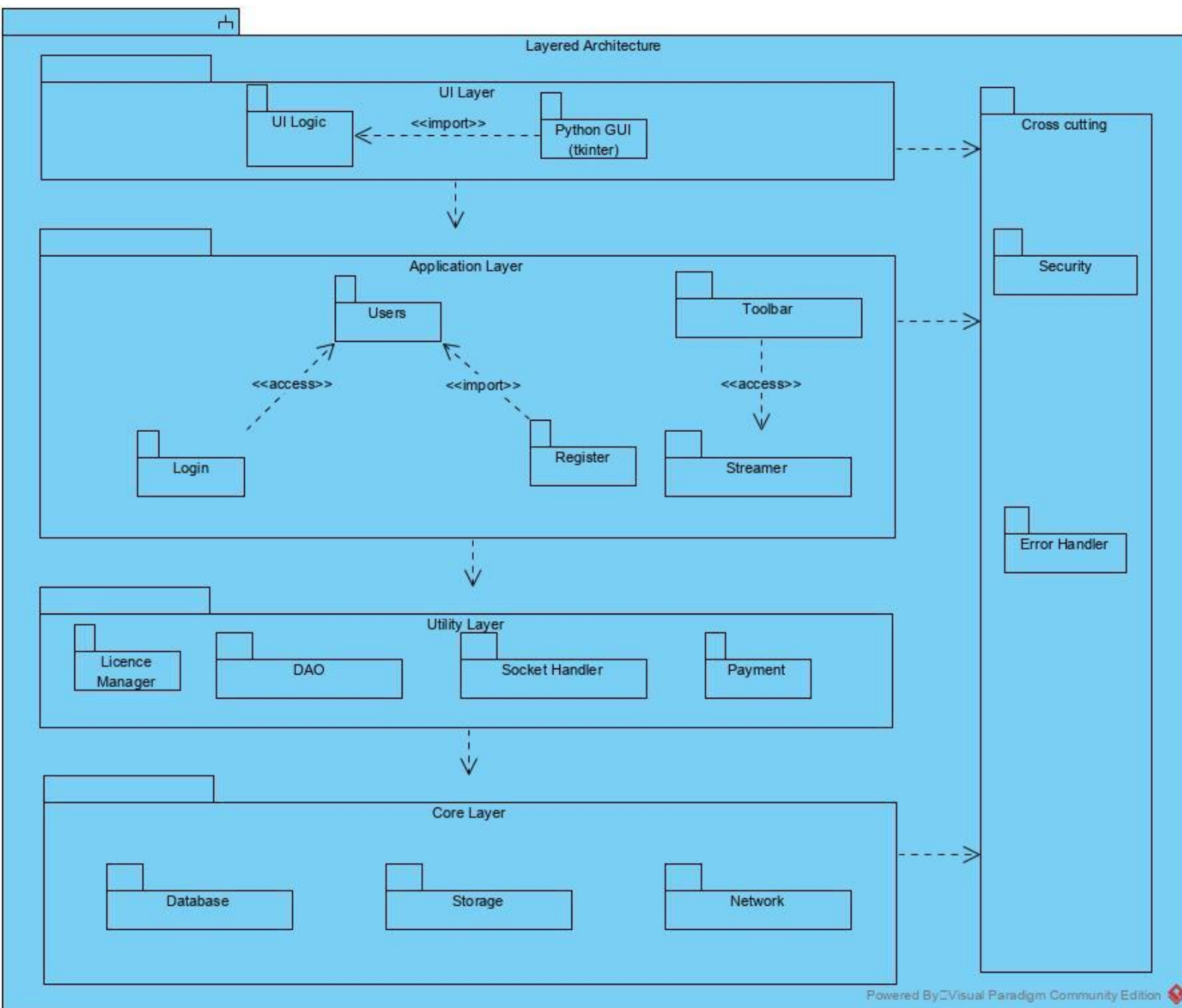
Microkernel Pattern: به کمک این الگو، میتوانیم بخش اصلی (Core System) که نظیر

اتصال کاربر مبدا به مقصد و مشاهده صفحه آن می باشد را پیاده سازی کنیم و باقی ابزارها نظیر اسکرین شات، ارسال پیام، ضبط ویدیو و ... را تحت Plug-in Component به عنوان component های کاملاً مستقل طراحی و به سیستم اضافه کنیم.

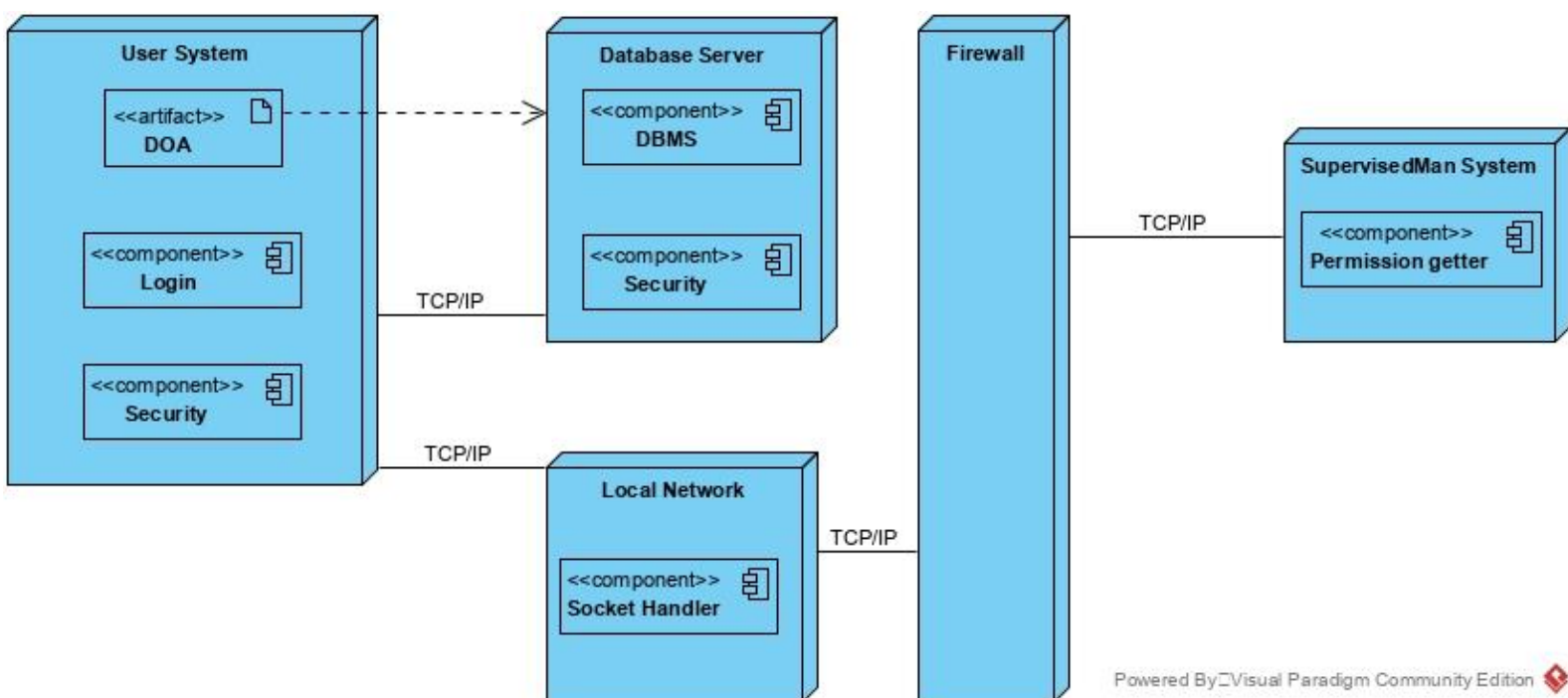
۳-۵- ارائه حداقل دو طرح برای معماری سیستم در قالب

طرح اول: در این طرح از استایل لایه ای برای معماری استفاده شده است.

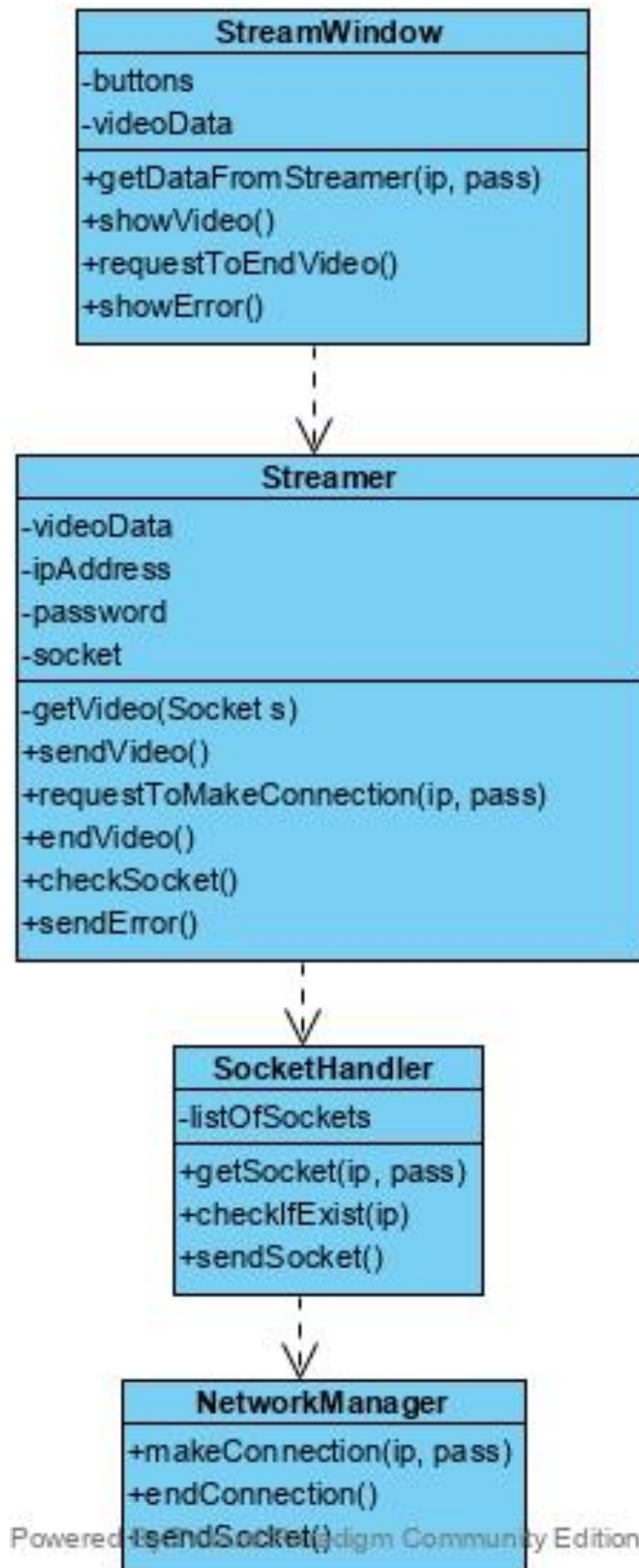
Package Diagram:



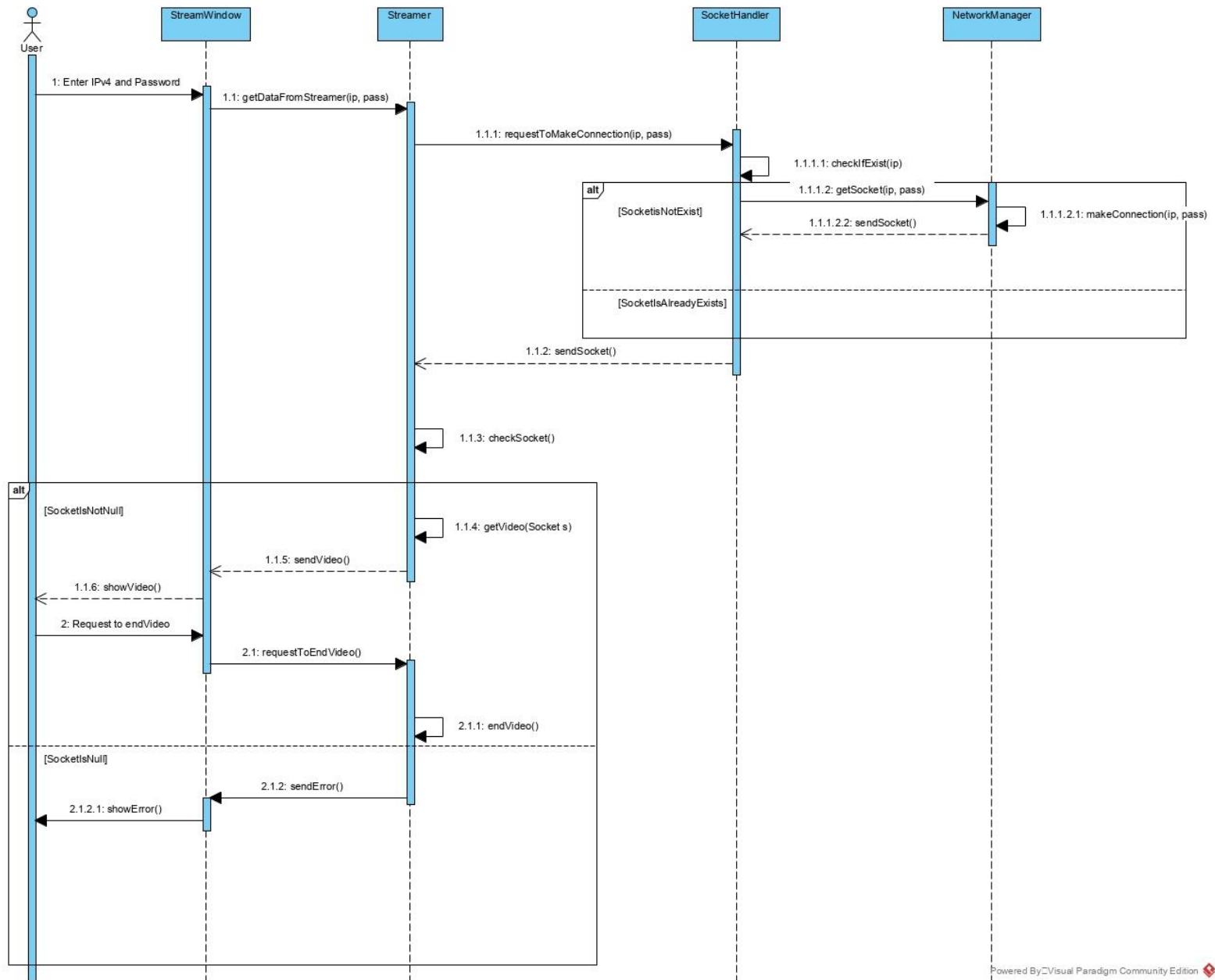
Deployment Diagram:



Class Diagram:

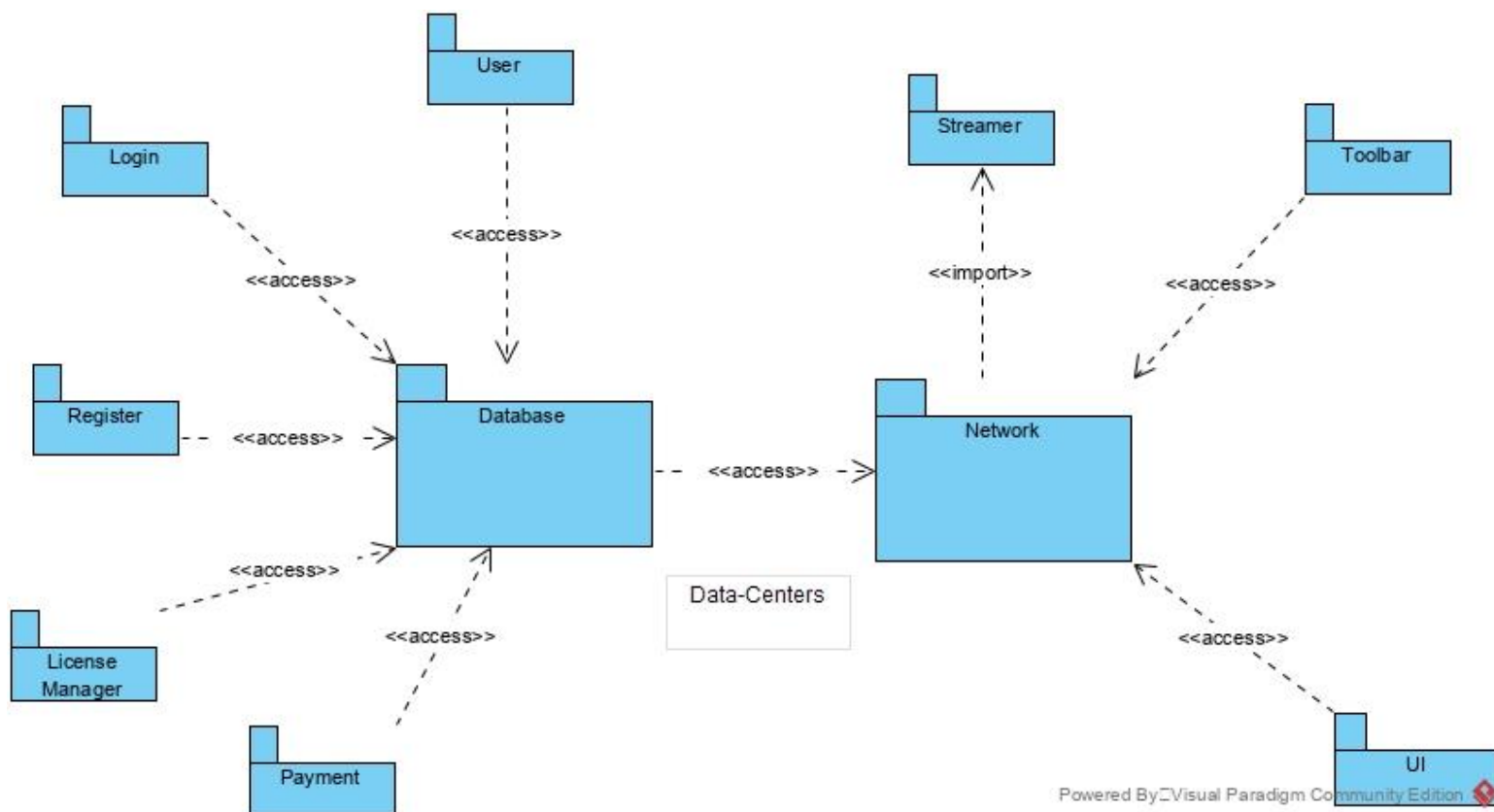


Sequence Diagram:



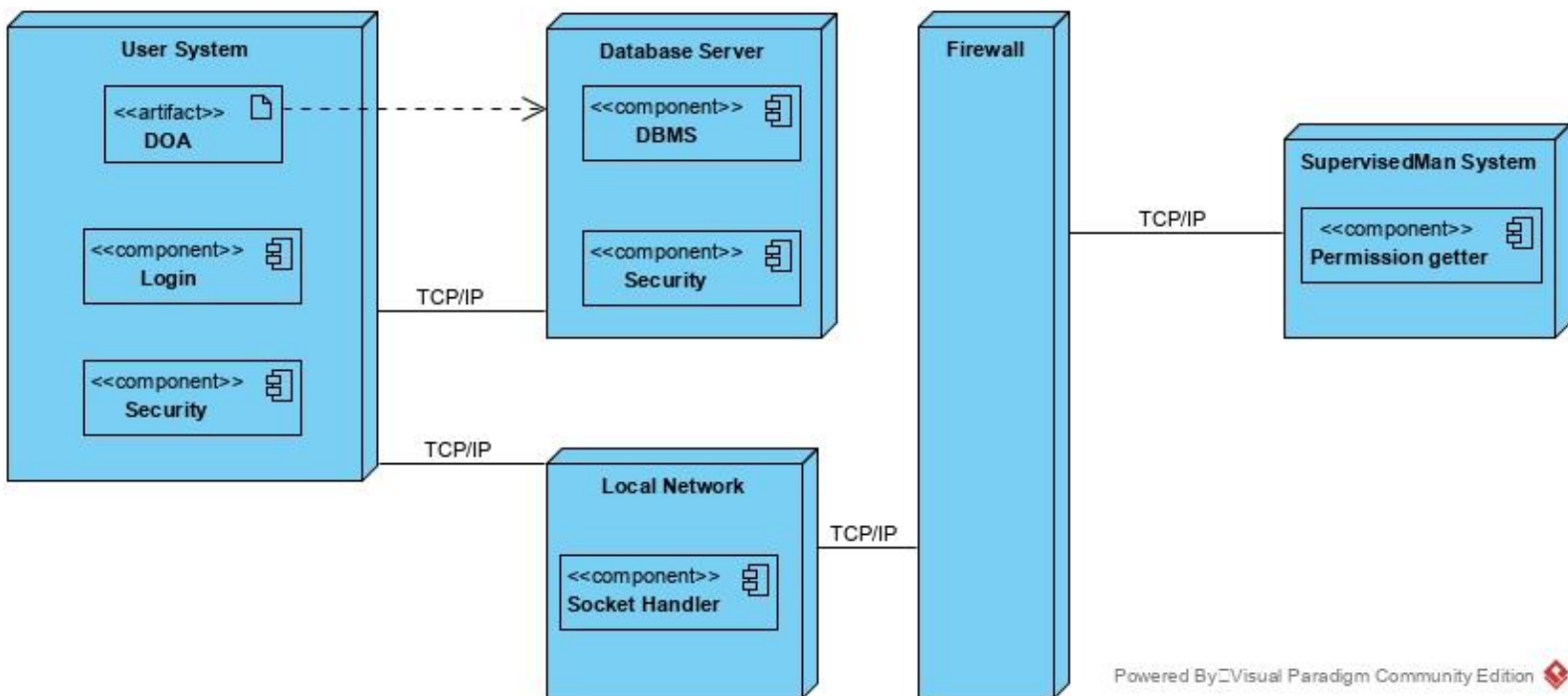
طرح دوم: در این طرح از استایل مشابه data-centered برای معماری استفاده شده است.

Package Diagram:

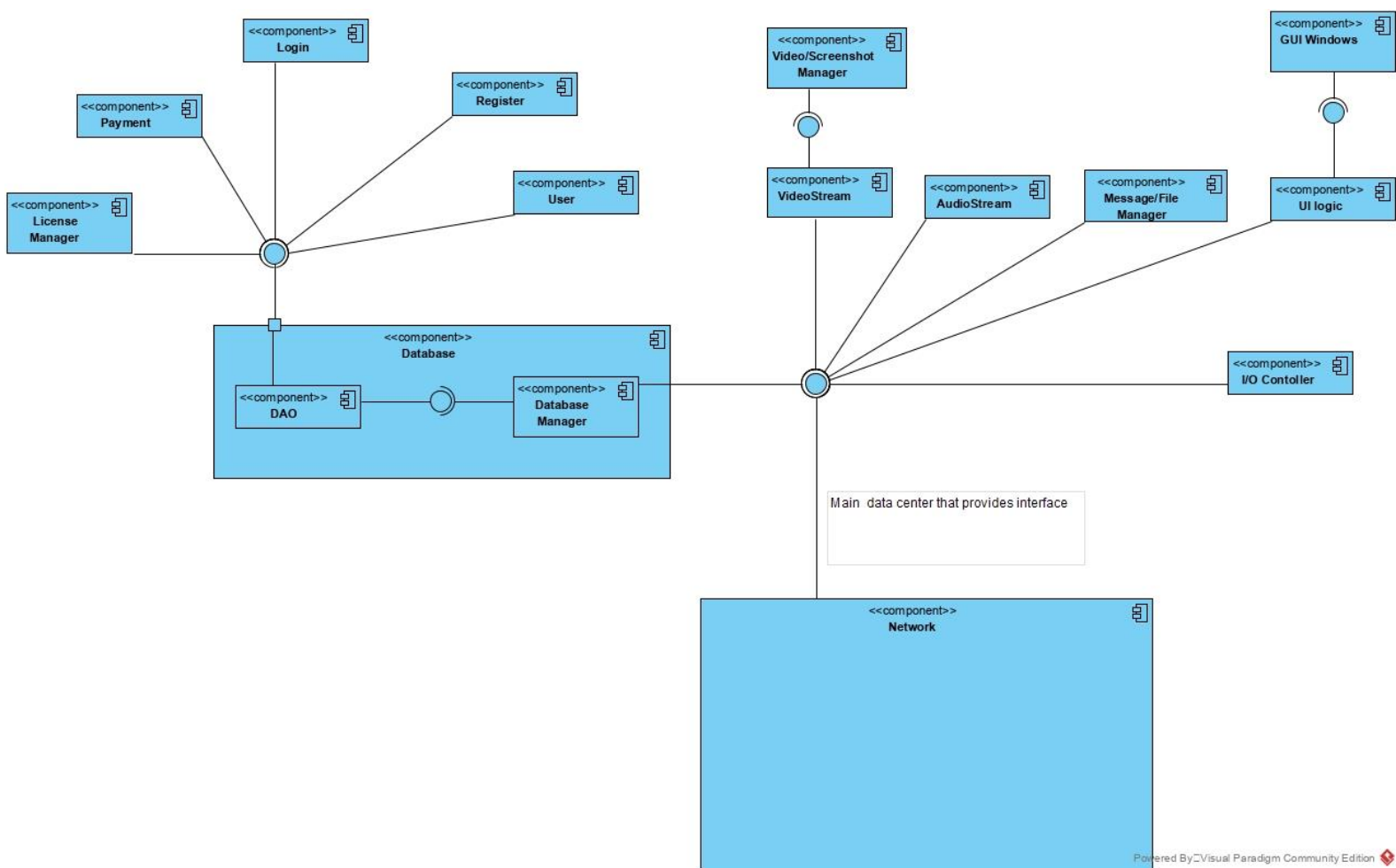


Powered By Visual Paradigm Community Edition

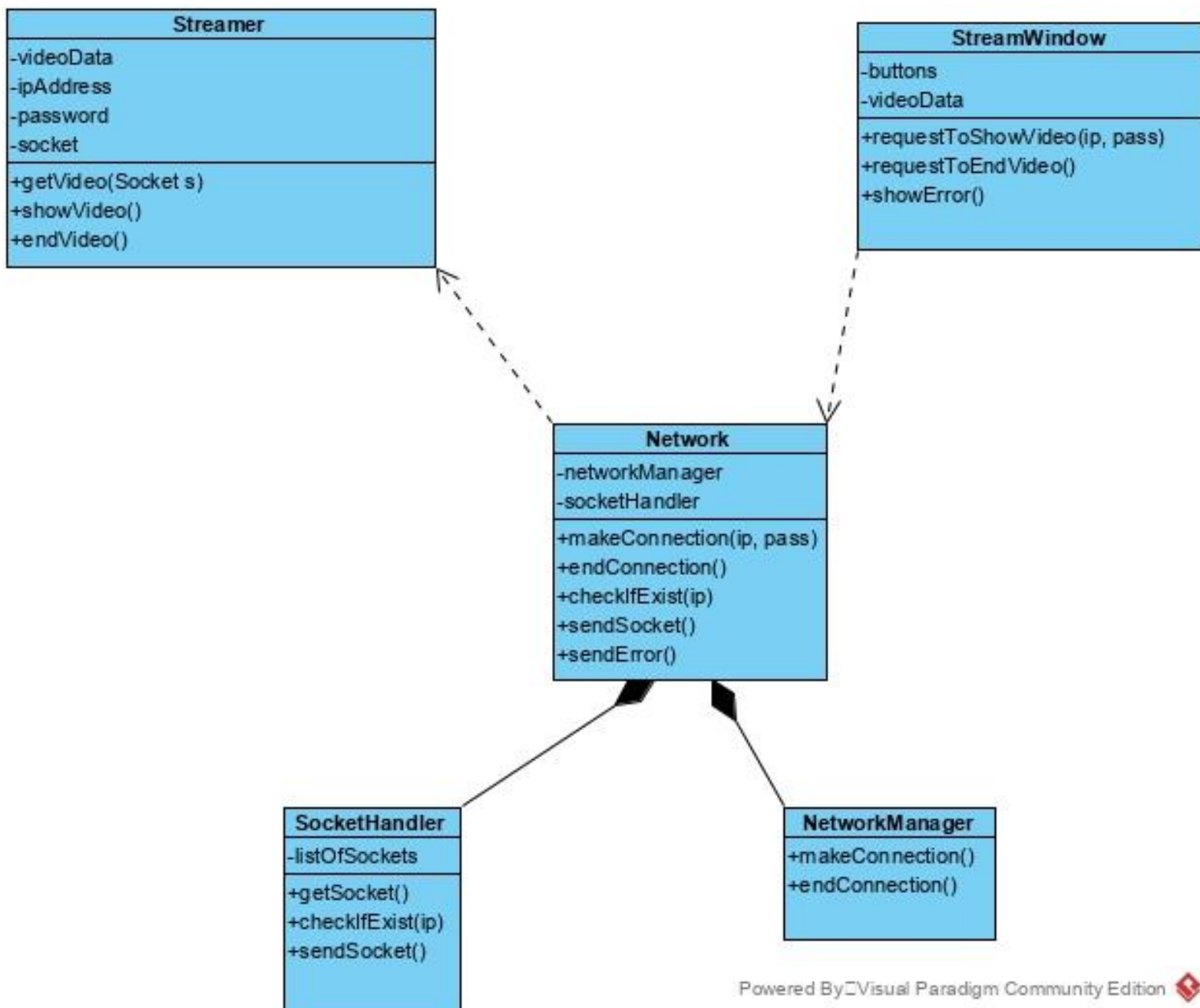
Deployment Diagram:



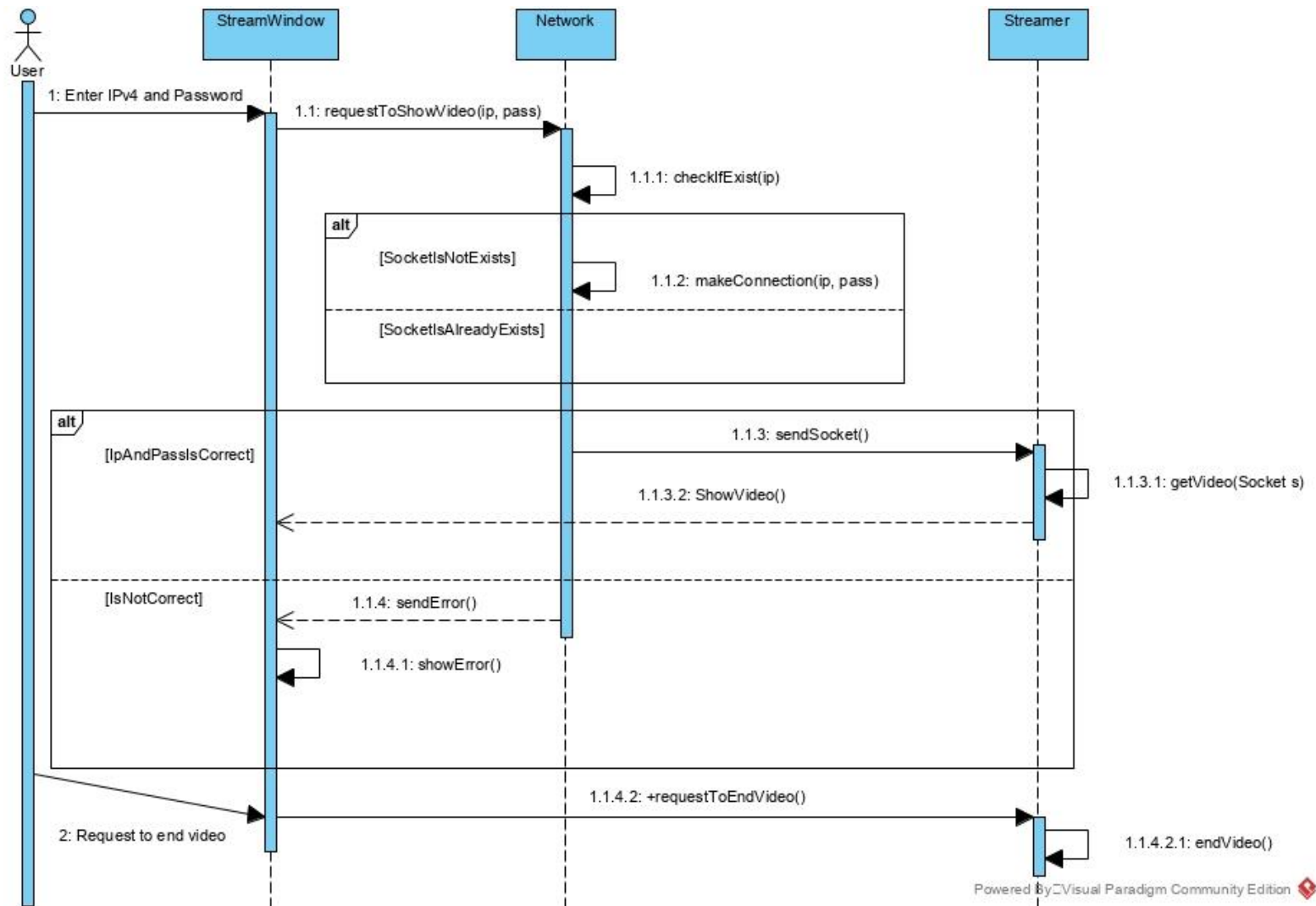
Component Diagram:



Class Diagram:



Sequence Diagram:



۴-۵- خروجی Technical Review

مسئول جلسه :

نظرات:

- مهیار سلامت: استفاده از معماری لایه ای قابل فهم تره و پیاده سازی راحت تره
- حسن کاظمی تهرانی: من با نظر مهیار موافقم و به نظرم Maintenance اش هم بهتره
- سروش مهربان براغوش: به نظرم معماری لایه ای Response time کمتری نسبت به Data-centered داره و بهتره از Data-centered استفاده کنیم.
- محمد مهدی منتظر: برنامه ما تحت شبکه داخلی هستش و تعداد کاربر ها از ۱۰۰ نفر بیشتر نمیشوند و Response-time اهمیت خیلی زیادی ندارد .

جمع بندی نظرات و تصمیم نهایی:

در بین دو طرح معماری داده شده، طرح لایه ای موافق گروه می باشد. زیرا که فهم آن برای گروه ساده تر است و با توجه به تعداد کاربر کم response time قابل قبولی نیز دارد.

فصل ششم

مدیریت پروژه

۱-۶- انتخاب ابزار CD/CI

	 Jenkins	 circleci	 TeamCity	 Bamboo	 GitLab
Open source	Yes	No	No	No	No
Ease of use & setup	Medium	Medium	Medium	Medium	Medium
Built-in features	3/5	4/5	4/5	4/5	4/5
Integration	★★★★★	★★★★	★★★★★	★★★★	★★★★★
Hosting	On premise & Cloud	On premise & Cloud	On premise	On premise & Bitbucket as Cloud	On premise & Cloud
Free version	Yes	Yes	Yes	Yes	Yes
Build Agent License Pricing	Free	From \$39 per month	From \$299 one-off payment	From \$10 one-off payment	From \$4 per month per user
Supported OSs	Windows, Linux, macOS, Unix-like OS	Linux or MacOS	Windows, Linux, macOS, Solaris, FreeBSD and more	Windows, Linux, macOS, Solaris	Linux distributions: Ubuntu, Debian, CentOS, Oracle Linux

پس از بررسی های انجام شده، ابزار های محبوب CD/CI در در نمودار فوق قابل مشاهده می باشد. پس از گفتگو بین اعضای تیم، ابزار مناسب Jenkins معرفی شد. زیرا:

- برخلاف باقی ابزار ها، این ابزار رایگان میباشد و از لحاظ اخلاقی استفاده آن مناسب است.
- بر روی سیستم های مختلف قابل اجراست و با توجه به این که فروش مهربان در گروه ویندوز دارد، می تواند با آن کار کند.
- اینترفیس آن use-friendly است
- کانفیگ کردن آن آسان است
- برای وضعیت build نوتیفیکیشن نشان میدهد.

۲-۶- تکمیل موارد متدلوژی چابک

۱-۲-۶- Sprint Map

با توجه به این که User Story ها بر اساس اولویت مرتب شده اند، از ۳ تا Sprint ۲ هفته ای برای توسعه انتخاب شده است، به طوری که :

۱- Sprint اول: یوزر استوری های ۱ تا ۶ انجام می شود.

۲- Sprint دوم: یوزر استوری ۷ تا ۱۵ انجام می شود.

۳- Sprint سوم: از ۱۶ تا ۲۱ انجام می شود.

User Stories - ۶-۲-۲

ID: us_1		
عنوان : گرفتن Permission		
<p>شرح: به عنوان کاربر، می‌خواهم اجازه وصل شدن به مقصد را داشته باشم تا امکان وصل شدن برایم فراهم شود و در صورت عدم دریافت permission به تعداد ۵ مرتبه، تا ۵ دقیقه توانایی اتصال مجدد را نداشته باشد</p>		<p>تأییدیه:</p> <p>Scope: اگر Permission از طرف سیستم مقصد داده شود امکان وصل شدن فراهم شود</p> <p>Not Scope: اگر Permission از طرف سیستم مقصد داده نشد، مدام برای گرفتن permission تلاش نکند.</p>
داستان های مربوطه: داشتن IP Address سیستم مقصد		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: بالا
وضعیت: در دست تولید	تلاش تخمینی: ۱ نفر/ساعت	اندازه دشواری: ساده
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_2		
عنوان : قابلیت Log گیری از کیبورد		
<p>شرح: به عنوان کاربر، میخوامم توانایی ذخیره Log کیبورد کامپیوتر را داشته باشم تا بتوانم آن را مشاهده کنم.</p>		<p>تأییدیه:</p> <p>Scope: پس از رسیدن به مرز ۵۰ کلمه عبارات در یک فایل txt ذخیره شوند تا بتوانیم آن را مشاهده کنیم.</p> <p>Not Scope: تا زمانی که به مرز ۵۰ کلمه نرسیده است اطلاعات داخل buffer ذخیره نشوند و اگر کاربر تقاضای خاموش کردن و یا restart کردن را داشت در صورت وجود داده در buffer، اطلاعات در فایل txt ذخیره شوند.</p>
داستان های مربوطه: وجود داده تایپ شده در کیبورد		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط (پیش نیاز us بعدی)
وضعیت: در دست تولید	تلاش تخمینی: ۶ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_3		
عنوان: اتصال به کامپیوتر هدف ثابت در شبکه داخلی		
<p>شرح: به عنوان کاربر، میخواهم به یک سیستم داخل شبکه ای از قبل تعریف شده وصل شوم تا Log کیبورد کاربر مقصد برایم ارسال شود و در کادری به نمایش در بیاید</p>		<p>تأییدیه:</p> <p>Scope: پس از نوشتن عبارت connect بتوان به سیستم مقصدی که از قبل تعریف شده است وصل شد و پس از هر ۲ دقیقه log کیبورد کامپیوتر مقصد در کنسول به نمایش دربیاید.</p> <p>پس از نوشتن عبارت end، تمامی داده های buffer از کامپیوتر مقصد ارسال و پس از دریافت log های باقی مانده اتصال قطع شود.</p> <p>Not Scope: پس از نوشتن عبارت connect در صورت نداشتن Permission اخطار عدم اجازه به نمایش در بیاید.</p> <p>اگر حین اتصال ارتباط قطع شود، پیامی جهت قطع شدن ناگهانی به نمایش در بیاید.</p>
داستان های مربوطه: داشتن IP Address سیستم مقصد، داشتن Permission وصل شدن، توانایی Log گرفتن		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: بالا
وضعیت: در دست تولید	تلاش تخمینی: ۴ نفر/ساعت	اندازه دشواری: ساده
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_4		
عنوان: وصل شدن به سیستم مقصد دلخواه در شبکه داخلی (با داشتن IP Address)		
<p>شرح: به عنوان کاربر می‌خواهم به IP Address مورد نظر خود وصل شوم تا بتوانم هر موقع که خواستم سیستم مقصد را عوض کنم.</p>		<p>تأییدیه:</p> <p>Scope: کاربر با وارد کردن IP Address کاربر مقصد در کنسول، به شبکه درخواست می‌دهد که اتصال برقرار شود و شبکه در صورت گرفتن permission از کاربر مقصد اتصال را برقرار و پس از اتصال مانند us_3 عمل می‌کند.</p> <p>Not Scope: اگر IP Address در شبکه داخلی موجود نباشد، شبکه خطاری جهت اشتباه بودن IP Address به نمایش در بیاید.</p> <p>در صورت نداشتن permission و قطع ناگهانی مانند us_3 عمل کند.</p> <p>اگر بیش از ۱۰ اتصال ناموفق داشتیم سیستم به مدت ۵ دقیقه Lock شود.</p>
داستان های مربوطه: داشتن IP Address سیستم مقصد، داشتن قابلیت وصل شدن شبکه داخلی		
نویسنده: Enigma	تاریخ: ۹۸/۸/۱۳	اولویت: بالا
وضعیت: در دست تولید	تلاش تخمینی: ۲ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_5		
عنوان: وجود password برای هر سیستم مقصد در نرم افزار		
<p>شرح: به عنوان کاربر میخوام در کنار IP address سیستم مقصد، password نیز برای هر سیستم داشته باشد تا هر کاربری توانایی درخواست وصل شدن به هر کاربر دیگر در شبکه را نداشته باشد.</p>		<p>تأییدیه:</p> <p>Scope: با وارد کردن IP Address و Password توسط کاربر، شبکه تلاش میکند به IP Address وصل شود و در صورت وصل شدن password را از کاربر مقصد گرفته و با پسورد وارد شده کاربر چک میکند و در صورت انطباق، دسترسی را به کاربر میدهد و پس از آن مانند us_3 عمل کند.</p> <p>Not Scope: رعایت us_4 not scope + قفل کردن سیستم در صورت وارد کردن پسورد اشتباه بیش از ۵ درخواست متعدد</p>
داستان های مربوطه: داشتن IP Address سیستم مقصد، داشتن قابلیت وصل شدن شبکه داخلی با IP Address		
نویسنده: Enigma	تاریخ: ۹۸/۸۹/۱۱	اولویت: بالا
وضعیت: در دست تولید	تلاش تخمینی: ۲ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_6		
عنوان : مشاهده سیستم مقصد به صورت Live stream در شبکه داخلی (بدون صدا)		
<p>شرح: به عنوان کاربر می‌خواهم پس از وصل شدن به سیستم مقصد به صورت Live Stream سیستم مقصد را مشاهده کنم تا در هر لحظه مطلع باشم چه اموری در حال انجام است.</p>		<p>تأییدیه:</p> <p>Scope: پس از وصل شدن به سیستمی که دارای Permission هستیم، به صورت Live Stream یک پنجره باز گردد و صفحه کاربر مقصد را از شبکه گرفته و نشان می‌دهیم و با بستن پنجره ارتباط قطع می‌شود.</p> <p>Not Scope: اگر به سیستم مقصد Permission نداریم، پنجره ای جهت نمایش باز نگردد.</p> <p>اگر حین کار اتصال قطع شود، پنجره بایستی بسته شود و اخطار قطع ناگهانی در کنسول به نمایش در بیاید.</p>
داستان های مربوطه: داشتن IP Address سیستم مقصد، قابلیت وصل شدن به سیستم مقصد با شبکه داخلی		
نویسنده: Enigma	تاریخ: ۹۸/۸/۱۳	اولویت: بالا
وضعیت: در انتظار تولید	تلاش تخمینی: ۱۵ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_7		
عنوان : گرفتن اسکرین شات از کامپیوتر مقصد		
<p>شرح: به عنوان کاربر میخواهم هر موقع که خواستم از سیستم مقصد اسکرین شات بگیرم تا یک تصویر از سیستم مقصد داشته باشم.</p>		<p>تأییدیه:</p> <p>Scope: کاربر تقاضای درخواست عکس گرفتن را میدهد، اسکرین شات گرفته می شود و در سیستم مبدا ذخیره میشود.</p> <p>Not Scope: اگر Permission ذخیره شدن در سیستم مبدا را نداشته باشیم و یا حافظه پر شده باشد خطاری به نمایش در بیاید و قابلیت ذخیره در درایو دیگر موجود باشد.</p> <p>اگر به محض عکس گرفتن سوکت قطع شود، عکس ناقص ذخیره نشود.</p> <p>با درخواست screenshot، از صفحه کاربر دیگری عکس نگیرد.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن، وجود دیتابیس		
نویسنده: Enigma	تاریخ: ۹۸/۸/۱۳	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۶ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکار سازی فرآیند تست: PyBuilder		

ID: us_8		
عنوان : ضبط ویدیو		
<p>شرح: به عنوان کاربر میخوام ویدیو از صفحه سیستم مقصد رکورد کنم تا مستند آن در سیستم من ذخیره شده شود.</p>		<p>تأییدیه:</p> <p>Scope: کاربر تقاضای ضبط ویدیو را میکند و فرایند ضبط آغاز میشود، سپس کاربر تقاضای قطع ویدیو را میکند و ویدیو مورد نظر در سیستم ذخیره می شود.</p> <p>Not Scope: اگر Permission ذخیره شدن در سیستم مبدا را نداشته باشیم و یا حافظه پر شده باشد خطاری به نمایش در بیاید و قابلیت ذخیره در درایو دیگر موجود باشد.</p> <p>اگر هنگام ضبط سوکت قطع شود، ضبط ویدیو تمام و در دیتابیس ذخیره شود.</p> <p>با کلیک آیکون ضبط، از صفحه کاربر دیگری ویدیو نگیرد.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن، وجود دیتابیس		
نویسنده: Enigma	تاریخ: ۹۸/۸/۱۳	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۸ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_9		
عنوان: ارسال پیام به سیستم مقصد		
<p>شرح: به عنوان کاربر میخواهم پیامی را به سیستم مقصد ارسال کنم تا پیام من در سیستم مقصد به نمایش در بیاید.</p>		<p>تأییدیه:</p> <p>Scope: کاربر تقاضای ارسال پیام میکند و یک text box جهت ارسال پیام باز میشود، کاربر پیام مورد نظر را وارد و ارسال میکند و به کمک شبکه به سیستم مقصد ارسال و مشاهده میشود.</p> <p>Not Scope: اگر هنگام ارسال socket قطع شود پیام ناقص ارسال نشود.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن.		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۶ نفر/ساعت	اندازه دشواری: دشوار
ابزار خودکار سازی فرآیند تست: PyBuilder		

ID: us_10		
عنوان: انتقال فایل از سیستم مبدا به سیستم مقصد		
<p>شرح: به عنوان کاربر می‌خواهم یک فایل از سیستم مبدا به سیستم مقصد ارسال کنم تا فایل مورد نظر در سیستم مقصد قرار گیرد.</p>		<p>تأییدیه:</p> <p>Scope: کاربر فایل مربوطه را درخواست به ارسال می‌دهد و فایل به کمک شبکه به سیستم مقصد ارسال می‌شود و در سیستم کاربر مقصد ذخیره می‌شود.</p> <p>Not Scope: هنگام قطع سوکت خطاری به نمایش در بیاید و فایل به کل ارسال نشود.</p> <p>اگر در سیستم مقصد permission ذخیره نداشته باشیم و یا حافظه پر باشد، کاربر مقصد انتخاب میکند که در کجا ذخیره شود.</p>
داستان های مربوطه: وجود قابلیت live stream، وجود داده در سیستم مبدا		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۱۲ نفر/ساعت	اندازه دشواری: دشوار
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_11		
عنوان: کنترل مانیتور سیستم هدف		
<p>شرح: به عنوان کاربر می‌خواهم قابلیت کنترل مانیتور سیستم هدف را داشته باشم تا هر موقع که خواستم آن را سیاه کنم.</p>		<p>تأییدیه:</p> <p>Scope: کاربر مبدا تقاضای سیاه کردن مانیتور سیستم هدف را می‌دهد و مانیتور سیستم هدف سیاه میشود، در صورت درخواست دوباره مانیتور سیستم هدف به حالت قبل باز میگردد</p> <p>Not Scope: در صورت فعال شدن این گزینه، نباید پنجره live stream نیز سیاه شود.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن.		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۱۲ نفر/ساعت	اندازه دشواری: دشوار
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_12		
عنوان : وصل شدن به چند سیستم دلخواه در شبکه داخلی		
<p>شرح: به عنوان کاربر میخواهم به چند سیستم وصل شوم تا بتوانم چند کاربر را همزمان مدیریت کنم.</p>		<p>تأییدیه:</p> <p>Scope: کاربر IPv4 و Password کاربر مقصد را وارد و تقاضای اتصال به سیستم مقصد را میکند، شبکه از سیستم مقصد Permission میخواهد و در صورت دادن Permission کاربر مقصد، سوکت مربوطه ارسال و پنجره Live Stream جدید باز میشود.</p> <p>Not Scope: اگر یک پنجره را بستیم، تمامی اتصالات قطع نشود و تنها اتصال مربوط به آن پنجره از بین برود.</p>
داستان های مربوطه: قابلیت وصل شدن به شبکه داخلی با IP Address و پسورد		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۳ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکار سازی فرآیند تست: PyBuilder		

ID: us_13		
عنوان: شنیدن صدای خروجی از کامپیوتر مقصد در شبکه داخلی		
<p>شرح: به عنوان کاربر می‌خواهم خروجی صوتی سیستم مقصد را داشته باشم تا خروجی ویدیو ها و آهنگ های در حال اجرا در سیستم را بشنوم.</p>		<p>تأییدیه:</p> <p>Scope: کاربر درخواست کنترل خروجی صدا را از نرم افزار میکند و نرم افزار تقاضا را به شبکه ارسال میکند، شبکه درخواست را به نرم افزار در سیستم مقصد منتقل میکند و از کاربر مقصد تقاضا می شود. در صورت تایید سوکت خروجی صدا به شبکه ارسال و به کاربر امکان کنترل نمایش داده میشود.</p> <p>Not Scope: اگر volume سیستم مقصد قطع شده باشد، نباید در شنیدن خروجی اختلال به وجود بیاید (باید مستقل باشد)</p> <p>هنگام فعال کردن خروجی صدا، باقی خروجی مربوط به سیستم های مقصد دیگر بایستی غیرفعال شوند تا تداخلی ایجاد نکند.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن. وصل شدن به چند سیستم مقصد در شبکه داخلی		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۹ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_14		
عنوان: کنترل ورودی کامپیوتر هدف		
<p>شرح: به عنوان کاربر میخوام قابلیت کنترل ورودی کامپیوتر (mouse, keyboard,...) هدف را داشته باشم تا بتوانم در سیستم هدف تغییری ایجاد کنم.</p>		<p>تأییدیه:</p> <p>Scope: کاربر درخواست کنترل ورودی مقصد را از نرم افزار میکند و نرم افزار تقاضا را به شبکه ارسال میکند، شبکه درخواست را به نرم افزار در سیستم مقصد منتقل میکند و از کاربر مقصد تقاضا می شود. در صورت تایید سوکت کنترل به شبکه ارسال و به کاربر امکان کنترل نمایش داده میشود.</p> <p>Not Scope: در صورت فعال شدن کنترل یه سیستم مقصد، باقی سیستم ها باید این قابلیت برایشان غیر فعال شود.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن. وصل شدن به چند سیستم مقصد در شبکه داخلی		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۲۵ نفر/ساعت	اندازه دشواری: دشوار
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_15		
عنوان : انتقال فایل از کامپیوتر مقصد به کامپیوتر مبدا		
<p>شرح: به عنوان کاربر یک فایل از سیستم مقصد را به سیستم خود ارسال کنم تا آن فایل را در سیستم خود داشته باشم.</p>		<p>تأییدیه:</p> <p>Scope: کاربر درخواست دریافت فایل را میدهد و نرم افزار از شبکه درخواست میکند، شبکه درخواست ارسال فایل از سیستم مقصد را میکند و فایل به شبکه و شبکه به سیستم مبدا ارسال میکند و در حافظه سیستم مبدا ذخیره میشود.</p> <p>Not Scope: هنگام قطع سوکت خطاری به نمایش در بیاید و فایل به کل ارسال نشود.</p> <p>اگر در سیستم permission ذخیره نداشته باشیم و یا حافظه پر باشد، کاربر انتخاب میکند که در کجا ذخیره شود.</p>
داستان های مربوطه: قابلیت کنترل سیستم مقصد		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: متوسط
وضعیت: در انتظار تولید	تلاش تخمینی: ۱۲ نفر/ساعت	اندازه دشواری: دشوار
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_16		
عنوان: گرفتن لیست app در یک سیستم		
<p>شرح: به عنوان کاربر میخوام لیست app های موجود در سیستم را داشته باشم تا بتوانم آن ها را رویت کنم</p>		<p>تأییدیه:</p> <p>Scope: کاربر مبدا به کمک نرم افزار درخواست دریافت app های موجود در سیستم مقصد را میدهد، شبکه درخواست را به سیستم مقصد میرساند و پس از Permission دادن کاربر مقصد، لیست App ها به کمک شبکه به سیستم مبدا ارسال و در فایل txt ذخیره می شود.</p> <p>Not Scope: اگر permission و یا فضای مورد نیاز جهت ذخیره را نداشتیم، یک پنجره جهت ذخیره در جای دیگر موجود باشد.</p>
داستان های مربوطه:		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۱	اولویت: پایین
وضعیت: در انتظار تولید	تلاش تخمینی: ۸ نفر/ساعت	اندازه دشواری: دشوار
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_17		
عنوان: ضبط ویدیو به محض انجام کار مشخص		
<p>شرح: به عنوان کاربر می‌خواهم به محض انجام یک کار مشخص توسط کاربر مشخص فرایند ضبط ویدیو آغاز شود تا اگر پشت سیستم نبودم و یا به هر دلیلی حواسم نبود، بتوانم بعداً ببینم طرف چه کاری کرده است.</p>		<p>تأییدیه:</p> <p>Scope: کاربر مبدا لیست app های دریافت شده از کاربر مقصد را مشاهده میکند و app های مورد نظر خود را انتخاب میکند.</p> <p>با اجرا شدن برنامه های مورد نظر کاربر تا زمان پایان برنامه فرایند ضبط میشود و در سیستم کاربر مبدا ذخیره می شود.</p> <p>Not Scope: اگر به هر دلیلی ارتباط قطع شد، فرایند ضبط قطع شود و همچنین با اجرای برنامه های تیک نخورده فرایند ضبط صورت نگیرد.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن، وجود قابلیت مشاهده app ها.		
نویسنده: Enigma	تاریخ: ۹۸/۸/۱۳	اولویت: پایین
وضعیت: در انتظار تولید	تلاش تخمینی: ۱۲ نفر/ساعت	اندازه دشواری: دشوار
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_18		
عنوان : داشتن GUI دستگاه مبدا		
<p>شرح: به عنوان کاربر میخوامم رابطه کاربری گرافیکی داشته باشم تا بتوانم با کلیک بر گزینه ها با آن کار کنم.</p>		<p>تأییدیه:</p> <p>Scope: پنجره ای گرافیکی پس از باز شدن نرم افزار باز شود و فیلدی جهت وارد IP Address و Password جهت اتصال باشد. و کاربر با کلیک بر connect در صورت اتصال پنجره ای جهت live stream موجود باشد.</p> <p>Not Scope: تا زمانی که IPv4 به صورت فرمت وارد نشده است و پسورد وارد نشده است تلاش برای اتصال برقرار نشود.</p>
داستان های مربوطه: وجود قابلیت Live stream کردن		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۳	اولویت: پایین
وضعیت: در انتظار تولید	تلاش تخمینی: ۱۲ نفر/ساعت	اندازه دشواری: متوسط
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_19		
عنوان : Login کردن در نرم افزار		
<p>شرح: به عنوان کاربر، می خواهم با وارد کردن نام کاربری و پسورد به سیستم وصل شوم تا از امکانات آن بتوانم استفاده کنم و هرکسی نتواند از سیستم من از نرم افزار استفاده کند.</p>		<p>تأییدیه:</p> <p>Scope: قبل باز شدن قسمت اصلی نرم افزار، پنجره ای جهت وارد کردن یوزرنیم و پسورد باشد که پس از وارد کردن یوزر نیم و پسورد کاربر تعریف شده در سیستم، در صورت تطابق کاربر وارد و پنجره اصلی نمایان می شود و در غیر این صورت اخطار و تلاش مجدد نمایش داده شود.</p> <p>Not Scope: اگر بیش از ۵ بار تلاش مضاعف صورت گیرد، برنامه به مدت ۲ دقیقه Lock شود. اگر فیلدی خالی بود عملیات لاگین صورت نگیرد و از کاربر خواسته شود که آن را وارد کند.</p>
داستان های مربوطه: وجود کاربر در دیتابیس سیستم		
نویسنده: Enigma	تاریخ: ۹۸/۸/۱۳	اولویت: پایین
وضعیت: در دست تولید	تلاش تخمینی: ۳ نفر/ساعت	اندازه دشواری: ساده
ابزار خودکار سازی فرآیند تست: PyBuilder		

ID: us_20		
عنوان : وصل شدن به درگاه پرداخت برای گرفتن License		
<p>شرح: به عنوان کاربر، می خواهیم برای خرید license مبلغی را بپردازم تا بتوانم در نرم افزار ثبت نام کنم</p>		<p>تأییدیه:</p> <p>Scope: در پنجره Login عبارتی برای خرید license موجود باشد که کاربر با کلیک بر آن به درگاه پرداخت متصل و پس از پرداخت لایسنس جدید تولید شده و به کاربر نشان داده می شود تا بتواند کپی کند.</p> <p>Not Scope: پس از پرداخت به یک URL خاص هدایت نشود(جهت امنیت)</p>
داستان های مربوطه: درگاه پرداخت شاپرک، پنجره Login		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۳	اولویت: پایین
وضعیت: در دست تولید	تلاش تخمینی: ۵ نفر/ساعت	اندازه دشواری: ساده
ابزار خودکارسازی فرآیند تست: PyBuilder		

ID: us_21		
عنوان : ثبت نام در نرم افزار		
<p>شرح: به عنوان کاربر، میخواهم به عنوان کاربر جدید توانایی ثبت نام را داشته باشم تا از امکانات نرم افزار بهره مند شوم.</p>		<p>تأییدیه:</p> <p>Scope: در پنجره ثبت نام، کاربر نام، نام خانوادگی، نام شرکت و لایسنس خریداری شده را وارد میکند. سیستم ثبت نام لایسنس را چک و در صورت معتبر بودن لایسنس ثبت نام انجام می شود و پنجره لاگین باز می شود.</p> <p>Not Scope: اگر لایسنس معتبر نبود و یا تکراری بود عملیات ثبت نام انجام نگردد و خطاری به نمایش بیاید. اگر بیش از ۵ بار جهت ثبت نام تلاش گردد به مدت ۲ دقیقه lock شود</p> <p>اگر فیلدی خالی بود عملیات ثبت نام صورت نگیرد و از کاربر بخواهد که وارد کند.</p>
داستان های مربوطه: قابلیت Login کردن		
نویسنده: Enigma	تاریخ: ۹۸/۹/۱۳	اولویت: پایین
وضعیت: در دست تولید	تلاش تخمینی: ۳ نفر/ساعت	اندازه دشواری: ساده
ابزار خودکارسازی فرآیند تست: PyBuilder		

DoD -۶-۲-۳

- Sprint اول:

- تست های اتوماتیک
- قبول کردن نماینده مشتری
- چک کردن Response Time

- Sprint دوم:

- تست های اتوماتیک
- قبول کردن نماینده مشتری
- چک کردن Response Time
- Cyclomatic Complexity از حد مورد نظر کمتر باشد

- Sprint سوم:

- تست های اتوماتیک
- قبول کردن نماینده مشتری
- چک کردن Response Time
- Refactor کردن کد ها
- Deploy کردن کد

DoR -۶-۲-۴

- Sprint اول:

- تخمین تلاش در Sprint
- وابستگی ها معلوم بشه تا یه وابستگی خارجی نیاد جلوی کار رو بگیره
- مشخص کردن Acceptance Criteria

- Sprint دوم:

- تخمین تلاش در Sprint
- مشخص کردن Performance Criteria
- مشخص کردن Acceptance Criteria

- Sprint سوم:

- تخمین تلاش در Sprint
- Story که کار میکنیم ارزش بیزینسی داشته باشه
- مشخص کردن Acceptance Criteria

۵-۲-۶- خروجی یک جلسه از Sprint Review

با توجه به این که تیم ما فرصت پیاده سازی را نداشت، جلسه Sprint Review تشکیل نشد.