

سنة الفجر

# **پروژه درس معماری نرم افزار**

**عنوان پروژه : بررسی اپلیکیشن iBartar**

**(اپلیکیشن یافتن کسب کار مبتنی بر موقعیت مکانی)**

**استاد راهنما : جناب آقای دکتر رضایی**

**دانشجویان :**

**بهناز اسلامی**

**نسیبه قویدل**

**زمستان ۹۵**

## ۱- هدف:

در این مستند، مروری جامع بر ساختار و معماری آی برتر، توسعه داده شده توسط تیم نرم افزاری گروه تبلیغاتی پیک برتر، خواهیم داشت. این مستندات با استفاده از دیدگاه ۴+۱ تولید شده است.

## ۲- معرفی اپلیکیشن انتخاب شده :

آی برتر برای صاحبان مشاغل و برای شهروندان به جهت پیدا کردن راهی برای خدمات و کسب و کار محلی مناسب است. به این صورت که با استفاده از آی برتر می توان با صرف کمترین زمان و هزینه کسب و کار مورد نیاز و اخبار مرتبط با آن را یافته و دنبال کنند. همچنین به صاحبان مشاغل کمک می کند با وارد کردن اطلاعات خود به صورت رایگان مورد توجه، استفاده کنندگان به صورت ویژه قرار بگیرند.

در واقع آی برتر همراه شهروندانی است که در شهر و به صورت روزانه تردد کرده و در شهر با توجه به نیازهای خود همانند فروشگاه، بیمارستان، دفاتر پستی، رستوران، کافی شاپ، پمپ بنزین، بانک یا هرگونه مرکز یا مکان دیگری را جستجو کرده و راههای دستیابی سریع را بیابند.

با توجه به مطالب فوق مبنی بر اهمیت این نرم افزار و راهگشایی آن در امور صاحبان مشاغل و شهروندان راههای دسترسی به این " آی برتر " به دو صورت ذیل می باشد :

۱- از طریق وب سایت مربوط به آن و

۲- از طریق اپلیکیشن موبایلی مربوط به آن در دونسخه آندروید و IOS

### ۳- چشم انداز:

به منظور مستندسازی تمام جنبه های معماری نرم افزار، در ادامه ی انجام فعالیتهای مربوط به پروژه؛ بخشهای زیر به ترتیب و به مرور زمان اضافه می گردند :

- تعریف قیدهای سیستم و تاکتیکهای استفاده شده
- تعریف نیازمندیهای کاربردی سیستم که بر معماری سیستم موثر می باشند.
- نمودارهای **Use Case**
- تعریف لایه ها و زیر سیستمهای اپلیکیشن
- تعریف فرآیندهای سیستم
- مدل سازی نحوه استقرار سیستم با استفاده از **Deployment Models**

#### ۴- نیازمندیها و قیدها

در این بخش نیازمندیها و اهداف سیستم که بر روی معماری نرم افزار موثرند، را توصیف کرده و سپس تاکتیکهای استفاده شده شرح داده می شوند:

- پلتفرم

آی برتر هم به صورت وب سایت و هم به صورت اپلیکیشن موبایل ( **Android** و **iOS** ) ارائه میشود. هسته اصلی این اپلیکیشن به زبان **Php**، (نسخه ۵,۶) نوشته شده است و بر روی **Webserver ۲,۲** **Apache** مستقر می باشد.

کلاینتها ( وبسایت و اپلیکیشنهای موبایل ) از طریق **API RESTful** با سرور در ارتباط هستند. جهت انجام امور مربوط به ذخیره سازی اطلاعات در آن از **MySQL** نسخه ۶,۵، و نمایه گذاری اطلاعات به منظور جستجو از **Elasticsearch** نسخه ۲,۰ و از **Redis ۰,۳** به عنوان حافظه نهان استفاده شده است.

- احراز هویت

قبل از پردازش هر درخواست، کاربران می بایست، احراز هویت گردند، به همین منظور برای احراز هویت کاربران از **oAuth ۰,۲** در کلاینت ها استفاده میشود.

- دسترس پذیری

حوزه عملکرد آی برتر در فاز اول استان تهران و در فاز دوم، تمامی شهرهای ایران بوده است. به همین خاطر دسترس پذیری اپلیکیشن برای شهروندان ایرانی در اولویت قرار دارد. لذا برای به وجود آوردن این دسترس پذیری سرورهای اپلیکیشن در داخل ایران با پهنای باند بالا انتخاب شده اند. برای دسترس پذیری مجموعه سرورها از مجموعه تاکتیکهای **Repair and Preparation** استفاده شده است.

از جمله این تاکتیک ها می توان **Exception**، **Passive Redundancy**، **Active Redundancy** و **Rollback**، **Handling** و **Degregation** اشاره کرد.

در سرور اصلی برای جلوگیری از بروز خطا، از **Exception Handling** و برای جلوگیری از **failed** شدن سایر مولفه های سرور در هنگام وقوع و رخ دادن خطا در یک از مولفه های سرور از تاکتیک **Rollback** استفاده می شود. بعبارتی در تاکتیک **Exception Handling** سعی بر این است که خطاهای رخ داده شده **Mask** بشوند.

در تاکتیک **Rollback** نیز همواره در حال گرفتن **Snapshot** از سیستم در وضعیت **stable** هستیم، تا در صورت بروز خطا سریعاً به وضع مطلوب قبلی بازگردیم. برای جلوگیری از عدم سرویس دهی به برخی از سرویس گیرندگان، در ساعات پر ترافیک، از تاکتیک **Degregation** استفاده میشود. در این تاکتیک در ساعات اوج ترافیک امکاناتی که دارای اهمیت کمتری هستند، (همانند: نمایه گذاریهای آنلاین) غیرفعال میشوند تا قسمتهای اصلی اپلیکیشن دچار مشکل و اختلال نگردند و همواره بتوانند پاسخگوی درخواستهای سرویس گیرنده گان در هر ساعتی از شبانه روز باشند.

### • سرعت عملکرد

با توجه به اینکه بیشترین کاربرد استفاده از این اپلیکیشن بر روی گوشی های موبایل و از طریق شبکه اینترنت تلفن همراه است، در نتیجه پاسخ هایی که از سمت سرور ارسال میشوند می بایست دارای کمترین حجم و بیشترین سرعت ممکن باشند. لذا برای اینکار از **JSON** جهت انتقال اطلاعات استفاده میشود که به نسبت **XML** دارای سربار کمتری است.

در واقع سرویس گیرندگان از طریق بستر **RESTful** از سرور اپلیکیشن سرویس میگیرند. همچنین برای جستجو و مشاهده در اطلاعات کسب و کارها - که اصلی ترین و مهمترین وظیفه این اپلیکیشن میباشد - از **Elasticsearch** استفاده شده که نتایج را تقریباً به صورت بلادرنگ (**RealTime**) ارسال میکند.

### • تداوم داده ها

تداوم داده ها در آی برتر از اهمیت بالایی برخوردار می باشد. به همین خاطر از دو تاکتیک **Active Redundancy** و **Passive Redundancy** که در بخش **دسترس پذیری** به آن اشاره گردید، استفاده شده. به همین منظور برای اطلاعات ذخیره شده در سرور **MySQL** تاکتیک **Passive Redundancy** و برای موتور جستجو تاکتیک **Active Redundancy** بکار گرفته شده است.

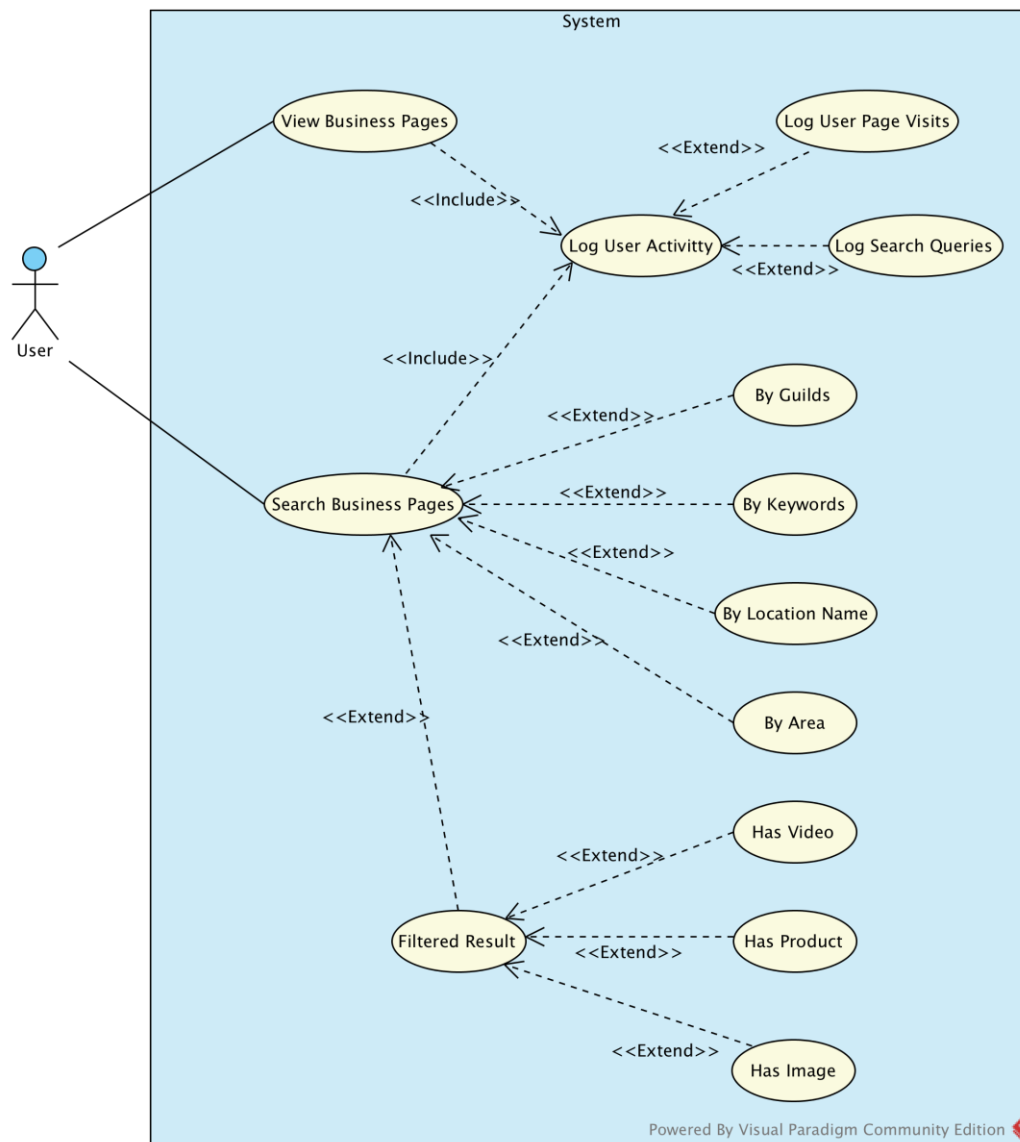
همواره جهت تداوم بخشیدن به داده ها یک سرور پشتیبان به جز سرور اصلی **MySQL** وجود دارد که موظف است، هر روز - زمانی که ترافیک و بار سرور در کمترین میزان خود قرار دارد - از اطلاعات **MySQL** پشتیبان گیری کرده و جداول خود را به روز رسانی نماید. که در این حالت، در صورت **failed** شدن سرور اصلی **MySQL** به سرور جایگزین درخواست خود را ارسال میکند.

برای موتور جستجو دو سرور جداگانه وجود دارد، که این دو سرور به صورت **Active** اطلاعات خود را با یکدیگر **Sync** می نمایند.

## ۵- Use-Case View :

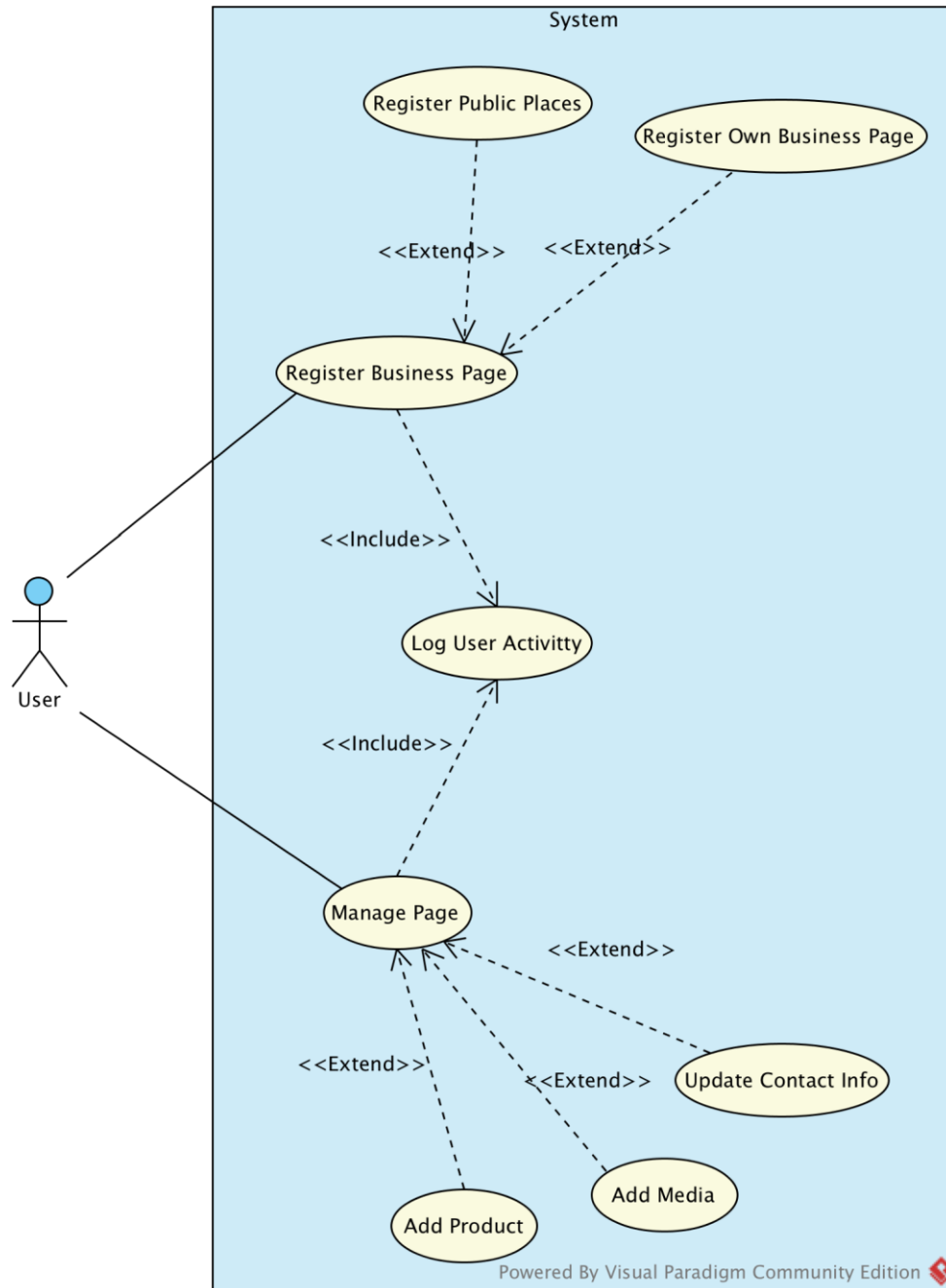
در این بخش به بررسی مدل‌های مورد کاربرد و سناریوهای اصلی و کلیدی اپلیکیشن پرداخته می‌شود.

### • جستجوی کسب و کار :

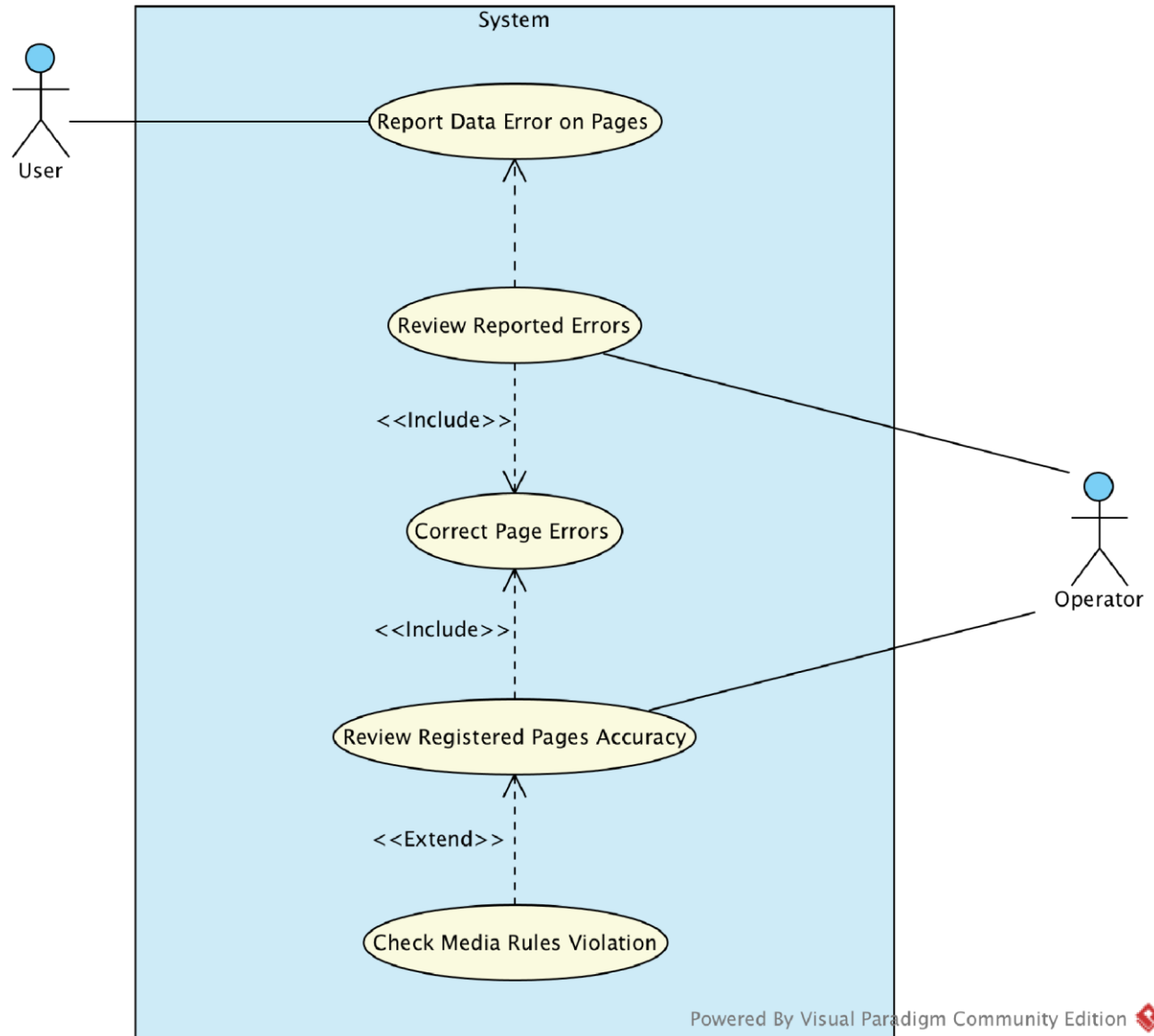




• ثبت و مدیریت کسب و کار :



• بررسی صحت اطلاعات :



## ۶- Logical View :

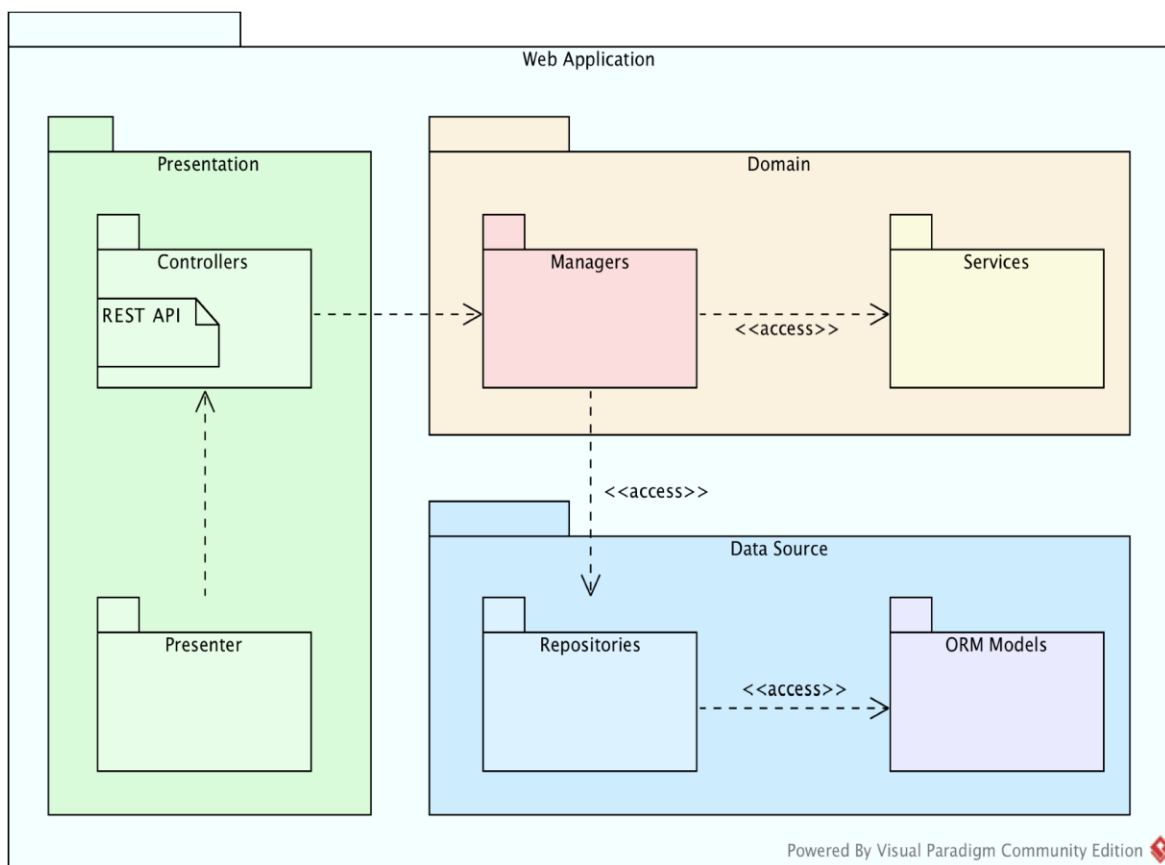
### • چشم انداز :

معماری هسته اصلی آی برتر، از رویکرد Design Driven Domain برگرفته شده است. این معماری شامل سه لایه زیر با وظایف مشخص شده، است :

۱- لایه Presentation : این لایه شامل کلاس ها و مؤلفه های مرزی است که وظیفه آنها دریافت درخواستها از سمت کلاینتها و ارائه پاسخ به آنها است. این لایه از دو بخش Controllers و Presenters تشکیل شده است.

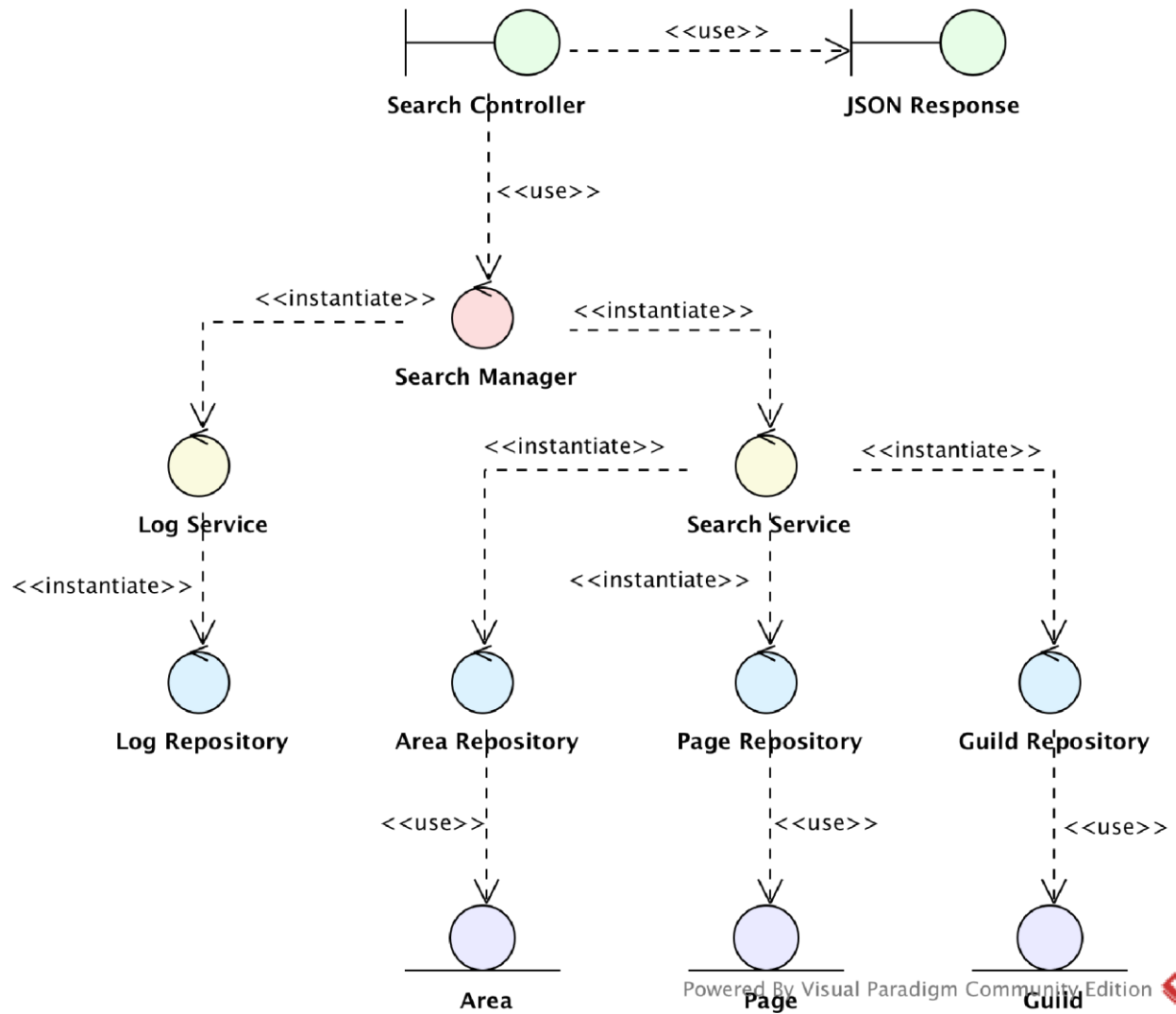
۲- لایه Domain : منطق اپلیکیشن در این لایه قرار میگیرد و شامل کلاسهای کنترلی است. این لایه از دو بخش Services و Managers تشکیل شده است.

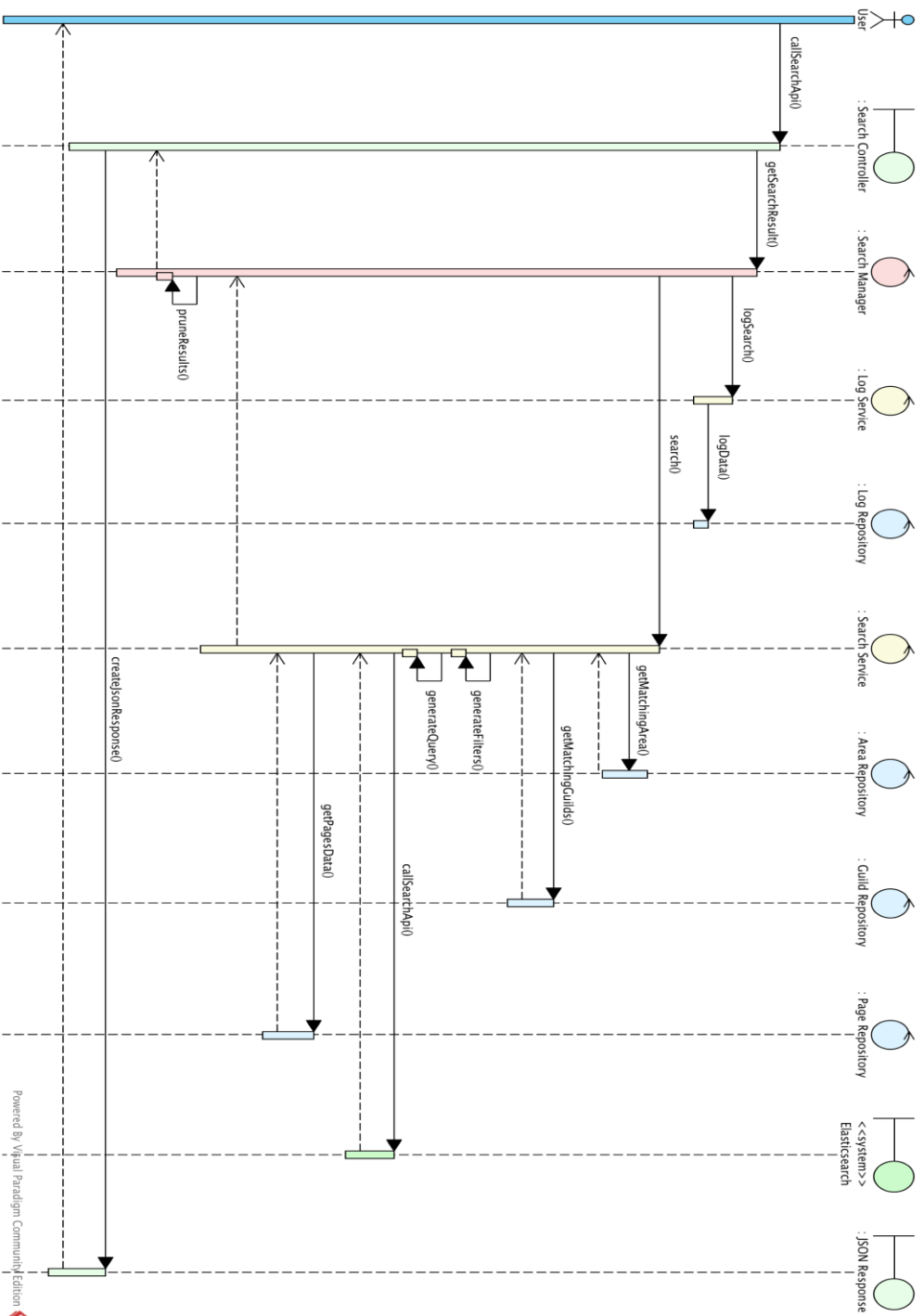
۳- لایه Source Data : در این لایه مدل های ORM و Entity های سیستم قرار میگیرند که وظیفهشان ذخیره سازی و بازیابی اطلاعات است. در این لایه تعدادی کلاس کنترلی وجود دارد که بر روی اطلاعات ورودی و خروجی پردازش انجام میدهند، که آنها را Repository مینامیم.



• جستجوی کسب و کار

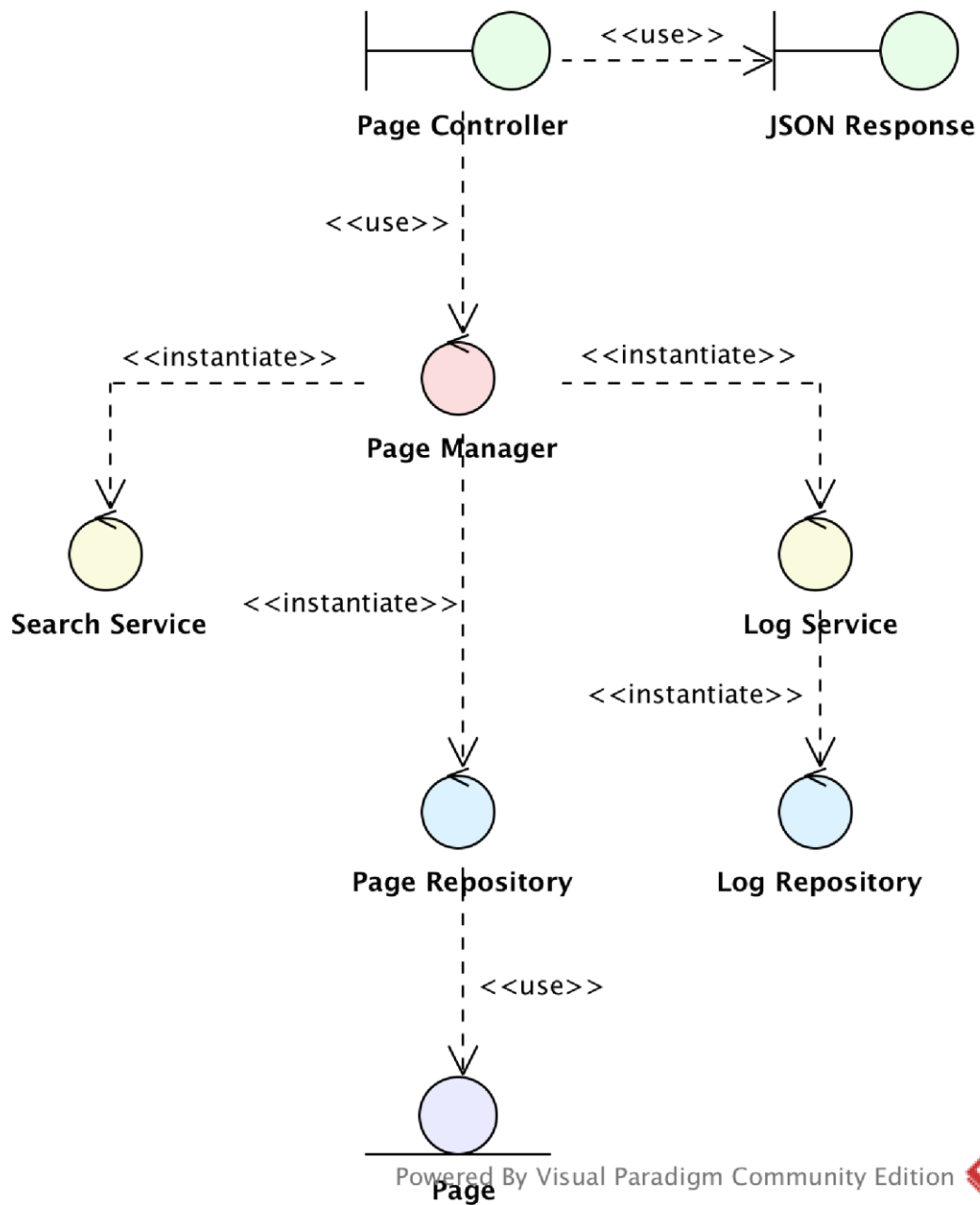
Class Diagram :

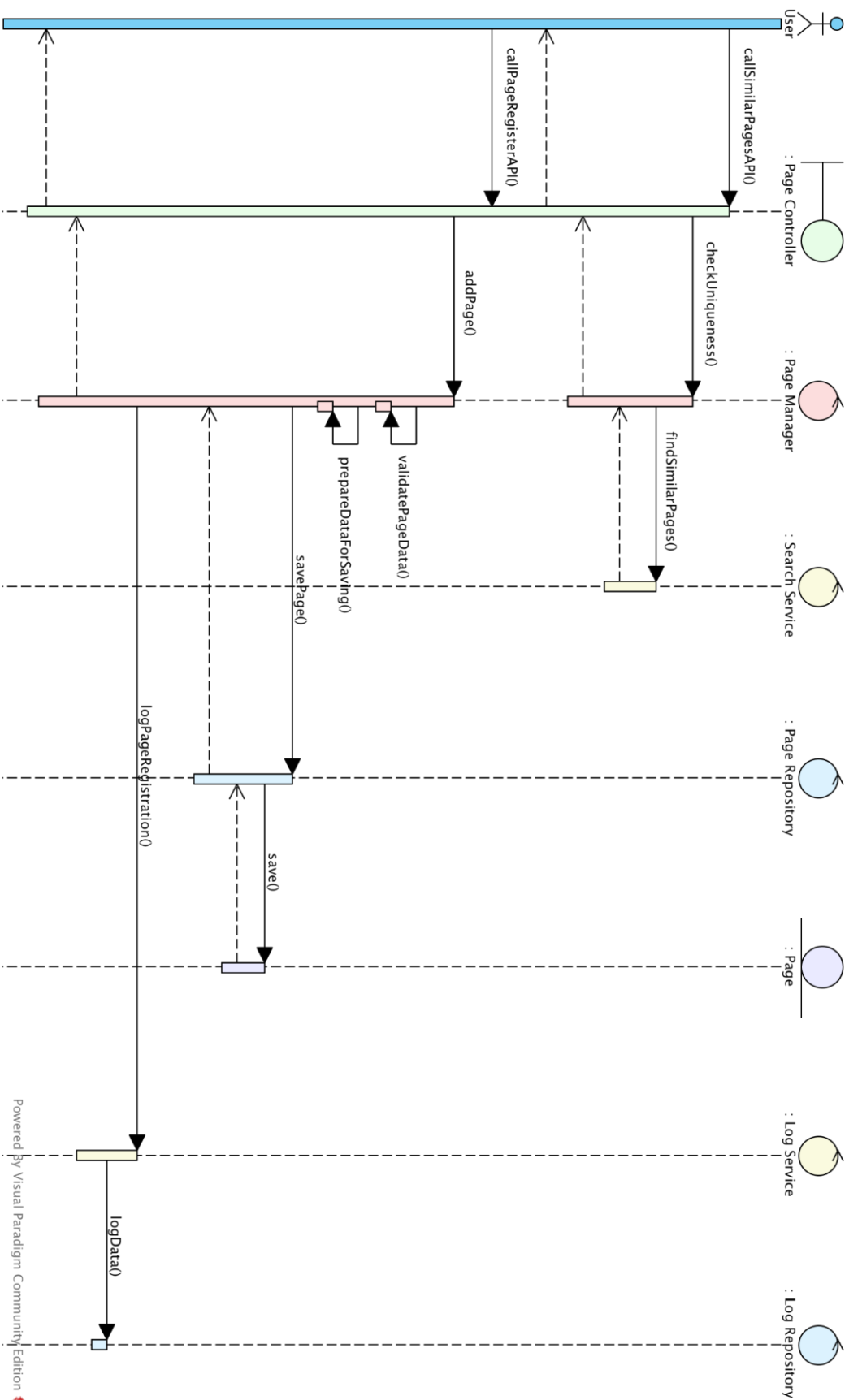




## • ثبت کسب و کار

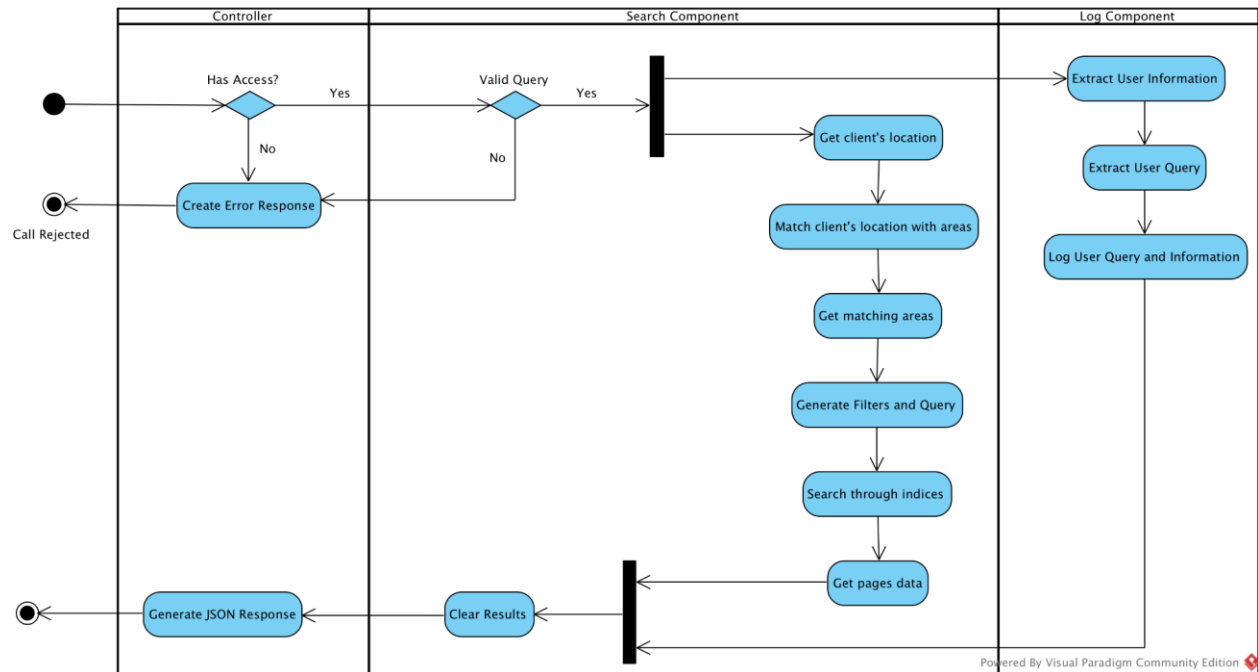
Class Diagram :





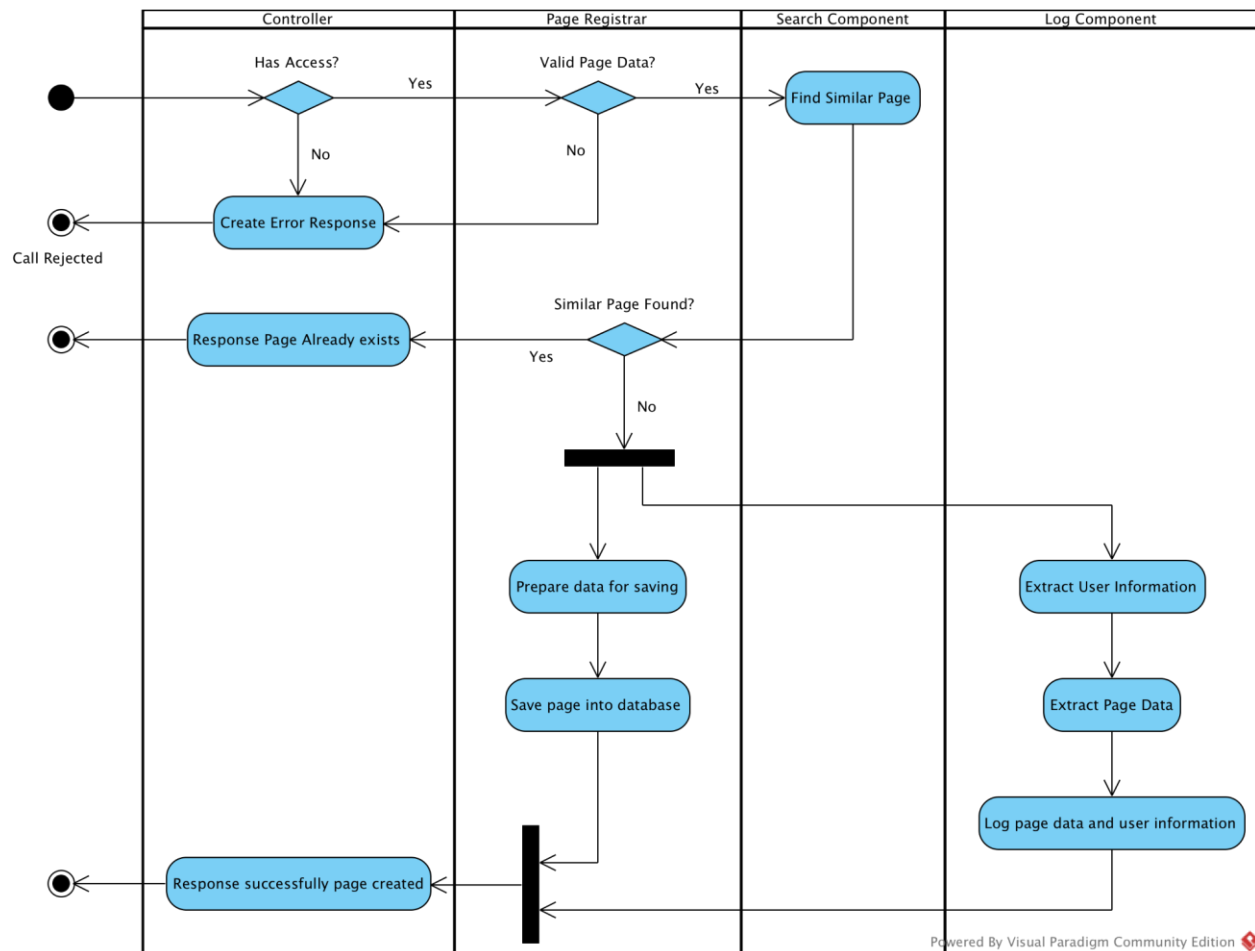
## ۸- Process View :

- جستجوی کسب و کار :



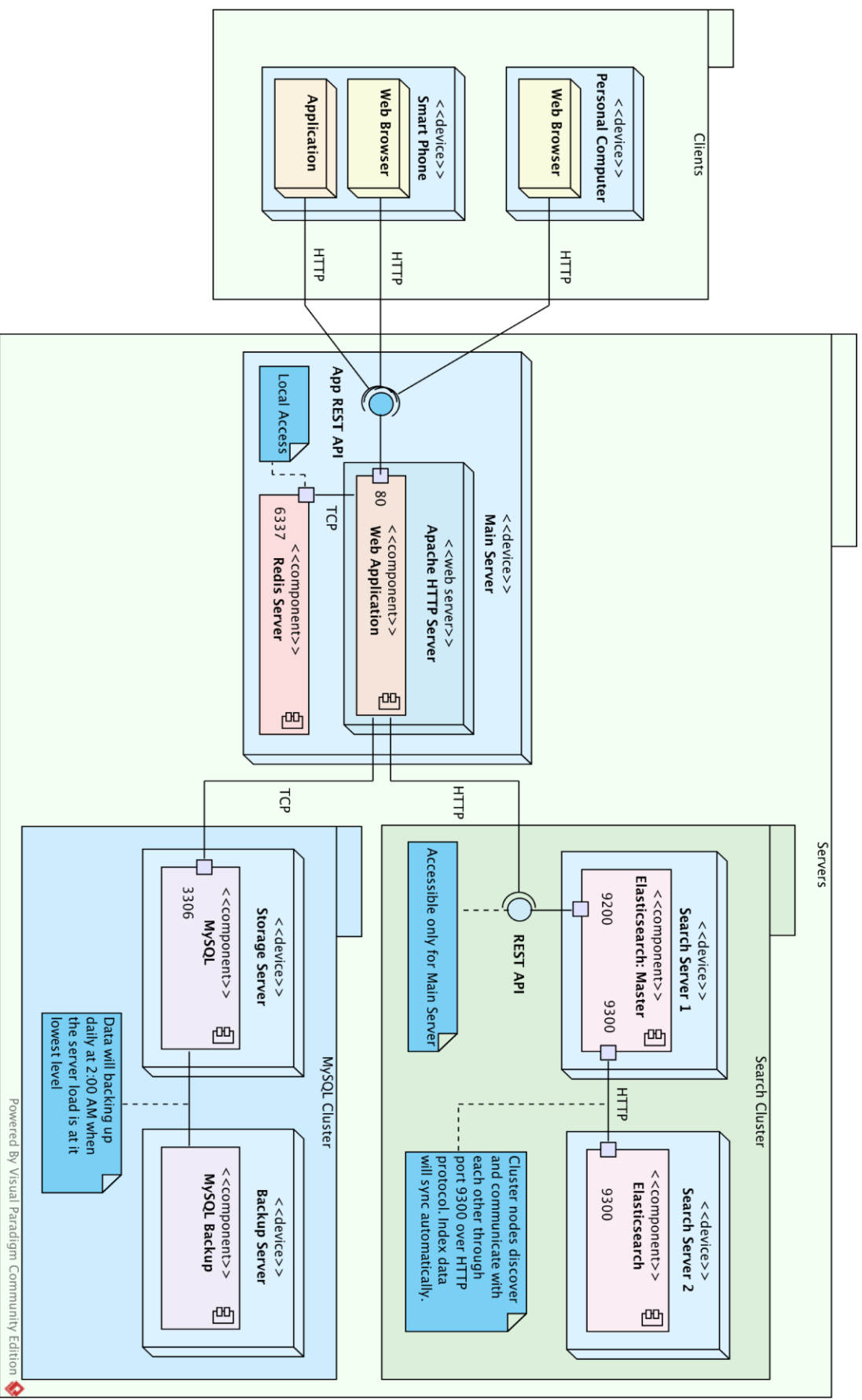


- **ثبت و مدیریت کسب و کار :**



## ۹- Deployment View :

- کلاینتها از طریق RESTful API با سرور (وب اپلیکیشن) بر روی بستر HTTP در ارتباط هستند.
- پایگاه داده MySQL بر روی سرور مجزا قرار دارد.
- اینترفیس Elasticsearch به صورت خصوصی بوده و تنها از سمت سرور اصلی قابل دسترسی است.
- ارتباط با موتور جستجو Elasticsearch از طریق REST API بر روی بستر HTTP برقرار میشود.
- سرورهای Elasticsearch با استفاده از درگاه ۹۳۰۰ در بستر HTTP با یکدیگر در ارتباطند. این ارتباط تنها برای سرورهایی با محدوده IP مشخص در دسترس است.



## ۱۰- Implementation View :

در این دیدگاه، ترکیب مولفه های سیستم می بایست ، ترسیم شوند. که این مولفه ها و ترکیب آنها با یکدیگر در نمودار قبلی در بخش Deployment View نمایش داده شدند. همچنین لایه های تشکیل دهنده اپلیکیشن در چشم انداز بخش View Logical ترسیم و توضیح داده شده اند.