



دانشگاه اصفهان  
دانشکده مهندسی کامپیوتر

گزارش پروژه درس تحلیل و طراحی سیستم‌ها

# سیستم کاریابی هوشمند

\* گروه ۵ \*

پدیدآورندگان:

محمد امین کیانی

یزدان افرا

محمد جعفری

متین عصب‌الظهور

علی حسینی‌فرد

امیررضا جهانگیری

استاد راهنما: جناب آقای دکتر محمدرضا شعرباف

نیمسال دوم تحصیلی ۱۴۰۱-۰۲

# فهرست مطالب

## فصل اول: نیازمندی‌های سیستم ..... ۵

### ۱-۱- مقدمه ..... ۵

۱-۱-۱- بیان مسئله ..... ۵

۱-۱-۲- هدف ..... ۶

۱-۱-۳- قلمرو ..... ۶

۱-۱-۴- تعاریف، سرنام‌ها و کوتاه‌نوشت‌ها ..... ۷

۱-۱-۵- مراجع ..... ۸

۱-۱-۶- طرح کلی ..... ۸

### ۱-۲- شرح کلی ..... ۹

۱-۲-۱- چشم‌انداز محصول (واسطه‌های سیستم، واسطه‌های سخت‌افزاری، واسطه‌های نرم‌افزاری و...) ..... ۹

۱-۲-۲- کارکرد محصول ..... ۱۳

۱-۲-۳- قوانین کسب‌وکار ..... ۱۴

۱-۲-۴- قیود ..... ۱۵

### ۱-۳- نیازمندی‌ها ..... ۱۶

۱-۳-۱- نیازمندی‌های کارکردی ..... ۱۶

۱-۳-۲- نیازمندی‌های کارایی ..... ۲۰

۱-۳-۳- قیود طراحی ..... ۲۱

۱-۳-۴- صفت‌های سیستم ..... ۲۱

۱-۳-۵- برنامه‌ی تکرار ..... ۲۳

## فصل دوم: مدل دامنه ..... ۲۴

### ۲-۱- شرح کلی: ..... ۲۴

### ۲-۲- جمع‌آوری اطلاعات دامنه کاربرد: ..... ۲۴

### ۲-۳- طوفان فکری: ..... ۲۴

۲۵.....	۲-۴- دسته‌بندی مفاهیم طوفان فکری:
۲۸.....	۲-۵- به تصویر کشیدن مدل دامنه:
۲۹.....	فصل سوم: طراحی معماری
۲۹.....	۳-۱- شرح کلی:
۲۹.....	۳-۲- فرایند طراحی معماری:
۳۳.....	۳-۳- نمودار بسته:
۳۳.....	۳-۴- اعمال قوانین طراحی نرم‌افزار:
۳۵.....	فصل چهارم: استنتاج مورد کاربردها از نیازمندی‌ها
۳۵.....	۴-۱- استنتاج مورد کاربردها از نیازمندی‌ها
۳۶.....	۴-۱-۱- شناسایی مورد کاربردها
۳۸.....	۴-۱-۲- تعیین قلمرو موارد کاربرد
۴۴.....	۴-۱-۳- ترسیم نمودار مورد کاربردها
۴۷.....	۴-۱-۴- بازبینی مورد کاربردها و نمودارها
۴۷.....	۴-۱-۵- تخصیص مورد کاربردها به تکرارها
۴۹.....	۴-۲- مدل‌سازی تعامل کنشگر-سیستم
۵۵.....	فصل پنجم: مدل‌سازی تعامل شیء
۵۶.....	۵-۱- سناریو و مدل تعامل شیء برای برای گام چهارم از UC01
۶۱.....	۵-۲- سناریو و مدل تعامل شیء برای برای گام سوم از UC23
۶۵.....	۵-۳- سناریو و مدل تعامل شیء برای برای گام سوم از UC18
۷۳.....	۵-۵- سناریو و مدل تعامل شیء برای برای گام چهارم از UC05
۷۶.....	۵-۶- سناریو و مدل تعامل شیء برای برای گام ششم از UC08
۸۰.....	فصل ششم: استنتاج نمودار کلاس طراحی

فصل هفتم: جمع‌بندی و انتقال تجارب ..... ۸۳

۷-۱- تجارب ..... ۸۳

۷-۲- ابزارها ..... ۸۴

۷-۲-۱- مدیریت پروژه ..... ۸۴

۷-۲-۲- ترسیم نمودارها و طراحی ..... ۸۵

۷-۲-۳- تولید محتوا ..... ۸۵

۷-۲-۴- راه‌های ارتباطی و جلسات ..... ۸۵

سخن آخر ..... ۸۶

# فصل اول: نیازمندی‌های سیستم

## 1-1-1-مقدمه

سیستم کاریابی به عنوان یکی از ابزارهای مهم در عرصه استخدام و کاریابی، به شرکت‌ها کمک می‌کند تا به سرعت و با کیفیت بالا، بازارکار را بررسی کنند و با جذب افراد ماهر و توانمند، باعث بهبود کارایی و افزایش سود شوند. در این راستا، یک سیستم کاریابی کارآمد و کاربردی باید توانایی پشتیبانی از تمام مراحل استخدام را داشته باشد، از جمله تبلیغ شغل، درخواست رزومه، ارائه آن به صاحب کار و برگزاری مصاحبه. برای پیاده‌سازی یک سیستم کاریابی کارآمد و موثر، باید نیازمندی‌هایی که از آن انتظار می‌رود را شناسایی کرده و تعریف کرد. این نیازمندی‌ها می‌توانند شامل امکانات مورد نیاز برای شرکت‌های استخدام کننده و کارجویان، ایجاد پروفایل کاربر، ارسال درخواست رزومه، مدیریت آسان و بدون مشکل لیست شغل‌ها و متقاضیان و همچنین امکان پیگیری وضعیت استخدام برای شرکت و کارجویان باشند. در این سند، نیازمندی‌های اساسی برای یک سیستم کاریابی موفق و کارآمد را تبیین خواهیم کرد.

## 1-1-1- بیان مسئله

سیستم کاریابی با چالش‌های مختلفی روبه‌روست که باید مدیریت آن‌ها به خوبی صورت پذیرد. یکی از مسائل اصلی، جمع‌آوری و ثبت اطلاعات متقاضیان است. این اطلاعات شامل اطلاعات شخصی، تحصیلی، تجربی و مهارتی است. به دلیل حجم بالای اطلاعات و نیاز به دسته‌بندی و دسته‌بندی شدن، این فرایند می‌تواند بسیار زمان‌بر و پرهزینه باشد.

مسئله دیگری که در سیستم کاریابی وجود دارد، توانایی هوشمندی سیستم در پیدا کردن و تخصیص دادن متقاضیان به شغل‌های مناسب است. برای انجام این کار، سیستم باید توانایی تحلیل داده‌های کاربران را داشته باشد و با استفاده از الگوریتم‌های مناسب، بهترین کاندیداهای ممکن را برای هر شغل پیشنهاد دهد.

همچنین، مسئله دیگری که در سیستم کاریابی وجود دارد، مدیریت فرایند مصاحبه و ارتباط با متقاضیان و کارفرمایان است. سیستم باید توانایی برقراری ارتباط با این دو گروه را داشته باشد و برای هر فرایند بهترین راه‌حل را ارائه دهد.

ساخت یک سیستم کاریابی هوشمند می‌تواند به شرکت‌هایی که تلاش می‌کنند برای جذب و استخدام بهترین مجریان و کارمندان، کمک کند. برخی از نیازهایی که ممکن است وجود داشته باشد عبارتند از:

- صرفه‌جویی در هزینه‌های بازاریابی عمومی و جذب کارمندان با استفاده از تکنولوژی‌های هوشمند.
- سرعت بیشتر در حلقه‌های استخدام و جذب کارمندان، که می‌تواند در هنگام شناسایی فرصت‌های کسب‌وکار، کمک کند شرکت‌ها سریعتر عمل کرده و رقابت را برنده شوند.
- بهبود تجربه کاربری متقاضیان کار و شرکت‌های کارفرما به وسیله ارائه پیشنهادات مناسب راجع به مشاغل خالی به تمامی کاربران.
- افزایش کیفیت و مهارت‌های جدید بین کارمندان با آموزش‌های جدید برای جذب شرکت‌های خوب.

به طور کلی، سیستم کاریابی هوشمند می‌تواند به شرکت‌هایی که به دنبال بهبود تجربه کاربری و کاهش هزینه‌های استخدام با افزایش سرعت و کیفیت کاریابی خود هستند، کمک کند.

## 2-1-1-هدف

هدف اصلی از تشکیل سیستم کاریابی هوشمند ایجاد یک سیستم عامل موثر برای یافتن ویژگی‌های مهم کاربران و جستجوی شغل‌های مناسب برای آن‌ها با استفاده از الگوریتم‌های هوشمند محاسبه شده است. با استفاده از این سیستم کاریابی هوشمند، امکان دسترسی به اطلاعات برترین شغل‌ها و مهارت‌های موردنیاز برای آن‌ها فراهم شده و به شکلی جامع و موثر ارائه می‌شود. این سیستم نه تنها به شرکت‌ها کمک می‌کند تا به راحتی جستجوی نیروی انسانی مناسب برای شغل‌های مختلف را انجام دهند، بلکه برای افرادی که به دنبال کار جدیدی هستند نیز امکان تفکیک شغل‌های مختلف بر اساس مهارت‌های موردنیاز و تخصص‌های مرتبط با آن‌ها را فراهم می‌کند. در نتیجه، این سیستم می‌تواند به بهبود کارآمدی و سرعت فرآیند استخدام کمک کند.

## 3-1-1-قلمرو

برای یک سیستم کاریابی، قلمرو (scope) به معنای تمام وظایف و عملیات مرتبط با کاریابی است که در آن پوشش داده می‌شود. این شامل مراحل مختلف کاریابی مانند جذب و انتخاب کارجویان، مصاحبه‌ها، بررسی مستندات، برگزاری آزمون‌ها، تعیین سطح مهارت‌ها، بررسی مراحل پیشرفت کاریابی، تعیین کمیته‌های داوری و اجرایی و کارهای مرتبط با گزارش‌گیری و ارائه آمارهای مرتبط با بخش کاریابی است. هدف از

تعیین قلمرو کاریابی، بهبود کیفیت افرادی است که به عنوان کارمند در سازمان جدید استخدام می‌شوند و بهبود عملکرد سازمان در بلند مدت است.

سیستم کاریابی به عنوان یک راهکار فناوری اطلاعات برای جستجوی و جذب بهترین استخدام‌کننده‌ها و برترین متقاضیان شغلی، دارای یک دامنه وسیع از عملیات است که نیازمندی به دقت بودن قلمرو آن می‌باشد. در سیستم کاریابی همان حد و محدوده فعالیت‌هایی است که طراحان و توسعه‌دهندگان سیستم برای آن تعیین می‌کنند. این محدوده می‌تواند شامل دامنه‌هایی مانند نوع شغل، حرفه، محل کار، تجربه کاری، سطح شرح شغل، حقوق و دیگر پارامترهایی باشد که برای جستجو، فیلتر کردن و همگام‌سازی اطلاعات شغلی بین شرکت‌ها و متقاضیان شغلی در سیستم کاریابی به کار می‌رفته است.

یکی از مزایای تعیین محدوده کاریابی (Scope در سیستم کاریابی)، کاهش دردسر در جستجوی شغل برای متقاضیان و کارفرمایان و البته افزایش سرعت در یافتن اطلاعات شغلی مورد نیاز است. همچنین، تعیین دقیق قلمرو سیستم کاریابی به کاربران اجازه می‌دهد که خیال راحت‌تری در زمینه پیدا کردن و خدمات ارائه شده توسط سیستم کاریابی داشته باشند. بنابراین، با توجه به اهمیت کلیدی قلمرو در سیستم کاریابی، این مسئله باید به دقت در نظر گرفته شود و برای آن یک بررسی کلی و جامع انجام شود.

#### 4-1-1- تعاریف، سرنام‌ها و کوتاه‌نوشت‌ها

- OOSE کوتاه‌شده واژه Object Oriented Software Engineering است.
- OOAD کوتاه‌شده واژه Object-Oriented Analysis and Design است.
- MBTI کوتاه‌شده واژه Myers Briggs Type Indicator است.
- NEO کوتاه‌شده واژه Neuroticism , Extraversion , Openness to Experience , Agreeableness & Conscientiousness است.
- کاربر: فردی است که به نحوی با سیستم در تعامل بوده، یعنی یا به آن ورودی می‌دهد یا از آن خروجی می‌گیرد و یا هر دو.
- مدیر سیستم : کاربری است که بالاترین سطح دسترسی را در سیستم دارد.
- SRS کوتاه‌شده واژه Software Requirement Specification است.

## 5-1-1-مراجع

- کونگ، دیوید سی: مهندسی نرم‌افزار شیء گرا (یک متودولوژی چابک یکنواخت) جلد اول. ترجمه: دکتر بهمن زمانی و دکتر افسانه فاطمی، ۱۳۹۴.

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEE Xplore Digital Library.

<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

- K. Jammalamadaka and V R. Krishna, "AGILE SOFTWARE DEVELOPMENT AND CHALLENGES", International Journal of Research in Engineering and Technology (IJRET)

- N. Rashid, " Applying Agile Methodologies on Large Software Projects", International Journal of Recent Research in Mathematics Computer Science and Information Technology.

- <https://jobinja.ir/>

- <https://jobvision.ir/>

- <https://www.irantalent.com/farsi>

- <https://jooiakar.com/>

## 6-1-1-طرح کلی

سیستم کاریابی هوشمند شامل چندین قابلیت است که از جمله آن‌ها می‌توان به موارد زیر اشاره کرد:

۱- جستجوی بیشترین تطبیق: از طریق کلمات کلیدی و مهارت‌های مورد نظر، سیستم کاریابی هوشمند با جستجوی بیشترین تطبیق، رزومه‌های کارجویان را با استانداردهای کاریابی شرکت‌های مختلف، مطابقت خواهد داد.

۲- طبقه‌بندی و مدیریت: طبقه‌بندی و مدیریت درخواست‌های کاریابی، ترتیب بندی کاندیدها بر اساس نیازهای شغلی شرکت‌ها و همچنین هوشمندی در طبقه‌بندی و مدیریت کاندیدهای برتر.

۳- مصاحبه‌های آنلاین: مصاحبه آنلاین با کارجویان از طریق سیستم کاریابی هوشمند، کمک می‌کند تا مراحل کاریابی برای شرکت‌ها به سادگی و انعطاف‌پذیری بیشتری صورت گیرد.



۴- تفحص هویت: تفحص هویت امکان دیگری است که توسط سیستم کاریابی هوشمند ارائه می‌شود. این قابلیت، هویت کارجویان را به صورت اتوماتیک بررسی کرده و هر نوع تقلب و دروغ را بررسی می‌کند. این بخش تنها به منظور ایجاد یک دید کلی از سیستم و کارکردهای آن تهیه شده است. در بخش بعد به قیود تعیین شده از سوی مشتری، نیازمندی‌های مختلف سامانه و صفات سیستم پرداخته و آنها را به طور مفصل تشریح کرده‌ایم.

## 2-1- شرح کلی

یک سامانه کاریابی هوشمند، یک پلتفرم کاملاً آنلاین است که با استفاده از الگوریتم‌های هوشمند، اطلاعات متقابل بین کارفرمایان و کارجویان را پردازش و بهینه می‌سازد.

در این سامانه، کارفرمایان می‌توانند شرح مشاغل خود را وارد کرده و از طریق الگوریتم‌های هوشمند، برای جذب بهترین کارجویان در بازه زمانی کوتاهی کمک بگیرند. همچنین کارجویان می‌توانند اطلاعات و رزومه خود را در این سامانه ثبت کنند. الگوریتم‌های هوشمند از این اطلاعات به منظور پیدا کردن شغل مناسب برای آنها استفاده می‌کنند. با بهینه‌سازی جستجوها و اطلاعات موجود، سامانه کاریابی هوشمند به کارجویان کمک می‌کند تا به سرعت کارهای مورد نظر خود را پیدا کنند. با فرآیند استخدام هوشمندتر، مدیران منابع انسانی تحت فشار کمتری قرار می‌گیرند و در نتیجه، فرایند استخدام بهینه‌تر شده و بازده آن افزایش می‌یابد.

## 1-2-1- چشم‌انداز محصول (واسط‌های سیستم، واسط‌های سخت-افزاری، واسط‌های نرم‌افزاری و...)

یک سیستم کاریابی یا ATS (Applicant Tracking System)، یک ابزار نرم‌افزاری است که برای کاهش زمان و هزینه‌های مربوط به پیدا کردن و استخدام کارمندان به کار می‌رود. با استفاده از این سیستم، شرکت‌ها می‌توانند رزومه‌ها و اطلاعات ارسالی کاندیدهای شغلی را جمع‌آوری و مدیریت کنند و مراحل استخدامی را بهبود دهند.

چشم‌انداز یک سیستم کاریابی معمولاً شامل تعیین وظایف و اولویت‌های مورد نیاز برای جذب نیروی انسانی به عنوان کارکنان در سازمان است. برای مثال، یک سیستم کاریابی می‌تواند برای هر شغل به نیروی انسانی با توانایی‌های خاص و با تجربه دست پیدا کند.

در طراحی و اجرای یک سیستم کاریابی، نیاز است تا معیارهای ارزیابی کاندیدها تعریف و پارامترهای جستجو تعیین شوند. فناوری‌های هوشمند مانند یادگیری عمیق از زیرمجموعه‌های یادگیری ماشین (یا machine

learning) می‌توانند به طور خودکار مهارت‌ها و تجربیات متقاضیان شغل را ارزیابی کنند، رزومه‌هایی را که به موارد مشخصی نیاز دارند، دریابند و یا متوجه مشابهت شغلی شخص با شغل مدنظر شده شوند تا شغل‌های احتمالا مناسب را به اشخاص از طریق ایمیل پیشنهاد دهند.

## **\*واسط‌های سیستم**

اطلاعات سازمان‌ها، کاربران، رزومه‌ها و مهارت‌ها همگی باید در پایگاه داده ذخیره شده تا برای اهداف مختلف سیستم مورد استفاده قرار گیرند.

اطلاع‌رسانی آگهی‌های شغلی جدید باید از طریق ایمیل به کاربران ارسال شود که نیازمند تشخیص آگهی مناسب با هوش مصنوعی و ارسال تعداد زیادی آگهی به افراد مختلف می‌باشد. به همین منظور به سرورهایی نیاز داریم که منابع مورد نیاز ما را جهت کارهای سنگین تامین کنند.

## **\*واسط‌های کاربر**

واسط کاربری یا UI باید به گونه‌ای طراحی شود که کاربران بتوانند به آسانی و با دسترسی آسان به اطلاعات مهم، از آن استفاده کنند. واسط کاربری نباید گیج کننده برای افراد مسن یا نابلد باشد، همچنین نباید آسانی بیش از حد واسط کاربری قدرت برنامه را در انجام کارها کم کند.

واسط کاربری باید به صورت یک وب اپلیکیشن در فضای وب همراه با نرم‌افزار موبایل همه‌جا به طور ۲۴ ساعته در دسترس همه کاربران باشد.

## **\*واسط‌های سخت‌افزاری**

واسط سخت‌افزاری باید پردازش سریع و قابل اعتمادی را در اختیار قرار دهد طوری که نیازهای امنیتی تامین شوند.

باید دارای کارایی بالا، مصرف برق کم و توسعه‌پذیر باشد به طوری که از بزرگی و پیچیدگی یک سیستم کارایی پشتیبانی شود.

قابلیت اتصال به اینترنت و شبکه‌های ارتباطی برای انجام تبادلات داده‌ها و ارتباط با سرویس‌های آنلاین را نیز داشته باشد.

البته نوع واسط‌های سخت‌افزاری طوری که بتواند نیازها را بر طرف کند، بسته به معماری و زیرساخت‌های نرم‌افزار می‌توانند متفاوت باشند.

## \*واسطه‌های نرم‌افزاری

برای دسترسی به سیستم کاریابی و استفاده از آن، نیاز به یک مرورگر وب (کروم، فایرفاکس و ..) وجود دارد. البته می‌توان از اپ موبایل برای دسترسی به سیستم نیز استفاده کرد.

استفاده از پایگاه داده بستگی به مقیاس پروژه و سلیقه توسعه دهندگان دارد. می‌توان از MySQL یا Oracle و یا پایگاه داده‌هایی که به صورت NoSQL هستند مانند MongoDB استفاده کرد.

## \*واسطه‌های ارتباطی

سیستم کاریابی از طریق وبسایت آن قابل دسترسی بوده و محرمانگی ارتباط بین کاربر و سرور با پروتکل HTTPS برقرار خواهد شد.

ارتباط مشتریان با ایمیل‌ها و شماره تلفن‌های مربوط به پشتیبانی باید برقرار باشد. همچنین گفتگو با پشتیبانی به صورت چت‌باکس می‌تواند روی وبسایت قرار گیرد.

## \*واسطه‌های حافظه

در سیستم کاریابی، واسطه‌های حافظه به عنوان یک نوع واسطه بین بخش‌های مختلف سیستم استفاده می‌شوند. این واسطه‌ها برای ذخیره و بازیابی اطلاعات مربوط به کارجویان و شرکت‌های استخدام کننده سایت کاربرد دارند.

برای دسترسی سریع‌تر به اطلاعات باید از حافظه حالت جامد (SSD) استفاده شود که سرعت بیشتری دارد.

## \*واسطه‌های عملیات

۱- ثبت نام و رزومه: واسطه‌ای که به کارجویان این امکان را می‌دهد تا رزومه خود را در سیستم ثبت کنند و اطلاعات شخصی خود را به روزرسانی کنند.

۲- جستجوی شغل: واسطه‌ای که به کارجویان این امکان را می‌دهد تا جستجوی شغل مورد نظر خود را از بین شغل‌های موجود در سیستم انجام دهند.

۳- آگهی استخدام: واسطه‌ای که به شرکت‌ها و کارفرمایان امکان می‌دهد آگهی استخدام خود را در سیستم منتشر کنند.

۴- ارزیابی کارکنان: واسطه‌ای که به کارفرمایان این امکان را می‌دهد تا ارزیابی کارکنان خود را در سیستم ثبت و نتایج آن را بازبینی کنند.

۵- انجام مصاحبه‌های آنلاین: واسطه‌ای که به شرکت‌ها و کارفرمایان امکان می‌دهد تا مصاحبه آنلاین با کارجویان را انجام دهند.

۶- مدیریت پرونده‌های کارکنان: واسطه‌ای که به کارفرمایان این امکان را می‌دهد تا پرونده‌های کارکنان خود را مدیریت و اطلاعات آنان را ثبت و به‌روزرسانی کنند.

۷- فیلترهای جستجوی پیشرفته: یکی از اصلی‌ترین واسطه‌های کاریابی، فیلتر جستجوی پیشرفته است. با استفاده از این واسطه، کارجویان می‌توانند جستجویی دقیق و هدفمند در بین شغل‌های موجود را صورت دهند.

۸- نمایش شغل‌های پیشنهاد شده: سامانه کاریابی می‌تواند با توجه به علاقه‌مندی‌های کارجویان، شغل‌های پیشنهاد شده‌ای به آن‌ها نمایش دهد.

۹- لیست شغل‌های مرتب شده: می‌توانید لیستی از شغل‌هایی که در سامانه شما وجود دارند را به کارجویان نشان دهید. این لیست بر اساس تناسب شغلی و تخصص کارجویان مرتب می‌شود.

۱۰- پروفایل کاربری: کارجویان باید بتوانند اطلاعات مربوط به تجربه کاری، تحصیلات، مهارت‌های تخصصی و... خود را در پروفایل کاربری خود به‌روزرسانی کنند.

۱۱- اعلانات: اعلان‌های جدید و ویژگی‌هایی مانند مصاحبه با کارفرمایان با استفاده از اعلانات و پیام‌ها به کارجویان اعلام می‌شود.

۱۲- پیام‌ها: در صورت بروز حوادث ناگوار یا درخواست مشتری، کارجویان باید بتوانند پیام داده و برای درخواست یا سوال خود پاسخ دریافت کنند.

### **\*نیازمندی‌های سازگار با محیط نصب**

این سامانه نیازمند استفاده از مرورگر وب روی دیوایس‌هایی همچون کامپیوترهای شخصی بوده و یا نیازمند نصب برنامه روی گوشی موبایل است.

### **\*مشخصات کاربران**

سیستم کاریابی به عنوان یک سامانه مدیریت منابع انسانی، دارای مشخصات کاربرانی است که در زیر به آن‌ها اشاره می‌کنیم:

۱. متقاضیان شغل: به طور کلی متقاضیان شغل می‌توانند اطلاعات شخصی و رزومه خود را در سیستم ثبت کنند. این شامل نام، سن، مهارت‌ها، تجربیات کاری، تحصیلات، شماره تماس و ایمیل آن‌ها است.

۲. شرکت‌های استخدام کننده: شرکت‌ها مشخصات خود از جمله نام، ایمیل و شماره تماس و همچنین فرصت‌های شغلی خود را در سامانه ثبت می‌کنند.

۳. مدیران سیستم: مدیران سیستم کارایی دسترسی به همه اطلاعات شرکت‌ها و متقاضیان شغل دارند و می‌توانند اقداماتی مانند حذف حساب کاربری و یا به‌روزرسانی اطلاعات را انجام دهند.

مشخصات کاربران یک سیستم کارایی عموماً شامل مشخصات زیر است:

۱. مشخصات شخصی: شامل نام، نام خانوادگی، شماره تماس، پست الکترونیکی، جنسیت و تاریخ تولد است.
۲. تحصیلات: شامل سابقه تحصیلی کاربر اعم از مدرسه، دانشگاه یا دانشگاه‌هایی که فارغ التحصیل شده است.
۳. تجربه کاری: شامل سوابق کاری کاربر به همراه نام شرکت، عنوان شغلی، تاریخ شروع و پایان کار، مسئولیت‌های انجام شده و دستاوردهایی که در طول این فرایند به دست آورده است.
۴. مهارت‌ها: شامل مهارت‌های فنی، رایانه‌ای و نرم‌افزاری کاربر است.
۵. موقعیت موردنظر: شامل توصیفی از موقعیت آن کاری که کاربر می‌خواهد درخواست بدهد.
۶. وضعیت فعلی: شامل وضعیت فعلی کاربر، اعم از اشتغال به کار، بیکاری یا موارد دیگر است.
۷. نمونه کارها: شامل نمونه کارهایی که کاربر انجام داده است و قابلیت نمایش به کارفرمایان را دارد.
۸. محل سکونت: شامل اطلاعات مکانی کاربر، به همراه آدرس، شهر و کشور است.
۹. درخواست‌های ارسالی: شامل درخواست‌هایی که کاربر از طریق سیستم کارایی ارسال کرده است و وضعیت آن‌ها را نشان می‌دهد.

## 2-2-1- کارکرد محصول

سیستم کارایی یک سیستم نرم‌افزاری است که برای ارتباط بین کارفرما و کارجویان مورد استفاده قرار می‌گیرد. در ادامه، برخی از کارکردهای اصلی یک سیستم کارایی ذکر می‌شوند:

- ۱- ثبت، نگهداری و به‌روزرسانی اطلاعات کارجویان:

کارجویان می‌توانند در سیستم کارایی اطلاعات خود را وارد و نگهداری کنند. این اطلاعات ممکن است شامل اطلاعات تماس، تحصیلات، سابقه کار و مهارت‌ها باشد.

- ۲- ثبت، نگهداری و به‌روزرسانی اطلاعات کارفرما:

کارفرما نیز می‌تواند در سیستم کارایی اطلاعات شرکت خود را وارد و نگهداری کند. این اطلاعات ممکن است شامل نام شرکت، نوع شغل، نوع حقوق و مزایا و موقعیت‌های شغلی باشد.

۳- جستجو و انتخاب کارجویان :

کارفرما می‌تواند برای یافتن نیرو، کارجویان را جستجو کرده و بهترین اشخاص را برای موقعیت شغلی مورد نظر انتخاب کند. امکاناتی مانند فیلترینگ، جستجوی پیشرفته، رتبه‌بندی و سایر روش‌ها جهت بررسی و انتخاب کارجویان در دسترس هستند.

۴- مدیریت مصاحبه‌ها:

شرکت‌ها و کارفرماها می‌توانند برنامه مصاحبه‌های خود را در سیستم ثبت و کارجویان را برای مصاحبه دعوت کنند.

۵- مدیریت رزومه‌های ارسالی:

کارجو رزومه خود را به سیستم کاریابی ارسال می‌کند و کارفرما می‌تواند همه رزومه‌های ارسالی را مدیریت کند تا بهترین نیروها را برای موقعیت شغلی مد نظر خود انتخاب نماید.

۶ - جستجوی کار :

کارجویان می‌توانند با استفاده از سیستم به دنبال فرصت‌های شغلی جدید بگردند.

۷- گزارش‌دهی :

سیستم کاریابی به شرکت‌ها و کارفرماها کمک می‌کند تا اطلاعاتی در مورد عملکرد پیشین و روند فعلی کاریابی خود دریافت کرده و در نتیجه تصمیم‌های بهتری را در زمینه استراتژی جذب نیروی خود اتخاذ نمایند.

## 3-2-1- قوانین کسب و کار

سیستم کاریابی از مهمترین ابزارهای جذب و انتخاب نیرو در سازمان‌هاست. برای بهره‌وری بهینه از این سیستم، قوانین کسب و کار باید به شکلی تدوین شود که زیرساخت مناسبی برای استفاده بهینه از این سیستم در سازمان فراهم کند.

۱. حفظ حریم خصوصی: باید اطمینان حاصل کنیم که اطلاعات حریم خصوصی مربوط به متقاضیان شغلی و شرکت‌ها رعایت شود و هرگونه انتشار، فروش یا انتقال آن‌ها به سایرین ممنوع است.

۲. عدم تبعیض: در انتخاب و استخدام، باید به همه متقاضیان دید یکسان داشته باشیم و هیچگونه تبعیض بر اساس جنسیت، سن، ملیت، دین یا معتقدات سیاسی انجام نشود.

۳. رعایت قوانین کار: باید در استفاده از سیستم کاریابی خود، قوانین کار و استخدام را رعایت و از هرگونه رفتار غیراخلاقی و غیرقانونی خودداری کرد.

۴. عدم انتشار اطلاعات درونی: هرگونه اطلاعات درونی مانند اطلاعات مالی، رمزهای دسترسی و رمزهای عبور را محفوظ نگه داریم و هرگونه انتشار آن‌ها به سایرین پیگرد قانونی دارد.

۵. رعایت حق مالکیت فکری: در صورت استفاده از نرم افزارها و برنامه‌های دیگر، باید حقوق مالکیت فکری مربوط به آن‌ها را رعایت کنید.

۶. پرداخت هزینه: در صورت استفاده از سیستم کاریابی از شرکت‌های دیگر باید هزینه مربوط به استفاده از آن‌ها پرداخت شود.

۷. عدم تبلیغات و پست نامرتب: باید اطمینان حاصل شود که هیچگونه انتخابات تبلیغاتی در سیستم کاریابی انجام نشود و هرگونه تبلیغات و ارسال پست غیرمجاز، اسپم یا هرگونه فرستنده نامعلوم، باید با دقت و سرعت مناسب متوقف شود.

۸. معتقدات سیاسی: باید اطمینان حاصل شود که عقاید کارکنان شرکت و قسمت‌هایی که مورد استفاده کاربران قرار می‌گیرد، براساس قوانین و معتقدات نظام جمهوری اسلامی ایران باشد.

## 4-2-1- قیود

دو دسته محدودیت وجود دارد یکی محدودیت‌های خود سیستم و دیگری محدودیت‌های کاربران سیستم که عبارت‌اند از:

### \* محدودیت‌های سیستم:

۱- محدودیت بودجه: سیستم کاریابی ممکن است با محدودیت بودجه مواجه باشد و در نتیجه قادر به انجام کامل تمام وظایف خود نباشد.

۲- محدودیت زمانی: سیستم کاریابی نیاز به نهاده‌های عظیمی دارد و در نتیجه ممکن است به محدودیت زمانی برخورد کند و کیفیت عملکردش کاهش یابد.

۳- مشکلات پشتیبانی فنی: سیستم کاریابی ممکن است با مشکلات فنی مواجه شود که کاربران به دلیل آن نتوانند به درستی از سیستم استفاده کنند.

۴- محدودیت مخاطب: سیستم کاریابی تنها برای کارجویان و صاحبان کسب و کارها مناسب است و افراد دیگر نمی‌توانند از آن استفاده کنند.

## \* محدودیت‌های کاربران:

۱- نیاز به تکنولوژی: سیستم کاریابی به تکنولوژی نیاز دارد تا بتوان از آن بهره برد، با این حال قابلیت دسترسی همه به تکنولوژی محدودیتی برای سیستم کاریابی به بار می‌آورد.

۲- اهداف شغلی متقاضیان: شاغلین ممکن است به دنبال شغلی در یک شهر خاص، حوزه کاری خاص، یا شرکت یا سازمان خاصی باشند و این ارتباط باعث می‌شود که نتوانند به شغل‌های دیگری دسترسی پیدا کنند.

۳- محدودیت‌های مالی: برخی متقاضیان ممکن است برای دسترسی به آگهی‌های شغلی هزینه کنند یا نیاز به اشتراک خاصی برای دسترسی به سایت کاریابی داشته باشند که این مسئله ممکن است برای برخی افراد محدودیت مالی ایجاد کند.

## 3-1- نیازمندی‌ها

سیستم نرم‌افزاری برای کاریابی هدفمند در سازمان‌ها و شرکت‌ها و صنایع مختلف طراحی شده دارای نیازمندی‌های متفاوتی است که بر اساس درخواست مشتری و پیشنهادات تیم ایجاد شده‌اند که به طور مفصل در ادامه آمده است.

### 1-3-1- نیازمندی‌های کارکردی<sup>1</sup>

سیستم نرم‌افزاری کاریابی<sup>۲</sup> باید برای داشتن عملکرد عالی و بهترین استفاده از ابزارهای موجود، برخی از نیازمندی‌های کارکردی را برآورده سازد که شامل موارد زیر می‌شوند:

#### \* لیست فعالیت‌های کارفرما :

(R۱) سیستم باید قابلیت ثبت نام و ورود به عنوان کارفرما را داشته باشد .

(R۲) سیستم باید امکان ثبت نام رابط شرکت و اطلاعات شخصی و سازمانی و امکان ویرایش اطلاعات را داشته باشد.

\* سیستم باید امکان ثبت شرکت و اطلاعات اولیه سازمان را داشته باشد.

\* سیستم باید امکان ثبت اطلاعات تکمیلی شرکت را داشته باشد.

(R۳) سیستم باید امکان پیش‌نویسی آگهی جدید را داشته باشد.

<sup>1</sup> Functional requirements

<sup>2</sup> Employment software system



- سیستم باید امکان معرفی شغل را داشته باشد.
- سیستم باید امکان مشخص کردن شرایط احراز شغل را داشته باشد.
- سیستم باید امکان شرح شغل و مزایا و حقوق پیشنهادی را داشته باشد.
- (R۴) سیستم باید قابلیت ارسال درخواست ثبت آگهی را داشته باشد.
- سیستم باید قابلیت ارسال درخواست را داشته باشد.
- سیستم باید قابلیت پرداخت آنلاین هزینه آگهی از طریق درگاه‌های پرداخت اینترنتی را (در صورت لزوم) داشته باشد.
- سیستم باید قابلیت مشاهده وضعیت درخواست را داشته باشد.
- (R۵) سیستم باید امکان بستن آگهی را داشته باشد.
- سیستم باید امکان حذف آگهی را داشته باشد.
- سیستم باید امکان آرشیو آگهی را داشته باشد.
- (R۶) سیستم باید قابلیت مشاهده و بررسی بانک رزومه سیستم و بررسی رزومه‌های بانک را به کارفرما ارائه دهد.
- سیستم باید قابلیت اعمال فیلترهای اختصاصی را داشته باشد.
- سیستم باید قابلیت دسته‌بندی و پوشه‌بندی رزومه‌ها را داشته باشد.
- (R۷) سیستم باید امکان بررسی رزومه‌های ارسال شده برای آگهی‌های ثبت شده را داشته باشد.
- سیستم باید امکان دسته‌بندی و پوشه‌بندی رزومه‌ها را داشته باشد.
- (R۸) سیستم باید به کارفرما امکان انتخاب کارجو را ارائه بدهد.
- سیستم باید قابلیت رتبه بندی کارجو با امتیازدهی به او طبق صلاح دید کارفرما در داشبورد خود کارفرما برای انتخاب بهترین کارجو از نظر کارفرما را داشته باشد.
- (R۹) سیستم باید امکان پیگیری وضعیت کارجوی انتخاب شده را به کارفرما بدهد.
- (R۱۰) سیستم باید قابلیت برگزاری جلسات حضوری یا آنلاین با کارجو برای مصاحبه را داشته باشد.
- (R۱۱) سیستم باید امکان تغییر یا بازیابی نام کاربری و گذرواژه در صورت فراموشی آن را به کارفرما بدهد.

- باید امکان تغییر یا بازیابی نام کاربری و گذرواژه از طریق ایمیل را داشته باشد.
- باید امکان تغییر یا بازیابی نام کاربری و گذرواژه از طریق پیامک را داشته باشد.

#### \* لیست فعالیت‌های کارجو:

- (R۱۲) سیستم باید قابلیت ثبت نام و ورود به عنوان کارجو را داشته باشد.
- (R۱۳) سیستم باید قابلیت جستجو بین محل کارهای متفاوت بر اساس فیلترهای استان، شهر، نزدیکی به محل سکونت کاربر را داشته باشد.
- (R۱۴) سیستم باید امکان مشاهده فرصت‌های شغلی بر اساس آیتم‌های زیر را به کارجو بدهد:

• میزان بازدید فرصت

• اعمال فیلترهای شخصی

• رزومه‌ی کارجو

• موقعیت مکانی

• نوع همکاری

• سازمان‌ها

- (R۱۵) سیستم باید امکان ساخت و تکمیل رزومه کاری با تقسیم‌بندی کلی و امکان ویرایش اطلاعات را به کارجو بدهد. مانند:

• اطلاعات اولیه

• سوابق تحصیلی

• سوابق شغلی

• زبان‌های تحت تسلط

• مهارت‌های نرم‌افزاری

• مهارت‌های تکمیلی

• ثبت معلولیت یا بیماری خاص

(R۱۶) سیستم باید به کارجو امکان شرکت در دوره‌های آموزشی جهت بهبود مهارت و رزومه و احتمال استخدام را بدهد.

(R۱۷) سیستم باید قابلیت تغییر یا بازیابی نام کاربری و گذرواژه در صورت فراموشی آن را داشته باشد.

• باید امکان تغییر یا بازیابی نام کاربری و گذرواژه از طریق ایمیل را داشته باشد.

• باید امکان تغییر یا بازیابی نام کاربری و گذرواژه از طریق پیامک را داشته باشد.

(R۱۸) سیستم باید به کارجو امکان درخواست را بدهد.

• باید امکان مشاهده درخواست‌ها با فیلترهای شخصی را داشته باشد.

• باید امکان مراجعه به تاریخچه درخواست‌ها را داشته باشد.

• باید امکان بررسی وضعیت درخواست‌های ارسال شده را داشته باشد.

(R۱۹) سیستم باید قابلیت ایجاد اطلاع‌رسان مشاغل مرتبط جدید برای کارجو را داشته باشد.

#### **\* لیست فعالیت‌های مشترک کارفرما و کارجو:**

(R۲۰) سیستم باید قابلیت بررسی داده‌های بازارکار را داشته باشد.

• باید امکان مشاهده گزارش‌های حقوق و دستمزد را داشته باشد.

• باید امکان مشاهده گزارش‌های آماری بازارکار را داشته باشد.

• باید امکان مشاهده و خرید کتاب‌های آموزشی را داشته باشد.

(R۲۱) سیستم باید به کاربران امکان شرکت در آزمون‌های خودشناسی و بررسی نتایج آن را بدهد.

• باید آزمون شخصیت‌سنجی MBTI<sup>۱</sup> داشته باشد.

• باید آزمون تیپ‌سنجی شغلی Holland<sup>۲</sup> داشته باشد.

• باید آزمون شخصیت‌شناسی NEO<sup>۲</sup> داشته باشد.

• باید آزمون هوش‌های چندگانه داشته باشد.

• باید آزمون هوش هیجانی Bar-On<sup>۲</sup> داشته باشد.

---

<sup>۱</sup> Myers Briggs Type Indicator

<sup>۲</sup> Neuroticism , Extraversion , Openness to Experience , Agreeableness & Conscientiousness (NEO)

R۲۲) سیستم باید امکان مشاهده سوابق جست‌وجوی کاربران را داشته باشد.

R۲۳) سیستم باید قابلیت ارسال انتقادات و پیشنهادات را داشته باشد.

R۲۴) سیستم باید راهنمای جامع درمورد همه بخش‌های سایت برای کاربران داشته باشد.

## 2-3-1-نیازمندی‌های کارایی<sup>1</sup>

در این بخش الزامات عددی ثابت و پویا را که بر روی نرم‌افزار یا تعامل انسان با نرم‌افزار نقش دارند، مشخص خواهیم کرد.

• زمان واکنش سیستم: درخواست‌های ارسال شده توسط کاربران باید به سرعت توسط سیستم پاسخ داده شوند به نحوی که درصد تراکنش‌ها در کمتر از یک ثانیه پردازش شود.

• سیستم باید در ۲۴ ساعت شبانه‌روز در دسترس باشد.

• سیستم باید به راحتی قابل استفاده و کاربرپسند بوده همچنین باید از UI و UX خوبی برخوردار باشد تا تعامل کاربران با سایت به راحتی امکان‌پذیر شود. (در این مورد باید این را هم در نظر بگیریم که ممکن است کاربران اهل روستاهایی که با تکنولوژی‌ها آشنا نیستند، باشند و برای آن‌ها هم باید استفاده از سایت به راحتی صورت بگیرد).

• سیستم باید سرعت بارگذاری بالایی داشته باشد و زمان پاسخ‌گویی به کاربران به منظور حفظ رضایت و جلب اعتماد آن‌ها نباید از حد انتظار بیشتر باشد.

• سیستم باید از سطح عالی امنیت داده‌ها و محافظت از اطلاعات کاربران اطمینان حاصل کند.

• سیستم باید قابلیت توسعه و افزودن ویژگی‌های جدید در آینده را داشته باشد.

• سیستم باید پشتیبانی داشته باشد و به‌روزرسانی سیستم در مدت زمان لازم انجام شود.

• سیستم باید آمادگی پاسخگویی به ده‌هزار کاربر در ثانیه را داشته باشد و درخواست‌های کاربران را بدون وقفه یا ناهماهنگی مدیریت کند.

• سیستم باید در ۹۹ درصد زمان‌ها در دسترس باشد و در صورت از دسترس خارج شدن با نمایش پیغام مناسب آنها را از علت خطا آگاه کند. همچنین تخمین زمانی که سامانه به حالت عادی خود باز خواهد گشت در آن ذکر شود.

---

<sup>1</sup>Performance Requirements

• سیستم باید اطلاعات مهم کاربران مانند شماره تلفن همراه و کد ملی را رمزنگاری شده به سمت سرور بفرستد تا هیچ خطای امنیتی آن را تهدید نکند.

• سیستم باید برای ورود کاربر، از کد CAPTCHA استفاده کند تا از اینکه فرد وارد شونده ربات نباشد، اطمینان حاصل کند.

• سیستم باید با مرورگرهای مختلف سازگار باشد.

• سیستم باید ساعت و تاریخ روز را در صفحه های مختلف سایت به کاربر نشان دهد.

• اگر سیستم هنگام گزارش گیری توسط کاربر (به جز عموم مردم) به مشکل برخورد، باید پس از رفع مشکل، اطلاعات را از ادامه آن گزارش دهد.

### 3-3-1- قیود طراحی<sup>1</sup>

- سامانه باید برای افراد با نیازمندی های خاص مثل افراد نابینا قابل دسترسی باشد.
- سامانه باید واکنش گرا<sup>2</sup> باشد.
- فرمت های بارگذاری رزومه و همچنین قسمت هایی مثل انتخاب زبان برنامه نویسی و ... باید به صورت انتخابی باشند و توسط مسئول سایت اضافه شوند نه اینکه توسط کاربر نوشته شود. ( یعنی کاربر از بین گزینه های موجود انتخاب کند. )
- قیدهای سازگاری : سامانه باید با سامانه های دیگر هم فراهم سازی شود و با داده های سازمان ارتباط برقرار کند.
- مدیر هر سطح تنها به اطلاعات و گزارش های مربوط به سطح خود دسترسی داشته باشد.
- قیدهای قابلیت استفاده : سامانه باید بسیار آسان و قابل استفاده برای کاربران باشد.
- قیدهای راندمان : سامانه باید در عملکرد خود بهینه باشد و راندمان داشته باشد.
- سامانه باید از زبان های مختلف پشتیبانی کند.
- سامانه باید قابلیت Dark mode و Light mode را داشته باشد.

### 3-3-1- صفتهای سیستم

صفتهای مهم یک سیستم کارایی عبارتند از:

---

<sup>1</sup> Design Constraints

<sup>2</sup> Responsive

(۱) قابلیت جستجو:

سیستم باید به کارجوها اجازه دهد تا جستجوی شغل‌های مختلف را بر اساس معیارهای مختلفی مانند مکان، مهارت‌های مورد نیاز، حقوق و غیره انجام دهند.

(۲) قابلیت درج آگهی استخدام:

سیستم باید به کارفرمایان اجازه دهد تا آگهی‌های استخدام خود را در سیستم قرار داده و اطلاعات مربوط به شغل، محل کار و حقوق را به کارجوها نشان دهد.

(۳) قابلیت مدیریت رزومه:

سیستم باید ابزارهای مورد نیاز برای مدیریت رزومه را فراهم کرده و اطلاعات آن‌ها را به روز نگه دارد.

(۴) سهولت استفاده:

سیستم باید برای استفاده کاربران گوناگون بسیار ساده باشد، به طوری که هم کارفرمایان و هم کارجوها به راحتی با آن کار کنند.

(۵) قابلیت شخصی‌سازی:

سیستم باید به کاربران اجازه دهد تا پروفایل شخصی خود را ویرایش کنند و نیازهای خود را در قالب اطلاعات اضافه و حذف کنند.

(۶) قابلیت ارتقاء:

سیستم باید برای به‌روزرسانی‌های نرم‌افزاری و ارتقاءهای فنی انعطاف‌پذیر باشد.

### 3-5-1- برنامه‌ی تکرار

جدول ۳-۵-۱- برنامه‌ی تکرار

تکرار ۴ (۳ هفته) /2/26 1402 /3/16 1402	تکرار ۳ (۳ هفته) 1402/2/4 1402/2/25	تکرار ۲ (۳ هفته) 1402/1/13 1402/2/3	تکرار ۱ (۳ هفته) /12/20 1401 1402/1/12	وابستگی	اولویت	نیازمندی‌ها
				-	1	R1
				1	1	R2
				1,2	1	R3
				1,3	1	R4
				1,2,4	2	R5
				15	2	R6
				15	1	R7
				1,12	1	R8
				1,12	2	R9
				1,12	2	R10
				1	2	R11
				-	1	R12
				12	2	R13
				4,12	1	R14
				12	2	R15
				12	3	R16
				12	2	R17
				12	2	R18
				4,20,21	3	R19
				4,6	1	R20
				4,6	3	R21
				6,15	1	R22
				1,12	3	R23
				-	3	R24

# فصل دوم: مدل دامنه

## 1-2- شرح کلی:

مدل سازی دامنه، یک فرایند مفهوم سازی برای کمک به تیم توسعه جهت فهم دامنه ی کاربرد است. این فرایند به شناسایی مفاهیم مهم دامنه، ویژگی های آن ها و ارتباط بین این مفاهیم کمک می کند. همچنین مدل سازی دامنه به شناسایی و رفع برداشتهای متفاوت کمک می کند.

\* فرایند مدل سازی طی پنج گام زیر انجام می پذیرد:

۱. جمع آوری اطلاعات دامنه ی کاربرد

۲. طوفان فکری

۳. دسته بندی نتایج طوفان فکری

۴. به تصویر کشیدن مدل دامنه

۵. مرور مدل دامنه

## 2-2- جمع آوری اطلاعات دامنه کاربرد:

در این گام، اطلاعاتی درباره دامنه کاربرد سامانه جمع آوری شد. از جمله فعالیت هایی که برای این گام انجام گرفت، می توان به بررسی سامانه های مشابه و ... اشاره کرد.

## 3-2- طوفان فکری:

پس از گام اول و جمع آوری اطلاعات، در طی یک جلسه با حضور همه ی اعضای تیم، مفاهیم مهم دامنه شناسایی شد و در نهایت نتایج آن در یک جدول دسته بندی شد.



## 4-2- دسته‌بندی مفاهیم طوفان فکری:

در این گام از مدل‌سازی دامنه، مفاهیم شناسایی شده، به کلاس‌ها، ویژگی‌ها، مقادیر ویژگی‌ها، و روابط، دسته‌بندی شدند. محصول نهایی این گام، فهرستی از کلاس‌ها و ویژگی‌های آن‌ها، و ارتباطات بین این کلاس‌ها می‌باشد که در قالب یک جدول (جدول ۲-۴) رسم شده است.

\* توضیحات سیستم کاریابی:

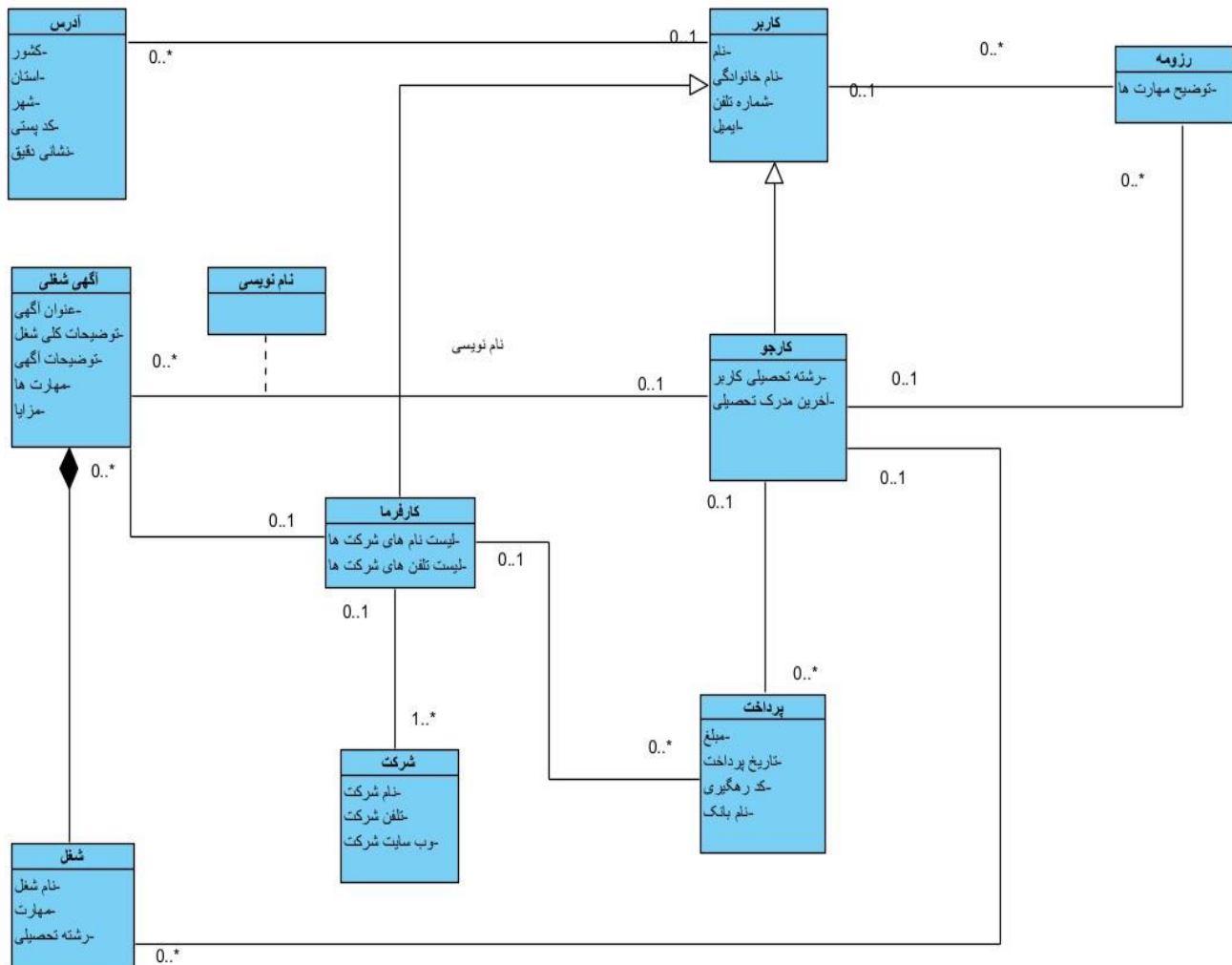
هر کاربر<sup>۱</sup> دارای نام<sup>۱</sup>، نام خانوادگی<sup>۱</sup>، شماره تلفن<sup>۱</sup>، ایمیل<sup>۱</sup> و آدرس<sup>۲</sup>، تاریخ تولد<sup>۱</sup>، وضعیت تاهل<sup>۱</sup>، جنسیت<sup>۱</sup> و بایوگرافی<sup>۱</sup> می‌باشد که می‌تواند به سیستم وارد شود (Log in) یا از سیستم خارج شود (Log out) و یا در سیستم ثبت نام کند (sign up). سیستم قابلیت اضافه کردن و ویرایش آدرس را دارد. آدرس<sup>۱</sup> شامل موارد کدپستی<sup>۱</sup>، کشور<sup>۱</sup>، استان<sup>۱</sup>، شهر<sup>۱</sup> و نشانی دقیق<sup>۱</sup> می‌باشد که هر کاربر می‌تواند تعداد صفر یا نامحدود<sup>۳</sup> از آن را داشته باشد. باید افراد بتوانند در آن شغل<sup>۱</sup>ها را اضافه یا کم کنند یا شغل‌ها را نمایش دهند. شغل‌ها شامل موارد نام شغل<sup>۱</sup>، مهارت<sup>۱</sup> و رشته تحصیلی<sup>۱</sup>. هر آگهی شغلی<sup>۱</sup> شامل موارد عنوان آگهی<sup>۱</sup>، توضیحات کلی شغل<sup>۱</sup>، توضیحات آگهی<sup>۱</sup>، مهارت‌ها<sup>۱</sup> و مزایا<sup>۱</sup> می‌باشد. این سامانه باید توانایی ساخت و ویرایش تعداد بسیاری<sup>۳</sup> آگهی<sup>۱</sup> را برای هر کارفرما<sup>۳</sup> را داشته باشد، همچنین هر آگهی یک شغل را به طور کامل در بر دارد<sup>۸</sup>. هر رزومه<sup>۱</sup> باید قابلیت ایجاد و بارگذاری داشته باشد و دارای اطلاعات کامل شخص<sup>۳</sup> و اطلاعات مهارت‌ها<sup>۱</sup> می‌باشد که هر کارجو تعداد صفر یا بی‌نهایت رزومه<sup>۳</sup> می‌تواند بسازد. هر کارفرما<sup>۱</sup> کاربری است<sup>۹</sup> که می‌تواند یک یا چندین شرکت<sup>۳</sup> داشته باشد و لیست نام شرکت‌هایش<sup>۱</sup> و شماره تلفن‌های آن‌ها<sup>۱</sup> را دارد. شرکت‌ها می‌توانند ثبت، حذف و ویرایش شوند همچنین هر شرکت<sup>۱</sup> دارای نام شرکت<sup>۱</sup>، تلفن<sup>۱</sup> و وبسایت<sup>۱</sup> می‌باشد. کارجو<sup>۱</sup> یک کاربر است<sup>۹</sup> که موارد رشته تحصیلی<sup>۱</sup> و آخرین مدرک تحصیلی<sup>۱</sup> را نیز دارد و قابلیت‌های ارسال رزومه به شرکت‌ها و جستجوی شغل را دارد. کارجو باید بتواند برای آگهی شغلی نام نویسی<sup>۱</sup> کند. برای پرداخت<sup>۱</sup> کردن، اطلاعات پرداخت ذخیره شده و عملیات پرداخت صورت می‌گیرد و موارد مبلغ<sup>۱</sup>، تاریخ پرداخت<sup>۱</sup>، کد رهگیری<sup>۱</sup> و نام بانک<sup>۱</sup> را ذخیره می‌شود، برای هر تعداد کارجو و کارفرما<sup>۴</sup>.

Brainstorming List	Classification Result	Rule
کاربر	(C) کاربر	۱(الف)
نام	(A) نام	۱(د)
نام خانوادگی	(A) نام خانوادگی	۱(د)
شماره تلفن	(A) شماره تلفن	۱(د)
ایمیل	(A) ایمیل	۱(د)
آدرس	(AG) آدرس	۲(الف)
آدرس	(C) آدرس	۱(الف)
کشور	(A) کشور	۱(د)
استان	(A) استان	۱(د)
شهر	(A) شهر	۱(د)
نشانی دقیق	(A) نشانی دقیق	۱(د)
تعداد صفر یا نامحدود	(AS) تعداد صفر یا نامحدود آدرس برای کاربر	۳(د)
شغل	(C) شغل	۱(الف)
نام شغل	(A) نام شغل	۱(د)
مهارت	(A) مهارت	۱(د)
رشته تحصیلی	(A) رشته تحصیلی	۱(د)
آگهی شغلی	(C) آگهی شغلی	۱(الف)
عنوان آگهی	(A) عنوان آگهی	۱(د)
توضیحات کلی شغل	(A) توضیحات کلی شغل	۱(د)
توضیحات آگهی	(A) توضیحات آگهی	۱(د)
مهارت‌ها	(A) مهارت‌ها	۱(د)
مزایا	(A) مزایا	۱(د)
رزومه	(C) رزومه	۱(الف)
اطلاعات کامل شخص	(AS) اطلاعات کامل شخص	۳(د)
اطلاعات مهارت‌ها	(A) اطلاعات مهارتی	۱(د)
تعداد صفر یا بی‌نهایت رزومه	(AS) تعداد صفر یا نامحدود رزومه برای کارجو	۳(د)
تعداد بسیاری آگهی برای هر کارفرما	(AS) تعداد بسیاری آگهی برای هر کارفرما	۳(د)
آگهی یک شغل در بر دارد	(AG) آگهی یک شغل در بر دارد	۸(ب)
کارفرما، کاربری است	(I) کارفرما یک کاربر است	۹(د)
شرکت	(C) شرکت	۱(الف)
یک یا چندین شرکت	(AS) یک یا چندین شرکت برای هر کارفرما	۳(د)
لیست نام شرکت‌ها	(A) لیست نام شرکت‌ها	۱(د)
لیست شماره تلفن شرکت‌ها	(A) لیست شماره تلفن شرکت‌ها	۱(د)
نام شرکت	(A) نام شرکت	۱(د)
تلفن	(A) تلفن شرکت	۱(د)

وبسایت	(A) وبسایت شرکت	۱(د)
کارجو	(C) کارجو	۱(الف)
کارجو یک کاربر است	(I) کارجو یک کاربر است	۹()
رشته تحصیلی	(A) رشته تحصیلی کارجو	۱(د)
آخرین مدرک تحصیلی	(A) آخرین مدرک تحصیلی	۱(د)
ارسال رزومه به شرکت‌ها	(A) ارسال رزومه به شرکت‌ها	۱(د)
جستجوی شغل	(A) جستجوی شغل	۱(د)
نام نویسی	(AC) نام نویسی	۱(ج)
پرداخت	(C) پرداخت	۱(الف)
مبلغ	(A) مبلغ	۱(د)
تاریخ پرداخت کد رهگیری	(A) تاریخ پرداخت کد رهگیری	۱(د)
نام بانک	(A) نام بانک	۱(د)
برای هر تعداد کارجو و کارفرما	(AS) پرداخت برای هر تعداد کارجو و کارفرما	۳()

## 5-2- به تصویر کشیدن مدل دامنه:

در این گام با استفاده از گام‌های قبل و اطلاعات به دست آمده، به وسیله نرم‌افزار Paradigm Visual مدل دامنه زیر (شکل ۵-۲-) رسم شد.



شکل ۵-۲-مدل دامنه

# فصل سوم: طراحی معماری

## 1-3- شرح کلی:

طراحی معماری سیستم نرم‌افزاری یکی از مراحل شناخته‌شده در توسعه نرم‌افزارهاست که برای اجرای بهتر و مستدام پروژه‌های نرم‌افزاری بسیار حائز اهمیت است. یک معماری صحیح، باعث کاهش هزینه‌ها، افزایش توسعه‌پذیری نرم‌افزار و بهره‌وری می‌شود و در پایان می‌تواند به موفقیت پروژه کمک کند. در این رابطه، طراحی معماری شامل مراحل، مفاهیم و الگوهای مختلفی است که با اطلاع‌رسانی شفاف و درک دقیق بخش‌های مختلف پروژه به عنوان یک رویکرد جامع، می‌تواند نتایج مطلوبی به همراه داشته باشد.

## 2-3- فرایند طراحی معماری:

فرایند طراحی معماری برای یک سیستم یا زیرسیستم نرم‌افزاری، یک فرایند شناختی تصمیم‌گیری است. این فرایند باید عوامل زیادی را در نظر بگیرد. نوع سیستمی که می‌خواهد توسعه داده شود، یک عامل مهم است. تجربه نشان داده است که نوع سیستم بر انتخاب سبک معماری موثر است. برای مثال یک نوع طراحی رایج ممکن است برای یک نوع سیستم با اهداف خاص مناسب نباشد. اهداف طراحی نیز بر انتخاب سبک معماری موثرند. در ادامه گام‌های طراحی معماری یک سیستم بررسی می‌شوند:

### • تعیین اهداف سیستم:

برخی از اهداف تعیین شده برای طراحی معماری سیستم کارایی به شرح زیر می‌باشند:

- **توسعه‌پذیری:** سیستم باید انعطاف‌پذیری لازم را درمقابل به‌روزرسانی‌ها و توسعه‌های احتمالی داشته باشد بدون این‌که باعث ایجاد اختلال در عملکرد آن شود.
- **مقیاس‌پذیری:** سیستم باید توانایی مدیریت تعداد فزاینده‌ای از کاربران و داده‌های شغلی، بدون به خطر انداختن عملکرد و کاهش کارایی، را داشته باشد.
- **امنیت:** سیستم باید توانایی حفاظت از داده‌های کاربران را داشته باشد و ارتباط امن بین لایه‌های مختلف و زیرسیستم‌ها را تضمین کند.

- **تحمل‌پذیری خطا:** سیستم باید توانایی بازیابی عملیات و اطلاعات را در صورت بروز خطا داشته باشد و به کار خود ادامه دهد.
- **قابلیت استفاده مجدد:** طراحی سیستم باید به گونه‌ای باشد که استفاده‌ی مجدد از کدها و اجزای سازنده‌ی آن در جهت بهبود زمان توسعه و کاهش هزینه‌ها ممکن باشد.
- **کاربرپسند بودن:** استفاده از محیط سیستم و تعامل کاربر با آن باید به راحتی امکان‌پذیر باشد.
- **قابلیت نگهداری:** معماری سیستم باید قابل فهم باشد و اصلاح آن سازگار با تغییر و توسعه‌ی نیازمندی‌ها به راحتی امکان‌پذیر باشد.

### ● تعیین نوع سیستم:

- سیستم کاریابی یک سیستم تعاملی است زیرا فرایندهای آن با ورودی‌های یک کنشگر شروع می‌شوند. این کنشگرها با توجه به نوع سیستم عموماً انسان هستند و از ورودی‌های مورد انتظار آن‌ها می‌توان به جستجوی شغل، به‌روزرسانی پروفایل و... اشاره کرد. یک فرایند در یک سیستم تعاملی با کنشگر شروع و به کنشگر ختم می‌شود. اتمام این فرایندها را می‌توان با خروجی‌های مناسب سیستم برای کاربر مانند توصیه‌ها و اعلان‌های شغلی، مشخص نمود.

### ● استفاده از سبک‌های معماری:

- با توجه به اهداف طراحی ذکر شده و تعاملی بودن سیستم، معماری چندلایه مناسب‌ترین گزینه برای طراحی سیستم است زیرا قابلیت جداسازی دغدغه‌ها را دارد و قابلیت توسعه‌پذیری برای هر لایه را به‌طور مستقل فراهم می‌کند. این ویژگی‌ها سیستمی کارآمدتر و قوی‌تر را تضمین می‌کنند که می‌تواند تعداد زیادی از کاربران و تعاملات را مدیریت کند.
- معماری چندلایه در حالت معمول از لایه‌های زیر تشکیل می‌شود که در این پروژه هم مورد استفاده قرار می‌گیرند:

(۱) لایه‌ی نمایش

(۲) لایه‌ی اشیای کسب و کار

(۳) لایه‌ی انباره‌ی مانا

(۴) لایه‌ی شبکه

\*در ادامه مزایای معماری چندلایه برای سیستم کاربایی به تفصیل شرح داده می‌شوند:

- **جداسازی دغدغه‌ها:** هر لایه مسئولیت خاصی دارد که درک، توسعه و نگهداری سیستم را آسان‌تر می‌کند.

- **مقیاس‌پذیری:** امکان توسعه‌ی مستقل هر لایه به سیستم اجازه می‌دهد تا کاربران و داده‌های شغلی بیشتری را بدون تأثیر بر عملکرد و کارایی کلی سیستم مدیریت کند.

- **توسعه‌پذیری:** طراحی ماژولار اجازه می‌دهد که تغییرات یا به‌روزرسانی در هر لایه، بدون تأثیر بر عملکرد کل سیستم، به صورت جداگانه انجام شود.

- **قابلیت استفاده‌ی مجدد:** اجزای سازنده و کدها را می‌توان در لایه‌های دیگر مورد استفاده‌ی مجدد قرار داد که باعث کاهش زمان توسعه و بهبود قابلیت نگهداری می‌شود.

## • تعیین واسط‌ها و زیرسیستم‌ها:

در این بخش لایه‌ها، زیرسیستم‌ها، توابع، واسط‌ها و رفتار تعاملی بین آن‌ها مشخص می‌شوند:

۱. **لایه‌ی Presentation:** این لایه دربرگیرنده‌ی رابط کاربری است و تعاملات کاربر با سیستم را مدیریت می‌کند. ممکن است شامل صفحات HTML، JavaScript، CSS یا مدل‌های نمایشی برای فهرست‌های شغلی، نتایج جستجو و پروفایل کاربر باشد. با لایه‌ی Application Logic برای پردازش ورودی‌های کاربر و به‌روزرسانی رابط کاربری تعامل دارد.

۲. **لایه‌ی Application Logic:** این لایه مسئول پردازش درخواست‌های کاربر، تصمیم‌گیری و مدیریت قوانین کسب‌وکار است. برای پیاده‌سازی این لایه می‌توان از فناوری‌هایی مانند Java، COBRA، DCOM و... استفاده نمود. این لایه خود به دو زیر لایه تقسیم می‌شود:

- **لایه‌ی Controller:** این لایه جریان داده‌های بین لایه‌ی Presentation و لایه Business objects را مدیریت می‌کند. پس از پردازش ورودی‌های کاربر، روش‌های منطقی تجاری مناسب برای انجام اقدامات لازم را فراخوانی و لایه‌ی ارائه را به‌روز می‌کند. با لایه‌ی Business objects برای انجام اقدامات بر اساس ورودی کاربر در تعامل است.

- لایه‌ی Business Objects: این لایه حاوی منطق اصلی برنامه است، مانند الگوریتم‌های تطبیق کار، احراز هویت کاربر و اعتبارسنجی داده‌ها. برای بازیابی و ذخیره‌ی داده‌ها با لایه‌ی Data Access تعامل دارد.

۴. لایه‌ی Data Access: این لایه ذخیره‌سازی و بازیابی داده‌ها را مدیریت و با پایگاه‌های داده (مانند MySQL، NoSQL یا PostgreSQL) یا سایر فناوری‌های ذخیره‌سازی داده کار می‌کند. به‌طور کلی می‌توان گفت این لایه واسطی برای تعامل لایه‌ی Application Logic و فناوری‌های ذخیره‌سازی داده است.

۵. لایه‌ی Network: این لایه ارتباط بین لایه‌ها و اجزای مختلف را مدیریت و انتقال امن و قالب‌بندی مناسب داده‌ها را تضمین می‌کند. به‌طور کلی واسطه‌های مورد نیاز برای ارسال و دریافت داده‌ها را برای هر لایه فراهم می‌کند. این لایه می‌تواند با استفاده از API ها یا سرویس‌های مختلف مانند ESB<sup>۱</sup> پیاده‌سازی شود.

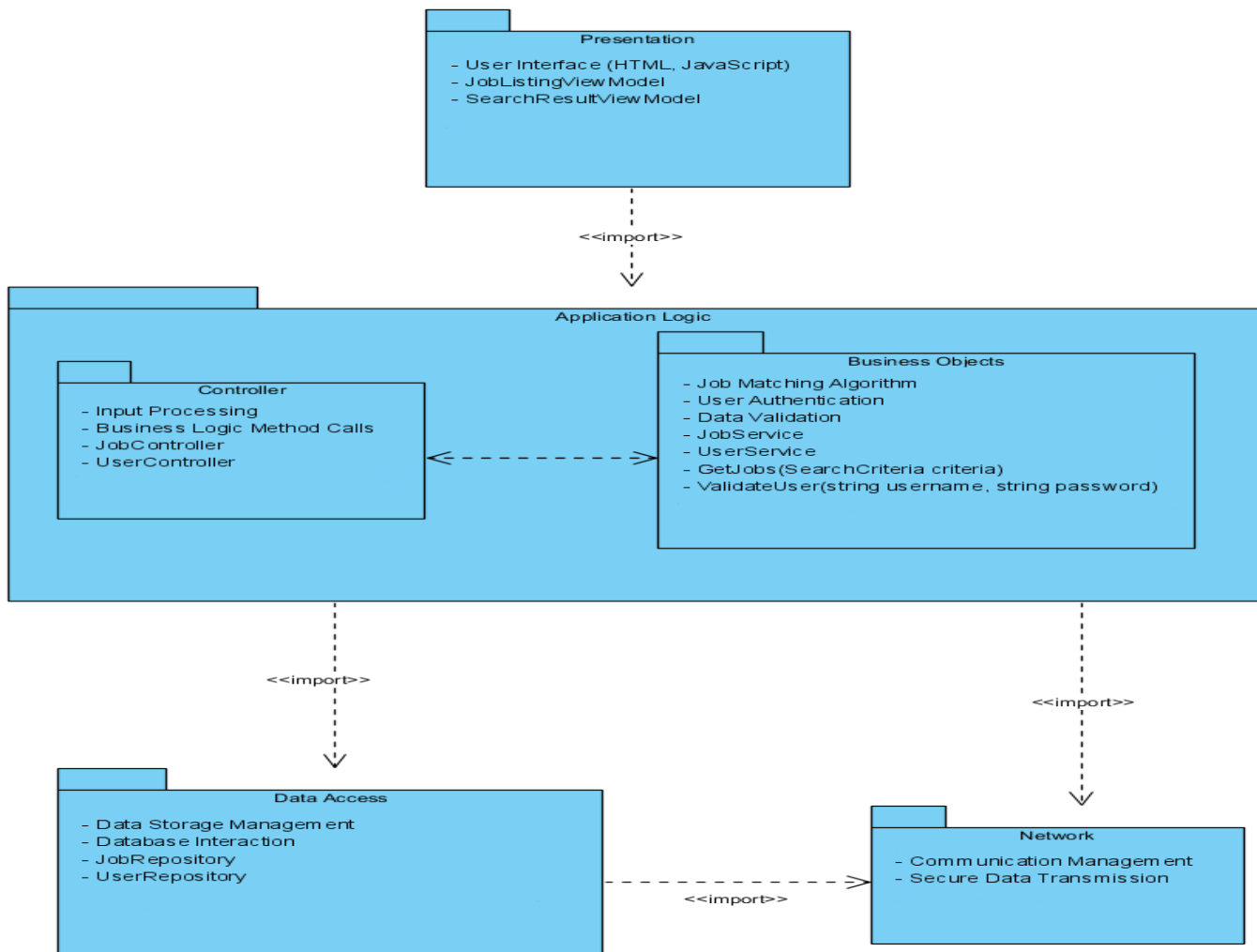
---

<sup>1</sup> Enterprise Service Bus



### 3-3- نمودار بسته:

برای نمایش اجزای سازنده و دست‌ساخته‌های نرم‌افزاری سیستم و نحوه تعامل بین آن‌ها به صورت ساخت‌یافته از نمودار بسته (شکل ۳-۳) استفاده می‌کنیم:



شکل ۳-۳- نمودار بسته

### 3-4- اعمال قوانین طراحی نرم افزار:

با اعمال این اصول در طراحی نرم‌افزار، سیستم کارایی قوی‌تر، قابل‌نگهداری و سازگارتر با تغییرات احتمالی خواهد بود که می‌تواند حجم زیادی از کاربران و تعاملات را مدیریت کند. این اصول به شرح زیر می‌باشند:

۱. طراحی برای تغییر: برای سازگاری با تغییرات احتمالی در آینده، سیستم باید با انعطاف طراحی شود. معماری چندلایه با سازماندهی سیستم به لایه‌های مجزا و مستقل، این انعطاف را فراهم می‌کند. با اعمال این

قانون با استفاده از این سبک معماری، جداسازی، جایگزینی، اصلاح و یا بهروزرسانی اجزای جداگانه‌ی سیستم بدون تأثیر بر کل سیستم آسان‌تر می‌شود. این قانون همچنین به اهداف سیستم از جمله توسعه‌پذیری، مقیاس‌پذیری، قابلیت نگهداری و قابلیت استفاده‌ی دوباره بهبود می‌بخشد. به عنوان مثال، الگوریتم تطابق شغلی را می‌توان به صورت یک ماژول جداگانه پیاده‌سازی کرد و به راحتی امکان بهروزرسانی یا جایگزینی آن را بدون تأثیر بر سایر قسمت‌های سیستم داشت.

۲. **جداسازی دغدغه‌ها:** همانطور که قبلاً ذکر شد، معماری چندلایه ذاتاً از جداسازی دغدغه‌ها پشتیبانی می‌کند. به دلیل این که هر لایه مسئولیت خاص خودش را دارد، تضمین می‌شود اجزای سیستم مستقل و متمرکز بر وظایف خاص خود عمل می‌کنند. این باعث بهبودی تمام اهداف تعیین شده در سیستم می‌شود.

۳. **پنهان‌سازی اطلاعات:** این اصل در معماری چندلایه با محدود کردن دسترسی کلی به داده‌های داخلی و منطق هر لایه اعمال می‌شود. اجرای مناسب کنترل دسترسی‌ها و تکنیک‌های کپسوله‌سازی می‌تواند عملکرد داخلی هر لایه و ماژول را پنهان کند. این امر احتمال وابستگی‌های ناخواسته را کاهش می‌دهد و سیستم را قابل نگهداری‌تر می‌کند.

۴. **چسبندگی زیاد:** اطمینان از این که هر ماژول یا اجزای نرم‌افزاری بر روی یک مسئولیت متمرکز شده‌اند، قابلیت درک و نگهداری سیستم را بهبود می‌بخشد و در نتیجه، به اهداف سیستم از جمله قابلیت نگهداری و توسعه‌پذیری کمک می‌کند. به عنوان مثال، الگوریتم تطبیق شغل باید یک ماژول جداگانه باشد که فقط بر روی یافتن مشاغل مورد جستجوی کاربران تمرکز دارد. معماری چندلایه به علت جداسازی سیستم به لایه‌های مستقل و با به حداقل رساندن تعاملات بین آن‌ها از این اصل پشتیبانی می‌کند.

۵. **جفت‌شدگی کم:** از آنجایی که معماری چندلایه تعاملات بین لایه‌ها را کاهش می‌دهد، وابستگی‌های متقابل بین ماژول‌ها و اجزای سیستم کم‌تر و به همین دلیل، امکان تغییرات و بهروزرسانی آسان‌تر فراهم می‌شود. به عنوان مثال، استفاده از واسط‌ها و تزریق وابستگی (dependency injection) می‌تواند با به حداقل رساندن جفت‌شدگی بین لایه‌ی Application Logic و لایه‌ی Data Access، به آن‌ها اجازه دهد به طور مستقل، توسعه یافته و به‌روز شوند. این اصل از اهداف سیستم از جمله توسعه‌پذیری، انعطاف‌پذیری و مقیاس‌پذیری پشتیبانی می‌کند.

۶. **ساده و احماقانه فرض کن:** سیستم باید با در نظر گرفتن سادگی طراحی شود و از پیچیدگی‌های غیرضروری جلوگیری کرد. معماری چندلایه این اصل را با سازماندهی سیستم به یک ساختار ساده و قابل درک اعمال می‌کند. تقسیم سیستم به لایه‌های مجزا و مستقل که هر کدام مسئولیت خاصی دارند، باعث می‌شود ساختار کلی سیستم قابل درک‌تر، مدیریت آن آسان‌تر و در نتیجه قابلیت نگهداری آن بهتر شود. این سادگی همچنین توسعه‌پذیری سیستم و قابلیت استفاده‌ی مجدد از اجزای آن را برای پروژه‌های دیگر آسان‌تر می‌کند.

# فصل چهارم: استنتاج مورد کاربردها از نیازمندی‌ها

در فصل‌های قبل روش‌ها و تکنیک‌هایی برای جمع‌آوری و تبیین نیازمندی‌ها معرفی کردیم. نیازمندی‌ها عبارتند از قابلیت‌هایی که سیستم باید ارائه دهد. بیان نیازمندی‌ها جمله‌های خبری هستند که تنها در خصوص قابلیت‌های سیستم صحبت می‌کنند ولی در خصوص چگونگی ارائه‌ی آن چیزی نمی‌گویند. "مورد کاربرد" راه‌حلی برای این مشکل ارائه می‌کند. قبل از آن باید بدانیم کنشگر چیست. "کنشگر" عبارت است از یک کسب‌وکار که به‌وسیله‌ی مجموعه‌ای از موجودیت‌های کسب‌وکاری یا ذی‌نفعان، که بیرون از سیستم بوده و با آن در تعامل هستند، بازی می‌شود. حال می‌توانیم به مفهومی خوبی از "مورد کاربرد" دست پیدا کنیم. مورد کاربرد یک فرایند کسب‌وکار است، با یک کنشگر شروع شده، با آن کنشگر خاتمه میابد، و یک وظیفه کسب‌وکاری را برای آن کنشگر به انجام می‌رساند. پس می‌توان گفت هر مورد کاربرد یک فرایند کسب‌وکاری است؛ باید توسط یک کنشگر آغاز گردد؛ باید توسط کنشگر خاتمه یابد به گونه‌ای که کنشگر بداند که فرایند کسب‌وکاری به طور موفقیت‌آمیز به پایان رسیده است؛ باید یک وظیفه کسب‌وکاری را برای کنشگر انجام دهند.

## 1-4- استنتاج مورد کاربردها از نیازمندی‌ها

همانطور که گفته شد، مورد کاربرد نیازمندی‌ها را پالایش کرده و یک طراحی از رفتار سیستم را مشخص می‌کند. بنابراین، مورد کاربردها باید از نیامندی‌ها استنتاج شده و نیازمندی‌ها را ارضا نمایند. در این قسمت، مراحل لازم برای استنتاج مورد کاربردها و زیر سیستم‌ها از نیازمندی‌های به دست آمده از فصل‌های قبل و تخصیص آن‌ها به تکرارها را ارائه می‌نماییم. به بیان دیگر ورودی این فصل سند SRS<sup>1</sup> و نیازمندی‌ها و اصلی‌ترین خروجی‌ها شامل نمودار مورد کاربرد، جدول ردیابی نیازمندی‌های مورد کاربرد و جدول تخصیص مورد کاربرد به تکرارها می‌باشند.

گام‌های استنتاج مورد کاربردها در زیر شرح داده شده‌اند؛ پنج گام وجود دارد:

- گام ۱: شناسایی مورد کاربردها
- گام ۲: تعیین قلمرو مورد کاربردها

<sup>1</sup> Software Requirements Specification

- گام ۳: ترسیم زمینه مورد کاربردها
- گام ۴: بازبینی مورد کاربردها و نمودارها
- گام ۵: تخصیص مورد کاربردها به تکرار

## 1-1-4-شناسایی مورد کاربردها

در این گام عبارت‌هایی که نشان‌دهنده‌ی فرایندهای کسب‌وکار هستند از روی نیازمندی‌ها شناسایی می‌شوند. شناسایی مورد کاربردها شامل چهار فعالیت می‌شود. ابتدا، اعضای تیم در نیازمندی‌ها به دنبال عبارت‌های فعلی-اسمی که مختص دامنه بودند، گشتند. این فعالیت در جهت استخراج مورد کاربردها با توجه به تعریف آن (در بالا به طور خلاصه توضیح داده شد) و مشخص کردن آن مورد به عنوان یک فرایند کسب‌وکاری بود. در گام دوم، برای پیدا کردن کنشگرهای هر مورد کاربرد به دنبال عبارت‌های اسمی گشتیم و نیز عبارت‌های اسمی که بیانگر زیرسیستم‌های مورد کاربرد بودند را مشخص کردیم تا چسبندگی زیرسیستم بهبود یابد. در ادامه جدولی شامل مورد کاربردهای بازچینی شده میان زیرسیستم‌ها به همراه کنشگر آمده است:

## 1-1-4-ماتریس ردیابی پذیرنی نیازمندی-مورد کاربرد

همانطور که پیش‌تر توضیح داده شد، مورد کاربردها از نیازمندی‌های پالایش شده استنتاج می‌شوند. از این رو RUTM<sup>۱</sup> برای ردیابی دوطرفه‌ی مورد کاربردها و نیازمندی‌ها رسم می‌شود. (ردیف‌ها بیانگر مورد کاربردها و ستون‌ها نشان‌دهنده نیازمندی‌ها می‌باشند.) در همین راستا تیم توسعه به کشیدن یک ماتریس ردیابی پذیرنی برای مورد کاربردها پرداخت، که نتیجه این فعالیت در جدول ۴-۱ آمده است.

UC32																				*	4
UC31																			*		5
UC30																		*			5
UC29																	*				4
UC28															*						3
UC27														*							4
UC26														*							3
UC25														*							2
UC24														*							2

<sup>1</sup> Requirements Use Case Traceability Matrix (RUTM)

UC23																*								5
UC22															*									3
UC21													*											3
UC20													*											3
UC19												*												4
UC18												*												4
UC17											*													1
UC16										*														2
UC15									*															5
UC14								*																5
UC13							*																	5
UC12						*																		5
UC11					*																			5
UC10					*																			5
UC09				*																				4
UC08				*																				4
UC07			*																					4
UC06			*																					4
UC05			*																					4
UC04		*																						3
UC03	*																							2
UC02	*																							2
UC01	*																							1

Requirements	Priority
R01	1
R02	2
R03	3
R04	4
R05	4
R06	5
R07	5
R08	5
R09	5
R10	5
R11	2
R12	1
R13	4
R14	4
R15	3
R16	5
R17	2
R18	3
R19	4
R20	5
R21	5
R22	4
R23	4
R24	5
UC Priority	

جدول 1-4 جدول ردیابی نیازمندی-مورد کاربرد

## 2-1-4-تعیین قلمرو موارد کاربرد

مورد کاربرد سطح بالا مشخص می‌کند که یک مورد کاربرد کی و کجا شروع شده و کی تمام می‌شود. به عبارت دیگر، قلمرو مورد کاربرد را مشخص می‌کند. تبیین قلمرو مورد کاربرد در واقع قانون توقف را تعریف می‌کند. برای هر مورد کاربرد، سه تصمیم گرفته می‌شود:

۱. مورد کاربرد کی شروع می‌شود؛
۲. کنشگر کجا اتفاق می‌افتد؛
۳. مورد کاربرد کی تمام می‌شود.

در طی مرحله تعیین قلمرو مورد کاربرد، تیم توسعه به مشخص کردن زمان، محل شروع شدن و همین‌طور خاتمه یافتن هر مورد کاربرد مجرد مشخص شده در مرحله قبل، پرداخت.

خروجی این گام که مورد کاربردهای سطح بالا هستند در ادامه آمده است:

UC01) ثبت‌نام کاربر به عنوان کارفرما:

- TUCBW: کاربر(کارفرما) بر روی پیوند ثبت‌نام به عنوان کارفرما کلیک می‌کند.
- TUCEW: در صورت موفقیت‌آمیز بودن ثبت‌نام، کاربر وارد صفحه شخصی خود می‌شود.

UC02) ثبت‌نام رابط شرکت

- TUCBW: کاربر(کارفرما) روی پیوند ثبت‌نام رابط شرکت کلیک می‌کند.
- TUCEW: در صورت موفقیت‌آمیز بودن ثبت‌نام، کاربر پیام « ثبت‌نام رابط شرکت موفقیت‌آمیز بود» را مشاهده می‌کند و در غیر این صورت، کاربر پیام « ثبت‌نام با مشکل مواجه شد» را مشاهده می‌کند.

### UC03) ویرایش اطلاعات رابط شرکت

- TUCBW: کاربر(کارفرما) بر روی پیوند ویرایش اطلاعات که زیر اطلاعات هر شرکت قرار دارد کلیک می‌کند.
- TUCEW: کاربر(کارفرما) صفحه‌ای که در آن توانایی تغییر دادن اطلاعات رابط شرکت وجود دارد را مشاهده می‌کند.

### UC04) پیش‌نویسی آگهی

- TUCBW: کاربر(کارفرما) روی پیوند ایجاد آگهی جدید کلیک می‌کند.
- TUCEW: کاربر (کارفرما) صفحه‌ای که در آن امکان نوشتن آگهی وجود دارد را مشاهده می‌کند.

### UC05) درخواست ثبت آگهی

- TUCBW: کاربر(کارفرما) روی پیوند ثبت آگهی کلیک می‌کند.
- TUCEW: کاربر نوار وضعیت درخواست آگهی را روی صفحه نمایش می‌بیند.

### UC06) وضعیت درخواست آگهی

- TUCBW: کاربر(کارفرما) روی پیوند وضعیت درخواست آگهی کلیک می‌کند.
- TUCEW: در صورت در دست بررسی بودن درخواست، کاربر پیام «در دست بررسی» و در صورت موافقت پیام «تایید شده» و پیوند پرداخت هزینه آگهی را مشاهده می‌کند و در صورت رد درخواست، «پیام رد شده» را مشاهده می‌کند.

### UC07) پرداخت آنلاین هزینه‌ی ثبت آگهی

- TUCBW: کاربر(کارفرما) روی پیوند پرداخت هزینه‌ی آگهی کلیک می‌کند.
- TUCEW: در صورت موفقیت‌آمیز بودن عملیات پرداخت، کاربر پیام «تراکنش با موفقیت انجام شد» را مشاهده می‌کند و در غیر این صورت، پیام «تراکنش انجام نشد» را مشاهده می‌کند.

### UC08) امکان حذف آگهی

- TUCBW: کاربر(کارفرما) روی پیوند حذف آگهی کلیک می‌کند.
- TUCEW: در صورت موفقیت‌آمیز بودن حذف، کاربر پیام «آگهی با موفقیت حذف شد» را مشاهده می‌کند و در غیر این صورت، پیام «آگهی حذف نشد» را مشاهده می‌کند.

### UC09) امکان آرشیو آگهی

- TUCBW: کاربر(کارفرما) روی پیوند آرشیو آگهی کلیک می‌کند.

- TUCEW: در صورت موفقیت‌آمیز بودن عملیات، کاربر پیام «آرشیو آگهی با موفقیت انجام شد» را مشاهده می‌کند و در غیر این صورت، کاربر پیام «آرشیو آگهی انجام نشد» را مشاهده می‌کند.

#### UC10) مشاهده‌ی رزومه توسط کاربر (کارفرما)

- TUCBW: کاربر (کارفرما) روی پیوند مشاهده‌ی رزومه کلیک می‌کند.
- TUCEW: کاربر، در صورت وجود رزومه، رزومه‌ها را مشاهده می‌کند و در غیر این صورت، پیام «هیچ رزومه‌ای جهت نمایش وجود ندارد» را مشاهده می‌کند.

#### UC11) اعمال فیلترهای شخصی روی رزومه‌ها

- TUCBW: کاربر (کارفرما) از بین لیست فیلترهای شخصی، یکی را انتخاب می‌کند.
- TUCEW: فیلتر انتخاب شده روی رزومه‌ها اعمال می‌شود و کاربر رزومه‌ها را طبق فیلتر انتخاب‌شده در صفحه نمایش، مشاهده می‌کند.

#### UC12) مشاهده‌ی رزومه‌های ارسال‌شده برای آگهی‌های ثبت‌شده

- TUCBW: کاربر (کارفرما) روی پیوند مشاهده رزومه‌های ارسال‌شده که زیر هر آگهی ثبت‌شده قرار دارد، کلیک می‌کند.
- TUCEW: در صورت وجود رزومه، کاربر رزومه‌ها را روی صفحه نمایش، مشاهده می‌کند و در غیر این صورت پیام «هیچ رزومه‌ای برای نمایش وجود ندارد» را مشاهده می‌کند.

#### UC13) انتخاب کارجو توسط کاربر (کارفرما)

- TUCBW: کاربر (کارفرما) روی پیوند انتخاب کارجو کلیک می‌کند.
- TUCEW: در صورت وجود کارجو، کاربر اسامی کارجویان را مشاهده می‌کند و امکان انتخاب کارجو برای کاربر (کارفرما) فراهم می‌شود و در غیر این صورت پیام «هیچ کارجویی برای نمایش وجود ندارد» را مشاهده می‌کند.

#### UC14) پیگیری وضعیت کارجوی انتخاب‌شده توسط کاربر (کارفرما)

- TUCBW: کاربر (کارفرما) روی پیوند وضعیت کارجوی انتخاب‌شده کلیک می‌کند.
- TUCEW: کاربر وضعیت کارجو را طبق شرایط روی صفحه نمایش، مشاهده می‌کند.

#### UC15) برگزاری جلسات حضوری یا آنلاین با کارجو برای مصاحبه

- TUCBW: کاربر (کارفرما) روی پیوند ایجاد جلسه کلیک می‌کند.



- TUCEW: کارجو پیام تشکیل جلسه را در صفحه شخصی خود و همچنین تلفن همراه خود مشاهده می کند و کاربر (کارفرما) پیام «جلسه ی جدید با موفقیت ایجاد شد» را مشاهده می کند.

#### UC16) تغییر یا بازیابی نام کاربری و گذرواژه

- TUCBW: کاربر (کارفرما) روی پیوند تغییر گذرواژه یا نام کاربری کلیک می کند.
- TUCEW: کاربر (کارفرما) صفحه ای شامل فیلدهای خالی نام کاربری جدید و گذرواژه ی قبلی و گذرواژه ی جدید و تکرار آن را مشاهده می کند.

#### UC17) ثبت نام و ورود کارجو به سامانه :

- TUCBW: کاربر (کارجو) بر روی پیوند ثبت نام یا ورود به سامانه کلیک می کند.
- TUCEW: کاربر (کارجو) پیام « ثبت نام شما موفقیت آمیز بود » یا « شما با موفقیت وارد شدید » و سپس صفحه ی پنل شخصی خود را مشاهده می کند.

#### UC18) جستجو بین محل کارهای متفاوت

- TUCBW: کاربر (کارجو) سیستم با کلیک روی پیوند جستجوی محل کار با وارد کردن استان و شهر و آدرس محل سکونت در فیلدهای مخصوص خود روی نشانه ی ذره بین کلیک می کند.
- TUCEW: کاربر (کارجو) فهرست جدول گونه ای از کار و محل کارها را که با فیلترهای استان و شهر و آدرس محل سکونت مطابقت دارد را مشاهده می کند .

#### UC19) مشاهده ی فرصت های شغلی بر اساس اطلاعات شخصی

- TUCBW: کاربر (کارجو) روی پیوند فرصت های شغلی کلیک می کند.
- TUCEW: کاربر (کارجو) فهرست فرصت های شغلی موجود بر اساس نوع همکاری و رزومه و موقعیت مکانی و سازمان ها را مشاهده می کند.

#### UC20) ساخت رزومه

- TUCBW: کاربر (کارجو) روی پیوند ساخت رزومه کلیک می کند.

- TUCEW : کاربر(کارجو) فیلدهای ضروری مثل اطلاعات اولیه، سوابق تحصیلی، سوابق شغلی، زبان‌های تحت تسلط، مهارت‌های تکمیلی، ثبت معلولیت یا بیماری خاص و ... را پر می‌کند سپس روی ثبت کلیک می‌کند و پیام «رزومه‌ی شما با موفقیت ساخته شد» را مشاهده می‌کند.

#### UC21) ویرایش رزومه

- TUCBW : کاربر(کارجو) روی پیوند ویرایش رزومه کلیک می‌کند.
- TUCEW : کاربر(کارجو) فیلدهای موردنظر مثل اطلاعات اولیه، سوابق تحصیلی، سوابق شغلی، زبان‌های تحت تسلط، مهارت‌های تکمیلی، ثبت معلولیت یا بیماری خاص و ... را ویرایش می‌کند سپس روی ثبت کلیک می‌کند و پیام «رزومه‌ی شما با موفقیت ویرایش شد» را مشاهده می‌کند.

#### UC22) شرکت در دوره‌های آموزشی جهت بهبود مهارت و رزومه

- TUCBW : کاربر(کارجو) روی پیوند دوره‌های آموزشی کلیک می‌کند.
- TUCEW : کاربر(کارجو) فهرست دوره‌های آموزشی را مشاهده می‌کند.

#### UC23) تغییر نام کاربری یا گذرواژه

- TUCBW : کاربر(کارجو) روی پیوند تغییر گذرواژه یا نام کاربری کلیک می‌کند.
- TUCEW : کاربر(کارجو) پیام «گذرواژه با موفقیت تغییر یافت» یا «لطفاً از گذرواژه قوی‌تر استفاده شود» یا «لطفاً از نام کاربری دیگری استفاده کنید.» را مشاهده می‌کند.

#### UC24) بازیابی گذرواژه

- TUCBW : کاربر(کارجو) روی پیوند بازیابی گذرواژه کلیک می‌کند.
- TUCEW : کاربر(کارجو) پیامک یا ایمیل حاوی رمز عبور جدید برای ورود دوباره را مشاهده می‌کند.

#### UC25) تاریخچه‌ی درخواست‌ها

- TUCBW : کاربر(کارجو) روی پروفایل خود سپس روی تاریخچه کلیک می‌کند.
- TUCEW : کاربر(کارجو) تاریخچه‌ی درخواست‌های خود را مشاهده می‌کند.

## UC26) وضعیت درخواست‌ها

- TUCBW : کاربر(کارجو) روی پروفایل خود سپس روی وضعیت درخواست‌ها کلیک می‌کند.
- TUCEW : کاربر(کارجو) فهرست درخواست‌ها و در کنار آن وضعیتشان را مشاهده می‌کند.

## UC27) اطلاع‌رسانی مشاغل مرتبط جدید

- TUCBW : کاربر(کارجو) در نوار بالایی روی پیوند اطلاع‌رسانی مشاغل مرتبط کلیک می‌کند.
- TUCEW : کاربر(کارجو) فهرست مشاغل مرتبط جدید را مشاهده می‌کند.

## UC28) بررسی داده‌های بازارکار

- TUCBW : کاربر(کارفرما،کارجو) روی پیوند داده‌های بازارکار کلیک می‌کند.
- TUCEW : کاربر(کارفرما،کارجو) گزارش‌های حقوق و دستمزد و گزارش‌های آماری بازارکار را مشاهده می‌کند.

## UC29) شرکت در آزمون‌های خودشناسی

- TUCBW : کاربر(کارفرما،کارجو) روی پیوند آزمون‌های خودشناسی و سپس روی یکی از پیوندهای زیر کلیک می‌کند:

الف) آزمون شخصیت‌سنجی MBTI

ب) آزمون تیپ‌سنجی شغلی Holland

ج) آزمون شخصیت‌شناسی NEO

د) آزمون هوش‌های چندگانه

ه) آزمون هوش هیجانی Bar-On

- TUCEW : کاربر(کارفرما،کارجو) صفحه‌ی تست سوالات مورد نظر را مشاهده می‌کند.

## UC30) مشاهده سوابق جست‌وجو

- TUCBW : کاربر(کارفرما،کارجو) روی پروفایل سپس روی پیوند سوابق جست‌وجو کلیک می‌کند.
- TUCEW : کاربر(کارفرما،کارجو) سوابق جست‌جوی خود را مشاهده می‌کند.

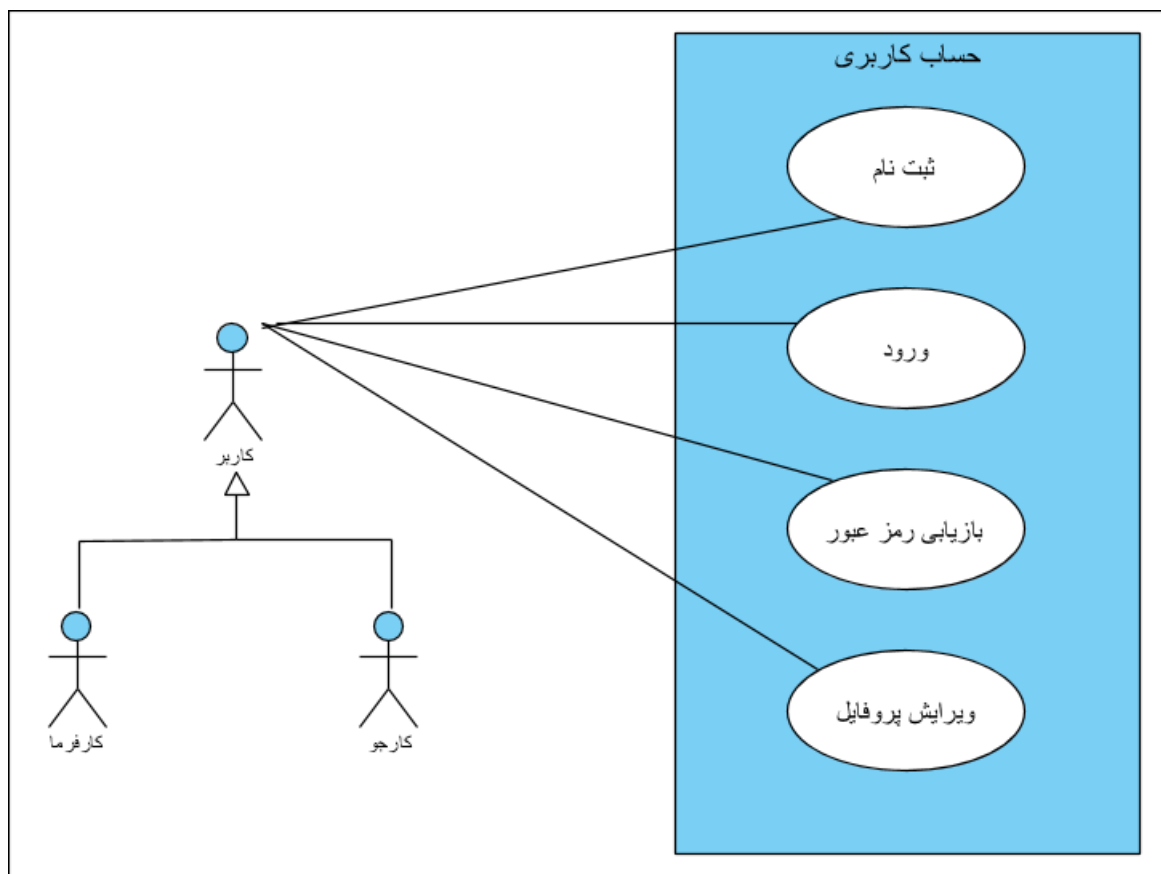
### UC31 ارسال انتقادات و پیشنهادات

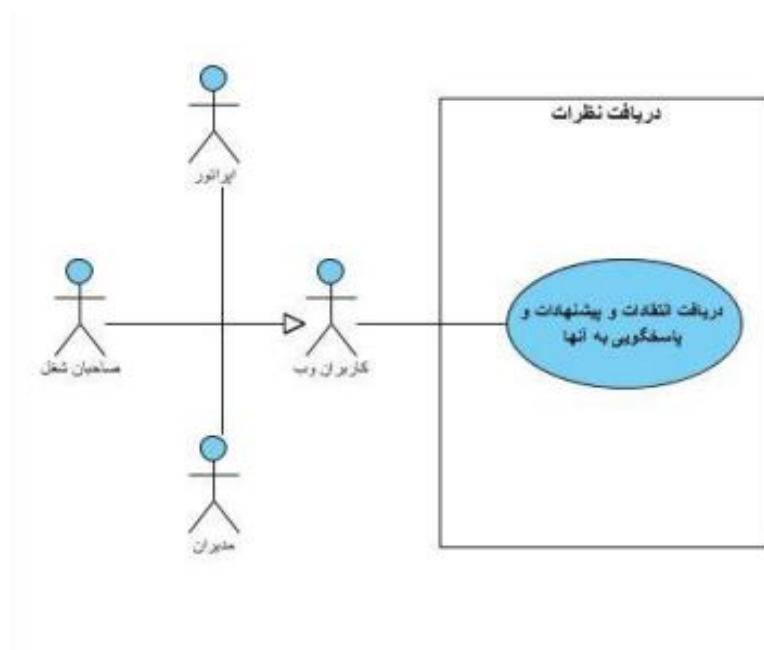
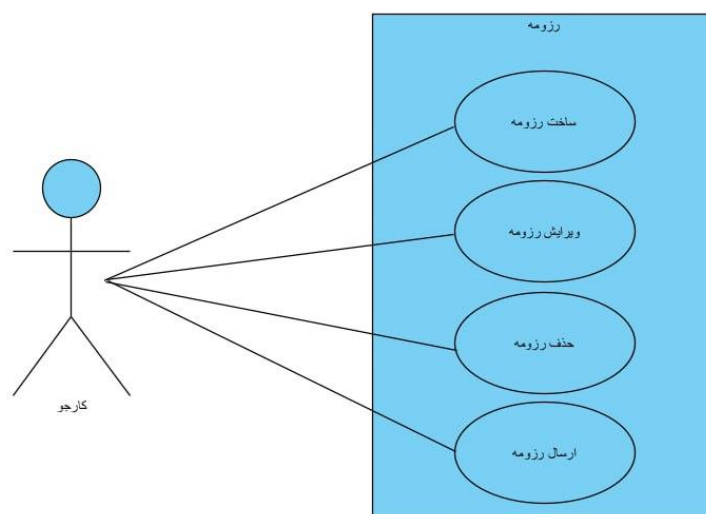
- TUCBW : کاربر(کارفرما،کارجو) روی پیوند انتقادات و پیشنهادات کلیک می کند و نظرات خود را ارسال می کند.
- TUCEW : کاربر(کارفرما،کارجو) پیام «نظرات شما با موفقیت ارسال شد و اقدامات لازم در صورت صلاحدید انجام می شود. با تشکر» را مشاهده می کند.

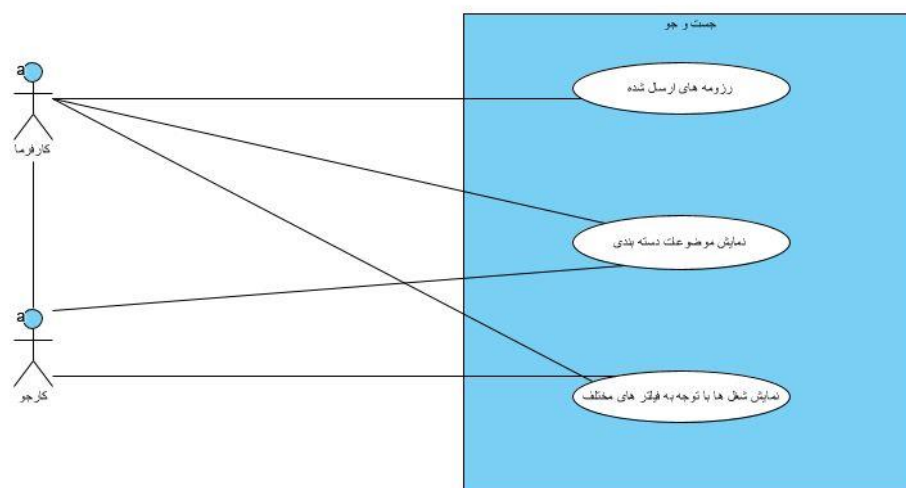
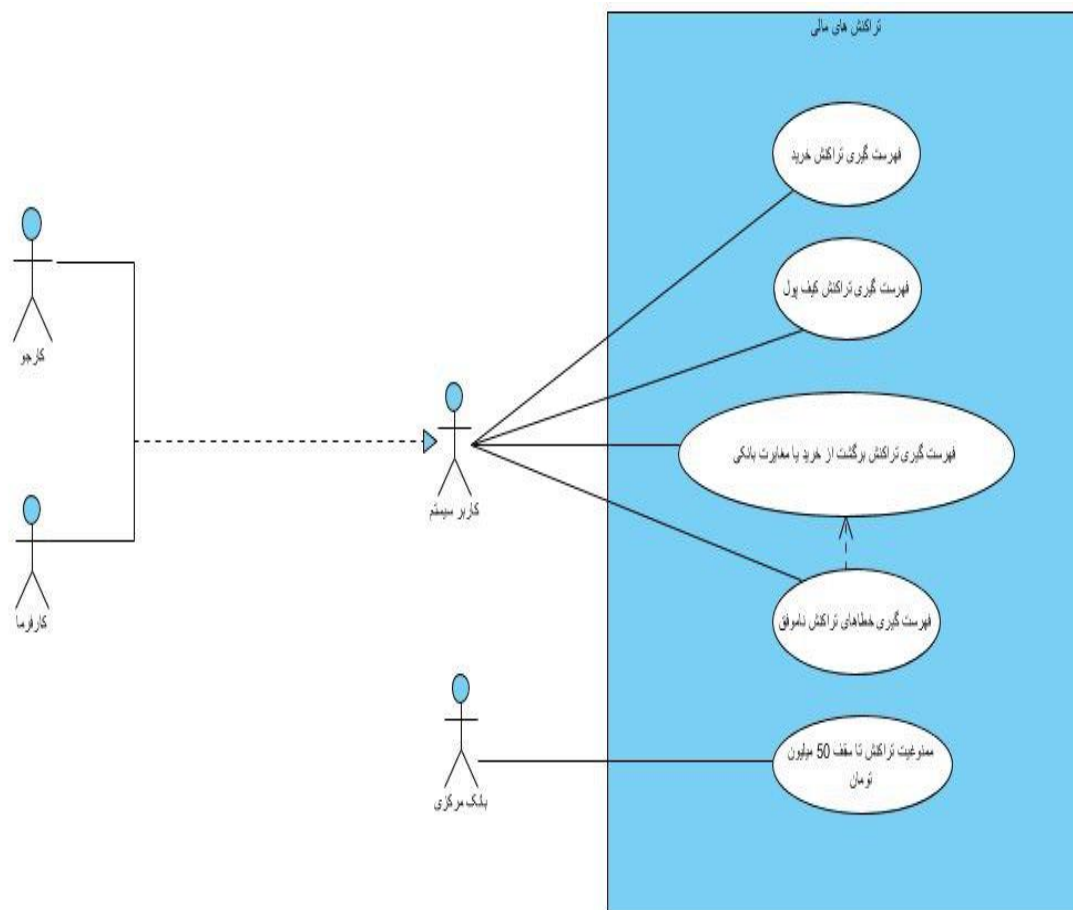
### UC32 راهنمای جامع

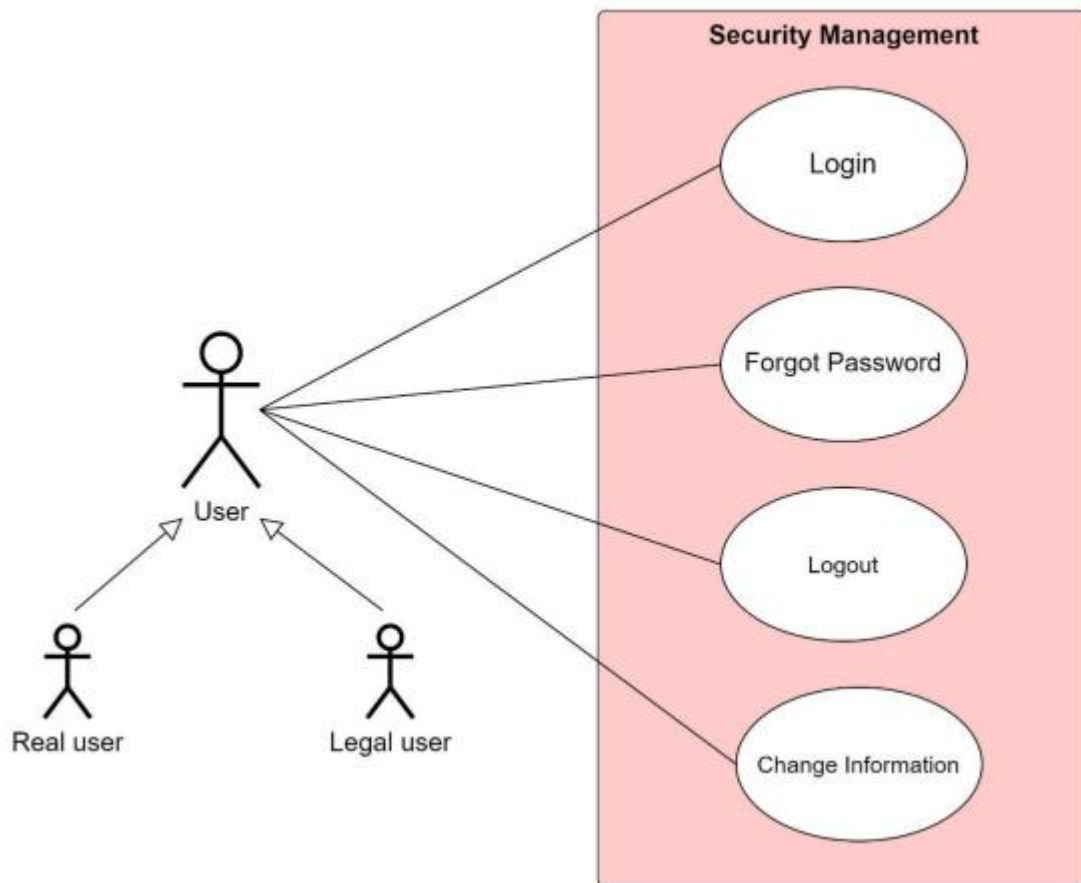
- TUCBW : کاربر(کارفرما،کارجو) در نوار بالایی روی پیوند راهنمای جامع کلیک می کند.
- TUCEW : کاربر(کارفرما،کارجو) راهنمای جامع سامانه را مشاهده می کند.

## 3-1-4- ترسیم نمودار مورد کاربردها









#### 4-1-4- بازبینی مورد کاربردها و نمودارها

در این گام، با توجه به فهرست بررسی مشخصات مورد کاربردها به بازبینی مورد کاربردهای مجرد، ماتریس ردیابی نیازمندی-مورد کاربردها، مورد کاربردهای سطح بالا و در انتها نمودار مورد کاربرد پرداختیم.

#### 4-1-5- تخصیص مورد کاربردها به تکرارها

تا کنون مورد کاربردها شناسایی و به تصویر کشیده شده‌اند و اولویت آن‌ها از روی اولویت نیازمندی‌ها به دست آمده‌است. حال باید یک زمان‌بندی برای توسعه و تحویل مورد کاربردها تولید شود. سپس در آن برنامه‌ریزی می‌شود که در هر تکرار چه مورد کاربردهایی توسعه یابند و تحویل مشتری داده شوند. جهت تهیه‌ی جدول تخصیص مورد کاربردها به تکرار، ابتدا میزان تلاش برای توسعه و تحویل هر مورد کاربرد را با توجه به اینکه این پروژه در تکرار سه هفته‌ای و توسط یک تیم ۶ نفره باید اجرا شود، تخمین زده شد. سپس با توجه به اولویت‌ها وابستگی‌های مورد کاربردها، جدول ۴-۲ تهیه شد.

Use Case	Priority	Effort (Person-Week)	Depend On	Iteration 1-3 wks	Iteration 2-3 wks	Iteration 3-3 wks
UC01	1	2	None	1	1	
UC02	2	4	UC01	2.5	1	0.5
UC03	2	5	UC02	2	1	2
UC04	3	2	UC02	1		1
UC05	4	8	UC04	5	1	2
UC06	4	3	UC05	2		1
UC07	4	3	UC05	1	1	1
UC08	4	7	UC05	3	3	2
UC09	4	4	UC05		4	
UC10	5	2	UC20		1	1
UC11	5	2	UC20		1.5	1.5
UC12	5	3	UC20	0.5	1	1.5
UC13	5	6	UC12	2	4	
UC14	5	5	UC13	0.5	1.5	3
UC15	5	4	UC13	1	0.5	1.5
UC16	2	3	UC01		1.5	1.5
UC17	1	2	None		0.25	1.75
UC18	4	2	UC07		0.5	1.5
UC19	4	1	UC20			1
UC20	3	6	UC17	6		
UC21	3	4	UC20	3	1	
UC22	3	3	UC17	2	1	
UC23	5	3	UC17		1	2
UC24	2	3	UC17		1.5	1.5
UC25	2	2	UC20			2
UC26	3	1	UC20			1
UC27	4	4	UC07		3	1
UC28	3	3	None		1.5	2.5
UC29	4	1	UC17	1		
UC30	5	1	UC18		0.25	0.75
UC31	5		None			
UC32	4		None			
Total Effort	-	99	-	33.5	33	34.5

Team Size = 6 Persons

جدول 2-4 تخصیص مورد کاربردها به تکرارها



## 2-4- مدل سازی تعامل کنشگر-سیستم

UC01 ثبت نام کاربر به عنوان کارفرما

	پیش شرط :-
کنشگر : کارفرما	سامانه : کاریابی هوشمند
	۰- سامانه صفحه‌ی قوانین و شرایط ثبت نام را نشان می دهد.
۱- TUCBW کارفرما روی پیوند «ثبت نام به عنوان کارفرما» کلیک می کند.	۲- سیستم صفحه‌ای برای دریافت اطلاعات کاربری (نام کاربری، ایمیل، شماره تلفن و...) را به کارفرما نشان می دهد.
۳- کارفرما اطلاعات خود را وارد می کند.	۴- سیستم با استفاده از ایمیل یا شماره تلفن احراز هویت می کند.
۵- کارفرما کد تایید ارسال شده را وارد می کند.	۶- سیستم یکی از اقدامات زیر را انجام می دهد: الف) اگر کد درست بود کارفرما به صفحه‌ی اصلی منتقل می شود. ب) اگر نادرست بود پیغام خطا چاپ می شود.
۷- TUCEW کارفرما صفحه‌ی اصلی و پروفایل خود را مشاهده می کند.	

جدول ۴-۲-۱- جدول مورد کاربرد گسترده شماره یک

## UC23) تغییر نام کاربری یا گذرواژه

پیش شرط : کاربر با موفقیت وارد حساب کاربری شده است.	
کنشگر : کاربر سامانه	سامانه : کاربایی هوشمند
۰- سامانه پیشخوان کاربر را نمایش می دهد.	
۱- TUCBW کاربر روی پیوند تغییر نام کاربری/گذرواژه کلیک می کند.	۲- مرورگر صفحه مربوط به تغییر نام کاربری/گذرواژه را نشان می دهد.
۳- کاربر در قسمت تغییر گذرواژه، گذرواژه قبلی و گذرواژه جدید و تکرار آن را وارد می کند. برای تغییر نام کاربری، نام کاربری جدید خود را در قسمت مربوطه وارد می کند و اگر قصد تغییر آن را ندارد همان نام کاربری قبلی را وارد می کند.	۴- سامانه گذرواژه/ نام کاربری ورودی را بررسی می کند. الف) در صورت قابل قبول بودن گذرواژه، رمز عبور کاربر تغییر می کند و پیام «گذرواژه با موفقیت تغییر یافت.» نمایش داده می شود. ب) در غیر این صورت پیام «لطفا از گذرواژه قوی تر استفاده کنید.» نمایش داده می شود. پ) در صورت تکراری یا غیر قابل قبول بودن نام کاربری پیام «لطفا از نام کاربری دیگری استفاده کنید.» نمایش داده می شود.
۵- TUCEW کاربر الف) پیام «گذرواژه با موفقیت تغییر یافت.» را مشاهده می کند. ب) پیام «لطفا از گذرواژه قوی تر استفاده کنید.» را مشاهده می کند. پ) پیام «لطفا از نام کاربری دیگری استفاده کنید.» را مشاهده می کند.	

جدول ۴-۲-۲- جدول مورد کاربرد گسترده شماره دو

## UC18) جستجو بین محل کار های متفاوت

پیش شرط : کاربر (کارجو) با موفقیت وارد حساب کاربری شده است.	
کنشگر : کاربر (کارجو) سامانه	سامانه : کار یابی هوشمند
۱- TUCBW کاربر روی پیوند «جستجوی محل کارها» کلیک می کند.	۲- مرورگر صفحه مربوط به جستجوی محل کارها نشان می دهد.
۳- کاربر : الف) روش خودکار را انتخاب می کند که در این صورت باید مکان یاب خود را روشن و روی گزینه ی تایید کلیک کند. ب) به روش دستی در قسمت های مربوطه، استان و شهر و آدرس محل سکونت خود را وارد می کند سپس روی گزینه ثبت کلیک می کند.	۴- با پیدا کردن موقعیت مکانی کاربر، سامانه فهرستی از نزدیک ترین محل کارها به همراه توضیحات کار را نمایش می دهد.
۵- TUCEW کاربر فهرستی از نزدیک ترین محل کارها به همراه توضیحات کار را مشاهده می کند.	

جدول ۴-۲-۳- جدول مورد کاربرد گسترده شماره سه

## UC24) بازیابی گذرواژه

پیش شرط : کاربر (کارجو) با موفقیت وارد حساب کاربری شده است.	
کنشگر : کاربر (کارجو) سامانه	سامانه : کاریابی هوشمند
۱- TUCBW کاربر روی پیوند بازیابی رمز عبور کلیک می کند.	۲- سامانه صفحه ای به شما نشان می دهد که می توانید برای دریافت کد تایید از دو طریق تلفن همراه یا ایمیل اقدام کنید.
۳- کاربر تلفن همراه یا ایمیل خود را وارد می کند و روی گزینه تایید کلیک می کند.	۴- سامانه جهت اعتبارسنجی کد تایید به تلفن همراه یا ایمیل کاربر ارسال می کند.
۵- کاربر کد تایید را وارد می کند.	۶- سامانه صفحه ای که گذرواژه جدید را دریافت می کند را نشان می دهد.
۷- کاربر گذرواژه جدید را وارد می کند و روی دکمه تایید کلیک می کند.	۸- سامانه : الف) در صورت قابل قبول بودن گذرواژه پیغام «گذرواژه ی شما با موفقیت تغییر یافت» را نشان می دهد. ب) در غیر این صورت پیغام «لطفا از گذرواژه قوی تر استفاده کنید» را نشان می دهد.
۹- TUCEW کاربر : الف) کاربر پیغام «گذرواژه ی شما با موفقیت تغییر یافت» را مشاهده می کند. ب) کاربر پیغام «لطفا از گذرواژه قوی تر استفاده کنید» را مشاهده می کند.	

جدول ۴-۲-۴- جدول مورد کاربرد گسترده شماره چهار

## UC05 درخواست ثبت آگهی

پیش نویس: کاربر (کارفرما) با موفقیت آگهی ای که مدنظر دارد را پیش نویسی کرده است.	
کنشگر: کاربر (کارفرما) سامانه	سامانه: کاریابی هوشمند
۱- TUCBW کاربر (کارفرما) روی پیوند ثبت آگهی کلیک می کند.	۲- مرورگر صفحه ی مربوط به ثبت آگهی را نمایش می دهد.
۳- کاربر: الف) روی پیوند ذخیره کلیک می کند تا درخواست ثبت آگهی خود را ارسال کند. ب) روی پیوند بازگشت کلیک می کند تا به مرحله قبل برود.	۴- در صورت اینکه کاربر روی پیوند ذخیره کلیک کرده باشد، سامانه صفحه ای که وضعیت درخواست کاربر را نشان می دهد را نمایش می دهد.
۵- TUCEW کاربر نوار وضعیت درخواست آگهی را روی صفحه نمایش می بیند.	

جدول ۴-۲-۵- جدول مورد کاربرد گسترده شماره پنج

## UC08) امکان حذف آگهی

پیش‌نویس: کاربر (کارفرما) با موفقیت آگهی موردنظر خود را ثبت کرده‌است.	
کنشگر: کاربر (کارفرما) سامانه	سامانه: کاربایی هوشمند
۱- TUCBW کاربر (کارفرما) روی پیوند حذف آگهی کلیک می‌کند.	۲- مرورگر صفحه‌ی مربوط به حذف آگهی را نمایش می‌دهد.
۳- کاربر الف) روی پیوند حذف کلیک می‌کند. ب) روی پیوند بازگشت کلیک می‌کند تا به مرحله‌ی قبل برود.	۴- در صورت اینکه کاربر روی پیوند حذف کلیک کرده باشد، سامانه پیام «آیا از انتخاب خود مطمئن هستید؟» را نشان می‌دهد.
۵- کاربر الف) روی پیوند «بله» کلیک می‌کند. ب) روی پیوند «خیر» کلیک می‌کند تا از حذف آگهی جلوگیری کند.	۶- در صورت اینکه کاربر روی پیوند «بله» کلیک کرده باشد، در صورت موفقیت‌آمیز بودن عملیات حذف، سامانه پیام «آگهی با موفقیت حذف شد» را نشان می‌دهد و در غیر این صورت، سامانه پیام «آگهی حذف نشد» را نشان می‌دهد.
۷- TUCEW در صورت موفقیت‌آمیز بودن حذف، کاربر پیام «آگهی با موفقیت حذف شد» را مشاهده می‌کند و در غیر این صورت، پیام «آگهی حذف نشد» را مشاهده می‌کند.	

جدول ۴-۲-۶- جدول مورد کاربرد گسترده شماره شش

# فصل پنجم: مدل سازی تعامل شیء

در این فصل به بیان و مدل سازی چگونگی تعامل اشیا یا یکدیگر پرداخته و نمودارهای توالی مرتبط با آنها را رسم می کنیم.

مدل سازی تعامل شیء برای درک فرایندهای کسب و کاری موجود و رفتارهای تعاملی اشیا می باشد. مورد کاربردها یک نمونه از فرایند کسب و کار هستند، این نوع مدل سازی با پردازش پس زمینه ی یک مورد کاربرد مرتبط می شود. در نگرش شیء گرا، دنیا و هم چنین سیستم نرم افزاری ما متشکل از اشیای مرتبط باهم و در تعامل باهم تصور می شود. این اشیا روابطی مانند ارث بری، انجمنی، تجمیع باهم دارند. تعامل بین آنها نیز به وسیله ی درخواست سرویس، صدا زدن یا انجام اعمال روی دیگر اشیا است. این تعامل و ارتباط اشیا به منظور به انجام رساندن فرایندهای کسب و کار می باشد. ورودی های این فصل مورد کاربردهای مجرد و گسترده حاصل از فعالیت های فصل پیش است. خروجی های این فصل شامل سناریو، جدول سناریو و نمودار توالی برای شش مورد کاربرد می باشد.

\* گام های مدل سازی تعاملی شیء به شرح ذیل است:

## ۱. جمع آوری اطلاعات درباره ی فرایندهای کسب و کار موجود

در ابتدا تیم توسعه باید با فرایندهای کسب و کاری آشنایی کافی داشته باشد. در فعالیت های انجام شده در فاز یک پروژه تا حد مناسبی با فرایندهای کسب و کاری و نیازمندی های مشتری آشنا شدیم و اطلاعات لازم را به دست آوردیم. در این گام تیم به جمع آوری اطلاعات لازمی که در مراحل قبل یافته نشده بودند، پرداخت.

## ۲. تبیین سناریوهای برای گام های غیربدیهی از مورد کاربردهای گسترده

در این گام، تیم توسعه گام های غیربدیهی از مورد کاربردهای گسترده ی تهیه شده در فصل قبل را شناسایی کرد و برای آنها سناریو نوشت. سناریو، دنباله ای از جملات تعامل شیء است. در واقع خروجی این گام، فهرستی از توصیف سناریوها است.

## ۳. ساخت جداول سناریو

تیم توسعه سناریوهای به دست آمده از گام قبل را به صورت جدولی نمایش می دهد که به آن جدول سناریو می گویند.

#### ۴. استنتاج نمودار توالی از جداول سناریو

در این مرحله، خروجی‌های گام‌های قبل به نمودار توالی تبدیل شدند. همچنین گونه‌ها و واسطه‌های اشیای مربوطه نیز تعیین شدند.

#### ۵. مرور مدل‌های تعامل شیء

در نهایت مدل‌های تعامل شیء از نظر سازگاری، کامل بودن و درستی توسط تیم توسعه مورد بازبینی و بازنگری قرار داده شدند. در ادامه خروجی‌های لازم برای هر یک از پنج مورد کاربرد انتخاب شده آورده شده است.

### 1-5- سناریو و مدل تعامل شیء برای برای گام چهارم از UC01

#### \* ورود کاربر به عنوان کارفرما

۳- کاربر مشخصات را وارد می‌کند و بر روی دکمه ورود کلیک می‌کند.

۴-۱ صفحه ورود (Login page) مشخصات را به کنترلگر ورود ارسال می‌کند.

۴-۲ کنترلگر ورود (Login Controller) یک پیغام خالی msg ایجاد می‌کند.

۴-۳ کنترلگر ورود با استفاده از کد ملی کاربر را از مدیر پایگاه داده درخواست می‌کند.

۴-۴ مدیر پایگاه داده (DBMgr) کاربر u را به کنترلگر ورود برمی‌گرداند.

۴-۵ کنترلگر ورود کاربر را با رمز عبور واریسی می‌کند.

۴-۶ اگر مشخصات صحیح باشد.

۵- کاربر رمز موقت ارسال شده را وارد می‌کند.

۶-۱ کنترلگر ورود رمز موقت را از صفحه ورود دریافت می‌کند.

۶-۲ کنترلگر ورود رمز موقت را واریسی می‌کند.

۶-۳ اگر رمز موقت وارد شده صحیح باشد.

۶-۳-۱ کنترلگر ورود شیء «ورود با موفقیت انجام شد» را روی msg می‌نویسد.

۶-۴ در غیر اینصورت



۶-۴-۱- کنترلگر ورود شیء «رمز موقت نادرست است» را روی msg می نویسد.

۶-۵- در غیر اینصورت

۶-۵-۱- کنترلگر ورود شیء «نام کاربری یا رمز عبور نادرست است» را روی msg می نویسد.

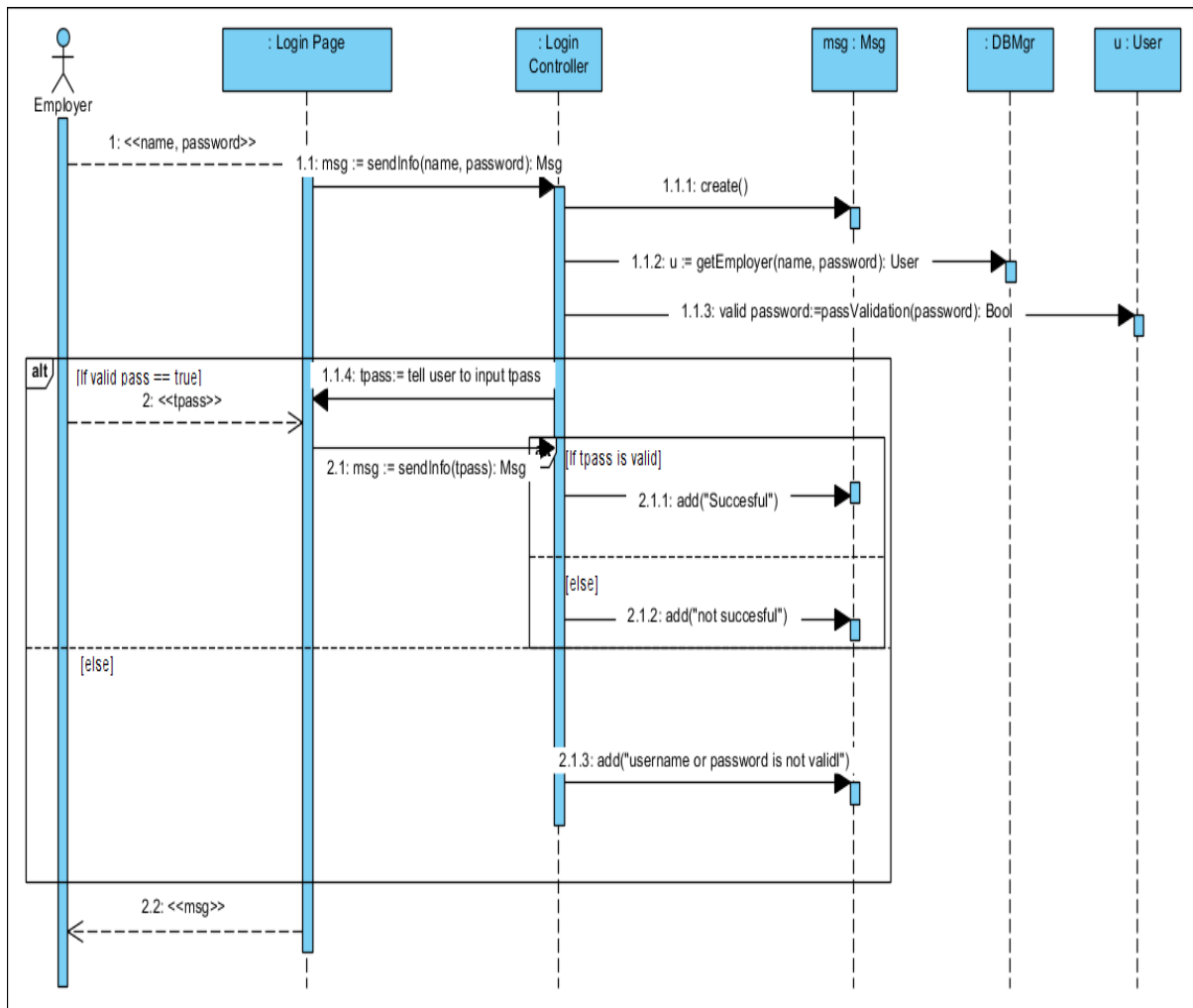
۶-۶- کنترلگر ورود پیام msg را به صفحه ورود برمی گرداند.

۶-۷- صفحه ورود پیام msg را به کاربر نمایش می دهد.

#	فاعل	کنش فاعل	دیگر داده ها / اشیاء	شیئی که کنش روی آن انجام می شود.
۳	کاربر	وارد می کند	مشخصات	صفحه ورود (login page)
۴-۱	صفحه ورود	ارسال می کند	مشخصات	کنترلگر ورود (login controller)
۴-۲	کنترلگر ورود	ایجاد می کند		پیغام (msg)
۴-۳	کنترلگر ورود	درخواست می کند	کد ملی	مدیر پایگاه داده
۴-۴	مدیر پایگاه داده	بر می گرداند	u کاربر	کنترلگر ورود
۴-۵	کنترلگر ورود	واریسی می کند	رمز عبور	کاربر
۴-۶	اگر مشخصات صحیح باشد			
۵	کاربر	وارد می کند	رمز موقت	صفحه ورود
۶-۱	کنترلگر ورود	دریافت می کند	رمز موقت	صفحه ورود
۶-۲	کنترلگر ورود	واریسی می کند	رمز موقت	کاربر
۶-۳	اگر رمز موقت وارد شده صحیح باشد			

پیغام	«ورود با موفقیت انجام شد»	اضافه می کند	کنترلگر ورود	۱-۳-۶
در غیر اینصورت				۴-۶
پیغام	«رمز موقت نادرست است»	اضافه می کند	کنترلگر ورود	۱-۴-۶
در غیر اینصورت				۵-۶
پیغام	«نام کاربری یا رمز عبور نادرست است»	اضافه می کند	کنترلگر ورود	۱-۵-۶
صفحه ورود	پیغام	بر می گرداند	کنترلگر ورود	۶-۶
کاربر	پیغام	نمایش می دهد	صفحه ورود	۷-۶

جدول ۱-۵ جدول سناریو مورد کاربرد یک



## \* الگوهای به کار رفته در نمودار UC01

### • الگوی کنترلرگر :

الگوی کنترلرگر در زمان نوشتن سناریوی مورد کاربرد اعمال شده است تا واسط گرافیکی را از اشیای کسب و کار مجزا نماید. مسئولیت کنترلرگر این است که بر اشیای کسب و کار نظارت کرده و یا آنها را هماهنگ کند تا گام‌های یک مورد کاربرد را انجام دهند. به این منظور، در نمودارهای توالی کنترلرگر مورد کاربرد، مورد استفاده قرار می‌گیرد. بر همین اساس در نمودار توالی ذکر شده از Login controller استفاده شده است و مسئولیت نمایش اطلاعات بر عهده Login page و همچنین مسئولیت پردازش درخواست‌های ایجاد شده بر عهده Login controller قرار گرفته است. در ادامه توضیح داده شده است که چگونه با اعمال این الگو، می‌توان طراحی نرم‌افزار را با استفاده از اصول طراحی، بهبود بخشید.

– طراحی برای تغییر: سیستم باید به گونه‌ای باشد که بتواند با تغییرات سازگار شود و یا به راحتی تغییر داده شود. با اعمال این الگو در نمودارهای توالی، واسط و رفتار تعاملی کنترلرگر تغییر نمی‌کند به همین دلیل تغییرات رخ داده در منطق کسب و کار تاثیر چندانی بر روی واسط گرافیکی نخواهد داشت. با تغییر در

نیازمندی‌ها تنها نیاز است که واسطه‌های جدید را اضافه کرده و اجازه دهیم هر کدام از آنها توابع متناسب از کنترلگر مربوطه را فراخوانی کنند.

— جداسازی دغدغه‌ها: جداسازی دغدغه‌ها توسط طراحی به خوبی پشتیبانی می‌شود زیرا هر قسمت تنها بر روی یک جنبه از نیازمندی‌ها تمرکز دارد. به عنوان مثال در نمودار توالی ذکر شده، `Login page` تنها مسئول نمایش اطلاعات به کاربر است درحالی که `Login controller` مسئول پردازش درخواست‌های ایجاد شده می‌باشد.

— چسبندگی زیاد: همانطور که در قسمت جداسازی دغدغه‌ها توضیح داده شد، با اعمال این الگو، واسط گرافیکی تنها مسئول نمایش اطلاعات به کاربر و کنترلگر مسئول پردازش درخواست‌های ایجاد شده می‌باشد. بنابراین طراحی به گونه‌ای است که توابع هر پیمانه، بیشترین درجه ارتباط با مسئولیت اصلی پیمانه را دارند و در نتیجه هر دو، چسبندگی تابعی دارند.

— طراحی اشیای نادان: با اعمال این الگو، واسط گرافیکی تنها از نمایش اطلاعات به کاربر و کنترلگر از پردازش مورد کاربرد مربوطه اطلاع دارد. بنابراین اصل ساده و احمقانه فرض کن و همچنین طراحی اشیای نادان به خوبی رعایت می‌شود.

— پنهان‌سازی اطلاعات: با در نظر گرفتن کلاس `DBMgr` در نمودارهای توالی، پایگاه داده از کنترلگر مخفی شده است. این کار، اصل پنهان‌سازی اطلاعات را رعایت کرده و از ایجاد کنترلگرهای متورم جلوگیری می‌کند.

— جفت‌شدگی کم: با در نظر گرفتن کلاس `DBMgr`، پایگاه داده از کنترلگر مجزا شده و در نتیجه تاثیر تغییرات، کم خواهد بود و اصل جفت‌شدگی کم رعایت می‌شود.

### • الگوی خالق:

الگوی خالق مشخص می‌کند که مسئولیت ایجاد یک شی باید بر عهده چه کسی گذاشته شود. از آن جایی که در این نمودار توالی، سازنده کلاس `User` به آدرس مرکز و همچنین کد معرف نیاز دارد و این اطلاعات در دست خود `Login controller` می‌باشد؛ بنابراین با اعمال این الگو، این کنترلگر برای ایجاد شی `User` انتخاب شده است. همچنین خود `Login controller` توانایی ایجاد یک شی از کلاس `Msg` را دارد؛ بنابراین با اعمال این الگو، این کنترلگر باید ایجادکننده شی `msg` باشد.

### • الگوی خبره:

هنگامی که یک کلاس اطلاعات لازم را برای انجام یا تایید عملی دارد. مانند اعتبار سنجی یک پسورد در فلش ۱.۱.۳ که شی‌ای از کلاس `User` این کار را انجام می‌دهد.

## 5-2- سناریو و مدل تعامل شیء برای برای گام سوم از UC23

### \* تغییر نام کاربری یا گذرواژه

۱۲- کاربر رمز عبور جدید خود را وارد می‌کند.

۱۳-۱- کنترلگر تغییر رمز عبور یک پیغام خالی msg ایجاد می‌کند.

۱۳-۲- کنترلگر تغییر رمز عبور با اتصال به پایگاه داده، شیء u مربوط به کاربر را از پایگاه داده دریافت می‌کند.

۱۳-۳- رمز وارد شده توسط کنترلگر تغییر رمز عبور واری می‌شود.

۱۳-۳-۱- اگر نتیجه مثبت باشد

۱۳-۳-۱-۱- کنترلگر تغییر رمز عبور، رمز عبور شیء u را به روزرسانی می‌کند.

۱۳-۳-۱-۲- کنترلگر تغییر رمز عبور، شیء u را در پایگاه داده به روزرسانی می‌کند.

۱۳-۳-۱-۳- کنترلگر تغییر رمز عبور پیام «گذرواژه با موفقیت تغییر یافت» را به msg اضافه می‌کند.

۱۳-۳-۱-۴- کنترلگر تغییر رمز عبور پیام msg را به واسط گرافیکی برمی‌گرداند.

۱۳-۳-۱-۵- واسط گرافیکی پیام msg را به کاربر نشان می‌دهد.

۱۳-۳-۲- اگر نتیجه منفی باشد

۱۳-۳-۲-۱- کنترلگر تغییر رمز عبور پیام «لطفاً از گذرواژه قوی‌تر استفاده کنید» را به msg اضافه می‌کند.

۱۳-۳-۲-۲- کنترلگر تغییر رمز عبور پیام msg را به واسط گرافیکی برمی‌گرداند.

۱۳-۳-۲-۳- واسط گرافیکی پیام msg را به کاربر نشان می‌دهد.

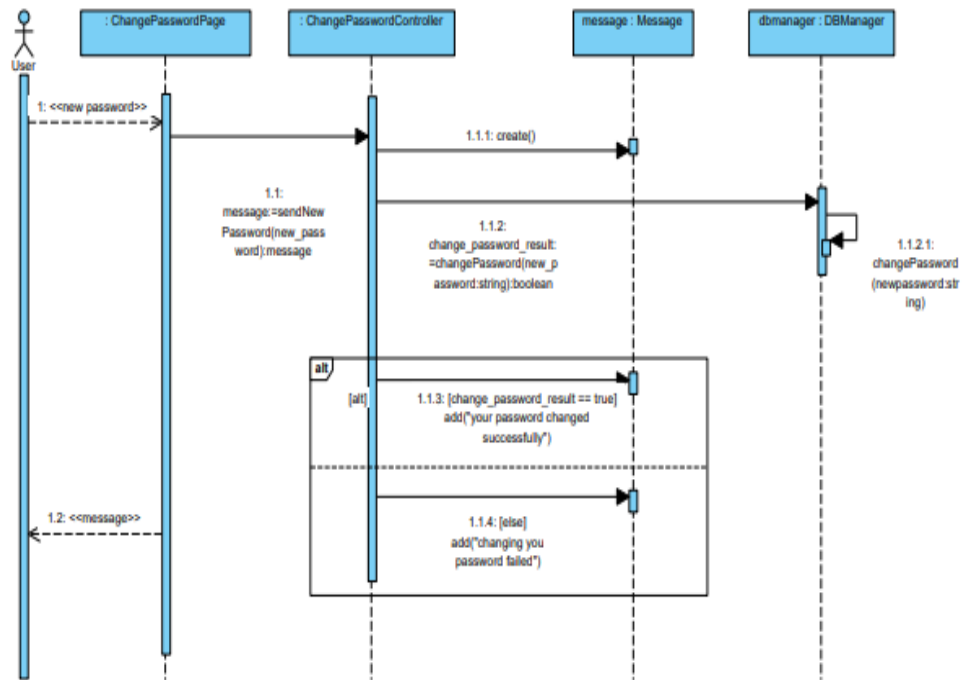
#	فاعل	کنش فاعل	دیگر داده ها / اشیا	شیئی که کنش روی آن انجام می‌شود.
۱۲	کاربر	وارد می‌کند	رمز عبور جدید	Change Password Page

پیغام(msg)		ایجاد می کند	کنترلگر تغییر رمز عبور	۱-۱۳
پایگاه داده	لاشی	دریافت می کند	کنترلگر تغییر رمز عبور	۲-۱۳
کنترلگر تغییر رمز عبور	رمز عبور جدید	وارسی می کند	کنترلگر تغییر رمز عبور	۳-۱۳
	اگر نتیجه مثبت باشد			۱-۳-۱۳
لاشی کاربر	رمز عبور	به روزرسانی می کند	کنترلگر تغییر رمز عبور	۱-۱-۳-۱۳
پایگاه داده	لاشی	به روزرسانی می کند	کنترلگر تغییر رمز عبور	۲-۱-۳-۱۳
Change Password Page	گذرواژه با موفقیت تغییر یافت	اضافه می کند	کنترلگر تغییر رمز عبور	۳-۱-۳-۱۳
کاربر	(msg)پیغام	بر می گرداند	کنترلگر تغییر رمز عبور	۴-۱-۳-۱۳
	(msg)پیغام	نشان می دهد	Change password page	۵-۱-۳-۱۳
اگر نتیجه منفی باشد				۲-۳-۱۳
(msg)پیغام	لطفا از گذرواژه قوی تر استفاده کنید	اضافه می کند	کنترلگر تغییر رمز عبور	۱-۲-۳-۱۳
Change Password Page	(msg)پیغام	بر می گرداند	کنترلگر تغییر رمز عبور	۲-۲-۳-۱۳

کاربر	نشان می‌دهد	نشان می‌دهد	Change password page	۳-۲-۳-۱۳
-------	-------------	-------------	----------------------	----------

جدول ۵-۲ جدول سناریو مورد کاربرد بیست و سه

۱



## \* الگوهای به کار رفته در نمودار UC23

### • الگوی کنترلر :

الگوی کنترلر در زمان نوشتن سناریوی مورد کاربرد اعمال شده است تا واسطه گرافیکی را از اشیای کسب و کار مجزا نماید. مسئولیت کنترلر این است که بر اشیای کسب و کار نظارت کرده و یا آنها را هماهنگ کند تا گام‌های یک مورد کاربرد را انجام دهند. به این منظور، در نمودارهای توالی کنترلر مورد کاربرد، مورد استفاده قرار می‌گیرد. بر همین اساس در نمودار توالی ذکر شده از Change PasswordController استفاده شده است و مسئولیت نمایش اطلاعات بر عهده Change PasswordPage و همچنین مسئولیت پردازش درخواست‌های ایجاد شده بر عهده Change PasswordController قرار گرفته است. در ادامه

توضیح داده شده است که چگونه با اعمال این الگو، می‌توان طراحی نرم‌افزار را با استفاده از اصول طراحی، بهبود بخشید.

— طراحی برای تغییر: سیستم باید به گونه‌ای باشد که بتواند با تغییرات سازگار شود و یا به راحتی تغییر داده شود. با اعمال این الگو در نمودارهای توالی، واسط و رفتار تعاملی کنترلر تغییر نمی‌کند به همین دلیل تغییرات رخ داده در منطق کسب‌وکار تاثیر چندانی بر روی واسط گرافیکی نخواهد داشت. با تغییر در نیازمندی‌ها تنها نیاز است که واسط‌های جدید را اضافه کرده و اجازه دهیم هر کدام از آنها توابع متناسب از کنترلر مربوطه را فراخوانی کنند.

— جداسازی دغدغه‌ها: جداسازی دغدغه‌ها توسط طراحی به خوبی پشتیبانی می‌شود زیرا هر قسمت تنها بر روی یک جنبه از نیازمندی‌ها تمرکز دارد. به عنوان مثال در نمودار توالی ذکر شده، `Change PasswordPage` تنها مسئول نمایش اطلاعات به کاربر است درحالی که `Change PasswordController` مسئول پردازش درخواست‌های ایجاد شده می‌باشد.

— چسبندگی زیاد: همانطور که در قسمت جداسازی دغدغه‌ها توضیح داده شد، با اعمال این الگو، واسط گرافیکی تنها مسئول نمایش اطلاعات به کاربر و کنترلر مسئول پردازش درخواست‌های ایجاد شده می‌باشد. بنابراین طراحی به گونه‌ای است که توابع هر پیمانه، بیشترین درجه ارتباط با مسئولیت اصلی پیمانه را دارند و در نتیجه هر دو، چسبندگی تابعی دارند.

— طراحی اشیای نادان: با اعمال این الگو، واسط گرافیکی تنها از نمایش اطلاعات به کاربر و کنترلر از پردازش مورد کاربرد مربوطه اطلاع دارد. بنابراین اصل ساده و احمقانه فرض کن و همچنین طراحی اشیای نادان به خوبی رعایت می‌شود.

— پنهان‌سازی اطلاعات: با در نظر گرفتن کلاس `DBManager` در نمودارهای توالی، پایگاه داده از کنترلر مخفی شده است. این کار، اصل پنهان‌سازی اطلاعات را رعایت کرده و از ایجاد کنترلرهای متورم جلوگیری می‌کند.

— جفت‌شدگی کم: با در نظر گرفتن کلاس `DBManager`، پایگاه داده از کنترلر مجزا شده و در نتیجه تاثیر تغییرات، کم خواهد بود و اصل جفت‌شدگی کم رعایت می‌شود.

### • الگوی خالق:

الگوی خالق مشخص می‌کند که مسئولیت ایجاد یک شی باید بر عهده چه کسی گذاشته شود. از آن جایی که در این نمودار توالی، سازنده کلاس `Change PasswordController` به آدرس مرکز و همچنین کد معرف نیاز دارد و این اطلاعات در دست خود `Change PasswordController` می‌باشد؛ بنابراین با اعمال این الگو، این کنترلر برای ایجاد شی `Change PasswordController` انتخاب شده است. همچنین



خود `Change PasswordController` توانایی ایجاد یک شیء از کلاس `Message` را دارد؛ بنابراین با اعمال این الگو، این کنترلر باید ایجادکننده شیء `message` باشد.

### • الگوی خبره:

هنگامی که یک کلاس اطلاعات لازم را برای انجام یا تایید عملی دارد. مانند فلش ۱.۱.۲ که شیء ای از کلاس `DBManager` این کار را انجام می‌دهد.

## 3-5- سناریو و مدل تعامل شیء برای برای گام سوم از UC18

### \* جست‌وجو بین محل کارهای متفاوت

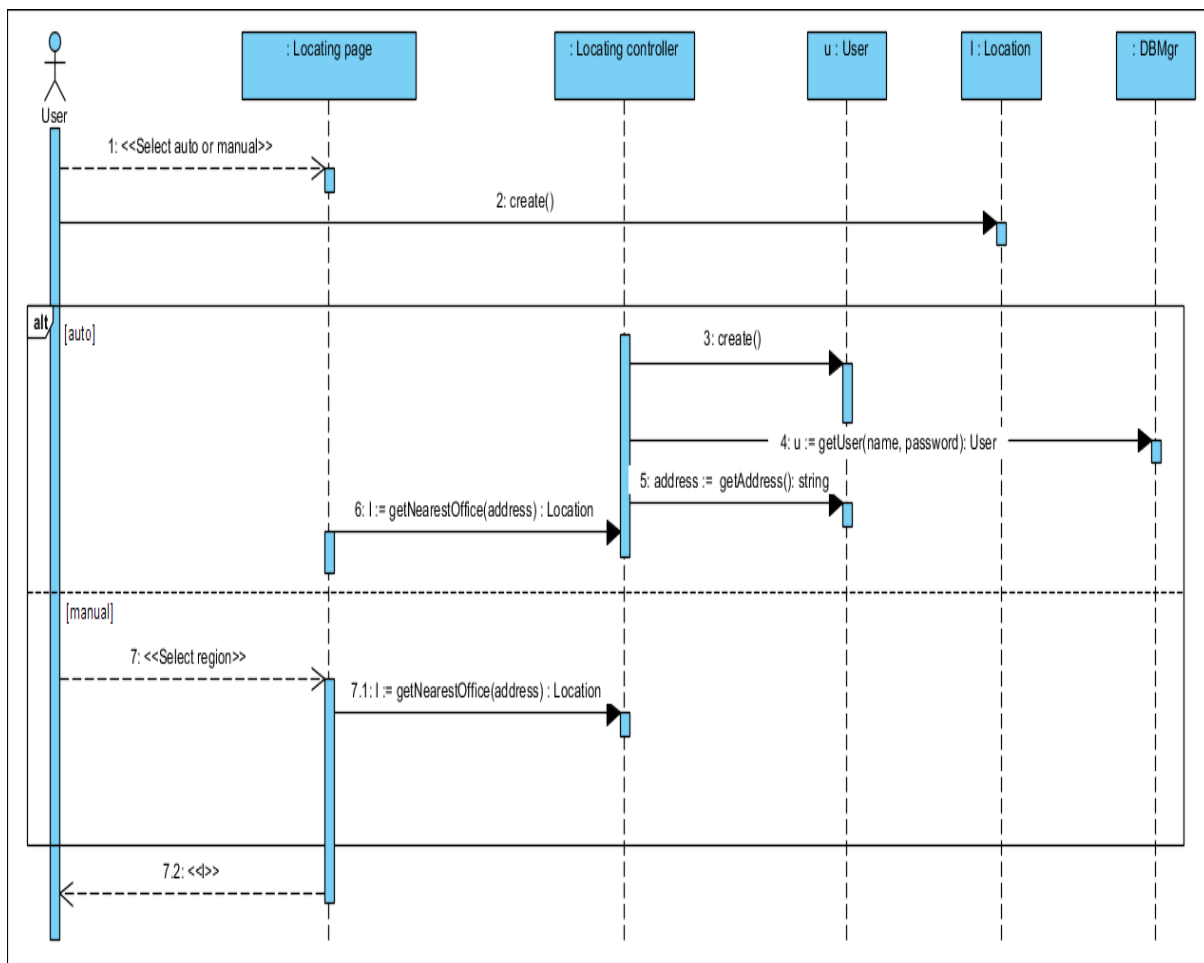
۱. سامانه صفحه‌ی *Location Page* را به کاربر نشان می‌دهد.
  - ۱.۱. کاربر روش خودکار را انتخاب می‌کند.
    - ۱.۱.۱. صفحه *Location Page* اطلاعات را به کنترل‌گر می‌فرستد.
    - ۱.۱.۲. کنترل‌گر یک شیء خالی با نام *l* از کلاس *Locating* ایجاد می‌کند.
    - ۱.۱.۳. کنترل‌گر یک شیء خالی با نام *u* از کلاس *User* ایجاد می‌کند.
    - ۱.۱.۴. کنترل‌گر با استفاده از کد ملی کاربر، اطلاعات کاربر را از مدیر پایگاه‌داده دریافت می‌کند.
      - ۱.۱.۴.۱. کنترل‌گر اطلاعات کاربر را در شیء *u* می‌نویسد.
      - ۱.۱.۴.۲. کنترل‌گر آدرس نزدیکترین محل کارها را نسبت به آدرس دریافتی از *u*، در شیء *l* می‌نویسد.
      - ۱.۱.۵. کنترل‌گر شیء *l* را به صفحه *Location Page* می‌فرستد.
      - ۱.۱.۶. *Location Page* آدرس‌های ذخیره شده در *l* را به کاربر نشان می‌دهد.
  - ۱.۲. کاربر گزینه دستی را انتخاب می‌کند.
    - ۱.۲.۱. صفحه *Location Page* اطلاعات وارد شده را به کنترل‌گر می‌فرستد.
    - ۱.۲.۲. کنترل‌گر یک شیء خالی با نام *l* از جنس کلاس *Location* ایجاد می‌کند.
    - ۱.۲.۳. کنترل‌گر آدرس نزدیکترین محل کارها را نسبت به آدرسی که دریافت کرده از `DBMgr` می‌گیرد و در شیء *l* قرار می‌دهد.

۱.۲.۴. کنترل گر شی ا را به صفحه *Location Page* می فرستد.

۲. صفحه *Location Page*, شی ا را نشان می دهد.

#	فاعل	کنش فاعل	دیگر داده ها / اشیا	شیء که کنش روی آن انجام میشود.
۱	سامانه	نشان می دهد	<i>Location Page</i>	<i>Location Page</i>
۱.۱	کاربر	انتخاب می کند	روش خودکار	<i>Location Page</i>
۱.۱.۱	<i>Location Page</i>	می فرستد	اطلاعات	کنترل گر
۱.۱.۲	کنترل گر	ایجاد می کند	شیء <i>I</i>	کلاس <i>Location</i>
۱.۱.۳	کنترل گر	ایجاد می کند	شیء <i>u</i>	کلاس <i>User</i>
۱.۱.۴	کنترل گر	دریافت می کند	اطلاعات	DBMgr
۱.۱.۴.۱	کنترل گر	می نویسد	اطلاعات	شیء <i>u</i>
۱.۱.۴.۲	کنترل گر	می نویسد	اطلاعاتی از شیء <i>u</i>	شیء <i>a</i>
۱.۱.۵	کنترل گر	می فرستد	شیء <i>a</i>	<i>Location Page</i>
۱.۱.۶	<i>Location Page</i>	نشان می دهد	شیء <i>a</i>	کاربر
۱.۲	کاربر	انتخاب می کند	روش دستی	<i>Location Page</i>
۱.۲.۱	<i>Location Page</i>	می فرستد	اطلاعات	کنترل گر
۱.۲.۲	کنترل گر	ایجاد می کند	شیء <i>a</i>	کلاس <i>Location</i>
۱.۲.۳	کنترل گر	می نویسد	اطلاعات	DBMgr
۱.۲.۴	کنترل گر	می فرستد	شیء <i>a</i>	<i>Location Page</i>
۲	<i>Location Page</i>	نشان می دهد	شیء <i>a</i>	کاربر

جدول ۳-۵ جدول سناریو مورد کاربرد هجده



## \* الگوهای به کار رفته در نمودار UC18

### • الگوی کنترلگر :

الگوی کنترلگر در زمان نوشتن سناریوی مورد کاربرد اعمال شده است تا واسط گرافیکی را از اشیای کسب و کار مجزا نماید. مسئولیت کنترلگر این است که بر اشیای کسب و کار نظارت کرده و یا آنها را هماهنگ کند تا گام‌های یک مورد کاربرد را انجام دهند. به این منظور، در نمودارهای توالی کنترلگر مورد کاربرد، مورد استفاده قرار می‌گیرد. بر همین اساس در نمودار توالی ذکر شده از **Locating controller** استفاده شده است و مسئولیت نمایش اطلاعات بر عهده **Locating page** و همچنین مسئولیت پردازش درخواست‌های ایجاد شده بر عهده **Locating controller** قرار گرفته است. در ادامه توضیح داده شده است که چگونه با اعمال این الگو، می‌توان طراحی نرم‌افزار را با استفاده از اصول طراحی، بهبود بخشید.

— طراحی برای تغییر: سیستم باید به گونه‌ای باشد که بتواند با تغییرات سازگار شود و یا به راحتی تغییر داده شود. با اعمال این الگو در نمودارهای توالی، واسط و رفتار تعاملی کنترلگر تغییر نمی‌کند به همین

دلیل تغییرات رخ داده در منطق کسب و کار تاثیر چندانی بر روی واسط گرافیکی نخواهد داشت. با تغییر در نیازمندی‌ها تنها نیاز است که واسط‌های جدید را اضافه کرده و اجازه دهیم هر کدام از آنها توابع متناسب از کنترلر مربوطه را فراخوانی کنند.

– جداسازی دغدغه‌ها: جداسازی دغدغه‌ها توسط طراحی به خوبی پشتیبانی می‌شود زیرا هر قسمت تنها بر روی یک جنبه از نیازمندی‌ها تمرکز دارد. به عنوان مثال در نمودار توالی ذکر شده، Locating page تنها مسئول نمایش اطلاعات به کاربر است در حالی که Locating controller مسئول پردازش درخواست‌های ایجاد شده می‌باشد.

– چسبندگی زیاد: همانطور که در قسمت جداسازی دغدغه‌ها توضیح داده شد، با اعمال این الگو، واسط گرافیکی تنها مسئول نمایش اطلاعات به کاربر و کنترلر مسئول پردازش درخواست‌های ایجاد شده می‌باشد. بنابراین طراحی به گونه‌ای است که توابع هر پیمانه، بیشترین درجه ارتباط با مسئولیت اصلی پیمانه را دارند و در نتیجه هر دو، چسبندگی تابعی دارند.

– طراحی اشیای نادان: با اعمال این الگو، واسط گرافیکی تنها از نمایش اطلاعات به کاربر و کنترلر از پردازش مورد کاربرد مربوطه اطلاع دارد. بنابراین اصل ساده و احمقانه فرض کن و همچنین طراحی اشیای نادان به خوبی رعایت می‌شود.

– پنهان‌سازی اطلاعات: با در نظر گرفتن کلاس DBMgr در نمودارهای توالی، پایگاه داده از کنترلر مخفی شده است. این کار، اصل پنهان‌سازی اطلاعات را رعایت کرده و از ایجاد کنترلرهای متورم جلوگیری می‌کند.

– جفت‌شدگی کم: با در نظر گرفتن کلاس DBMgr، پایگاه داده از کنترلر مجزا شده و در نتیجه تاثیر تغییرات، کم خواهد بود و اصل جفت‌شدگی کم رعایت می‌شود.

### • الگوی خالق:

الگوی خالق مشخص می‌کند که مسئولیت ایجاد یک شی باید بر عهده چه کسی گذاشته شود. از آن جایی که در این نمودار توالی، سازنده کلاس User به آدرس مرکز و همچنین کد معرف نیاز دارد و این اطلاعات در دست Locating controller می‌باشد؛ بنابراین با اعمال این الگو، این کنترلر برای ایجاد شی User انتخاب شده است. همچنین Locating controller توانایی ایجاد یک شی از کلاس Msg را دارد؛ بنابراین با اعمال این الگو، این کنترلر باید ایجادکننده شی msg باشد.

## ۴-۵- سناریو و مدل تعامل شی برای برای گام سوم از UC24

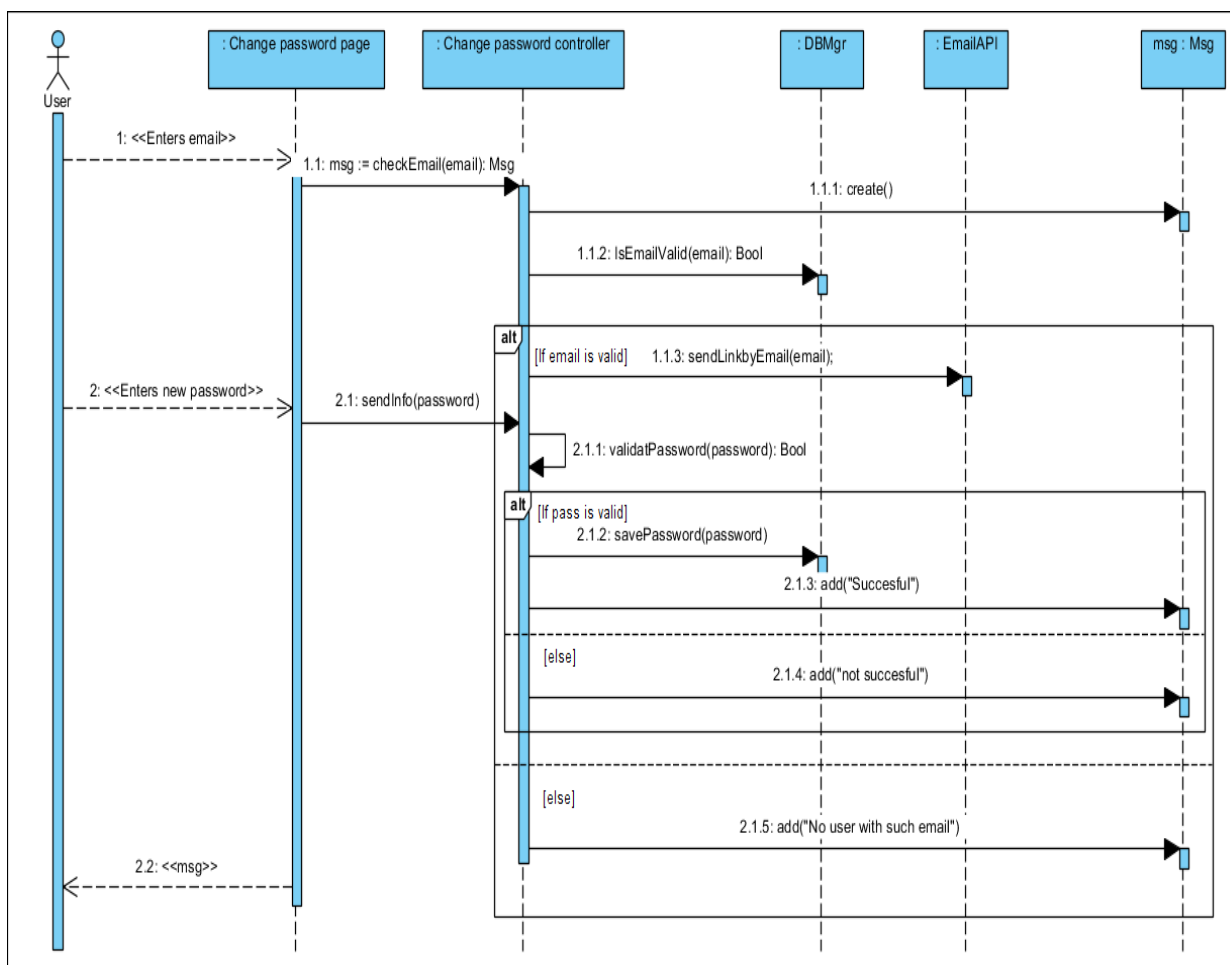
## \* بازیابی گذرواژه

۱. سامانه صفحه `change password` را به کاربر نشان می‌دهد.
۲. صفحه تغییر رمز ( `change password` ) ایمیل را از کاربر می‌گیرد و به کنترل‌گر ارسال می‌کند.
۳. کنترل‌گر یک پیام `msg` ایجاد می‌کند.
۴. کنترل‌گر با استفاده از ایمیل، کاربر را از مدیر پایگاه داده درخواست می‌کند.
  - ۴.۱. اگر ایمیل وجود داشت
    - ۴.۱.۱. کنترل‌گر لینک تغییر گذرواژه را برای ایمیل کاربر می‌فرستد.
    - ۴.۱.۱.۱. کاربر در پیوند ارسال‌شده رمز جدید را وارد می‌کند.
    - ۴.۱.۲. صفحه تغییر رمز ( `Change password page` )، رمز جدید را به کنترل‌گر تغییر رمز ارسال می‌کند.
    - ۴.۱.۳. کنترل‌گر رمز جدید را در `DBMgr` ذخیره می‌کند.
    - ۴.۱.۴. اگر رمز با موفقیت تغییر کرد،
    - ۴.۱.۵. کنترل‌گر تغییر رمز پیام «گذرواژه شما با موفقیت بازیابی شد» را به `msg` اضافه می‌کند.
  - ۴.۲. در غیر اینصورت،
    - ۴.۲.۱. کنترل‌گر تغییر رمز پیام « لطفاً از گذرواژه قوی تر استفاده کنید » را به `msg` اضافه می‌کند.
    - ۴.۲.۲. کنترل‌گر تغییر رمز `msg` را به صفحه تغییر رمز برمی‌گرداند.
    - ۴.۲.۳. صفحه تغییر رمز پیام `msg` را به کاربر نمایش می‌دهد.
- ۴.۳. اگر ایمیل وجود نداشت
  - ۴.۳.۱. کنترل‌گر تغییر رمز یک پیام خالی `msg2` ایجاد می‌کند.
  - ۴.۳.۲. کنترل‌گر پیام « کاربری با چنین ایمیلی یافت نشد » را به `msg2` اضافه می‌کند.
  - ۴.۳.۳. کنترل‌گر `msg` را به صفحه تغییر رمز برمی‌گرداند.
  - ۴.۳.۴. صفحه تغییر رمز پیام `msg` را به کاربر نمایش می‌دهد.

#	فاعل	کنش فاعل	دیگر داده ها / اشیا	شیئی که کنش روی آن انجام می‌شود
۱	سامانه	نمایش می‌دهد	<code>Change password</code>	<code>Change password</code>
۲	صفحه تغییر رمز	ارسال می‌کند	ایمیل	کنترل‌گر
۳	کنترل‌گر	ایجاد می‌کند		پیغام <code>msg</code>
۴	کنترل‌گر	درخواست می‌کند	ایمیل	مدیر پایگاه داده

اگر ایمیل وجود داشت				۴.۱
ایمیل کاربر	پیوند تغییر گذرواژه	ارسال می کند	کنترلگر	۴.۱.۱
صفحه تغییر رمز (Change password page)	رمز جدید	وارد و تایید می کند	کاربر	۴.۱.۱.۱
کنترلگر تغییر رمز (changr password controller)	رمز جدید	ارسال می کند	صفحه تغییر رمز	۴.۱.۲
مدیر پایگاه داده (DBMgr)	رمز جدید	ذخیره می کند	کنترلگر تغییر رمز عبور	۴.۱.۳
اگر رمز با موفقیت تغییر کرد				۴.۱.۴
پیغام msg	رمز شما با موفقیت بازیابی شد	اضافه می کند	کنترلگر تغییر رمز عبور	۴.۱.۵
در غیر اینصورت				۴.۲
پیغام msg	عملیات بازیابی رمز شما با شکست مواجه شد	اضافه می کند	کنترلگر تغییر رمز عبور	۴.۲.۱
صفحه تغییر رمز	پیغام msg	بر می گرداند	کنترلگر تغییر رمز عبور	۴.۲.۲
کاربر	پیغام msg	نمایش می دهد	صفحه تغییر رمز	۴.۲.۳
اگر ایمیل وجود نداشت				۴.۳
پیغام msg2		ایجاد می کند	کنترلگر	۴.۳.۱
پیغام msg2	کاربری با چنین ایمیلی یافت نشد	اضافه می کند	کنترلگر	۴.۳.۲
صفحه تغییر رمز	پیغام msg2	بر می گرداند	کنترلگر	۴.۳.۳
کاربر	پیغام msg2	نمایش می دهد	صفحه تغییر رمز	۴.۳.۴

جدول ۴-۵ جدول سناریو مورد کاربرد بیست و چهار



## \* الگوهای به کار رفته در نمودار UC24

### • الگوی کنترلر :

الگوی کنترلر در زمان نوشتن سناریوی مورد کاربرد اعمال شده است تا واسط گرافیکی را از اشیای کسب و کار مجزا نماید. مسئولیت کنترلر این است که بر اشیای کسب و کار نظارت کرده و یا آنها را هماهنگ کند تا گام‌های یک مورد کاربرد را انجام دهند. به این منظور، در نمودارهای توالی کنترلر مورد کاربرد، مورد استفاده قرار می‌گیرد. بر همین اساس در نمودار توالی ذکر شده از Change password controller استفاده شده است و مسئولیت نمایش اطلاعات بر عهده Change password page و همچنین مسئولیت پردازش درخواست‌های ایجاد شده بر عهده Change password controller قرار گرفته است. در ادامه توضیح داده شده است که چگونه با اعمال این الگو، می‌توان طراحی نرم‌افزار را با استفاده از اصول طراحی، بهبود بخشید.

— طراحی برای تغییر: سیستم باید به گونه‌ای باشد که بتواند با تغییرات سازگار شود و یا به راحتی تغییر داده شود. با اعمال این الگو در نمودارهای توالی، واسط و رفتار تعاملی کنترلر تغییر نمی‌کند به همین دلیل تغییرات رخ داده در منطق کسب‌وکار تاثیر چندانی بر روی واسط گرافیکی نخواهد داشت. با تغییر در نیازمندی‌ها تنها نیاز است که واسط‌های جدید را اضافه کرده و اجازه دهیم هر کدام از آنها توابع متناسب از کنترلر مربوطه را فراخوانی کنند.

— جداسازی دغدغه‌ها: جداسازی دغدغه‌ها توسط طراحی به خوبی پشتیبانی می‌شود زیرا هر قسمت تنها بر روی یک جنبه از نیازمندی‌ها تمرکز دارد. به عنوان مثال در نمودار توالی ذکر شده، `Change password` page تنها مسئول نمایش اطلاعات به کاربر است در حالی که `Change password controller` مسئول پردازش درخواست‌های ایجاد شده می‌باشد.

— چسبندگی زیاد: همانطور که در قسمت جداسازی دغدغه‌ها توضیح داده شد، با اعمال این الگو، واسط گرافیکی تنها مسئول نمایش اطلاعات به کاربر و کنترلر مسئول پردازش درخواست‌های ایجاد شده می‌باشد. بنابراین طراحی به گونه‌ای است که توابع هر پیمانه، بیشترین درجه ارتباط با مسئولیت اصلی پیمانه را دارند و در نتیجه هر دو، چسبندگی تابعی دارند.

— طراحی اشیای نادان: با اعمال این الگو، واسط گرافیکی تنها از نمایش اطلاعات به کاربر و کنترلر از پردازش مورد کاربرد مربوطه اطلاع دارد. بنابراین اصل ساده و احمقانه فرض کن و همچنین طراحی اشیای نادان به خوبی رعایت می‌شود.

— پنهان‌سازی اطلاعات: با در نظر گرفتن کلاس `DBMgr` در نمودارهای توالی، پایگاه داده از کنترلر مخفی شده است. این کار، اصل پنهان‌سازی اطلاعات را رعایت کرده و از ایجاد کنترلرهای متورم جلوگیری می‌کند.

— جفت‌شدگی کم: با در نظر گرفتن کلاس `DBMgr`، پایگاه داده از کنترلر مجزا شده و در نتیجه تاثیر تغییرات، کم خواهد بود و اصل جفت‌شدگی کم رعایت می‌شود.

### • الگوی خالق:

الگوی خالق مشخص می‌کند که مسئولیت ایجاد یک شی باید بر عهده چه کسی گذاشته شود. از آن جایی که در این نمودار توالی، سازنده کلاس `Change password controller` به آدرس مرکز و همچنین کد معرف نیاز دارد و این اطلاعات در دست خود `Change password controller` می‌باشد؛ بنابراین با اعمال این الگو، این کنترلر برای ایجاد شی از خود `Change password controller` انتخاب شده است. همچنین خود `Change password controller` توانایی ایجاد یک شی از کلاس `Msg` را دارد؛ بنابراین با اعمال این الگو، این کنترلر باید ایجادکننده شی `msg` باشد.



## • الگوی خبره:

هنگامی که یک کلاس اطلاعات لازم را برای انجام یا تایید عملی دارد. Change password controller پسورد ها را توسط فلشی که در داخل آن قرار دارد، چک می کند.

## 5-5- سناریو و مدل تعامل شیء برای برای گام چهارم از UC05

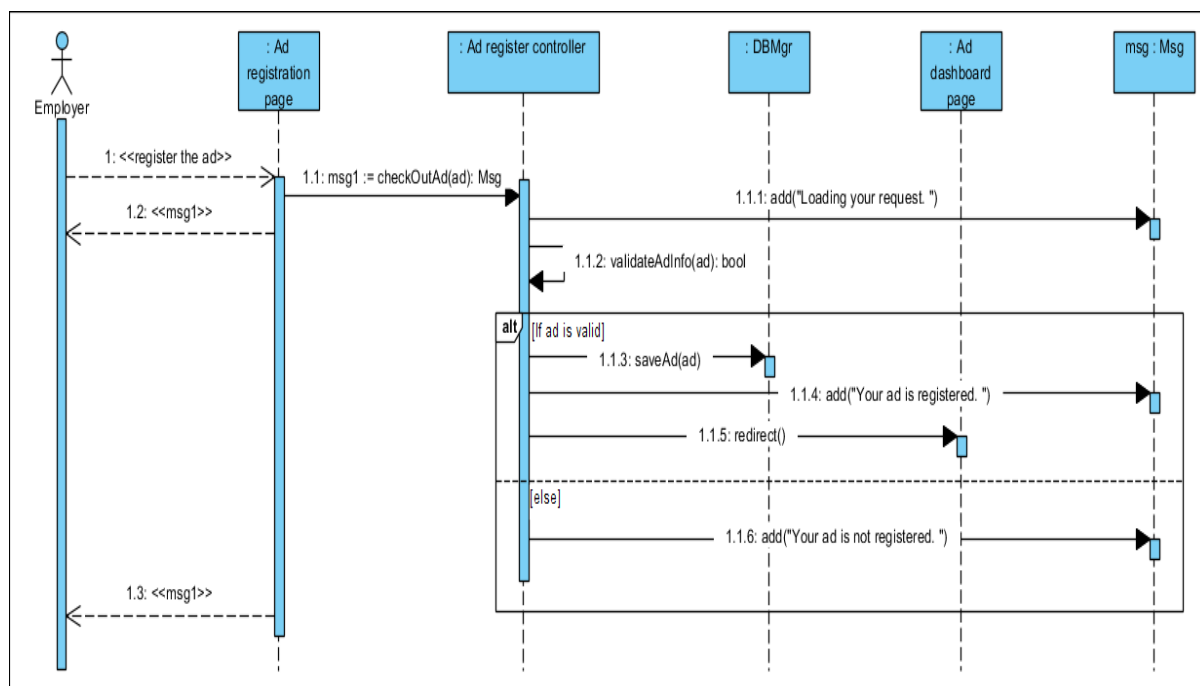
### \* درخواست ثبت آگهی

۱. کاربر روی پیوند «ذخیره» کلیک می کند.
۲. صفحه درخواست ثبت آگهی، اطلاعات را به کنترل گر ارسال می کند.
۳. کنترل گر یک پیغام خالی *msg ۱* ایجاد می کند.
۴. کنترل گر روی *msg ۱* پیغام «در دست بررسی» را می نویسد.
۵. کنترل گر *msg ۱* را به صفحه وضعیت درخواست ثبت آگهی ارسال می کند.
۶. صفحه وضعیت درخواست ثبت آگهی *msg ۱* را نمایش می دهد.
۷. در صورتی که درخواست کاربر پذیرفته شود،
  - ۷.۱. کنترل گر روی *msg ۱* پیغام «پذیرفته شده» را می نویسد.
  - ۷.۲. کنترل گر شیء آگهی را در *DBMgr* ذخیره می کند.
۸. در صورتی که درخواست کاربر پذیرفته نشود،
  - ۸.۱. کنترل گر روی *msg ۱* پیغام «رد شده» را می نویسد.
۹. کنترل گر *msg ۱* را به صفحه وضعیت درخواست ثبت آگهی ارسال می کند.
۱۰. صفحه وضعیت درخواست ثبت آگهی *msg ۱* را نمایش می دهد.

#	فاعل	کنش فاعل	دیگر داده ها/اشیا	شیء ای که کنش روی آن انجام می شود
۱	کاربر	کلیک می کند	گزینه ذخیره	صفحه وضعیت درخواست ثبت آگهی
۲	صفحه وضعیت درخواست ثبت آگهی	ارسال می کند	اطلاعات	کنترل گر صفحه درخواست ثبت آگهی
۳	کنترل گر	ایجاد می کند	پیغام <i>msg ۱</i>	<i>msg ۱</i>
۴	کنترل گر	می نویسد	پیغام «در دست بررسی»	<i>msg ۱</i>
۵	کنترل گر	ارسال می کند	<i>msg ۱</i>	صفحه وضعیت درخواست ثبت آگهی

صفحه وضعیت درخواست آگهی	<i>msg ۱</i>	نمایش می‌دهد	صفحه وضعیت درخواست ثبت آگهی	۶
			در صورتی که درخواست کاربر پذیرفته شود	۷
	<i>msg ۱</i>	پیغام «پذیرفته شده»	کنترل‌گر	۷.۱
	<i>DBMgr</i>	ذخیره می‌کند	کنترل‌گر	۷.۲
			در صورتی که درخواست کاربر پذیرفته نشود	۸
	<i>msg ۱</i>	پیغام «رد شده»	کنترل‌گر	۸.۱
صفحه وضعیت درخواست ثبت آگهی	<i>msg ۱</i>	ارسال می‌کند	کنترل‌گر	۹
صفحه وضعیت درخواست آگهی	<i>msg ۱</i>	نمایش می‌دهد	صفحه وضعیت درخواست ثبت آگهی	۱۰

جدول ۵-۵ سناریو مورد کاربرد پنج



## \* الگوهای به کار رفته در نمودار UC05

- الگوی کنترل‌گر:

الگوی کنترلر در زمان نوشتن سناریوی مورد کاربرد اعمال شده است تا واسط گرافیکی را از اشیای کسبوکار مجزا نماید. مسئولیت کنترلر این است که بر اشیای کسبوکار نظارت کرده و یا آنها را هماهنگ کند تا گام‌های یک مورد کاربرد را انجام دهند. به این منظور، در نمودارهای توالی کنترلر مورد کاربرد، مورد استفاده قرار می‌گیرد. بر همین اساس در نمودار توالی ذکر شده از Ad register controller استفاده شده است و مسئولیت نمایش اطلاعات بر عهده Ad registration page و همچنین مسئولیت پردازش درخواست‌های ایجاد شده بر عهده Ad register controller قرار گرفته است. در ادامه توضیح داده شده است که چگونه با اعمال این الگو، می‌توان طراحی نرم‌افزار را با استفاده از اصول طراحی، بهبود بخشید.

— طراحی برای تغییر: سیستم باید به گونه‌ای باشد که بتواند با تغییرات سازگار شود و یا به راحتی تغییر داده شود. با اعمال این الگو در نمودارهای توالی، واسط و رفتار تعاملی کنترلر تغییر نمی‌کند به همین دلیل تغییرات رخ داده در منطق کسبوکار تاثیر چندانی بر روی واسط گرافیکی نخواهد داشت. با تغییر در نیازمندی‌ها تنها نیاز است که واسط‌های جدید را اضافه کرده و اجازه دهیم هر کدام از آنها توابع متناسب از کنترلر مربوطه را فراخوانی کنند.

— جداسازی دغدغه‌ها: جداسازی دغدغه‌ها توسط طراحی به خوبی پشتیبانی می‌شود زیرا هر قسمت تنها بر روی یک جنبه از نیازمندی‌ها تمرکز دارد. به عنوان مثال در نمودار توالی ذکر شده، Ad registration page تنها مسئول نمایش اطلاعات به کاربر است در حالی که Ad register controller مسئول پردازش درخواست‌های ایجاد شده می‌باشد.

— چسبندگی زیاد: همانطور که در قسمت جداسازی دغدغه‌ها توضیح داده شد، با اعمال این الگو، واسط گرافیکی تنها مسئول نمایش اطلاعات به کاربر و کنترلر مسئول پردازش درخواست‌های ایجاد شده می‌باشد. بنابراین طراحی به گونه‌ای است که توابع هر پیمانه، بیشترین درجه ارتباط با مسئولیت اصلی پیمانه را دارند و در نتیجه هر دو، چسبندگی تابعی دارند.

— طراحی اشیای نادان: با اعمال این الگو، واسط گرافیکی تنها از نمایش اطلاعات به کاربر و کنترلر از پردازش مورد کاربرد مربوطه اطلاع دارد. بنابراین اصل ساده و احمقانه فرض کن و همچنین طراحی اشیای نادان به خوبی رعایت می‌شود.

— پنهان‌سازی اطلاعات: با در نظر گرفتن کلاس DBMgr در نمودارهای توالی، پایگاه داده از کنترلر مخفی شده است. این کار، اصل پنهان‌سازی اطلاعات را رعایت کرده و از ایجاد کنترلرهای متورم جلوگیری می‌کند.

— جفت‌شدگی کم: با در نظر گرفتن کلاس DBMgr، پایگاه داده از کنترلر مجزا شده و در نتیجه تاثیر تغییرات، کم خواهد بود و اصل جفت‌شدگی کم رعایت می‌شود.

## • الگوی خالق:

الگوی خالق مشخص می‌کند که مسئولیت ایجاد یک شی باید بر عهده چه کسی گذاشته شود. از آن جایی که در این نمودار توالی، سازنده کلاس `dashboard page` به آدرس مرکز و همچنین کد معرف نیاز دارد و این اطلاعات در دست `Ad register controller` می‌باشد؛ بنابراین با اعمال این الگو، این کنترلگر برای ایجاد شی `dashboard page` انتخاب شده است. همچنین `Ad register controller` توانایی ایجاد یک شی از کلاس `Msg` را دارد؛ بنابراین با اعمال این الگو، این کنترلگر باید ایجادکننده شی `msg` باشد.

## • الگوی خبره:

هنگامی که یک کلاس اطلاعات لازم را برای انجام یا تایید عملی دارد. مانند فلش ۱.۱.۲ که نمونه‌ای از این کار را انجام می‌دهد.

## 5-6- سناریو و مدل تعامل شی برای گام ششم از UC08

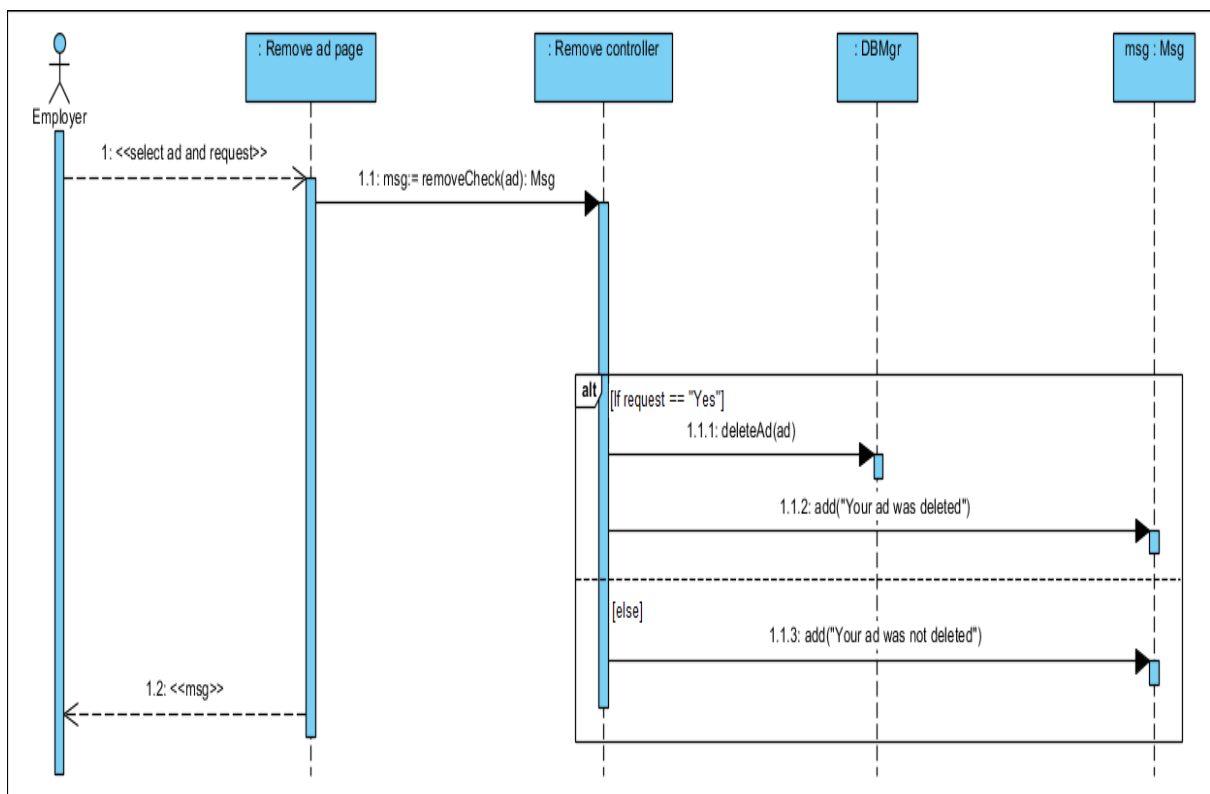
### \* امکان حذف آگهی

۱. کاربر یک گزینه را انتخاب می‌کند.
- ۱.۱. صفحه حذف آگهی گزینه انتخاب شده را به کنترلگر می‌فرستد.
- ۱.۱.۱. در صورتی که گزینه «بله» انتخاب شده باشد
  - ۱.۱.۱.۱. کنترلگر آگهی انتخاب شده را از `DBMgr` حذف می‌کند.
  - ۱.۱.۱.۲. کنترلگر یک شی `msg` ایجاد می‌کند.
  - ۱.۱.۱.۳. کنترلگر روی `msg` پیام «حذف آگهی با موفقیت انجام شد» را می‌نویسد.
  - ۱.۱.۱.۴. کنترلگر `msg` را به صفحه حذف آگهی ارسال می‌کند.
  - ۱.۱.۱.۵. صفحه حذف آگهی `msg` را نمایش می‌دهد.
- ۱.۱.۲. در غیر این صورت
  - ۱.۱.۲.۱. حذف آگهی انجام نمی‌شود.

#	فاعل	کنش فاعل	دیگر داده‌ها/اشیا	شی‌ای که کنش روی آن انجام می‌شود
۱	کاربر	انتخاب می‌کند	یک گزینه	صفحه حذف آگهی
۱.۱	صفحه حذف آگهی	می‌فرستد	گزینه انتخاب شده	کنترلگر
۱.۱.۱	در صورتی که گزینه «بله» انتخاب شده باشد			
۱.۱.۱.۱	کنترلگر	حذف می‌کند	آگهی انتخاب شده	<code>DBMgr</code>

۱.۱.۱.۲	کنترل گر	ایجاد می کند	شیء <i>msg</i> ۱	<i>msg</i> ۱
۱.۱.۱.۳	کنترل گر	می نویسد	پیام «حذف آگهی با موفقیت انجام شد»	<i>msg</i> ۱
۱.۱.۱.۴	کنترل گر	ارسال می کند	<i>msg</i> ۱	صفحه حذف آگهی
۱.۱.۱.۵	صفحه حذف آگهی	نمایش می دهد	<i>msg</i> ۱	صفحه حذف آگهی
۱.۱.۲	در غیر این صورت			
۱.۱.۲.۱	حذف آگهی انجام نمی شود.			

جدول ۵-۶ جدول سناریو مورد کاربرد هشت



## \* الگوهای به کار رفته در نمودار UC08

### • الگوی کنترلگر :

الگوی کنترلگر در زمان نوشتن سناریوی مورد کاربرد اعمال شده است تا واسطه گرافیکی را از اشیای کسب و کار مجزا نماید. مسئولیت کنترلگر این است که بر اشیای کسب و کار نظارت کرده و یا آنها را هماهنگ کند تا گام‌های یک مورد کاربرد را انجام دهند. به این منظور، در نمودارهای توالی کنترلگر مورد کاربرد،

مورد استفاده قرار می‌گیرد. بر همین اساس در نمودار توالی ذکر شده از Remove controller استفاده شده است و مسئولیت نمایش اطلاعات بر عهده Remove ad page و همچنین مسئولیت پردازش درخواست‌های ایجاد شده بر عهده Remove controller قرار گرفته است. در ادامه توضیح داده شده است که چگونه با اعمال این الگو، می‌توان طراحی نرم‌افزار را با استفاده از اصول طراحی، بهبود بخشید.

— طراحی برای تغییر: سیستم باید به گونه‌ای باشد که بتواند با تغییرات سازگار شود و یا به راحتی تغییر داده شود. با اعمال این الگو در نمودارهای توالی، واسط و رفتار تعاملی کنترلر تغییر نمی‌کند به همین دلیل تغییرات رخ داده در منطق کسب‌وکار تاثیر چندانی بر روی واسط گرافیکی نخواهد داشت. با تغییر در نیازمندی‌ها تنها نیاز است که واسط‌های جدید را اضافه کرده و اجازه دهیم هر کدام از آنها توابع متناسب از کنترلر مربوطه را فراخوانی کنند.

— جداسازی دغدغه‌ها: جداسازی دغدغه‌ها توسط طراحی به خوبی پشتیبانی می‌شود زیرا هر قسمت تنها بر روی یک جنبه از نیازمندی‌ها تمرکز دارد. به عنوان مثال در نمودار توالی ذکر شده، Remove ad page تنها مسئول نمایش اطلاعات به کاربر است درحالی که Remove controller مسئول پردازش درخواست‌های ایجاد شده می‌باشد.

— چسبندگی زیاد: همانطور که در قسمت جداسازی دغدغه‌ها توضیح داده شد، با اعمال این الگو، واسط گرافیکی تنها مسئول نمایش اطلاعات به کاربر و کنترلر مسئول پردازش درخواست‌های ایجاد شده می‌باشد. بنابراین طراحی به گونه‌ای است که توابع هر پیمانه، بیشترین درجه ارتباط با مسئولیت اصلی پیمانه را دارند و در نتیجه هر دو، چسبندگی تابعی دارند.

— طراحی اشیای نادان: با اعمال این الگو، واسط گرافیکی تنها از نمایش اطلاعات به کاربر و کنترلر از پردازش مورد کاربرد مربوطه اطلاع دارد. بنابراین اصل ساده و احمقانه فرض کن و همچنین طراحی اشیای نادان به خوبی رعایت می‌شود.

— پنهان‌سازی اطلاعات: با در نظر گرفتن کلاس DBMgr در نمودارهای توالی، پایگاه داده از کنترلر مخفی شده است. این کار، اصل پنهان‌سازی اطلاعات را رعایت کرده و از ایجاد کنترلرهای متورم جلوگیری می‌کند.

— جفت‌شدگی کم: با در نظر گرفتن کلاس DBMgr، پایگاه داده از کنترلر مجزا شده و در نتیجه تاثیر تغییرات، کم خواهد بود و اصل جفت‌شدگی کم رعایت می‌شود.

### • الگوی خالق:

الگوی خالق مشخص می‌کند که مسئولیت ایجاد یک شی باید بر عهده چه کسی گذاشته شود. از آن جایی که در این نمودار توالی، سازنده کلاس Remove controller به آدرس مرکز و همچنین کد معرف نیاز

دارد و این اطلاعات در دست خود Remove controller می باشد؛ بنابراین با اعمال این الگو، این کنترلر برای ایجاد شیء Remove controller انتخاب شده است. Remove controller خود توانایی ایجاد یک شیء از کلاس Msg را دارد؛ بنابراین با اعمال این الگو، این کنترلر باید ایجادکننده شیء msg باشد.

# فصل ششم: استنتاج نمودار کلاس طراحی

نمودار کلاس طراحی<sup>۱</sup> یک نمودار UML است که از روی مدل‌های رفتاری و مدل دامنه به دست می‌آید. این نمودار یک نقشه‌ی طراحی است که فعالیت‌های بعدی پیاده‌سازی، آزمون و یکپارچه‌سازی را تسهیل می‌کند؛ از این رو تهیه‌ی DCD بسیار مهم است. ورودی‌های این فصل نمودارهای توالی و مدل دامنه تهیه شده در فصول قبلی و خروجی این فصل نمودار کلاس طراحی شامل تمام کلاس‌ها، اعمال و روابطی که برای ارضای نیازمندی‌ها است.

\* گام‌های استنتاج DCD از روی نمودار توالی طراحی عبارتند از :

۱. شناسایی کلاس‌ها

۲. شناسایی متدها

۳. شناسایی ویژگی‌ها

۴. شناسایی رابطه‌ها

۵. مرور DCD

در این بخش، با استفاده از نمودارهای توالی رسم شده (فصل پنجم) و همچنین مدل دامنه (فصل دوم)، به تبیین کلاس‌ها، صفت‌ها، توابع و ارتباط میان این کلاس‌ها پرداخته شد و بر اساس آن نمودار کلاس طراحی شکل ۶-۱ رسم شد. نمودار کلاس برای تجسم، توصیف و مستندسازی جنبه‌های مختلف سیستم رسم شده است و همچنین برای ساخت کد اجرایی نرم‌افزار به صورت شی‌گرا در مراحل بعدی مورد استفاده قرار می‌گیرد.

---

<sup>1</sup> Design Class Diagram (DCD)





\* سازماندهی کلاس ها برحسب سبک معماری سیستم

لایه کنترلر	لایه دیتابیس	لایه منطق کسب و کار	لایه گرافیکی
LoginController ChangePasswordController AdRegisterController RemoveController LocatingController	DBManager	User Employer Ad Message Location UserAccount EmailAPI	ChangePasswordPage AdRegistrationPage LoginPage LocatingPage RemoveAdPage AdDashboard

# فصل هفتم: جمع‌بندی و انتقال تجارب

این فصل از پروژه به عنوان فصل نهایی، به جمع‌بندی و انتقال تجارب تخصیص داده شده است. مباحث تدریس شده و کتاب مرجع، اصول مهندسی نرم‌افزار را بر اساس فناوری شیء‌گرا تعریف نموده و یک متدولوژی نوین در توسعه سیستم‌های نرم‌افزاری را به ما آموخت. این درس و متدولوژی مطرح شده در آن، منجر به یادگیری نحوه صحیح مدیریت پروژه‌ها برای اعضای گروه شد که یاری‌رسان ما در آینده و در محیط کار خواهد بود. در ابتدا یعنی فاز اول، به مقدمه و شرحی کلی از پروژه و تبیین نیازمندی‌ها پرداخته و در ادامه با طوفان فکری اعضا و تشکیل یک مدل دامنه‌ی یکپارچه و سپس طراحی معماری از سیستم و استنتاج موردکاربردها با توجه به نیازمندی‌های تبیین شده، سعی در پیشبرد اهداف به سوی مدل سازی تعاملی داشته ایم.

## 1-7- تجارب

- استفاده از ابزارهای گوناگونی که برای رسم نمودارها، نگارش متن‌ها، مدیریت پروژه و زمان‌بندی و... در این پروژه استفاده شد، بسیار آموزنده بود. یعنی برای مثال استفاده از Paradigm Visual برای ترسیم نمودارها و همچنین وبسایت Trello جهت انجام هماهنگی‌های لازم، جهت دادن به کار گروهی و مدیریت و زمان‌بندی فعالیت‌ها، تجربه‌ای جدید و کارآمد بود.
- کار گروهی در قالب گروه شش نفره، تجربه‌ای جدید و سازنده بود. در ابتدا به دلیل تعداد زیاد نفرات و فرض آنکه با تقسیم مناسب حجم کار بین همه‌ی اعضا، از فشار کار روی هر یک از اعضا کاسته می‌شود، مسرت‌بخش به نظر می‌آمد؛ اما با گذشت زمان و آشنایی بیشتر با اولویت‌ها، نظرات و عادات افراد، امیدها برای ایجاد هماهنگی قابل قبول و تعدیل بار و فشار روانی و جسمی برای به سرانجام رساندن این پروژه، کم رنگ و کم رنگ‌تر شد. در نهایت ایجاد هماهنگی و تقسیم وظایف به طور یکسان و عادلانه با چالش‌ها و مشکلاتی رو به رو بود. با همراهی، همدلی و تعهد اعضا تیم و همین‌طور افزایش جلسه‌های تیمی، شناخت اعضا از یکدیگر افزایش یافت و همین شناخت نسبی، به پخش شدن تسک‌های متناسب با هر فرد کمک شایانی کرد و با تقسیم پروژه به قسمت‌های کوچک و با تخصیص آن به اعضا گروه توانستیم از فشارهای موجود به سختی اما با موفقیت عبور نماییم. یعنی با توجه به عدم همکاری قبلی تیم با هم، فرآیند تثبیت کار گروهی کمی زمان‌بر بود ولی با کمک شناخت صحیح اعضا تیم نسبت به کار گروهی، تیم توانست شاکله اصلی خود را ساخته و هماهنگی و کارایی اعضا با یکدیگر به شکل قابل توجهی افزایش یافت. به طور کلی برای انجام وظایف در این پروژه اعضای تیم به زیرتیم‌هایی دو یا سه نفره تقسیم شدند. برای بعضی از بخش‌ها مانند تبیین نیازمندی‌ها، جلسه طوفان فکری، مدل دامنه و نوشتن سناریوها برای نمودارهای توالی، همه اعضای تیم با همکاری یکدیگر این بخش‌ها را به پایان رساندند. قابل ذکر است که با توجه به

اینکه متدولوژی مدنظر در این پروژه، متدولوژی چابک بوده است، به صورت مداوم جلساتی برای هماهنگی میان زیرتیم‌ها برگزار می‌شد و افراد در هماهنگی کامل بودند.

- با توجه به محدودیت‌های زمانی موجود، مدیریت زمان و تقسیم وظایف به نحو مناسب، یکی دیگر از تجربیات آموزنده ما در این پروژه بود.

- برگزاری جلسات به صورت آنلاین و حضوری انجام گرفت و سعی بر این بود که تا حد امکان جلسات آنلاین نیز مشابه جلسات حضوری و با مشارکت همه افراد تیم انجام بگیرد. جلسات گروهی در ابتدا با فواصل زمانی نسبتاً زیاد برگزار می‌شدند؛ اما به مرور و همچنین با امتحان کردن شیوه‌های مختلف به روش‌های دیگری برای هم‌فکری و همکاری و همچنین برنامه‌ریزی برای پروژه روی آوردیم و در ماه آخر چندین جلسه با حضور جمعی از اعضای گروه برگزار شد که پیشرفت انجام کار در آنها مشهود بود.

- همزمانی تدریس دروس مورد نیاز برای انجام هر مرحله از پروژه و درگیر کردن مستمر اعضا با مطالب و الزام استفاده از این آموزه‌ها در انجام یک پروژه واقعی، تجربه ارزشمند و چالش برانگیزی بود.

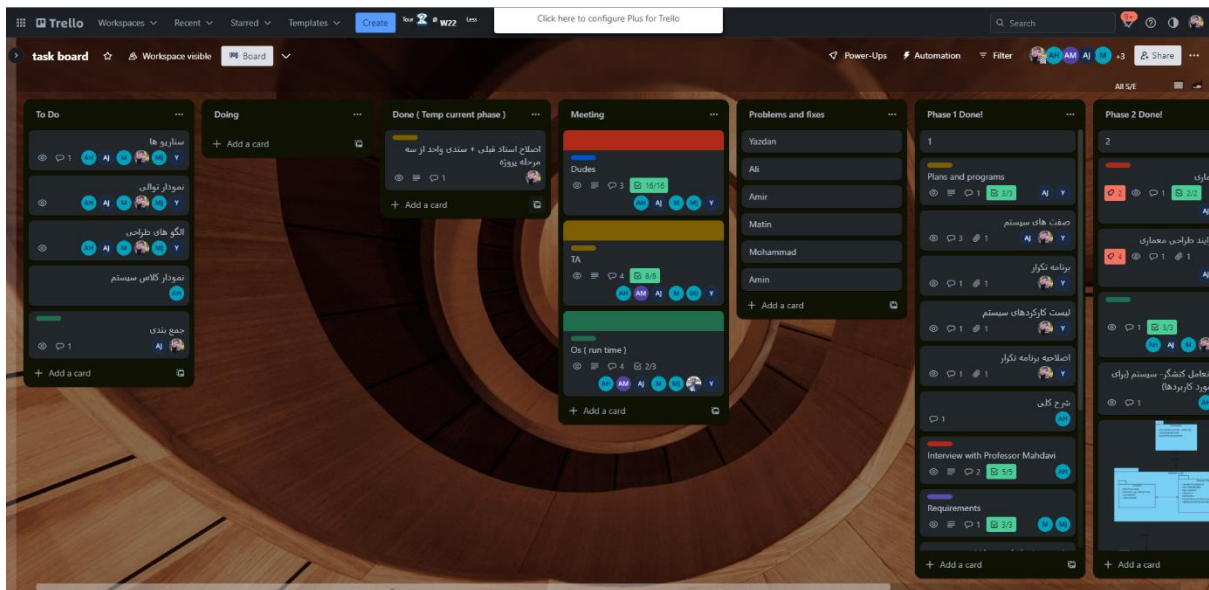
- نکته بارز این پروژه‌ای که تمام اعضا تقریباً کل ترم با آن درگیر بودند، "واقعی" و "کاربردی" آن است. همین‌طور انجام پروژه‌ای در این ابعاد برای تمام اعضای گروه، تجربه متفاوت و ارزنده‌ای بود و ما را با جنبه‌های مختلف انجام یک پروژه تقریباً واقعی به بهترین شکل آشنا کرد. شناخت اهمیت و پیچیدگی طراحی و برنامه‌ریزی برای پروژه‌ها و همچنین انواع فرآیندها و متدهای توسعه نرم‌افزار از مواردیست که میتوان به آن اشاره کرد.

## 2-7- ابزارها

### 1-2-7- مدیریت پروژه

برای مدیریت بهتر و تقسیم فعالیت‌های گوناگون بین اعضای گروه از Trello استفاده شد. همچنین استاد محترم و دستیار آموزشی درس نیز به منظور نظارت بر روند فعالیت اعضا، نحوه همکاری گروه و ارزیابی عملکرد آن‌ها به فضای کار اضافه شدند.

در صفحه‌ی بعدی نمایی از فضای کار در Trello را نشان داده ایم:



شکل ۱-۲-۷- تصویر صفحه تیم در نرم افزار ترلو

## 7-2-2- ترسیم نمودارها و طراحی

برای ترسیم نمودارهایی مثل Class Diagram, Package Diagram, Domain Model از نرم افزار Visual Paradigm و همچنین برای طراحی شکل ها از وبسایت draw.io استفاده شد.

## 7-2-3- تولید محتوا

برای نگارش گزارش از Word و Google docs و برای آماده سازی اسلایدهای ارائه از Power Point استفاده شده است.

## 7-2-4- راه های ارتباطی و جلسات

ابزار ارتباطی گروه ما برای ایجاد هماهنگی های لازم و تعامل، بستر تلگرام بود. همچنین تمام جلسات غیر حضوری در Google Meet برگزار شدند.

## سخن آخر

در پروژه سامانه کاریابی، بسیاری از مفاهیم و تجربیات مرتبط با تحلیل داده‌ها، طراحی و توسعه نرم‌افزار، بهینه‌سازی الگوریتم‌ها و ارتباط با مشتریان و کاربران را فرا گرفتیم.

به نظرمان این تجربیات و مهارت‌هایی که در پروژه کسب کردیم، به طور قطعی به ما در آینده کمک خواهند کرد تا در زمینه‌های مرتبط با تحلیل داده‌ها، توسعه نرم‌افزار و روابط با مشتریان و کاربران، موفقیت بیشتری داشته باشیم. همچنین، مهارت‌هایی که در پروژه به دست آوردیم، می‌توانند در بسیاری از صنایع و شرکت‌ها مورد استفاده قرار بگیرند، به عنوان مثال در تحلیل داده‌های مالی، برنامه‌ریزی بازاریابی و مدیریت پروژه‌های تحقیق و توسعه. به طور خلاصه:

۱. **تجربه‌ی کار با یک تیم بزرگ:** در این پروژه با تیمی از برنامه‌نویسان، طراحان و کنترل‌گران پروژه همکاری کردیم و تجربه‌ی کار با یک تیم بزرگ را کسب کردیم.

۲. **مدیریت پروژه:** در این پروژه با مدیریت پروژه و برنامه‌ریزی زمانی آشنا شدیم و دانشی در این زمینه کسب کردیم.

۳. **طراحی و پیاده‌سازی سیستم‌های پیچیده:** این پروژه شامل طراحی و پیاده‌سازی یک سیستم پیچیده بود و ما تجربه‌ی کار با سیستم‌های بزرگ را کسب کردیم.

۴. **تجربه‌ی کار با تکنولوژی‌های جدید:** در این پروژه از تکنولوژی‌های جدیدی استفاده شده بود که پیش‌تر بیان کردیم و ما تجربه‌ی کار با آن‌ها را کسب کردیم.

۵. **توانایی حل مسئله و ارائه‌ی راه‌حل:** در این پروژه برای حل مسئله‌های پیش آمده، به دقت توجه کردیم و راه‌حل‌های مختلفی را برای آن‌ها پیشنهاد کردیم.

\* در نهایت باید از جناب آقای دکتر محمدرضا شهرباف برای تدریس فوق‌العاده و کمک‌های بی‌دریغشان در به ثمر رساندن این پروژه قدردانی کنیم. همچنین از جناب آقای مأمّن پوش بابت کمک‌های ارزنده ایشان تشکر به عمل آوریم. \*

