

به نام خدا



دانشگاه صنعتی شریف
درس طراحی شیگرای سیستم‌ها

سند سامانه مدیریت توزیع غذا

امیرحسین عربزاده (۹۸۱۰۵۹۰۸)

امیرمحمد فخیمی (۹۹۱۷۰۵۳۱)

محمدامین لطفی (۹۸۱۷۱۰۷۵)

محمدمهدی میرزایی (۹۹۱۷۱۰۲۲)

فهرست مطالب

2.....	فهرست مطالب
5.....	لیست نیازمندی‌های وظیفه‌ای.....
8.....	لیست نیازمندی‌های غیروظیفه‌ای.....
10.....	معماری سیستم
11.....	توصیف کنشگرها
12.....	نمودار موارد کاربرد
12.....	زیرسیستم احراز هویت و دسترسی
13.....	زیرسیستم مدیریت مشتریان.....
14.....	زیرسیستم مدیریت تهیه‌کنندگان غذا
15.....	زیرسیستم مدیریت پیکها.....
16.....	توصیف موارد کاربرد.....
16.....	زیرسیستم احراز هویت و دسترسی
21.....	زیرسیستم مدیریت مشتریان.....
49.....	زیرسیستم مدیریت تهیه‌کنندگاه غذا
70.....	زیرسیستم مدیریت پیکها.....
77.....	نمودارهای فعالیت
77.....	زیرسیستم احراز هویت و دسترسی
80.....	زیرسیستم مدیریت مشتریان.....
98.....	زیرسیستم مدیریت تهیه‌کنندگاه غذا
111.....	زیرسیستم مدیریت پیکها.....
115.....	نمودار کلاس‌های تحلیل
116.....	نمودارهای توالی تحلیل
116.....	زیرسیستم احراز هویت و دسترسی
120.....	زیرسیستم مدیریت مشتریان.....
137.....	زیرسیستم مدیریت تهیه‌کنندگاه غذا
150.....	زیرسیستم مدیریت پیکها.....
154.....	نمودار بسته
155.....	نمودار کلاس‌های طراحی
155.....	مولفه‌ی User

156.....	مولفه‌ی Order
157.....	مولفه‌ی Authentication
158.....	مولفه‌ی Payment
159.....	مولفه‌ی Product
160.....	مولفه‌ی Delivery
161.....	نمودارهای توالی طراحی
161.....	زیرسیستم احراز هویت و دسترسی
164.....	زیرسیستم مدیریت مشتریان
181.....	زیرسیستم مدیریت تهیه‌کنندگاه غذا
194.....	زیرسیستم مدیریت پیکها
196.....	نمودار مولفه
197.....	سندهای ریسک
197.....	آنواع ریسکها
197.....	ریسک‌های فنی
197.....	ریسک‌های مدیریتی
197.....	ریسک‌های سازمانی
198.....	ریسک‌های خارجی
198.....	ریسک‌های اضافی
199.....	اولویت‌بندی ریسکها
200.....	تحلیل دنیای واقع
201.....	تحلیل اسم/ فعل
203.....	CRC کارت‌های
208.....	نمونه‌ی اولیه‌ی واسط کاربری قابل اجرا
208.....	زیرسیستم احراز هویت و دسترسی
209.....	زیرسیستم مدیریت مشتریان
210.....	زیرسیستم مدیریت تهیه‌کنندگان غذا
211.....	زیرسیستم مدیریت پیکها
212.....	DishDash تصاویر رابط کاربری سامانه‌ی
212.....	زیرسیستم احراز هویت و دسترسی
213.....	زیرسیستم مدیریت تهیه‌کنندگان غذا
214.....	زیرسیستم مدیریت مشتریان
215.....	زیرسیستم مدیریت پیکها

216.....	گزارش استفاده از الگوهای طراحی در بکاند
220.....	شمای پایگاه داده
221.....	نمودار استقرار
222.....	فهرست بررسی فاز تدقیق
223.....	فهرست بررسی جزئی‌تر فاز تدقیق
225.....	فهرست بررسی نمودارهای فعالیت
226.....	فهرست بررسی نمودار کلاس‌های تحلیل
227.....	فهرست بررسی نمودارهای توالی تحلیل
228.....	فهرست بررسی نمودار بسته
229.....	فهرست بررسی نمودار کلاس‌های طراحی
230.....	فهرست بررسی نمودارهای توالی طراحی
231.....	فهرست بررسی نمودار مولفه
232.....	فهرست بررسی پیاده‌سازی
233.....	فهرست بررسی شمای پایگاه داده
234.....	فهرست بررسی الگوها
235.....	فهرست بررسی محصولات
236.....	برنامه‌ی زمان‌بندی پروژه
240.....	واژه‌نامه
244.....	لينک‌ها

لیست نیازمندی‌های وظیفه‌ای

شناسه	زیرسیستم	موردن کاربرد	اولویت	توضیحات
۱	احراز هویت و دسترسی	ورود	M	
۲	احراز هویت و دسترسی	ثبت‌نام	M	
۳	احراز هویت و دسترسی	خروج	M	
۴	احراز هویت و دسترسی	حذف حساب کاربری	W	
۵	مدیریت مشتریان	مشاهده‌ی لیست رستوران‌ها	M	
۶	مدیریت مشتریان	مشاهده‌ی غذا و قیمت یک رستوران	M	
۷	مدیریت مشتریان	افزودن غذا به سبد خرید همان رستوران	M	امکان افزودن چند غذا با تعداد متفاوت از هر کدام، وجود دارد.
۸	مدیریت مشتریان	مشاهده‌ی سبدهای خرید	M	
۹	مدیریت مشتریان	ویرایش سبد خرید	S	امکان کم یا زیاد کردن و یا حذف سبد خرید وجود داشته باشد.
۱۰	مدیریت مشتریان	انتخاب و مشاهده‌ی یک سبد خرید	M	
۱۱	مدیریت مشتریان	پرداخت	M	بعد از پرداخت موفق، سفارش ثبت می‌شود. توجه کنید که پرداخت شامل هزینه‌ی پیک نیز می‌شود.
۱۲	مدیریت مشتریان	مشاهده‌ی وضعیت فعلی سفارش	M	
۱۳	مدیریت مشتریان	امکان ویرایش اطلاعات	M	اطلاعات حساب کاربری خود را می‌تواند ویرایش کند. (نام، نشانی)
۱۴	مدیریت مشتریان	مشاهده‌ی امتیاز هر رستوران	S	

	S	امتیازدهی به سبد خرید	مدیریت مشتریان	۱۵
	S	مشاهده‌ی تاریخچه‌ی سفارش‌ها	مدیریت مشتریان	۱۶
	C	مشاهده‌ی نظرات کاربران هر رستوران	مدیریت مشتریان	۱۷
	C	مشاهده‌ی توضیحات غذای یک رستوران	مدیریت مشتریان	۱۸
	C	نظردهی به سبد خرید	مدیریت مشتریان	۱۹
	C	مشاهده‌ی محل فعلی سفارش	مدیریت مشتریان	۲۰
	W	امتیازدهی به پیک	مدیریت مشتریان	۲۱
رستوران امکان مشاهده‌ی لیست سفارشات با وضعیت‌های «آماده‌ی ارسال» و «در حال ارسال» را دارد.	M	دیدن لیست سفارشات	مدیریت تهیه‌کنندگان غذا	۲۲
رستوران امکان مشاهده‌ی لیست سفارشات با وضعیت «تحویل شده» را دارد.	C	دیدن لیست تاریخچه‌ی سفارشات	مدیریت تهیه‌کنندگان غذا	۲۳
رستوران فقط می‌تواند از وضعیت «آماده‌ی ارسال» به «در حال ارسال» تغییر دهد.	M	تغییر دادن وضعیت هر سفارش	مدیریت تهیه‌کنندگان غذا	۲۴
موارد هر سفارش در یک فایل جمع‌آوری و دریافت می‌شود.	M	چاپ فاکتور	مدیریت تهیه‌کنندگان غذا	۲۵
	M	درخواست پیک	مدیریت تهیه‌کنندگان غذا	۲۶
به صورت خودکار هنگامی که وضعیت سفارش از «آماده‌ی ارسال» به «در حال ارسال» تغییر کند، سفارش را به پیک با وضعیت آزاد	M	تعیین خودکار یک پیک برای هر سفارش	مدیریت تهیه‌کنندگان غذا	۲۷

تخصیص می‌دهد.				
	C	مشاهده‌ی پیک‌های حمل‌کننده‌ی سفارشات آن رستوران	مدیریت تهیه‌کنندگان غذا	۲۸
۱. به صورت درخواست دادن توسط رستوران و قبول یا رد کردن توسط پیک ۲. تخصیص اتوماتیک بر اساس موقعیت جغرافیایی	W	تعیین هوشمند یک پیک برای هر سفارش	مدیریت تهیه‌کنندگان غذا	۲۹
افزودن غذا شامل اسم، قیمت، عکس، و توضیحات است.	M	افزودن غذا به منو	مدیریت تهیه‌کنندگان غذا	۳۰
	S	حذف غذا از منو	مدیریت تهیه‌کنندگان غذا	۳۱
	C	تعیین موجودی هر غذا	مدیریت تهیه‌کنندگان غذا	۳۲
	S	ویرایش غذا در منو	مدیریت تهیه‌کنندگان غذا	۳۳
	C	افزودن دسته‌بندی برای هر غذا	مدیریت تهیه‌کنندگان غذا	۳۴
پیک فقط می‌تواند از وضعیت «در حال ارسال» به «تحویل شده» تغییر دهد و پس از تغییر، وضعیت پیک به آزاد تغییر می‌کند.	M	تغییر دادن وضعیت هر سفارش	مدیریت پیک‌ها	۳۵
	C	ارسال موقعیت به صورت زنده برای مشتری	مدیریت پیک‌ها	۳۶
	W	ارسال موقعیت به صورت زنده برای رستوران	مدیریت پیک‌ها	۳۷
	W	تغییر وضعیت بین فعال و غیرفعال	مدیریت پیک‌ها	۳۸

لیست نیازمندی‌های غیروظیفه‌ای

اولویت	توضیحات	نیازمندی
کارایی		
M	سیستم باید به خواست مشتریان در کمتر از ۳ ثانیه پاسخ دهد.	زمان پاسخ
S	سیستم باید بتواند با تعداد زیادی از کاربران همزمان بدون مشکل کار کند.	مقیاس پذیری
C	سیستم باید حداقل قادر به پردازش ۱۰۰۰ سفارش در دقیقه باشد.	توانایی پردازش
قابلیت اعتماد		
M	سیستم باید ۲۴/۷ در دسترس باشد.	در دسترس بودن
S	سیستم باید در برابر خطاهای مقاوم بوده و بدون از دست داده به صورت شیوه‌ایی بازیابی شود.	مقاومت در برابر خطأ
M	پشتیبان‌گیری منظم از داده‌ها باید گرفته شود و سیستم باید مکانیزم‌هایی برای بازیابی سریع در صورت از دست رفتن داده داشته باشد.	پشتیبان‌گیری ^۱ و بازیابی
امنیت		
M	کاربران باید پیش از دسترسی به هرگونه عملکردی به طور امن احراز هویت شوند.	احراز هویت
M	دسترسی به عملکردهای مختلف باید بر اساس نقش و مجوزهای کاربران باشد.	اختیار دسترسی
M	داده‌های حساس مانند اطلاعات کاربران و اطلاعات پرداخت باید در طول انتقال و ذخیره‌سازی رمزگذاری شوند.	رمزگذاری داده
راحتی استفاده		
S	رابط کاربری باید شیوا و آسان برای کاربران با توانایی‌های فنی مختلف باشد.	رابط کاربری شیوا
مقیاس پذیری		

¹ Backup

M	برنامه باید با پلتفرم های مختلفی مانند iOS، Android و مرورگرهای وب سازگار باشد.	سازگاری با پلتفرم
S	برنامه باید در دستگاههای مختلف با اندازه و رزولوشن صفحه‌ی متفاوت به طور شیوه‌ای عمل کند.	سازگاری با دستگاه
قابلیت نگهداری		
M	مستندات جامع برای ایجادکنندگان و مدیران باید ارائه شود تا سیستم را بفهمند و نگهداری کنند.	مستندسازی
ادغام		
M	برنامه باید به طور شیوا با درگاههای پرداخت، سرویس‌های نقشه‌برداری و سایر API‌های سرویس‌های خارجی ادغام شود.	ادغام با سرویس‌های خارجی
S	سیستم باید قابلیت ادغام با سیستم‌های قدیمی مورد استفاده توسط رستوران‌ها و خدمات پیک را داشته باشد.	سازگاری با سیستم‌های قدیمی

معماری سیستم

فناوری‌های استفاده شده

زبان برنامه‌نویسی سمت سرور: Java | Spring Boot

سیستم مدیریت پایگاه داده: PostgreSQL

فناوری رابط کاربری: TypeScript | Next.js

مؤلفه‌های معماری

بک‌اند (Java | Spring Boot + PostgreSQL)

بک‌اند Java: یک زبان برنامه‌نویسی قدرتمند و گسترده است که به دلیل قابلیت اطمینان و مقیاس‌پذیری خود شناخته شده است. Java مسئولیت منطق کسب و کار، پردازش داده و ارتباط با رابط کاربری و پایگاه داده را بر عهده دارد. یک چهارچوب معروف Java است که بک‌اند برنامه‌های تحت وب را مدیریت می‌کند.

پایگاه داده PostgreSQL

به عنوان سامانه مدیریت پایگاه داده رابطه‌ای (RDBMS) برای ذخیره اطلاعاتی مانند اطلاعات کاربر، جزئیات رستوران، موارد غذایی، سفارش‌ها، و ... استفاده می‌شود. PostgreSQL پایداری، عملکرد و انعطاف‌پذیری را برای مدیریت داده‌های ساختاری ارائه می‌دهد.

فرانتاند (TypeScript | Next.js)

چهارچوب TypeScript | Next.js: فرانتاند سامانه مدیریت توزیع غذا با استفاده از Next.js که یک چهارچوب React برای ساخت برنامه‌های وب با اجرای سمت سرور (SSR) است، ساخته خواهد شد. Next.js مزایایی از قبیل بهینه‌سازی SEO، بارگذاری سریع صفحات و تجربه کاربری فوق العاده‌ای را ارائه می‌دهد. این اجازه را برای ایجاد رابطه‌های کاربری پویا و تعاملی با مولفه‌های React فراهم می‌کند. TypeScript یک زبان با قدرتمندی‌های JavaScript به همراه type است که جلوی انواع مشکلات را در زمان ایجاد می‌گیرد.

تصویف کنشگرها

توصیف	کنشگر
مشتری فردی است که برای ثبت سفارش، مشاهده اطلاعات رستوران‌ها، و مدیریت حساب خود با سیستم تعامل دارد.	مشتری

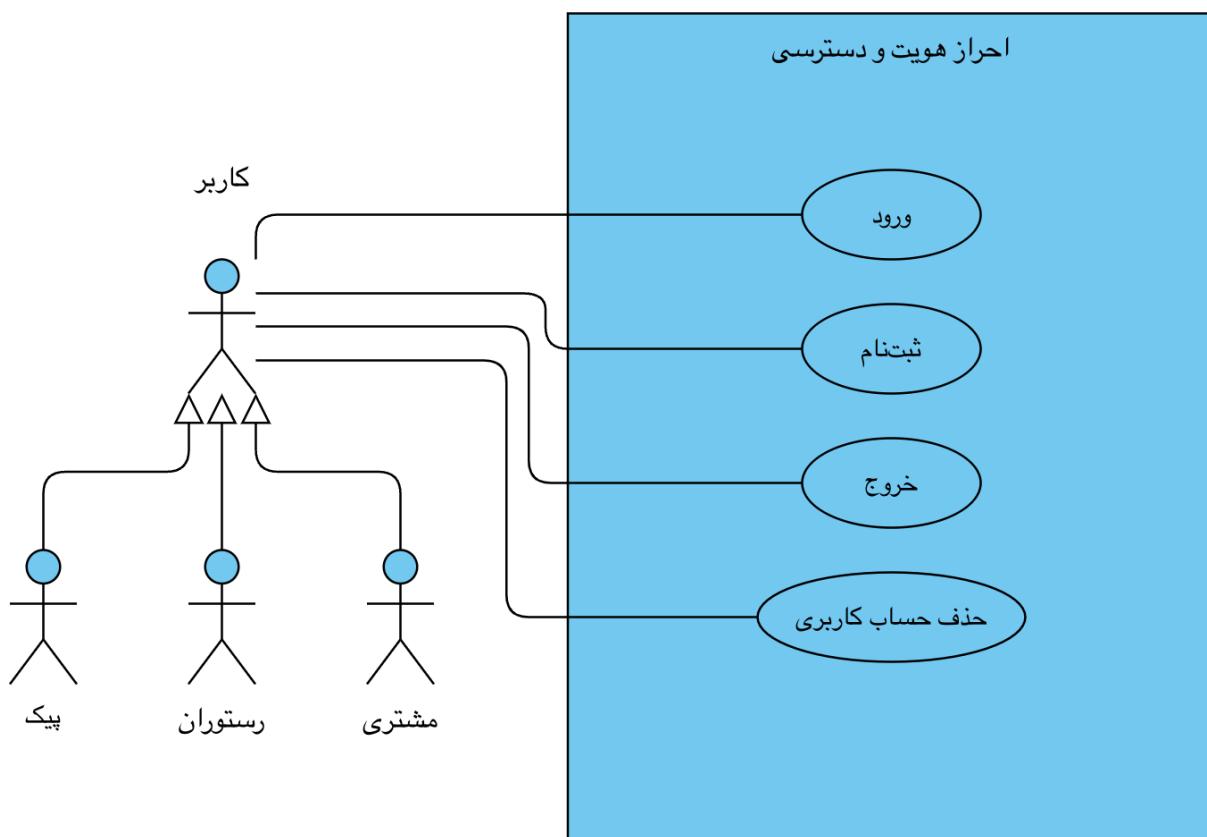
توصیف	کنشگر
رستوران برای مدیریت منو، مشاهده و آماده‌سازی سفارشات، و بهروزرسانی وضعیت سفارشات با سیستم تعامل دارد.	رستوران

توصیف	کنشگر
پیک برای دریافت و تحویل سفارشات، بهروزرسانی وضعیت سفارش، و ارائه ارسال موقعیت مکانی زنده با سیستم تعامل دارد.	پیک

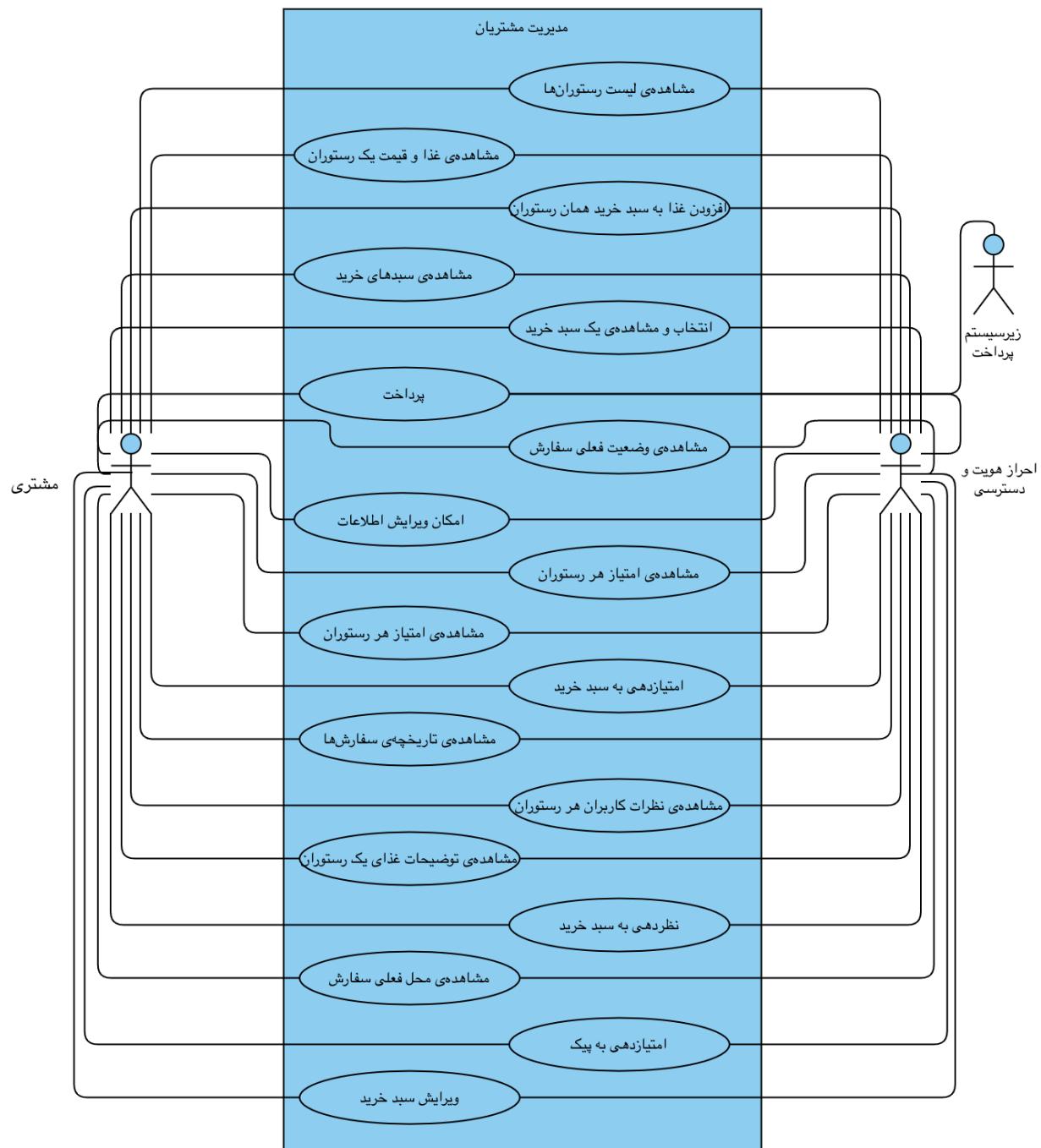
توصیف	کنشگر
سیستم پرداخت مسئول پردازش تراکنش‌های مالی و پرداخت‌ها در سیستم است. این سیستم تعاملات مالی بین مشتریان و رستوران‌ها را مدیریت کرده و اطمینان حاصل می‌کند که تمامی پرداخت‌ها به درستی انجام و ثبت شوند.	زیرسیستم پرداخت

نمودار موارد کاربرد

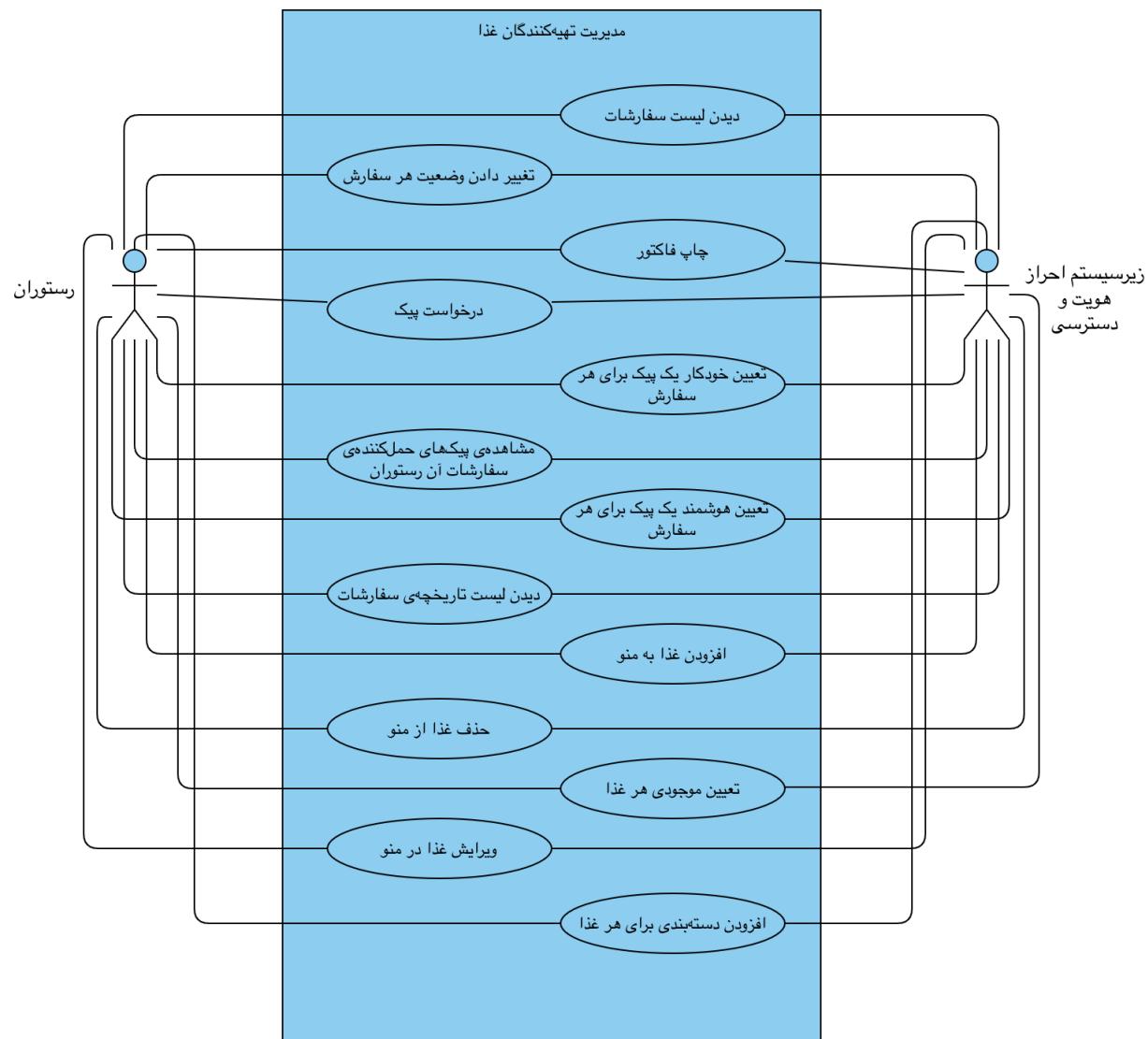
زیرسیستم احراز هویت و دسترسی



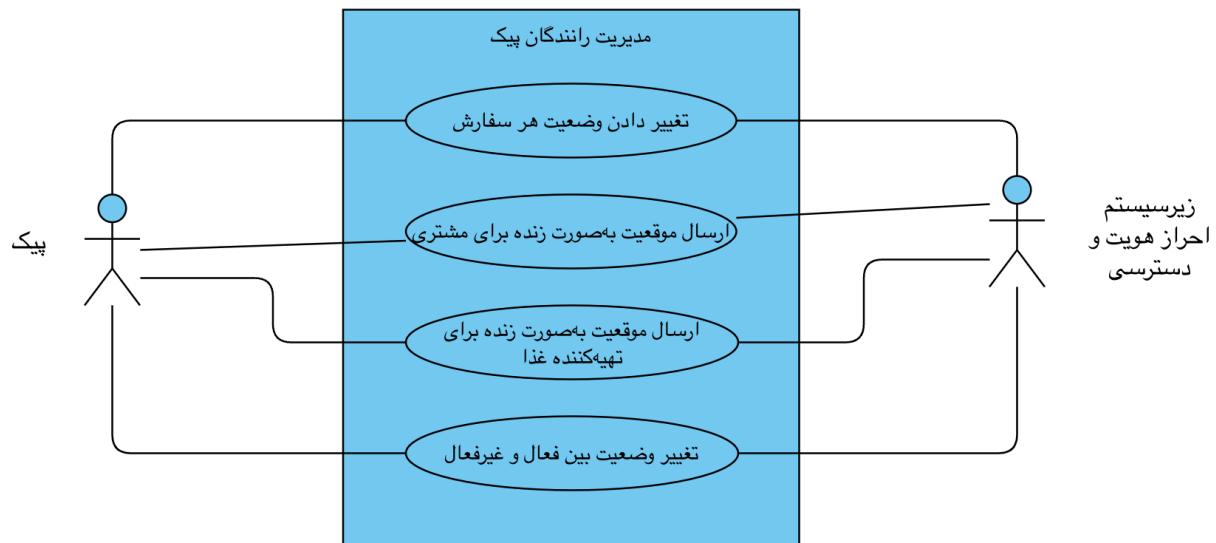
زیرسیستم مدیریت مشتریان



زیرسیستم مدیریت تهیه‌کنندگان غذا



زیرسیستم مدیریت پیک‌ها



توصیف موارد کاربرد

زیرسیستم احراز هویت و دسترسی

مورد کاربرد	ورود
شناسه	۱
توضیح مختصر	کنشگر (مشتری، رستوران، پیک) می‌تواند برای دسترسی به سیستم وارد حساب کاربری خود شود.
کنشگر(های) اولیه	مشتری، رستوران، پیک
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> • کنشگر (مشتری، رستوران، پیک) یک حساب کاربری ثبت شده دارد. • این کنشگر یک نام کاربری و رمز عبور معتبر دارد.
روند اصلی	<ol style="list-style-type: none"> 1. کنشگر فرایند ورود را آغاز می‌کند. 2. سیستم از کنشگر می‌خواهد که نام کاربری و رمز عبور خود را وارد کند. 3. کنشگر نام کاربری و رمز عبور خود را وارد می‌کند. 4. سیستم اعتبارسنجی اطلاعات کنشگر را انجام می‌دهد. 5. تا زمانی که اطلاعات معتبر نباشد: <ol style="list-style-type: none"> a. سیستم پیام خطای نمایش می‌دهد. 6. سیستم به کنشگر اجازه‌ی ورود می‌دهد و کاربر وارد سیستم می‌شود.
پس‌نیازها	<ul style="list-style-type: none"> • کنشگر (مشتری، رستوران، پیک) وارد حساب کاربری خود شده است.
روند جایگزین	

ثبتنام	مورد کاربرد
۲	شناسه
کنشگر (مشتری، رستوران، پیک) می‌تواند یک حساب کاربری جدید در سیستم ثبت کند.	توضیح مختصر
مشتری، رستوران، پیک	کنشگر(های) اولیه
	کنشگر(های) ثانویه
• کنشگر اطلاعات ثبتنام معتبر (مانند ایمیل) دارد.	پیش‌نیازها
1. کنشگر فرایند ثبتنام را آغاز می‌کند. 2. سیستم از کنشگر می‌خواهد اطلاعات ثبتنام خود را وارد کند (مانند نام کاربری رمز عبور). 3. کنشگر اطلاعات مورد نیاز را وارد می‌کند. 4. سیستم اطلاعات را اعتبارسنجی می‌کند. 5. تا زمانی که اطلاعات معتبر نباشد: a. سیستم پیام خطای نمایش می‌دهد. 6. سیستم یک حساب کاربری جدید برای کنشگر ایجاد می‌کند.	روند اصلی
• سیستم یک حساب کاربری برای کنشگر ایجاد می‌کند.	پس‌نیازها
	روند جایگزین

مورد کاربرد	خروج
شناسه	۳
توضیح مختصر	کنشگر (مشتری، رستوران، پیک) می‌تواند از برنامه خارج شود.
کنشگر(های) اولیه	مشتری، رستوران، پیک
کنشگر(های) ثانویه	
پیش‌نیازها	● کنشگر (مشتری، رستوران، پیک) وارد سیستم شده باشد.
روند اصلی	1. کنشگر گزینه خروج از برنامه را انتخاب می‌کند. 2. سیستم کنشگر را از سیستم خارج می‌کند.
پس‌نیازها	● کنشگر (مشتری، رستوران، پیک) از سیستم خارج شده است.
روند جایگزین	

مورد کاربرد	حذف حساب کاربری
شناسه	۴
توضیح مختصر	کنشگر (مشتری، رستوران، پیک) می‌تواند حساب کاربری خود را از سیستم حذف کند.
کنشگر(های) اولیه	مشتری، رستوران، پیک
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● کنشگر (مشتری، رستوران، پیک) وارد سیستم شده باشد.
روند اصلی	<ol style="list-style-type: none"> 1. کنشگر گزینه‌ی حذف حساب کاربری را انتخاب می‌کند. 2. سیستم از کنشگر می‌خواهد که حذف حساب کاربری را تأیید یا لغو کند. 3. اگر کنشگر گزینه‌ی تأیید را انتخاب کند: <ol style="list-style-type: none"> a. سیستم حساب کاربری کنشگر و تمام داده‌های مرتبط را حذف می‌کند. 4. در غیر این صورت: <ol style="list-style-type: none"> a. حساب کاربری فعال باقی می‌ماند.
پس‌نیازها	<ul style="list-style-type: none"> ● حساب کاربری و تمام داده‌های مرتبط با کنشگر (مشتری، رستوران، پیک) به طور دائمی از سیستم حذف شده است.
روند جایگزین	<ul style="list-style-type: none"> ● لغو فرآیند حذف حساب کاربری

حذف حساب کاربری:لغو فرآیند حذف حساب کاربری	روند جایگزین
1.۴	شناسه
کنشگر (مشتری، رستوران، پیک) می‌تواند از حذف حساب کاربری خود منصرف شود و فرآیند را لغو کند.	توضیح مختصر
مشتری، رستوران، پیک	کنشگر(های) اولیه
	کنشگر(های) ثانویه
<ul style="list-style-type: none"> ● کنشگر (مشتری، رستوران، پیک) وارد سیستم شده باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. 2. کنشگر گزینه‌ی لغو حذف حساب کاربری را انتخاب می‌کند. 	روند جایگزین
<ul style="list-style-type: none"> ● کنشگر از فرآیند حذف حساب کاربری خارج شده و حساب کاربری‌اش حذف نمی‌شود. 	پس‌نیازها

زیرسیستم مدیریت مشتریان

مورد کاربرد	مشاهده‌ی لیست رستوران‌ها
شناسه	۵
توضیح مختصر	مشتری لیستی از رستوران‌های موجود را مشاهده می‌کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری گزینه مشاهده‌ی لیست رستوران‌ها را انتخاب می‌کند. 2. سیستم لیست رستوران‌های موجود را نمایش می‌دهد.
پس‌نیازها	<ul style="list-style-type: none"> ● مشتری لیست رستوران‌های موجود را مشاهده می‌کند. ● مشتری می‌تواند به جزئیات بیشتری درباره یک رستوران خاص دسترسی پیدا کند.
روند جایگزین	<ul style="list-style-type: none"> ● رستورانی وجود ندارد

روند جایگزین	مشاهده‌ی لیست رستوران‌ها: رستورانی وجود ندارد
شناسه	1.5
توضیح مختصر	سیستم به مشتری اطلاع می‌دهد که لیستی از رستوران‌های برای نمایش وجود ندارد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد.
روند جایگزین	<ol style="list-style-type: none"> 1. روند جایگزین بعد از مرحله‌ی 1 روند اصلی شروع می‌شود. 2. هیچ رستورانی وجود ندارد و سیستم پیامی مبنی بر عدم وجود رستوران نمایش می‌دهد.
پس‌نیازها	<ul style="list-style-type: none"> ● لیست خالی از رستوران‌ها به همراه پیام

مشاهده‌ی غذا و قیمت یک رستوران	مورد کاربرد
۶	شناسه
مشتری منو و قیمت‌های یک رستوران انتخاب شده را مشاهده می‌کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● مشتری وارد سیستم شده باشد.	پیش‌نیازها
1. مشتری یک رستوران را برای مشاهده‌ی منو و قیمت‌ها انتخاب می‌کند. 2. سیستم منوی رستوران، شامل غذاها و قیمت‌ها را نمایش می‌دهد.	روند اصلی
● مشتری منو و قیمت‌های رستوران انتخاب شده را مشاهده می‌کند.	پس‌نیازها
● رستوران هیچ غذایی ندارد	روند جایگزین

روند جایگزین	مشاهده‌ی غذا و قیمت یک رستوران: رستوران هیچ غذایی ندارد
شناسه	۱.۶
توضیح مختصر	سیستم به مشتری اطلاع می‌دهد که رستوران مربوطه، غذایی برای نمایش ندارد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	مشتری وارد سیستم شده باشد.
روند جایگزین	1. روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. 2. رستوران هیچ غذایی ندارد و سیستم پیامی مبنی بر عدم وجود غذا نمایش می‌دهد.
پس‌نیازها	

افزودن غذا به سبد خرید همان رستوران	مورد کاربرد
۷	شناسه
مشتری غذاهای را از یک رستوران به سبد خرید همان رستوران اضافه می‌کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. مشتری یک رستوران را انتخاب کرده باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. مشتری یک یا چند غذا را از منوی رستوران انتخاب می‌کند. 2. مشتری تعداد هر مورد انتخاب شده را مشخص می‌کند. 3. سیستم موارد انتخاب شده را به سبد خرید مشتری اضافه می‌کند. 4. سیستم هزینه‌ی کل اقلام در سبد خرید را به روزرسانی می‌کند. 	روند اصلی
<ul style="list-style-type: none"> غذاهای انتخاب شده به سبد خرید مشتری اضافه می‌شود. هزینه‌ی کل اقلام در سبد خرید به روزرسانی می‌شود. 	پس‌نیازها
	روند جایگزین

مشاهده‌ی سبدهای خرید	مورد کاربرد
۸	شناسه
مشتری سبدهای خرید خود را مشاهده می‌کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
• مشتری وارد سیستم شده باشد.	پیش‌نیازها
1. مشتری گزینه مشاهده سبدهای خرید خود را انتخاب می‌کند. 2. سیستم لیست سبدهای خرید مشتری را نمایش می‌دهد.	روند اصلی
• مشتری لیست سبدهای خرید خود را مشاهده می‌کند.	پس‌نیازها
• مشتری هیچ سبد خریدی ندارد	روند جایگزین

روند جایگزین	مشاهده‌ی سبدهای خرید: مشتری هیچ سبد خریدی ندارد
شناسه	1.۸
توضیح مختصر	سیستم به مشتری اطلاع می‌دهد که سبد خریدی برای مشاهده ندارد
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	● مشتری وارد سیستم شده باشد.
روند جایگزین	1. روند جایگزین بعد از مرحله‌ی 1 روند اصلی شروع می‌شود. 2. مشتری هیچ سبد خریدی ندارد و سیستم پیامی را نمایش می‌دهد که مبنی بر عدم وجود سبدهای خرید است.
پس‌نیازها	

مورد کاربرد	ویرایش سبد خرید
شناسه	٩
توضیح مختصر	مشتری می‌تواند یک سبد خرید خاص را ویرایش کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری یک یا چند سبد خرید داشته باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری گزینه‌ی مشاهده سبدهای خرید خود را انتخاب می‌کند. 2. سیستم لیست سبدهای خرید مشتری را نمایش می‌دهد. 3. مشتری یک سبد خرید خاص را انتخاب می‌کند. 4. مشتری تعداد غذা�های آن سبد خرید را کم، زیاد، و یا حذف می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● مشتری محتوای سبد خرید انتخاب شده را طبق آخرین تغییراتی که اعمال کرده است، مشاهده می‌کند.
روند جایگزین	

اننتخاب و مشاهده‌ی یک سبد خرید	مورد کاربرد
۱۵	شناسه
مشتری می‌تواند یک سبد خرید خاص را انتخاب کرده و محتوای آن را مشاهده کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. مشتری یک یا چند سبد خرید داشته باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. مشتری گزینه‌ی مشاهده سبدهای خرید خود را انتخاب می‌کند. 2. سیستم لیست سبدهای خرید مشتری را نمایش می‌دهد. 3. مشتری یک سبد خرید خاص را انتخاب می‌کند. 4. سیستم محتوای سبد خرید انتخاب شده به همراه مجموع قیمت آن را نمایش می‌دهد. 	روند اصلی
<ul style="list-style-type: none"> مشتری محتوای سبد خرید انتخاب شده را مشاهده می‌کند. 	پس‌نیازها
	روند جایگزین

پرداخت	مورد کاربرد
۱۱	شناسه
مشتری پس از انتخاب غذاها و افزودن آنها به سبد خرید، اقدام به پرداخت می‌کند و سفارش ثبت می‌شود.	توضیح مختصر
مشتری	کنشگر(های) اولیه
زیرسیستم پرداخت	کنشگر(های) ثانویه
<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری یک یا چند سبد خرید داشته باشد. 	پیش‌نیازها
<ul style="list-style-type: none"> .1 مشتری یک سبد خرید را انتخاب می‌کند. .2 مشتری گزینه‌ی پرداخت را انتخاب می‌کند. .3 سیستم جزئیات سبد خرید را به همراه هزینه‌ی کل نمایش می‌دهد. .4 مشتری گزینه‌ی پرداخت را انتخاب می‌کند. .5 مشتری به درگاه پرداخت هدایت می‌شود. .6 مشتری اطلاعات پرداخت را وارد می‌کند. .7 سیستم پرداخت را تأیید و پرداش می‌کند. .8 سیستم سفارش را ثبت کرده و جزئیات سفارش را نمایش می‌دهد. 	روند اصلی
<ul style="list-style-type: none"> ● سفارش ثبت شده و مشتری تأییدیه‌ی پرداخت را دریافت می‌کند. 	پس‌نیازها
<ul style="list-style-type: none"> ● پرداخت موفق نباشد. 	روند جایگزین

پرداخت: پرداخت موفق نباشد.	روند جایگزین
1.1) شناسه	
سیستم به مشتری اطلاع می‌دهد و امکان تلاش مجدد را فراهم می‌کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری سبد خرید داشته باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. روند جایگزین بعد از مرحله‌ی 7 روند اصلی شروع می‌شود. 2. سیستم پرداخت را از طریق درگاه پرداخت پردازش می‌کند. 3. اگر پرداخت ناموفق باشد، سیستم پیامی مبنی بر عدم موفقیت پرداخت نمایش می‌دهد. 4. مشتری می‌تواند مجدداً تلاش کند. 	روند جایگزین
	پس‌نیازها

مشاهده‌ی وضعیت فعلی سفارش	مورد کاربرد
۱۲	شناسه
مشتری می‌تواند وضعیت فعلی سفارش خود را مشاهده کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. مشتری سفارش فعال داشته باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. مشتری گزینه مشاهده وضعیت سفارش را انتخاب می‌کند. 2. سیستم سفارش فعال مشتری را نمایش می‌دهد. 3. سیستم وضعیت فعلی سفارش انتخاب شده را نمایش می‌دهد. 	روند اصلی
<ul style="list-style-type: none"> مشتری وضعیت فعلی سفارش خود را مشاهده می‌کند. 	پس‌نیازها
	روند جایگزین

مورد کاربرد	امکان ویرایش اطلاعات
شناسه	۱۳
توضیح مختصر	مشتری می‌تواند اطلاعات حساب کاربری خود را ویرایش کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	● مشتری وارد سیستم شده باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری گزینه ویرایش اطلاعات حساب کاربری را انتخاب می‌کند. 2. سیستم فرم اطلاعات حساب کاربری فعلی را نمایش می‌دهد. 3. مشتری اطلاعات مورد نظر خود را ویرایش می‌کند. 4. مشتری تغییرات را تأیید می‌کند. 5. تا زمانی که اطلاعات معتبر نباشد: <ol style="list-style-type: none"> a. سیستم پیام خطای نمایش می‌دهد. 6. سیستم اطلاعات ویرایش شده را ذخیره می‌کند. 7. سیستم پیامی مبنی بر موفقیت ویرایش اطلاعات نمایش می‌دهد.
پس‌نیازها	● اطلاعات حساب کاربری مشتری به روزرسانی می‌شود.
روند جایگزین	

مورد کاربرد	مشاهده‌ی امتیاز هر رستوران
شناسه	۱۴
توضیح مختصر	مشتری می‌تواند امتیازهای داده شده به هر رستوران را مشاهده کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری رستورانی را انتخاب می‌کند. 2. سیستم میانگین امتیازهای داده شده به رستوران را نمایش می‌دهد.
پس‌نیازها	<ul style="list-style-type: none"> ● مشتری میانگین امتیازهای داده شده به رستوران را مشاهده می‌کند.
روند جایگزین	<ul style="list-style-type: none"> ● رستوران هیچ امتیازی نداشته باشد.

روند جایگزین	مشاهده‌ی امتیاز هر رستوران: رستوران هیچ امتیازی نداشته باشد
شناسه	1.1۴
توضیح مختصر	سیستم به مشتری اطلاع می‌دهد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	● مشتری وارد سیستم شده باشد.
روند جایگزین	1. روند جایگزین بعد از مرحله‌ی ۲ روند اصلی شروع می‌شود. 2. سیستم بررسی می‌کند که آیا امتیازی برای رستوران موجود است یا خیر. 3. اگر هیچ امتیازی موجود نباشد، سیستم پیامی مبنی بر عدم وجود امتیاز نمایش می‌دهد.
پس‌نیازها	

مورد کاربرد	امتیازدهی به سبد خرید
شناسه	۱۵
توضیح مختصر	مشتری می‌تواند به سبد خرید خود امتیاز دهد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری سفارشی را تکمیل کرده باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری گزینه مشاهده‌ی تاریخچه‌ی سفارش‌های خود را انتخاب می‌کند. 2. مشتری گزینه امتیازدهی به آن سفارش را انتخاب می‌کند. 3. سیستم بخش امتیازدهی را نمایش می‌دهد. 4. مشتری امتیاز خود را وارد و تأیید می‌کند. 5. سیستم امتیاز داده شده را ذخیره می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● امتیاز داده شده به سبد خرید ذخیره می‌شود.
روند جایگزین	<ul style="list-style-type: none"> ● مشتری قبلًا به سفارش امتیاز داده باشد.

امتیازدهی به سبد خرید: مشتری قبلاً به سفارش امتیاز داده باشد.	روند جایگزین
1.15	شناسه
اگر مشتری قبلاً به سفارشی امتیاز داده باشد، سیستم پیامی مبنی بر امتیازدهی قبلی نمایش می‌دهد.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری سفارشی را تکمیل کرده باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. روند جایگزین بعد از مرحله‌ی ۳ روند اصلی شروع می‌شود. 2. سیستم بررسی می‌کند که آیا مشتری قبلاً به سفارشی امتیاز داده است یا خیر. 3. اگر مشتری قبلاً امتیاز داده باشد، سیستم پیامی مبنی بر امتیازدهی قبلی نمایش می‌دهد. 	روند جایگزین
	پس‌نیازها

مشاهده‌ی تاریخچه‌ی سفارش‌ها	مورد کاربرد
۱۶	شناسه
مشتری می‌تواند تاریخچه‌ی سفارش‌های قبلی خود را مشاهده کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● مشتری وارد سیستم شده باشد.	پیش‌نیازها
1. مشتری گزینه مشاهده‌ی تاریخچه‌ی سفارش‌ها را انتخاب می‌کند. 2. سیستم لیست سفارش‌های قبلی مشتری را نمایش می‌دهد.	روند اصلی
● مشتری تاریخچه‌ی سفارش‌های قبلی خود را مشاهده می‌کند.	پس‌نیازها
● مشتری هیچ سفارش قبلی نداشته باشد.	روند جایگزین

روند جایگزین	مشاهده‌ی تاریخچه‌ی سفارش‌ها؛ مشتری هیچ سفارش قبلی نداشته باشد.
شناسه	1.16
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود سفارش‌های قبلی نمایش می‌دهد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	● مشتری وارد سیستم شده باشد.
روند جایگزین	1. روند جایگزین بعد از مرحله‌ی ۲ روند اصلی شروع می‌شود. 2. سیستم بررسی می‌کند که آیا مشتری سفارش قبلی دارد یا خیر. 3. اگر مشتری هیچ سفارش قبلی نداشته باشد، سیستم پیامی مبنی بر عدم وجود سفارش‌های قبلی نمایش می‌دهد.
پس‌نیازها	پیام عدم وجود سفارش‌های قبلی نمایش داده می‌شود.

مشاهده‌ی نظرات کاربران هر رستوران	مورد کاربرد
۱۷	شناسه
مشتری می‌تواند نظرات کاربران دیگر را درباره رستوران‌ها مشاهده کند.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● مشتری وارد سیستم شده باشد.	پیش‌نیازها
1. مشتری رستورانی را انتخاب می‌کند. 2. مشتری گزینه‌ی مشاهده‌ی نظرات کاربران را انتخاب می‌کند. 3. سیستم نظرات کاربران درباره رستوران را نمایش می‌دهد.	روند اصلی
● مشتری نظرات کاربران درباره رستوران را مشاهده می‌کند.	پس‌نیازها
● رستوران هیچ نظری نداشته باشد.	روند جایگزین

روند جایگزین	مشاهدهی نظرات کاربران هر رستوران: رستوران هیچ نظری نداشته باشد.
شناسه	1.17
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود نظرات نمایش می‌دهد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	● مشتری وارد سیستم شده باشد.
روند جایگزین	1. روند جایگزین بعد از مرحله‌ی ۳ روند اصلی شروع می‌شود. 2. سیستم بررسی می‌کند که آیا رستوران نظری دارد یا خیر. 3. اگر رستوران هیچ نظری نداشته باشد، سیستم پیامی مبنی بر عدم وجود نظرات نمایش می‌دهد.
پس‌نیازها	پیام عدم وجود نظرات نمایش داده می‌شود.

مورد کاربرد	مشاهده‌ی توضیحات غذای یک رستوران
شناسه	۱۸
توضیح مختصر	مشتری می‌تواند برای هر غذای رستوران، توضیحات آن غذا را مشاهده کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	مشتری وارد سیستم شده باشد.
روند اصلی	1. مشتری رستورانی را انتخاب می‌کند. 2. منو به مشتری نمایش داده می‌شود. 3. بخشی از توضیحات در منو به مشتری نمایش داده می‌شود. 4. مشتری یک غذا را انتخاب می‌کند. 5. توضیحات کامل غذا نمایش داده می‌شود.
پس‌نیازها	مشتری توضیحات غذای یک رستوران را مشاهده می‌کند.
روند جایگزین	غذا هیچ توضیحی نداشته باشد.

روند جایگزین	مشاهده‌ی توضیحات غذای یک رستوران: غذا هیچ توضیحی نداشته باشد.
شناسه	1.18
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود توضیحات نمایش می‌دهد.
کنшگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	● مشتری وارد سیستم شده باشد.
روند جایگزین	1. روند جایگزین بعد از مرحله‌ی ۵ روند اصلی شروع می‌شود. 2. سیستم بررسی می‌کند که آیا غذای انتخابی توضیحی دارد یا خیر. 3. اگر غذای انتخابی هیچ توضیحی نداشته باشد، سیستم پیامی مبنی بر عدم وجود توضیحات نمایش می‌دهد.
پس‌نیازها	پیام عدم وجود توضیحات نمایش داده می‌شود.

مورد کاربرد	نظردهی به سفارش
شناسه	۱۹
توضیح مختصر	مشتری می‌تواند نظر خود را راجع به سبد خرید خود از رستوران ثبت کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری سفارش تکمیل شده‌ای داشته باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری گزینه‌ی مشاهده‌ی تاریخچه‌ی سفارش‌های خود را انتخاب می‌کند. 2. مشتری یک سفارش را انتخاب می‌کند. 3. سیستم فرم نظردهی را نمایش می‌دهد. 4. مشتری نظر خود را وارد و تأیید می‌کند. 5. سیستم نظر داده شده را ذخیره می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● سیستم نظر داده شده را ذخیره می‌کند.
روند جایگزین	<ul style="list-style-type: none"> ● مشتری قبلًاً راجع به سفارش نظر داده باشد.

روند جایگزین	نظردهی به سفارش: مشتری قبلاً راجع به سفارش نظر داده باشد
شناسه	1.19
توضیح مختصر	سیستم پیامی مبنی بر نظردهی قبلی نمایش می‌دهد.
کنшگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری سفارش تکمیل شده‌ای داشته باشد.
روند جایگزین	<ol style="list-style-type: none"> 1. روند جایگزین بعد از مرحله‌ی ۴ روند اصلی شروع می‌شود. 2. سیستم بررسی می‌کند که آیا مشتری قبلاً راجع به این سفارش نظر داده است یا خیر. 3. اگر مشتری قبلاً نظر داده باشد، سیستم پیامی مبنی بر نظردهی قبلی نمایش می‌دهد.
پس‌نیازها	<ul style="list-style-type: none"> ● پیام نظردهی قبلی نمایش داده می‌شود.

مورد کاربرد	مشاهده‌ی محل فعلی سفارش
شناسه	۲۰
توضیح مختصر	مشتری می‌تواند محل فعلی سفارش خود را لحظه به لحظه رصد کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	پیک
بیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری سفارش تکمیل نشده‌ای داشته باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری سفارش فعلی خود را انتخاب می‌کند. 2. محل فعلی سفارش به او نشان داده می‌شود.
پس‌نیازها	<ul style="list-style-type: none"> ● مشتری محل فعلی سفارش را مشاهده می‌کند.
روند جایگزین	

مورد کاربرد	امتیازدهی به پیک
شناسه	۲۱
توضیح مختصر	مشتری می‌تواند به پیک‌هایی که تا به آن سفارش‌های او را آورده‌اند، امتیاز دهد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری سفارش تکمیل شده‌ای داشته باشد.
روند اصلی	<ol style="list-style-type: none"> 1. مشتری گزینه‌ی مشاهده‌ی تاریخچه‌ی سفارش‌های خود را انتخاب می‌کند. 2. مشتری گزینه‌ی امتیازدهی به پیک را انتخاب می‌کند. 3. سیستم بخش امتیازدهی را نمایش می‌دهد. 4. مشتری امتیاز خود را وارد و تأیید می‌کند. 5. سیستم امتیاز داده شده را ذخیره می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● امتیاز داده شده به پیک ذخیره می‌شود.
روند جایگزین	<ul style="list-style-type: none"> ● مشتری قبلاً به پیک در آن سفارش امتیاز داده باشد.

امتیازدهی به پیک: مشتری قبلاً به پیک در آن سفارش امتیاز داده باشد	روند جایگزین
۱.۲۱	شناسه
سیستم پیامی مبنی بر امتیازدهی قبلی نمایش می‌دهد.	توضیح مختصر
مشتری	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● مشتری گزینه‌ی امتیازدهی به پیک را انتخاب می‌کند.	پیش‌نیازها
۱. این روند جایگزین بعد از مرحله‌ی ۳ روند اصلی شروع می‌شود. ۲. سیستم پیامی مبنی بر امتیازدهی قبلی نمایش می‌دهد.	روند جایگزین
	پس‌نیازها

زیرسیستم مدیریت تهیه‌کنندگاه غذا

دیدن لیست سفارشات	مورد کاربرد
۱۲	شناسه
رستوران می‌تواند لیست سفارش‌های مشتریان خود را مشاهده کند.	توضیح مختصر
رستوران	کنشگر(های) اولیه
	کنشگر(های) ثانویه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
1. رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب می‌کند. 2. سیستم لیست سفارش‌های مشتریان از آن رستوران را نشان می‌دهد.	رونده اصلی
• رستوران لیست سفارش‌های مشتریان خود را مشاهده می‌کند.	پس‌نیازها
• رستوران هیچ سفارشی نداشته باشد.	رونده جایگزین

روند جایگزین	دیدن لیست سفارشات: رستوران هیچ سفارشی نداشته باشد
شناسه	۱.۲۲
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> • رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب کند.
روند جایگزین	<ol style="list-style-type: none"> 1. این روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. 2. سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
پس‌نیازها	

دیدن لیست تاریخچه‌ی سفارشات	مورد کاربرد
۲۳	شناسه
رستوران می‌تواند لیست تاریخچه‌ی سفارش‌های خود را مشاهده کند.	توضیح مختصر
رستوران	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● رستوران وارد سیستم شده باشد.	پیش‌نیازها
1. رستوران گزینه‌ی مشاهده‌ی لیست تاریخچه‌ی سفارش‌ها را انتخاب می‌کند. 2. سیستم لیست تاریخچه‌ی سفارش‌های آن رستوران را نشان می‌دهد.	روند اصلی
● رستوران لیست تاریخچه‌ی سفارش‌های خود را مشاهده می‌کند.	پس‌نیازها
● رستوران هیچ سفارشی نداشته باشد.	روند جایگزین

روند جایگزین	دیدن لیست تاریخچه‌ی سفارشات: رستوران هیچ سفارشی نداشته باشد
شناسه	۱.۲۳
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود هیچ تاریخچه‌ی سفارشی نمایش می‌دهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	• رستوران گزینه‌ی مشاهده‌ی لیست تاریخچه‌ی سفارش‌ها را انتخاب کند.
روند جایگزین	۱. این روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. ۲. سیستم پیامی مبنی بر عدم وجود هیچ تاریخچه‌ی سفارشی نمایش می‌دهد.
پس‌نیازها	

مورد کاربرد	تغییر دادن وضعیت هر سفارش
شناسه	۲۴
توضیح مختصر	rstوران می‌تواند وضعیت هر سفارش را از «آماده‌ی ارسال» به «در حال ارسال» تغییر دهد.
کنشگر(های) اولیه	rstوران
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> rstوران وارد سیستم شده باشد.
روند اصلی	<ol style="list-style-type: none"> rstوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب می‌کند. سیستم لیست سفارش‌های مشتریان از rstوران را نشان می‌دهد. rstوران یک سفارش را انتخاب می‌کند. سیستم گزینه‌ی «تغییر وضعیت سفارش از آماده‌ی ارسال به در حال ارسال» را به rstوران نشان می‌دهد. rstوران وضعیت سفارش را تغییر می‌دهد.
پس‌نیازها	<ul style="list-style-type: none"> وضعیت سفارش تبدیل به «در حال ارسال» شده باشد.
روند جایگزین	<ul style="list-style-type: none"> rstوران هیچ سفارشی نداشته باشد

روند جایگزین	تغییر دادن وضعیت هر سفارش: رستوران هیچ سفارشی نداشته باشد
شناسه	۱.۲۴
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> • رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب کند.
روند جایگزین	<ol style="list-style-type: none"> 1. این روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. 2. سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
پس‌نیازها	

مورد کاربرد	چاپ فاکتور
شناسه	۲۵
توضیح مختصر	رستوران می‌تواند فاکتور هر سفارش را چاپ کند.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	● رستوران وارد سیستم شده باشد.
روند اصلی	1. رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب می‌کند. 2. سیستم لیست سفارش‌های مشتریان از آن رستوران را نشان می‌دهد. 3. رستوران یک سفارش را انتخاب می‌کند. 4. سیستم گزینه‌ی «چاپ فاکتور» را به رستوران نشان می‌دهد. 5. رستوران گزینه‌ی چاپ فاکتور را انتخاب می‌کند. 6. سیستم فاکتور را چاپ می‌کند.
پس‌نیازها	● فاکتور سفارش چاپ شده باشد.
روند جایگزین	● رستوران هیچ سفارشی نداشته باشد.

چاپ فاکتور: رستوران هیچ سفارشی نداشته باشد	روند جایگزین
1.۲۵	شناسه
سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.	توضیح مختصر
رستوران	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب کند.	پیش‌نیازها
1. این روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. 2. سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.	روند جایگزین
	پس‌نیازها

مورد کاربرد	درخواست پیک
شناسه	۲۶
توضیح مختصر	رستوران می‌تواند برای هر سفارش درخواست پیک بدهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	پیک
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران وارد سیستم شده باشد.
روند اصلی	<ol style="list-style-type: none"> 1. رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب می‌کند. 2. سیستم لیست سفارش‌های مشتریان از آن رستوران را نشان می‌دهد. 3. رستوران یک سفارش را انتخاب می‌کند. 4. رستوران گزینه‌ی «تغییر وضعیت سفارش از آماده‌ی ارسال به در حال ارسال» را انتخاب می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● وضعیت سفارش به «در حال ارسال» تغییر کرده باشد.
روند جایگزین	<ul style="list-style-type: none"> ● رستوران هیچ سفارشی نداشته باشد.

روند جایگزین	درخواست پیک: رستوران هیچ سفارشی نداشته باشد
شناسه	۱.۱۶
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	• رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب کند.
روند جایگزین	۱. این روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. ۲. سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
پس‌نیازها	

تعیین خودکار یک پیک برای هر سفارش	مورد کاربرد
۲۷	شناسه
برای سفارش یک پیک به صورت خودکار تخصیص داده می‌شود.	توضیح مختصر
رستوران	کنشگر(های) اولیه
پیک	کنشگر(های) ثانویه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
1. رستوران گزینه‌ی «تغییر وضعیت سفارش از آماده‌ی ارسال به در حال ارسال» را انتخاب می‌کند. 2. سیستم از بین پیک‌های فعال، دنبال یک پیک آزاد می‌گردد. 3. سیستم از بین پیک‌های پیدا شده یک پیک را به صورت خودکار به آن سفارش تخصیص می‌دهد. 4. سیستم به پیک انتخاب شده اطلاعات سفارش را ارسال می‌کند. 5. وضعیت پیک به «مشغول» تغییر پیدا می‌کند. 6. سیستم به مشتری اطلاع می‌دهد که پیک انتخاب شده و سفارش در حال آماده‌سازی برای تحویل است.	روند اصلی
• برای سفارش ثبت شده یک پیک تخصیص داده شود.	پس‌نیازها
• هیچ پیک فعالی نداشته باشیم.	روند جایگزین

روند جایگزین	تعیین خودکار یک پیک برای هر سفارش: هیچ پیک فعالی نداشته باشیم
شناسه	۱.۳۷
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود پیک نمایش می‌دهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	پیک
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران گزینه‌ی «تغییر وضعیت سفارش از آماده‌ی ارسال به در حال ارسال» را انتخاب می‌کند.
روند جایگزین	<ol style="list-style-type: none"> 1. این روند جایگزین بعد از مرحله‌ی ۲ روند اصلی شروع می‌شود. 2. سیستم پیامی مبنی بر عدم وجود پیک نمایش می‌دهد.
پس‌نیازها	

مشاهده‌ی پیک‌های حمل‌کننده‌ی سفارشات آن رستوران	مورد کاربرد
۲۸	شناسه
در این مورد رستوران می‌تواند لیست پیک‌هایی که سفارشات رستوران را حمل می‌کنند، مشاهده کند.	توضیح مختصر
رستوران	کنیشگر(های) اولیه
	کنیشگر(های) ثانویه
<ul style="list-style-type: none"> ● رستوران وارد سیستم شده باشد. ● یک یا چند سفارش باید «در حال ارسال» باشند. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب می‌کند. 2. سیستم لیست سفارش‌های مشتریان از آن رستوران را نشان می‌دهد. 3. رستوران یک سفارش را انتخاب می‌کند. 4. سیستم پیکی که سفارش رستوران را حمل می‌کند نمایش می‌دهد. 	رونده اصلی
<ul style="list-style-type: none"> ● سیستم پیک در حال حمل سفارش را نمایش می‌دهد. 	پس‌نیازها
<ul style="list-style-type: none"> ● رستوران هیچ سفارشی نداشته باشد. 	رونده جایگزین

روند جایگزین شناسه	مشاهده‌ی پیکه‌های حمل‌کننده‌ی سفارشات آن رستوران: رستوران هیچ سفارشی نداشته باشد
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران گزینه‌ی مشاهده‌ی لیست سفارش‌ها را انتخاب کند.
روند جایگزین	<ol style="list-style-type: none"> 1. این روند جایگزین بعد از مرحله‌ی ۱ روند اصلی شروع می‌شود. 2. سیستم پیامی مبنی بر عدم وجود سفارش نمایش می‌دهد.
پس‌نیازها	

تعیین هوشمند یک پیک برای هر سفارش	مورد کاربرد
۲۹	شناسه
در این مورد کاربرد، سیستم به صورت هوشمند و خودکار پیک مناسبی را برای هر سفارش تعیین می‌کند.	توضیح مختصر
رستوران	کنشگر(های) اولیه
پیک	کنشگر(های) ثانویه
● رستوران وارد سیستم شده باشد.	پیش‌نیازها
1. رستوران گزینه‌ی «تغییر وضعیت سفارش از آماده‌ی ارسال به در حال ارسال» را انتخاب می‌کند. 2. سیستم لیستی از پیک‌های موجود و در دسترس را تهیه می‌کند. 3. سیستم با استفاده از الگوریتم‌های هوشمند، پیک‌های موجود را بر اساس معیارهایی مانند نزدیکی به رستوران، زمان تخمینی تحویل، وضعیت ترافیک و ظرفیت حمل ارزیابی می‌کند. 4. سیستم به صورت هوشمند پیک مناسبی را از بین پیک‌های ارزیابی شده انتخاب می‌کند. 5. سیستم به پیک انتخاب شده اطلاعات سفارش را ارسال می‌کند. 6. وضعیت پیک به «مشغول» تغییر پیدا می‌کند. 7. سیستم به مشتری اطلاع می‌دهد که پیک انتخاب شده و سفارش در حال آماده‌سازی برای تحویل است.	روند اصلی
● سیستم یک پیک را به صورت هوشمند برای تحویل سفارش انتخاب می‌کند.	پس‌نیازها
● هیچ پیک فعال و آزادی نداشته باشیم.	روند جایگزین

روند جایگزین	تعیین هوشمند یک پیک برای هر سفارش: هیچ پیک فعال و آزادی نداشته باشیم
شناسه	۱.۲۹
توضیح مختصر	سیستم پیامی مبنی بر عدم وجود پیک نمایش می‌دهد.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	پیک
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران گزینه‌ی «تغییر وضعیت سفارش از آماده‌ی ارسال به در حال ارسال» را انتخاب می‌کند.
روند جایگزین	<ol style="list-style-type: none"> 1. این روند جایگزین بعد از مرحله‌ی ۲ روند اصلی شروع می‌شود. 2. سیستم پیامی مبنی بر عدم وجود پیک نمایش می‌دهد.
پس‌نیازها	

افزودن غذا به منو	مورد کاربرد
۳۵	شناسه
در این مورد کاربرد، رستوران غذایی را به منوی خود اضافه می‌کند.	توضیح مختصر
رستوران	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● رستوران وارد سیستم شده باشد.	پیش‌نیازها
1. رستوران گزینه‌ی «افزودن غذا به منو» را انتخاب می‌کند. 2. رستوران اطلاعات غذا مانند اسم، قیمت، عکس، و توضیحات را وارد می‌کند. 3. رستوران گزینه‌ی «ثبت اطلاعات» را انتخاب می‌کند.	روند اصلی
● غذا به منوی رستوران افزوده می‌شود.	پس‌نیازها
	روند جایگزین

مورد کاربرد	حذف غذا از منو
شناسه	۳۱
توضیح مختصر	در این مورد کاربرد، رستوران غذایی را از منوی خود حذف می‌کند.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران وارد سیستم شده باشد. ● در منوی رستوران حداقل یک غذا وجود داشته باشد.
روند اصلی	<ol style="list-style-type: none"> 1. رستوران گزینه‌ی «مشاهده منو» را انتخاب می‌کند. 2. رستوران غذایی را که می‌خواهد حذف کند را انتخاب می‌کند. 3. رستوران گزینه‌ی «حذف غذا» را انتخاب می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● غذا از منوی رستوران حذف می‌شود.
روند جایگزین	

تعیین موجودی هر غذا	مورد کاربرد
۳۲	شناسه
در این مورد کاربرد، رستوران برای هر غذای خود موجودی تعیین می‌کند.	توضیح مختصر
رستوران	کنشگر(های) اولیه
	کنشگر(های) ثانویه
<ul style="list-style-type: none"> ● رستوران وارد سیستم شده باشد. ● در منوی رستوران حداقل یک غذا وجود داشته باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. رستوران گزینه‌ی «مشاهده منو» را انتخاب می‌کند. 2. رستوران غذایی را که می‌خواهد برای آن موجودی تعیین کند را انتخاب می‌کند. 3. رستوران موجودی را تعیین می‌کند. 4. رستوران گزینه‌ی «ثبت موجودی» را انتخاب می‌کند. 	روند اصلی
<ul style="list-style-type: none"> ● موجودی غذا تعیین می‌شود. 	پس‌نیازها
	روند جایگزین

مورد کاربرد	ویرایش غذا در منو
شناسه	۳۳
توضیح مختصر	در این مورد کاربرد، رستوران اطلاعات غذا را ویرایش می‌کند.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران وارد سیستم شده باشد. ● در منوی رستوران حداقل یک غذا وجود داشته باشد.
روند اصلی	<ol style="list-style-type: none"> 1. رستوران گزینه‌ی «مشاهده منو» را انتخاب می‌کند. 2. رستوران غذایی را که می‌خواهد ویرایش کند را انتخاب می‌کند. 3. رستوران اطلاعات جدید را وارد می‌کند. 4. رستوران گزینه‌ی «ثبت اطلاعات» را انتخاب می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● اطلاعات غذا ویرایش می‌شود.
روند جایگزین	

مورد کاربرد	افرودن دسته‌بندی برای هر غذا
شناسه	۳۴
توضیح مختصر	در این مورد کاربرد، رستوران برای هر غذا دسته‌بندی مشخص می‌کند.
کنشگر(های) اولیه	رستوران
کنشگر(های) ثانویه	
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران وارد سیستم شده باشد. ● در منوی رستوران حداقل یک غذا وجود داشته باشد.
رونده اصلی	<ol style="list-style-type: none"> 1. رستوران گزینه‌ی «مشاهده منو» را انتخاب می‌کند. 2. رستوران غذایی را که می‌خواهد برای آن دسته‌بندی مشخص کند را انتخاب می‌کند. 3. رستوران دسته‌بندی را مشخص می‌کند. 4. رستوران گزینه‌ی «ثبت اطلاعات» را انتخاب می‌کند.
پس‌نیازها	<ul style="list-style-type: none"> ● دسته‌بندی به غذا افزوده می‌شود.
رونده جایگزین	

زیرسیستم مدیریت پیک‌ها

مورد کاربرد	تغییر دادن وضعیت هر سفارش
شناسه	۳۵
توضیح مختصر	در این مورد کاربرد، پیک می‌تواند وضعیت سفارشات تخصیص داده شده به خود را تغییر دهد.
کنشگر(های) اولیه	پیک
کنشگر(های) ثانویه	مشتری، رستوران
پیش‌نیازها	<ul style="list-style-type: none"> ● پیک وارد سیستم شده باشد. ● سفارشی باید به پیک تخصیص داده شده باشد.
رونده اصلی	<ol style="list-style-type: none"> 1. پیک، سفارش «در حال ارسال» خود را مشاهده می‌کند. 2. پیک، وضعیت سفارش را به «تحویل شده» تغییر می‌دهد. 3. سیستم به مشتری و رستوران وضعیت جدید سفارش را اطلاع می‌دهد.
پس‌نیازها	<ul style="list-style-type: none"> ● مشتری و رستوران بتوانند وضعیت جدید سفارش را مشاهده کنند.
رونده جایگزین	

ارسال موقعیت به صورت زنده برای مشتری	مورد کاربرد
۳۶	شناسه
در این مورد کاربرد، پیکها می‌توانند موقعیت مکانی خود را به صورت زنده به مشتری ارسال کنند تا مشتری بتواند وضعیت و محل دقیق پیک را در هر لحظه مشاهده کند.	توضیح مختصر
پیک	کنشگر(های) اولیه
مشتری، رستوران	کنشگر(های) ثانویه
<ul style="list-style-type: none"> ● سفارشی باید به پیک تخصیص داده شده باشد. ● پیک باید اجازه اشتراک‌گذاری موقعیت مکانی خود را داده باشد. 	پیش‌نیازها
<ol style="list-style-type: none"> 1. به پیک سفارشی واگذار می‌شود. 2. پیک موقعیت مکانی زنده خود را به مشتری ارسال می‌کند. 	رونده اصلی
<ul style="list-style-type: none"> ● مشتری باید امکان پیگیری موقعیت مکانی زنده پیک را در هر لحظه داشته باشد. 	پس‌نیازها
<ul style="list-style-type: none"> ● پیک نتواند موقعیت مکانی خود را به اشتراک بگذارد (مثلًاً به دلیل مشکلات فنی یا نبود سیگنال GPS). 	رونده جایگزین

ارسال موقعیت به صورت زنده برای مشتری: پیک نتواند موقعیت مکانی خود را به اشتراک بگذارد.	روند جایگزین
۱.۳۶	شناسه
سیستم پیامی مبنی بر عدم توانایی به اشتراک گذاری موقعیت مکانی توسط پیک را به مشتری نمایش می‌دهد.	توضیح مختصر
پیک	کنشگر(های) اولیه
مشتری	کنشگر(های) ثانویه
• پیک قابلیت اشتراک گذاری موقعیت مکانی خود را فعال می‌کند.	پیش‌نیازها
۱. این روند جایگزین بعد از مرحله ۲ روند اصلی شروع می‌شود. ۲. سیستم پیامی مبنی بر عدم توانایی به اشتراک گذاری موقعیت مکانی، توسط پیک را به مشتری نمایش می‌دهد.	روند جایگزین
	پس‌نیازها

مورد کاربرد	ارسال موقعیت به صورت زنده برای رستوران
شناسه	۳۷
توضیح مختصر	در این مورد کاربرد، پیک می‌تواند موقعیت مکانی خود را به صورت زنده به رستوران ارسال کند تا رستوران بتواند وضعیت و محل دقیق پیک را در هر لحظه مشاهده کند.
کنشگر(های) اولیه	پیک
کنشگر(های) ثانویه	رستوران
پیش‌نیازها	<ul style="list-style-type: none"> ● سفارشی باید به پیک تخصیص داده شده باشد.
روند اصلی	<ol style="list-style-type: none"> 1. به پیک سفارشی واگذار می‌شود. 2. پیک قابلیت اشتراک‌گذاری موقعیت مکانی زنده خود را فعال می‌کند. 3. پیک موقعیت مکانی زنده خود را به رستوران ارسال می‌کند.
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران باید امکان پیگیری موقعیت مکانی زنده پیک را در هر لحظه داشته باشد.
روند جایگزین	<ul style="list-style-type: none"> ● در صورتی که پیک نتواند موقعیت مکانی خود را به اشتراک بگذارد (مثلًاً به دلیل مشکلات فنی یا نبود سیگنال GPS).

رسال موقعیت به صورت زنده برای رستوران: پیک نتواند موقعیت مکانی خود را به اشتراک بگذارد.	روند جایگزین
۱.۳۷	شناسه
سیستم پیامی مبنی بر عدم توانایی به اشتراک گذاری موقعیت مکانی توسط پیک را به رستوران نمایش می‌دهد.	توضیح مختصر
پیک	کنشگر(های) اولیه
رستوران	کنشگر(های) ثانویه
• پیک قابلیت اشتراک گذاری موقعیت مکانی خود را فعال می‌کند.	پیش‌نیازها
۱. این روند جایگزین بعد از مرحله‌ی ۲ روند اصلی شروع می‌شود. ۲. سیستم پیامی مبنی بر عدم توانایی به اشتراک گذاری موقعیت مکانی توسط پیک را به رستوران نمایش می‌دهد.	روند جایگزین
	پس‌نیازها

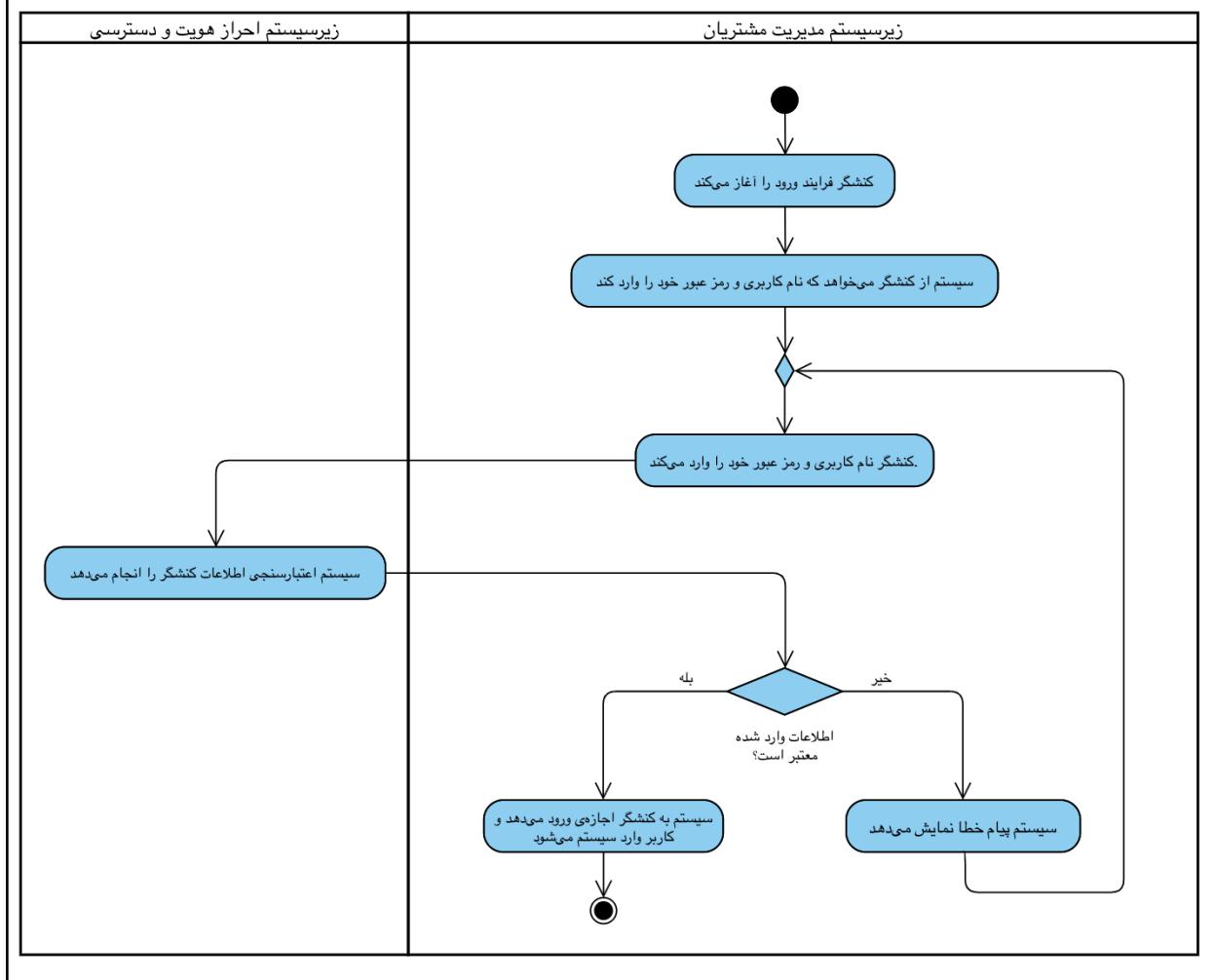
تغییر وضعیت بین فعال و غیرفعال	مورد کاربرد
۳۸	شناسه
پیک‌ها می‌توانند وضعیت حضور خود را به صورت فعال یا غیر فعال تنظیم کنند. این قابلیت به پیک‌ها اجازه می‌دهد که مشخص کند آیا در حال حاضر آماده دریافت سفارشات جدید هستند یا خیر.	توضیح مختصر
پیک	کنشگر(های) اولیه
	کنشگر(های) ثانویه
● پیک وارد سیستم شده است.	پیش‌نیازها
1. پیک وضعیت حضور خود را به «فعال» یا «غیر فعال» تغییر می‌دهد. 2. اگر پیک حضور خود را فعال کند. a. سیستم وضعیت پیک را به عنوان "آماده برای دریافت سفارشات" ثبت می‌کند. b. سیستم پیک را به لیست پیک‌های آماده برای دریافت سفارشات اضافه می‌کند. 3. در غیر این صورت a. سیستم وضعیت پیک را به عنوان "غیر فعال" ثبت می‌کند. b. سیستم پیک را از لیست پیک‌های آماده برای دریافت سفارشات خارج می‌کند.	رونده اصلی
● وضعیت فعال یا غیر فعال بودن پیک ویرایش شود.	پس‌نیازها
● در صورتی که پیک وضعیت حضور خود را به "غیر فعال" تغییر دهد، اما سفارشات جاری پیک هنوز تحويل داده نشده باشد، سیستم پیام خطایی مبنی بر عدم امکان تغییر وضعیت نمایش می‌دهد.	روندهایگزین

روند جایگزین	تغییر وضعیت بین فعال و غیرفعال: پیک سفارش جاری فعال داشته باشد.
شناسه	۱.۳۸
توضیح مختصر	اگر پیک سفارش جاری فعال داشته باشد، سیستم پیامی مبنی بر عدم امکان تغییر وضعیت نمایش می‌دهد.
کنشگر(های) اولیه	مشتری
کنشگر(های) ثانویه	
پیش‌نیازها	• پیک وضعیت حضور خود را به «غیر فعال» تغییر می‌دهد
روند جایگزین	۱. این روند جایگزین بعد از مرحله‌ی ۳ روند اصلی شروع می‌شود. ۲. سیستم پیام خطابی مبنی بر عدم امکان تغییر وضعیت نمایش می‌دهد.
پس‌نیازها	

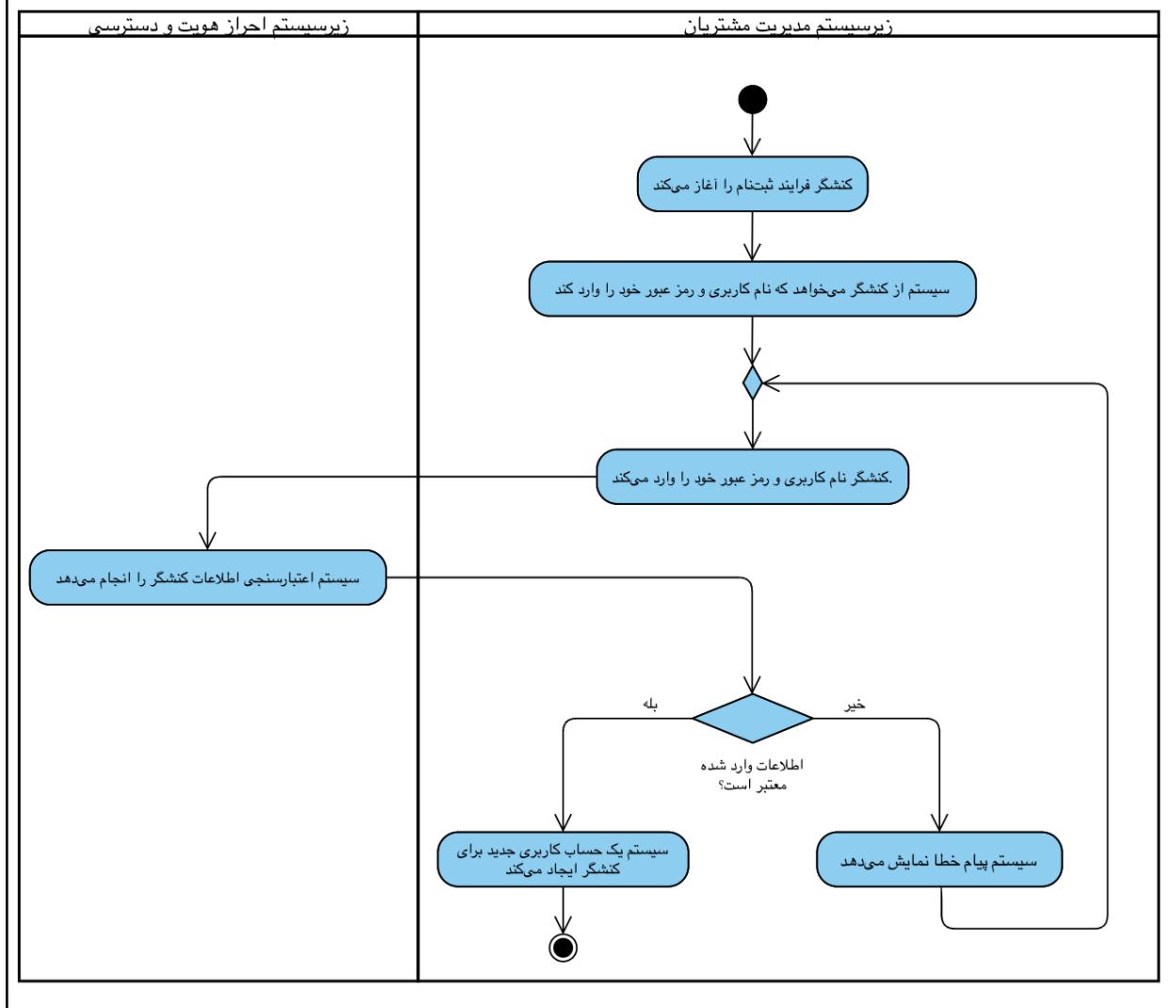
نمودارهای فعالیت

زیرسیستم احراز هویت و دسترسی

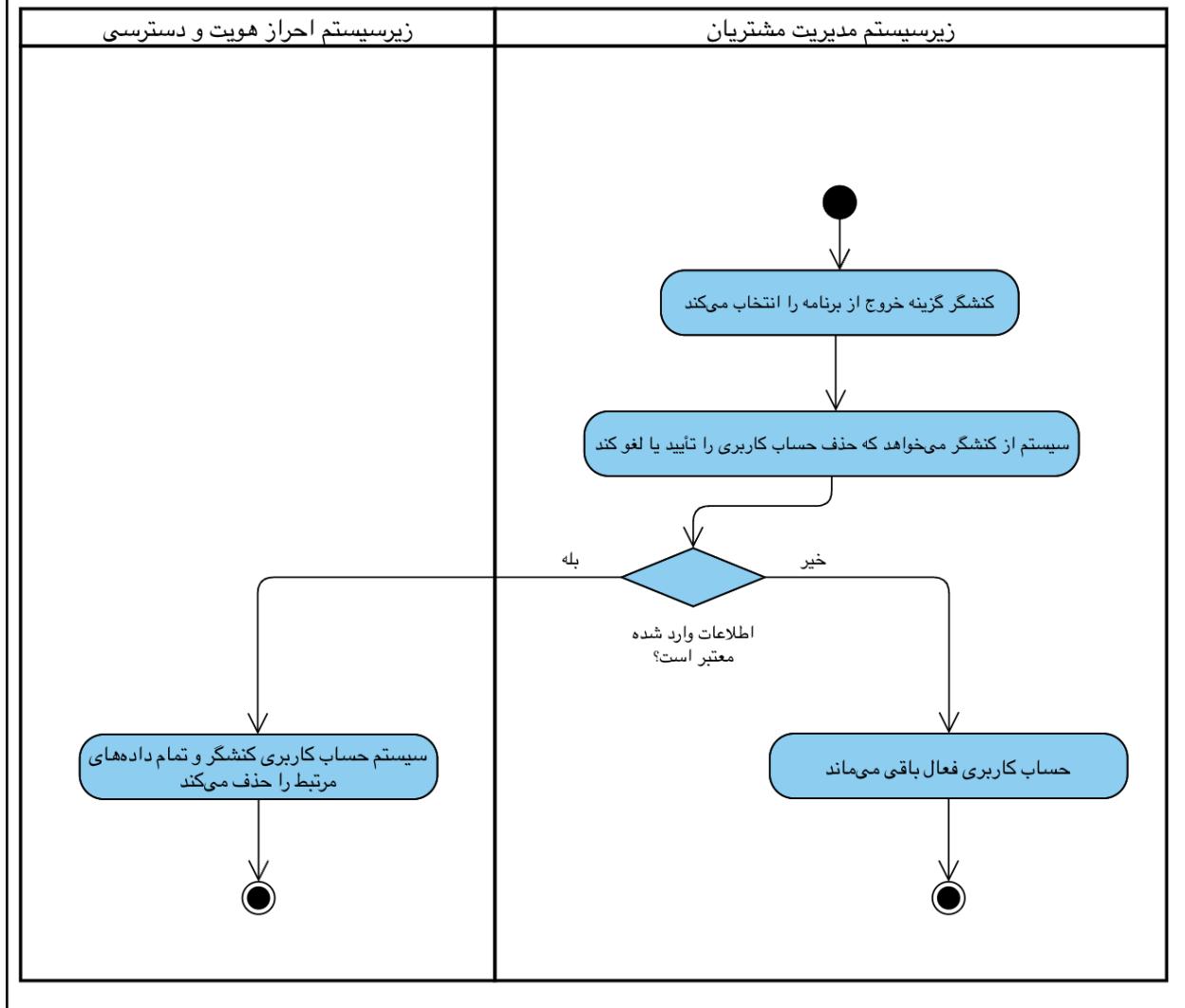
ورود	مورد کاربرد
۱	شناسه
<ul style="list-style-type: none"> کنشگر (مشتری، رستوران، پیک) یک حساب کاربری ثبت شده دارد. این کنشگر یک نام کاربری و رمز عبور معتبر دارد. 	پیش‌نیازها
<ul style="list-style-type: none"> کنشگر (مشتری، رستوران، پیک) وارد حساب کاربری خود شده است. 	پس‌نیازها



ثبتنام	مورد کاربرد
۲	شناسه
<ul style="list-style-type: none"> کنشگر اطلاعات ثبت نام معتبر (مانند ایمیل) دارد. 	پیش‌نیازها
<ul style="list-style-type: none"> سیستم یک حساب کاربری برای کنشگر ایجاد می‌کند. 	پس‌نیازها

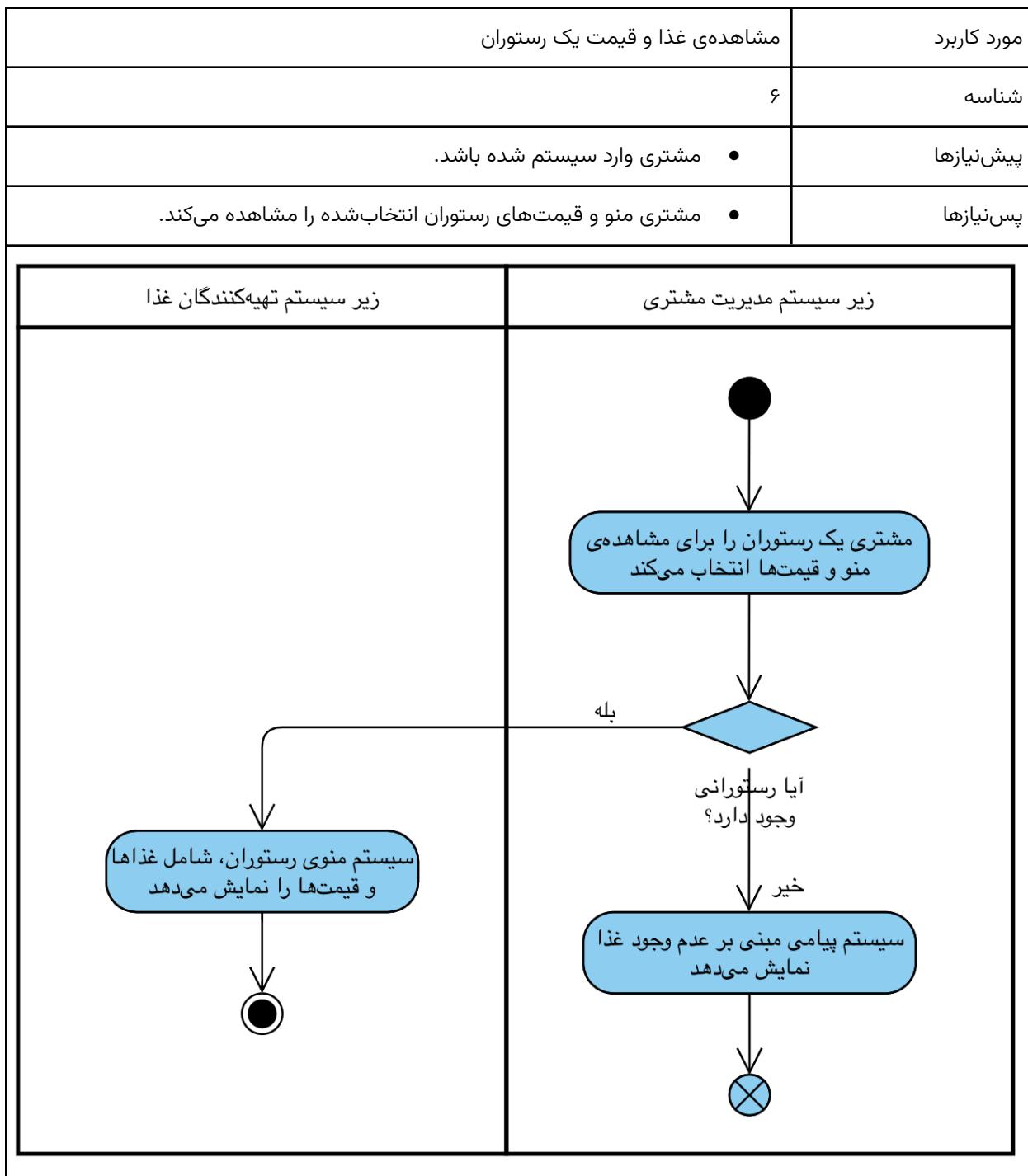


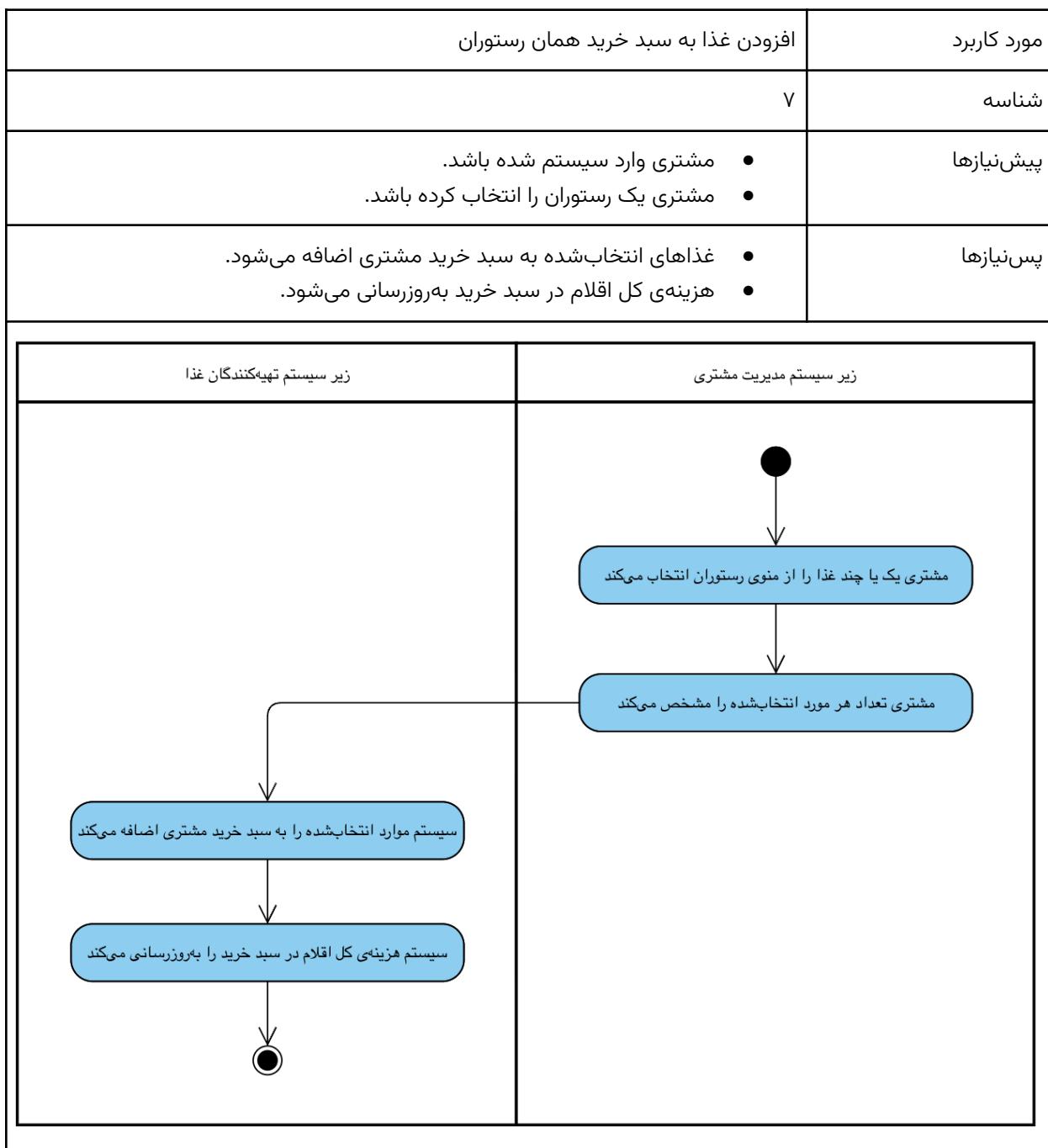
خروج	مورد کاربرد
۳	شناسه
• کنشگر (مشتری، رستوران، پیک) وارد سیستم شده باشد.	پیش‌نیازها
• کنشگر (مشتری، رستوران، پیک) از سیستم خارج شده است.	پس‌نیازها



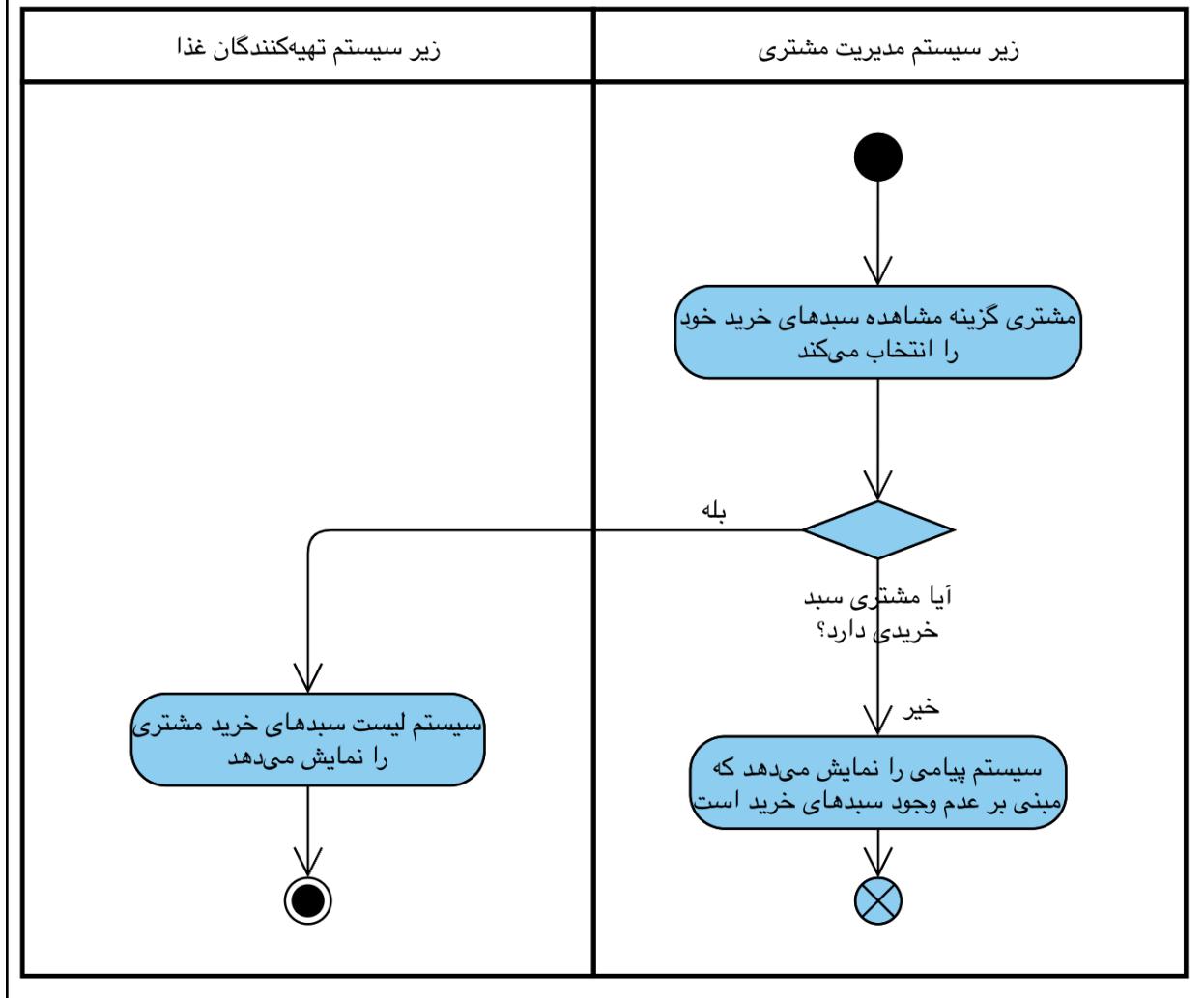
زیرسیستم مدیریت مشتریان

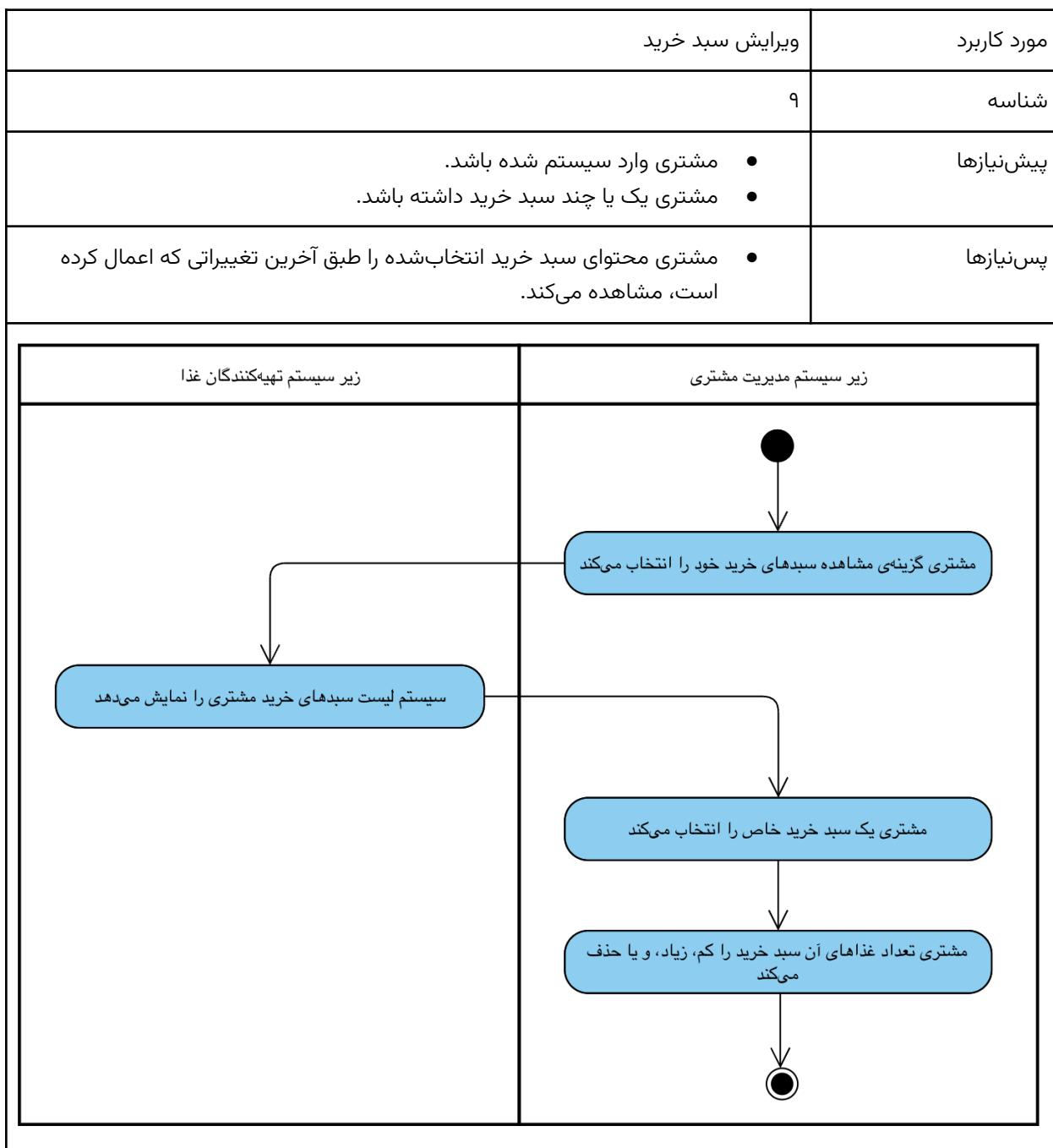
مشاهده‌ی لیست رستوران‌ها	مورد کاربرد
۵	شناسه
مشتری وارد سیستم شده باشد.	پیش‌نیازها
مشتری لیست رستوران‌های موجود را مشاهده می‌کند. مشتری می‌تواند به جزئیات بیشتری درباره یک رستوران خاص دسترسی پیدا کند.	پس‌نیازها
<pre> graph TD Start(()) --> Select{مشتری گزینه مشاهده لیست رستوران‌ها را انتخاب می‌کند} Select -- بله --> ShowList[سیستم لیست رستوران‌های موجود را نمایش می‌دهد] ShowList --> End(()) Select -- خیر --> NoShow[سیستم پیامی مبنی بر عدم وجود رستوران نمایش می‌دهد] NoShow --> End </pre>	



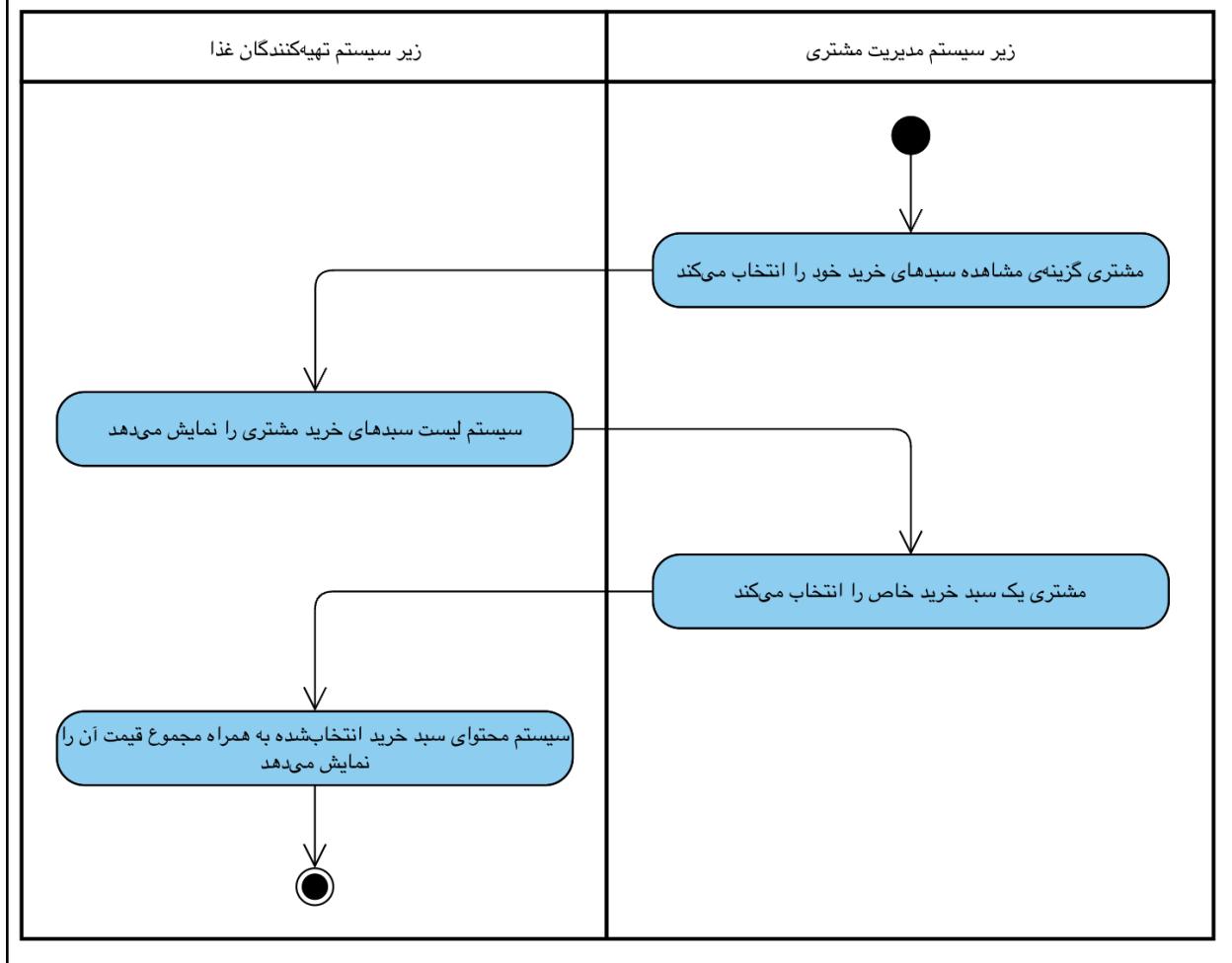


مشاهده‌ی سبدهای خرید	مورد کاربرد
•	شناسه
مشتری وارد سیستم شده باشد.	پیش‌نیازها
مشتری لیست سبدهای خرید خود را مشاهده می‌کند.	پس‌نیازها

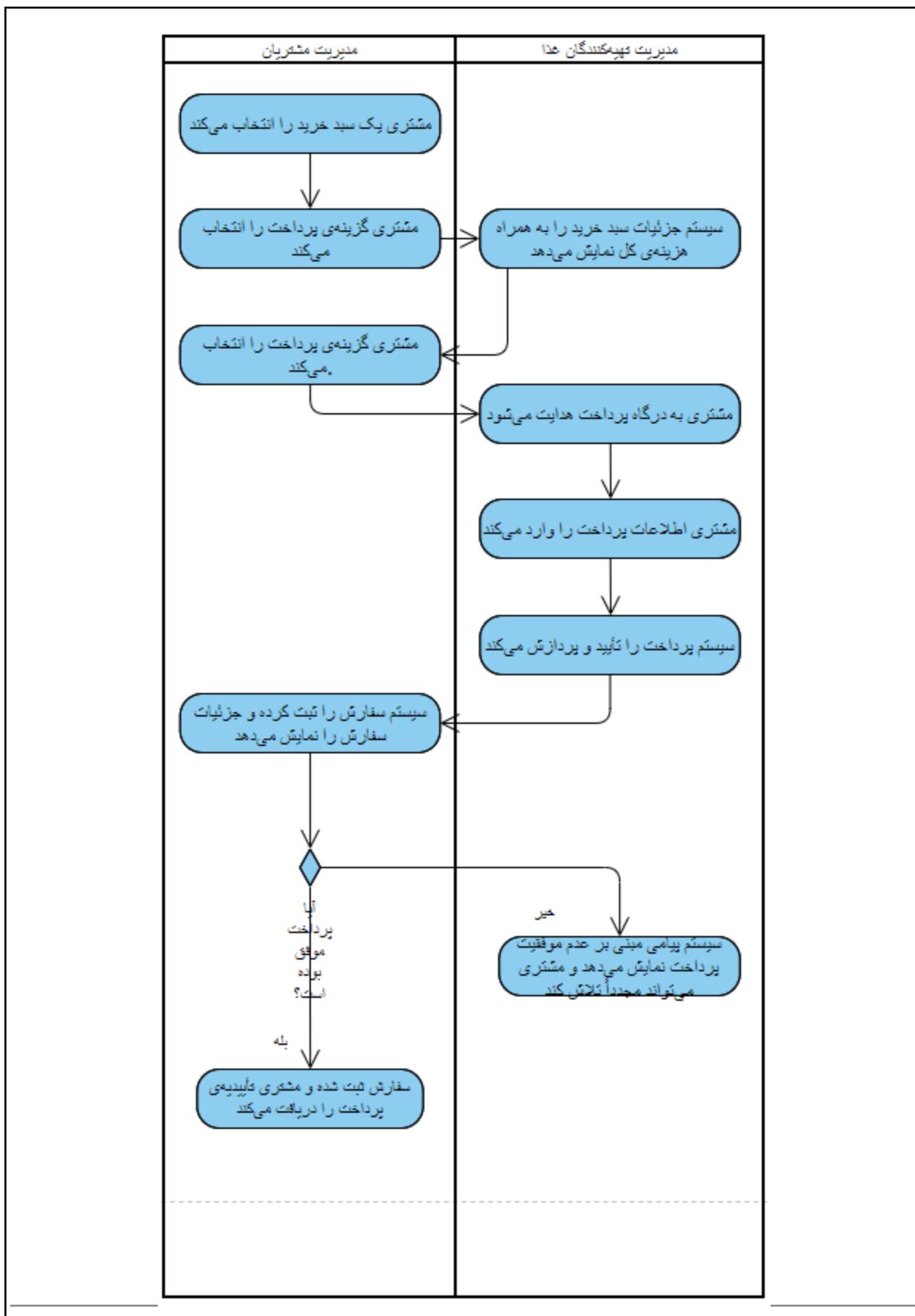




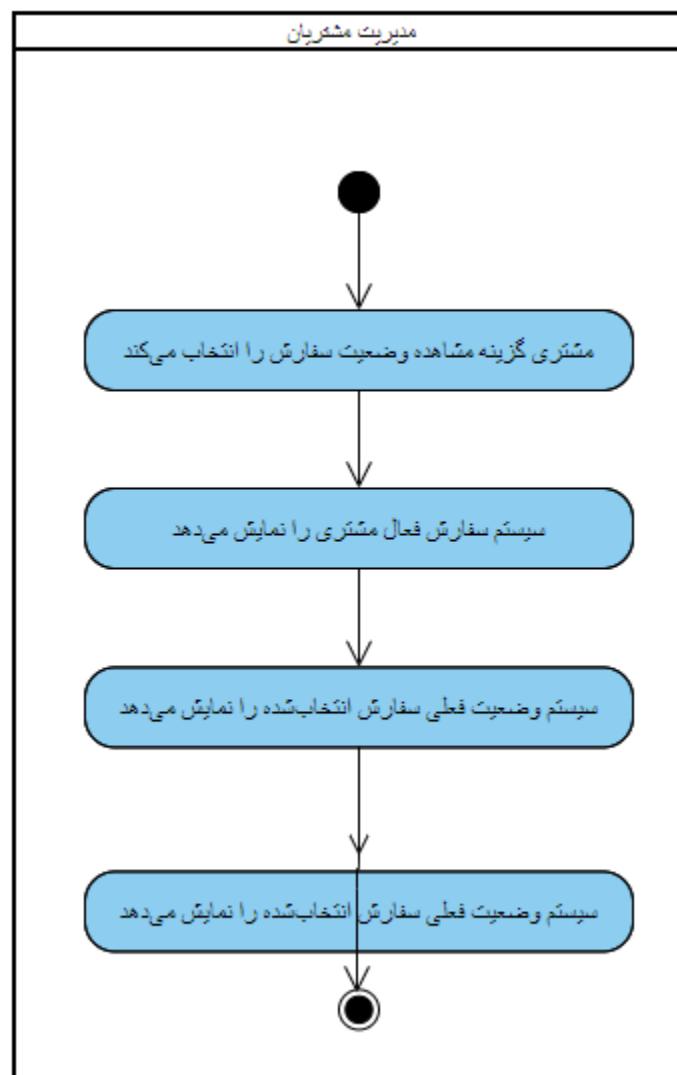
انتخاب و مشاهده‌ی یک سبد خرید	مورد کاربرد
۱۵	شناسه
<p>مشتری وارد سیستم شده باشد.</p> <ul style="list-style-type: none"> ● مشتری یک یا چند سبد خرید داشته باشد. 	پیش‌نیازها
<p>مشتری محتوای سبد خرید انتخاب شده را مشاهده می‌کند.</p> <ul style="list-style-type: none"> ● 	پس‌نیازها



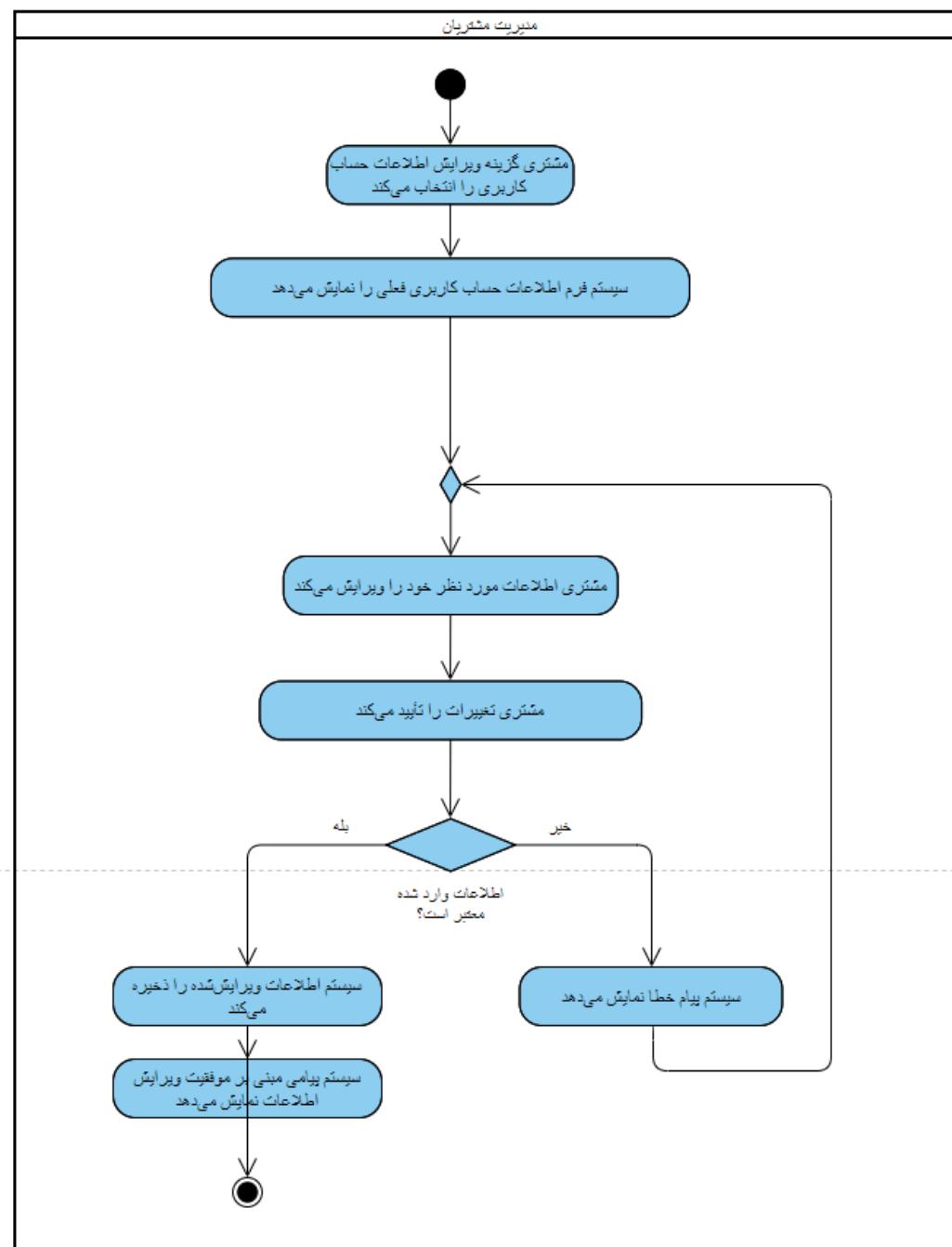
پرداخت	مورد کاربرد
۱۱	شناسه
<ul style="list-style-type: none"> ● مشتری وارد سیستم شده باشد. ● مشتری یک یا چند سبد خرید داشته باشد. 	پیش‌نیازها
<ul style="list-style-type: none"> ● سفارش ثبت شده و مشتری تأییدیه‌ی پرداخت را دریافت می‌کند. 	پس‌نیازها



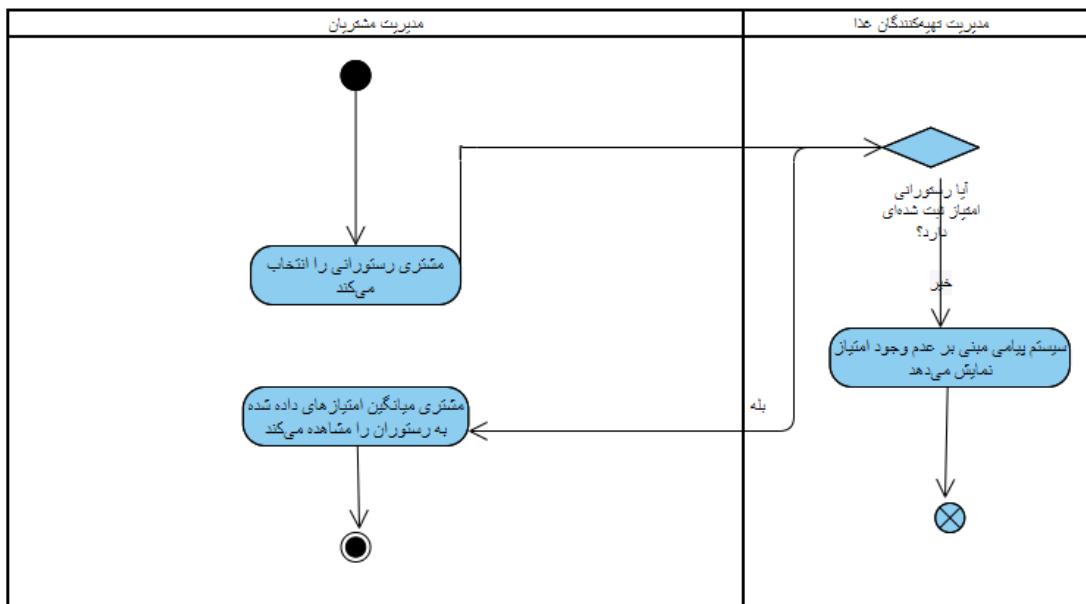
مورد کاربرد	مشاهده‌ی وضعیت فعلی سفارش
شناسه	۱۲
پیش‌نیازها	<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. مشتری سفارش فعال داشته باشد.
پس‌نیازها	<ul style="list-style-type: none"> مشتری وضعیت فعلی سفارش خود را مشاهده می‌کند.



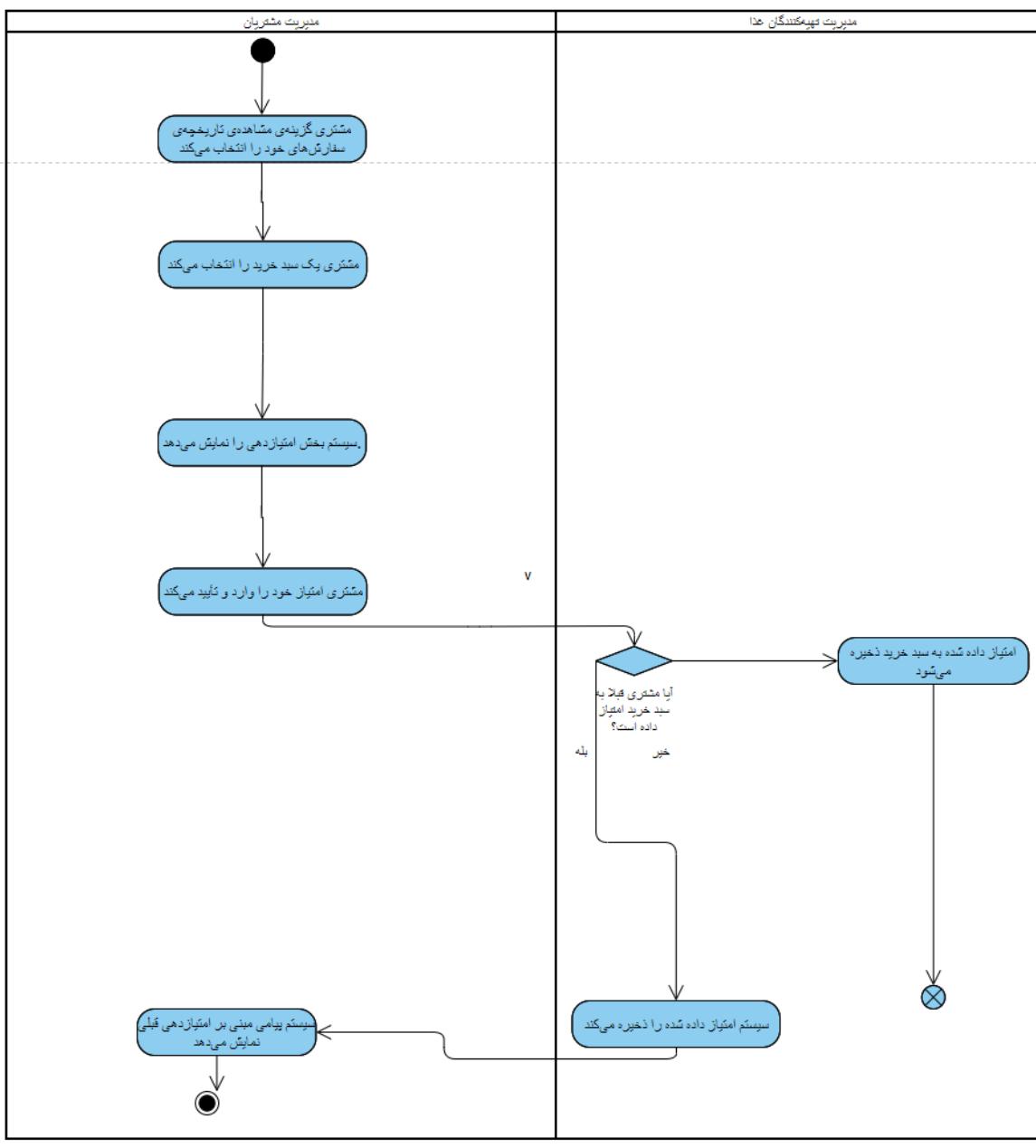
امکان ویرایش اطلاعات	مورد کاربرد
۱۳	شناسه
• مشتری وارد سیستم شده باشد.	پیش‌نیازها
• اطلاعات حساب کاربری مشتری به روزرسانی می‌شود.	پس‌نیازها



مشاهده‌ی امتیاز هر رستوران	مورد کاربرد
۱۴	شناسه
• مشتری وارد سیستم شده باشد.	پیش‌نیازها
• مشتری میانگین امتیازهای داده شده به رستوران را مشاهده می‌کند.	پس‌نیازها

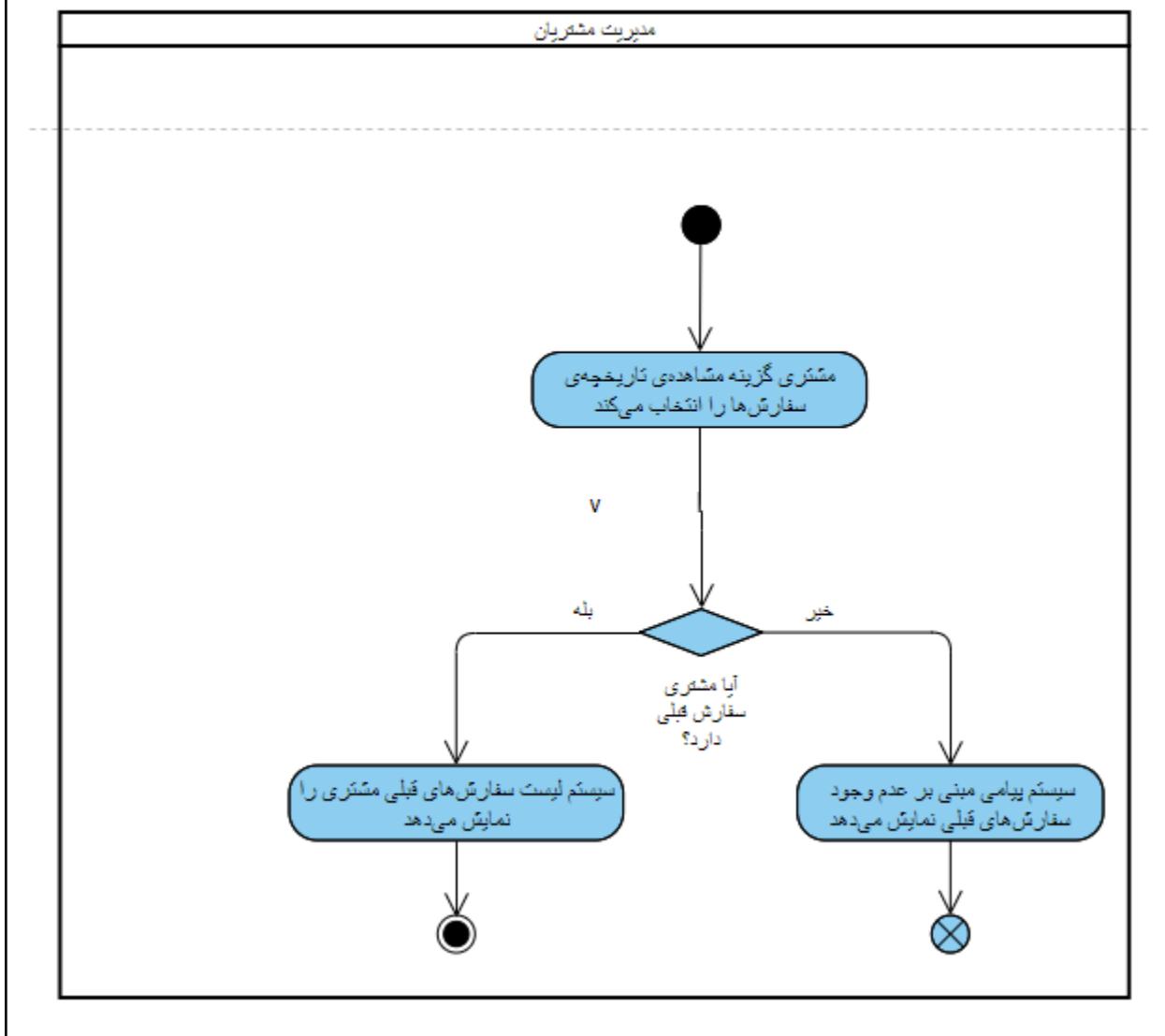


امتیازدهی به سبد خرید	مورد کاربرد
۱۵	شناسه
<p>مشتری وارد سیستم شده باشد.</p> <p>مشتری سفارشی را تکمیل کرده باشد.</p>	<ul style="list-style-type: none"> پیش‌نیازها پس‌نیازها
امتیاز داده شده به سبد خرید ذخیره می‌شود.	

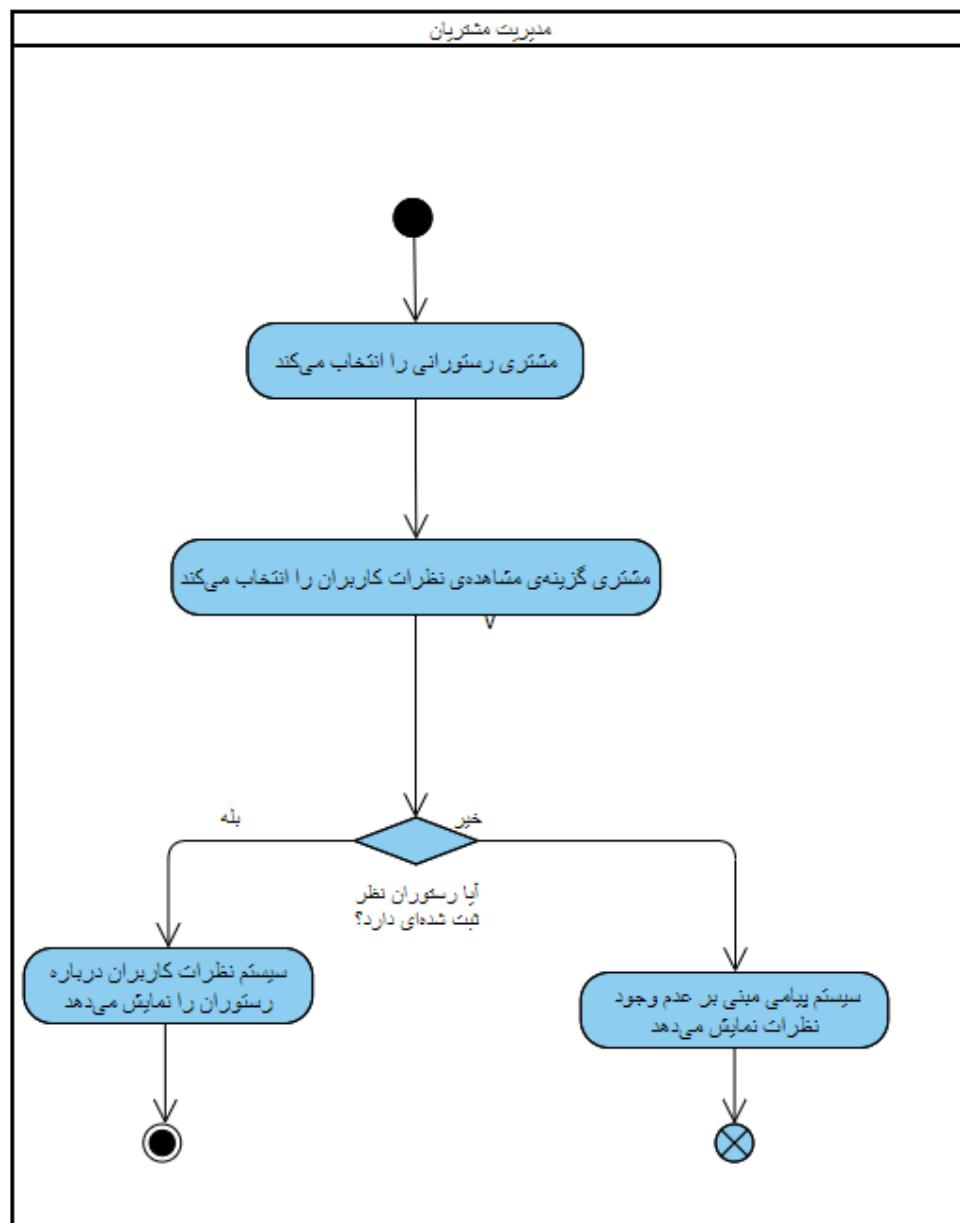


امتیازدهی به سبد خرید	مورد کاربرد

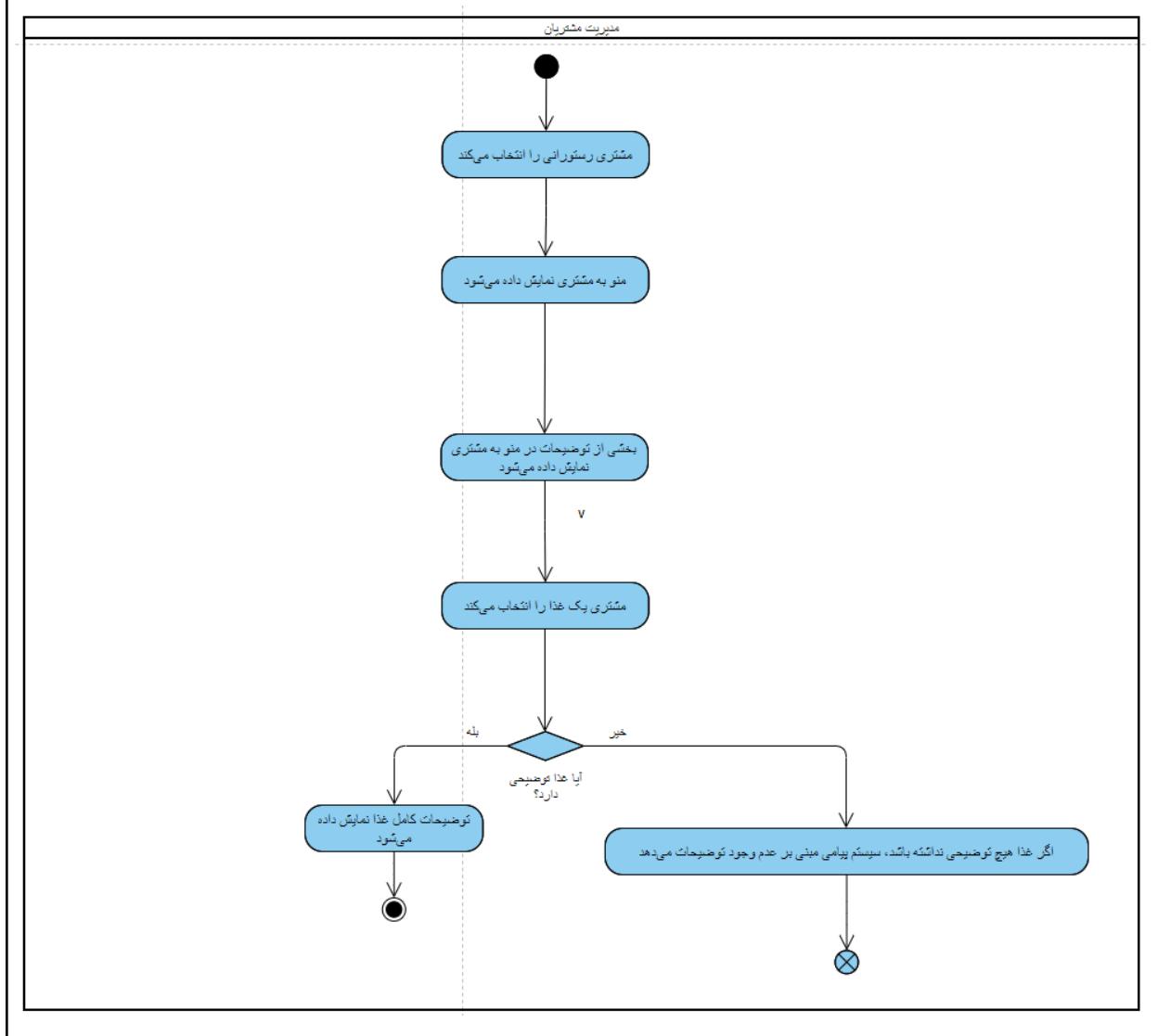
مشاهده‌ی تاریخچه‌ی سفارش‌ها	مورد کاربرد
۱۶	شناسه
• مشتری وارد سیستم شده باشد.	پیش‌نیازها
• مشتری تاریخچه‌ی سفارش‌های قبلی خود را مشاهده می‌کند.	پس‌نیازها



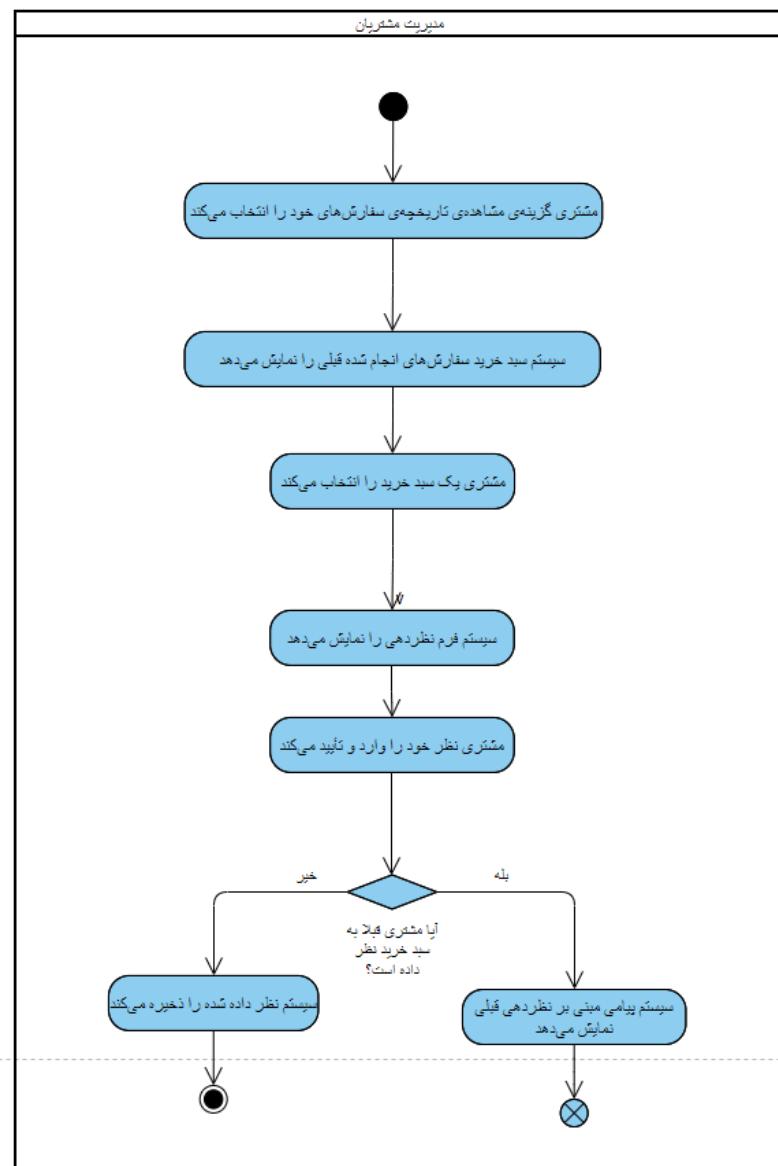
مشاهده‌ی نظرات کاربران هر رستوران	مورد کاربرد
۱۷	شناسه
<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. 	پیش‌نیازها
<ul style="list-style-type: none"> مشتری نظرات کاربران درباره رستوران را مشاهده می‌کند. 	پس‌نیازها



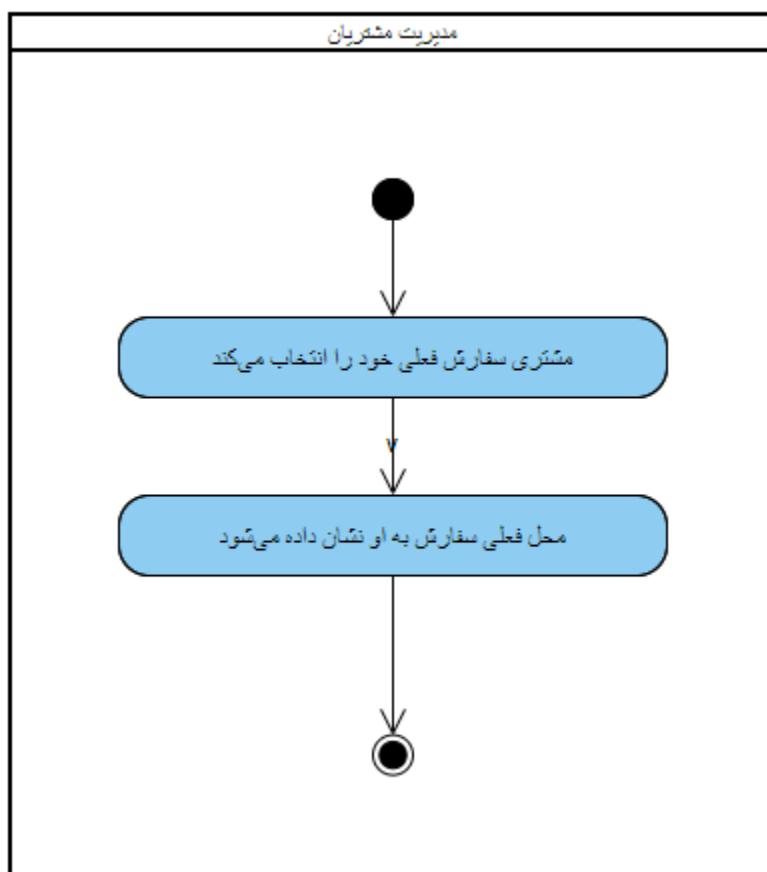
مشاهده‌ی توضیحات غذای یک رستوران	مورد کاربرد
۱۸	شناسه
● مشتری وارد سیستم شده باشد.	پیش‌نیازها
● مشتری توضیحات غذای یک رستوران را مشاهده می‌کند.	پس‌نیازها



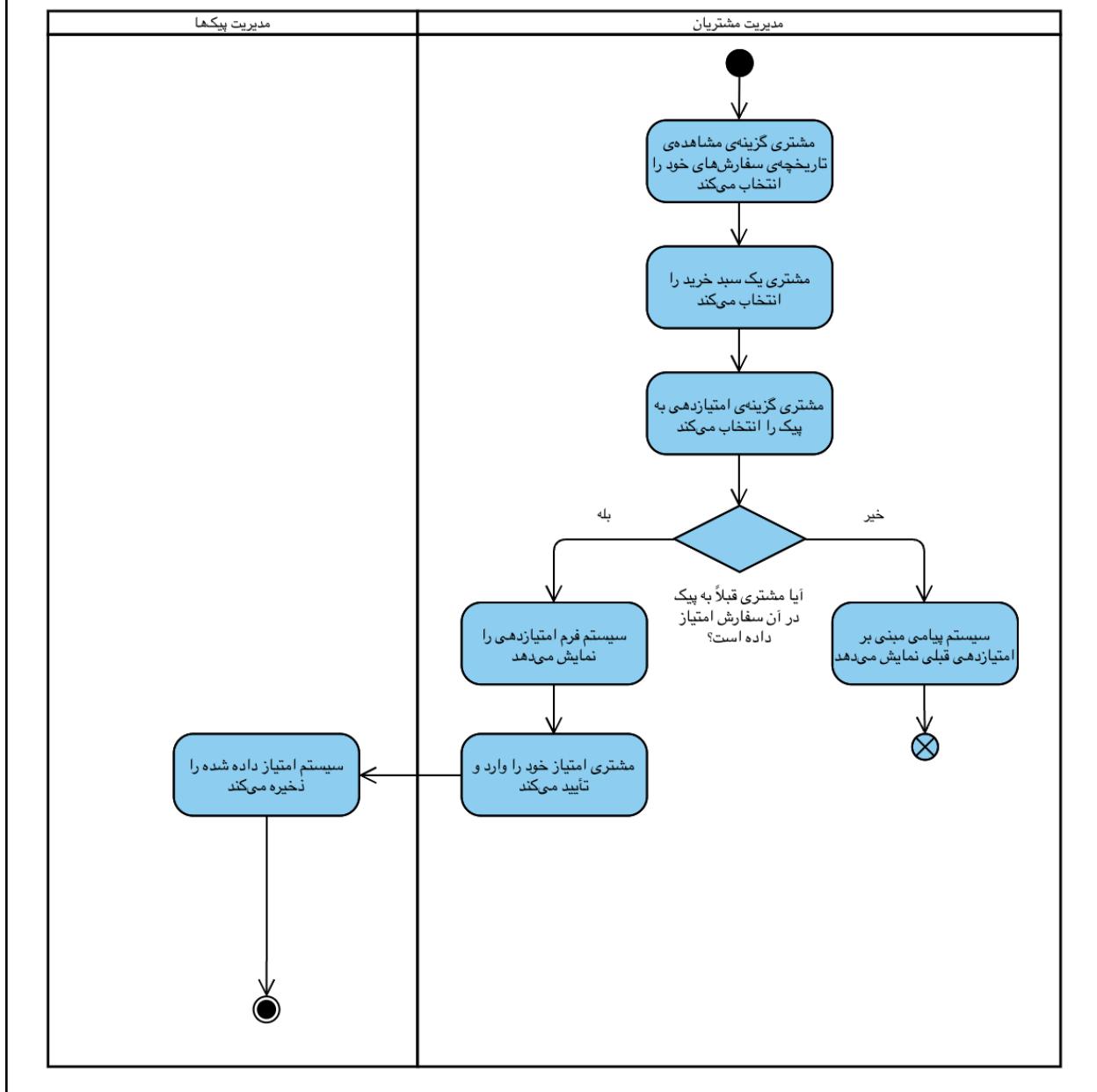
نظردهی به سفارش	مورد کاربرد
۱۹	شناسه
<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. مشتری سفارش تکمیل شده‌ای داشته باشد. 	پیش‌نیازها
سیستم نظر داده شده را ذخیره می‌کند.	پس‌نیازها



مشاهده‌ی محل فعلی سفارش	مورد کاربرد
۲۵	شناسه
<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. مشتری سفارش تکمیل نشده‌ای داشته باشد. 	پیش‌نیازها
• مشتری محل فعلی سفارش را مشاهده می‌کند.	پس‌نیازها

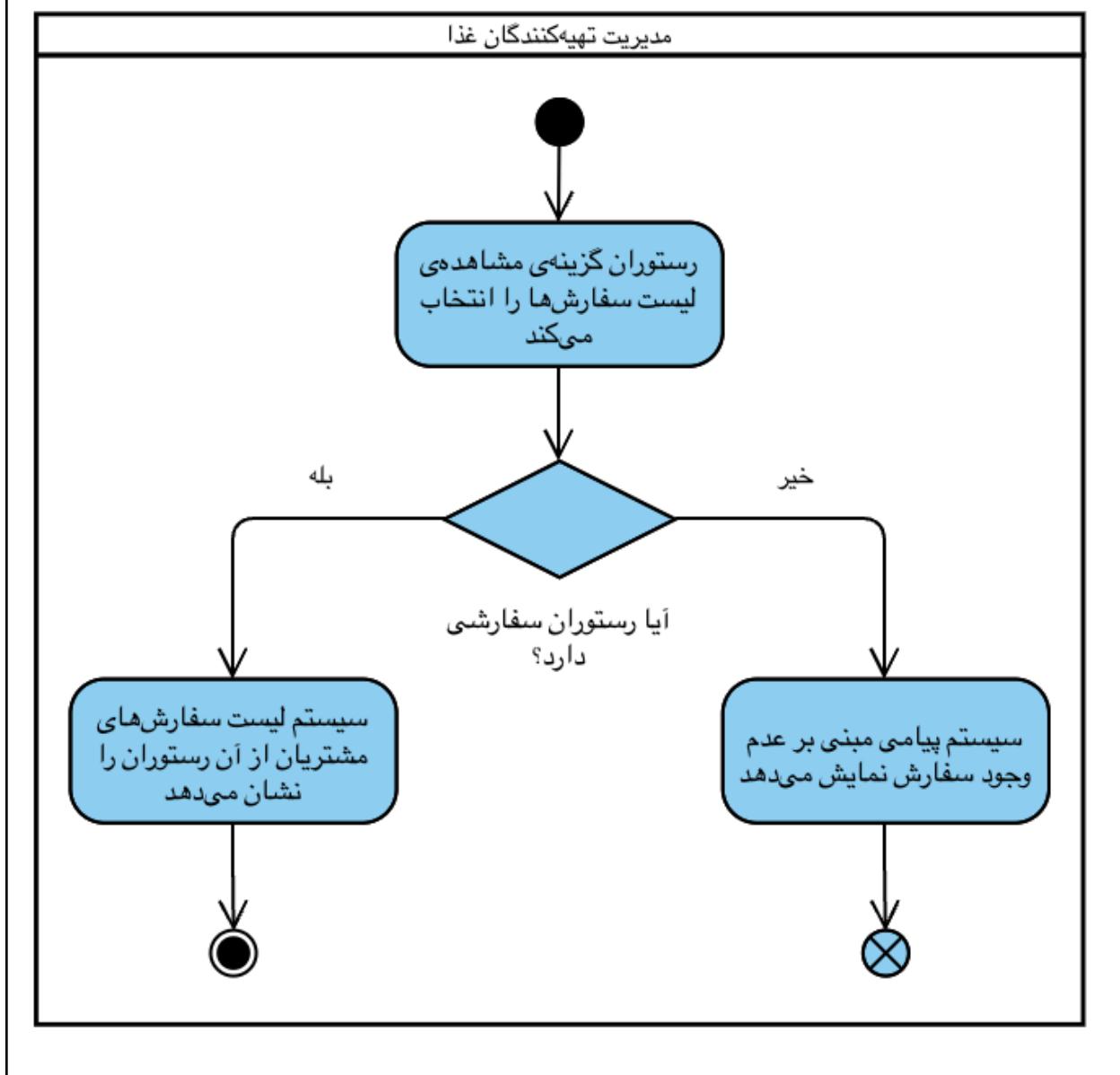


امتیازدهی به پیک	مورد کاربرد
۲۱	شناسه
<ul style="list-style-type: none"> مشتری وارد سیستم شده باشد. مشتری سفارش تکمیل شده‌ای داشته باشد. 	پیش‌نیازها
• امتیاز داده شده به پیک ذخیره می‌شود.	پس‌نیازها

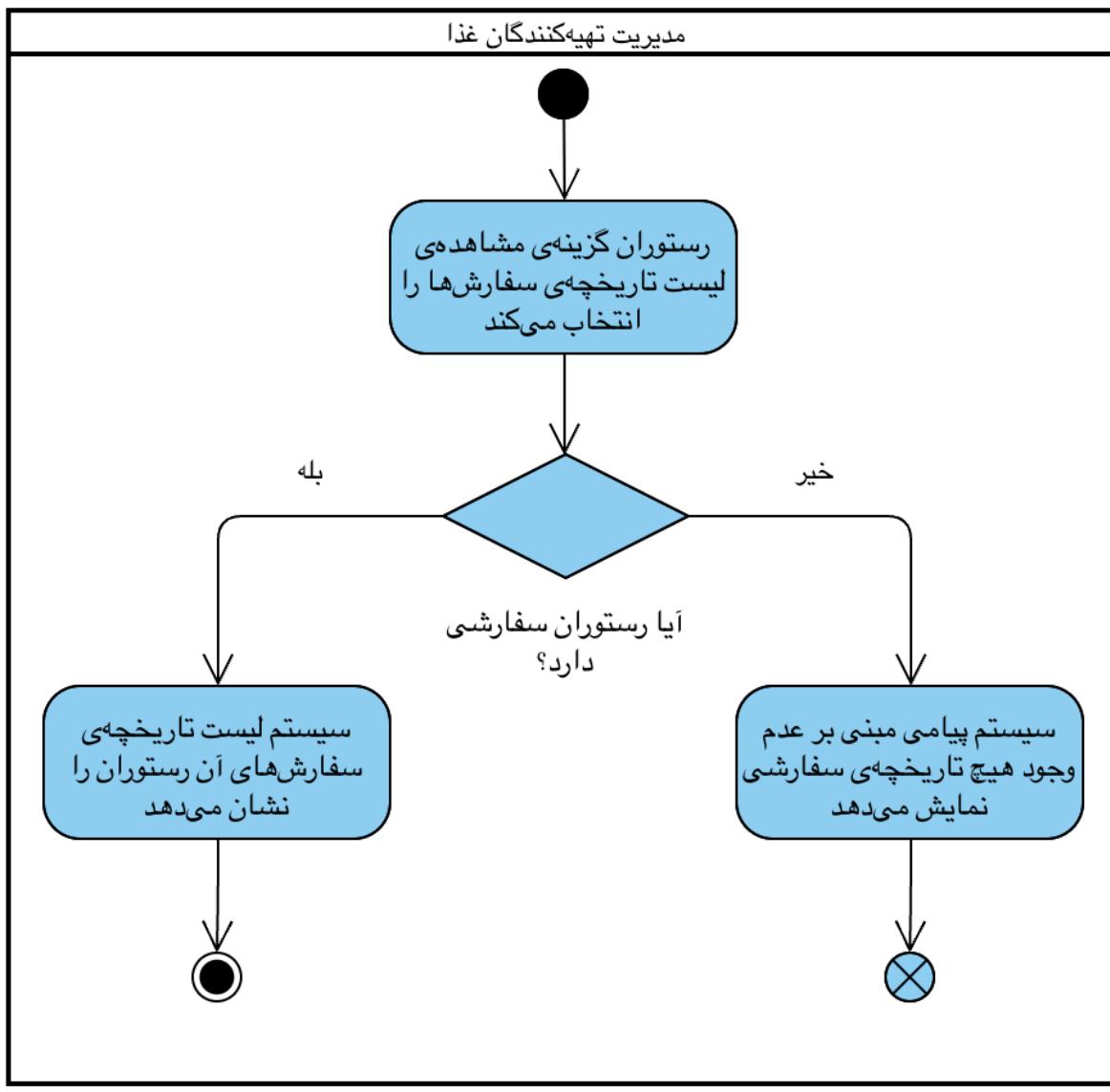


زیرسیستم مدیریت تهیه‌کنندگان غذا

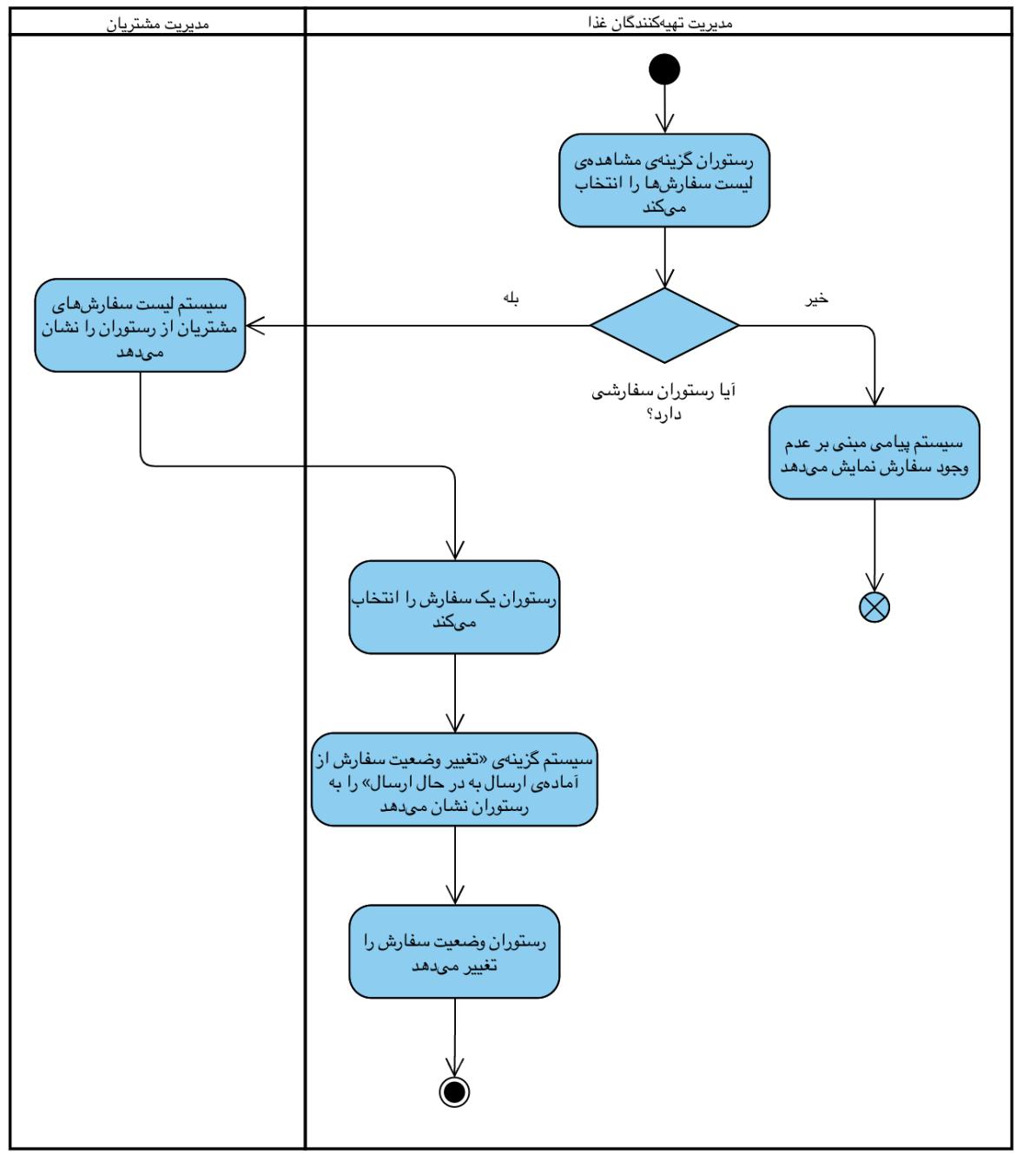
دیدن لیست سفارشات	مورد کاربرد
۲۲	شناسه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
• رستوران لیست سفارش‌های مشتریان خود را مشاهده می‌کند.	پس‌نیازها



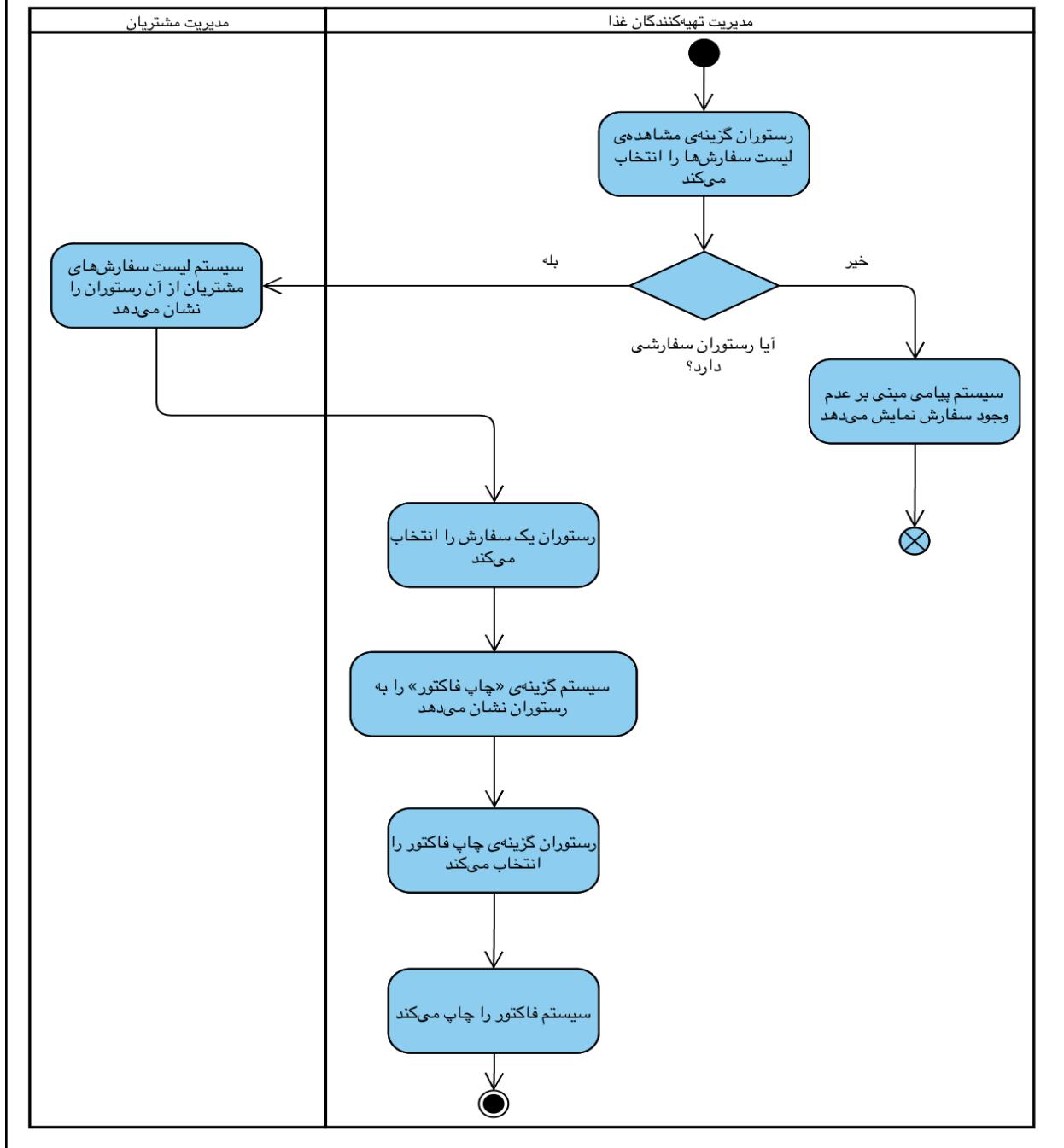
دیدن لیست تاریخچه‌ی سفارشات	مورد کاربرد
۲۳	شناسه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
• رستوران لیست تاریخچه‌ی سفارش‌های خود را مشاهده می‌کند.	پس‌نیازها



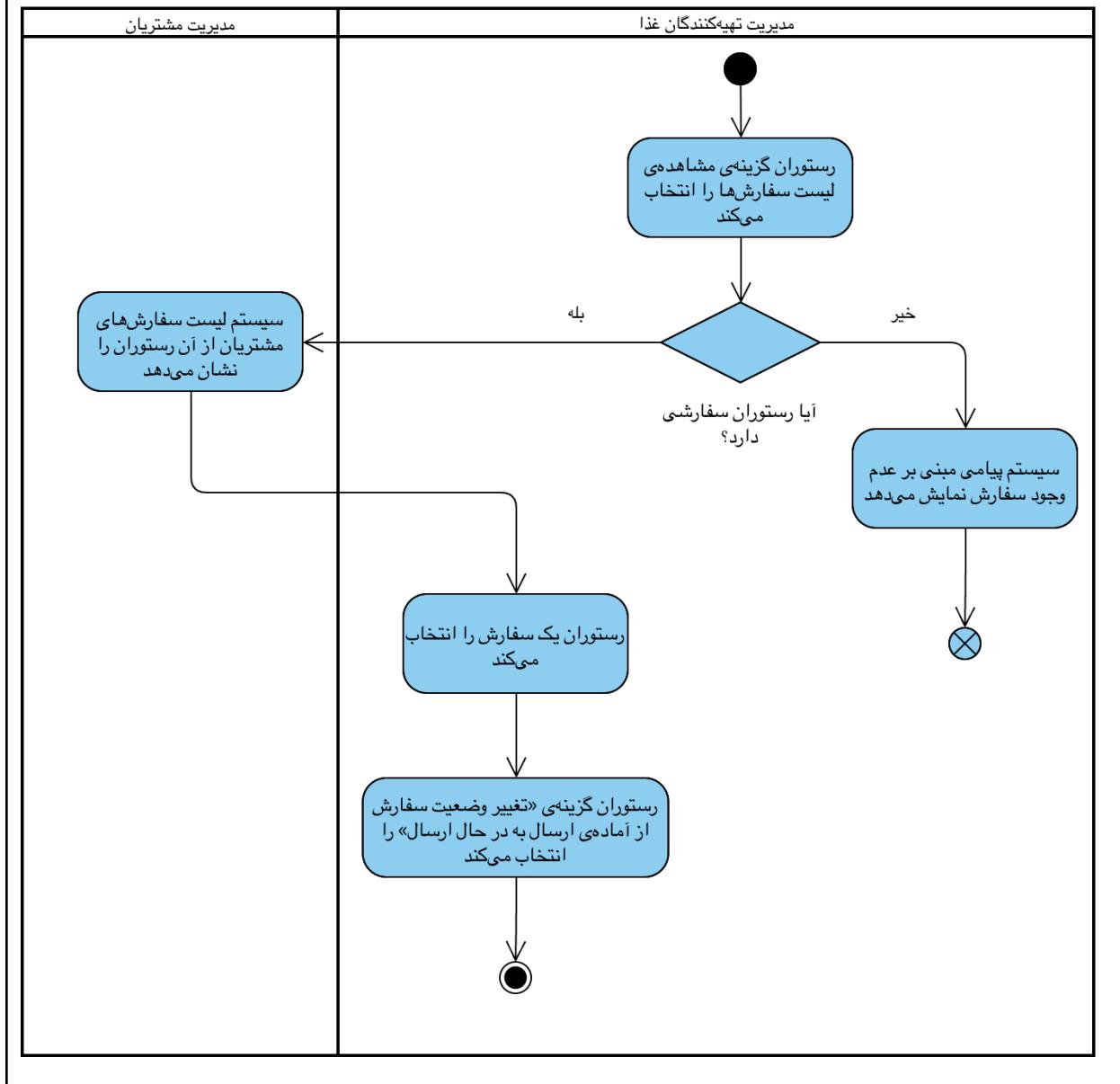
تغییر دادن وضعیت هر سفارش	مورد کاربرد
۲۴	شناسه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
• وضعیت سفارش تبدیل به «در حال ارسال» شده باشد.	پس‌نیازها



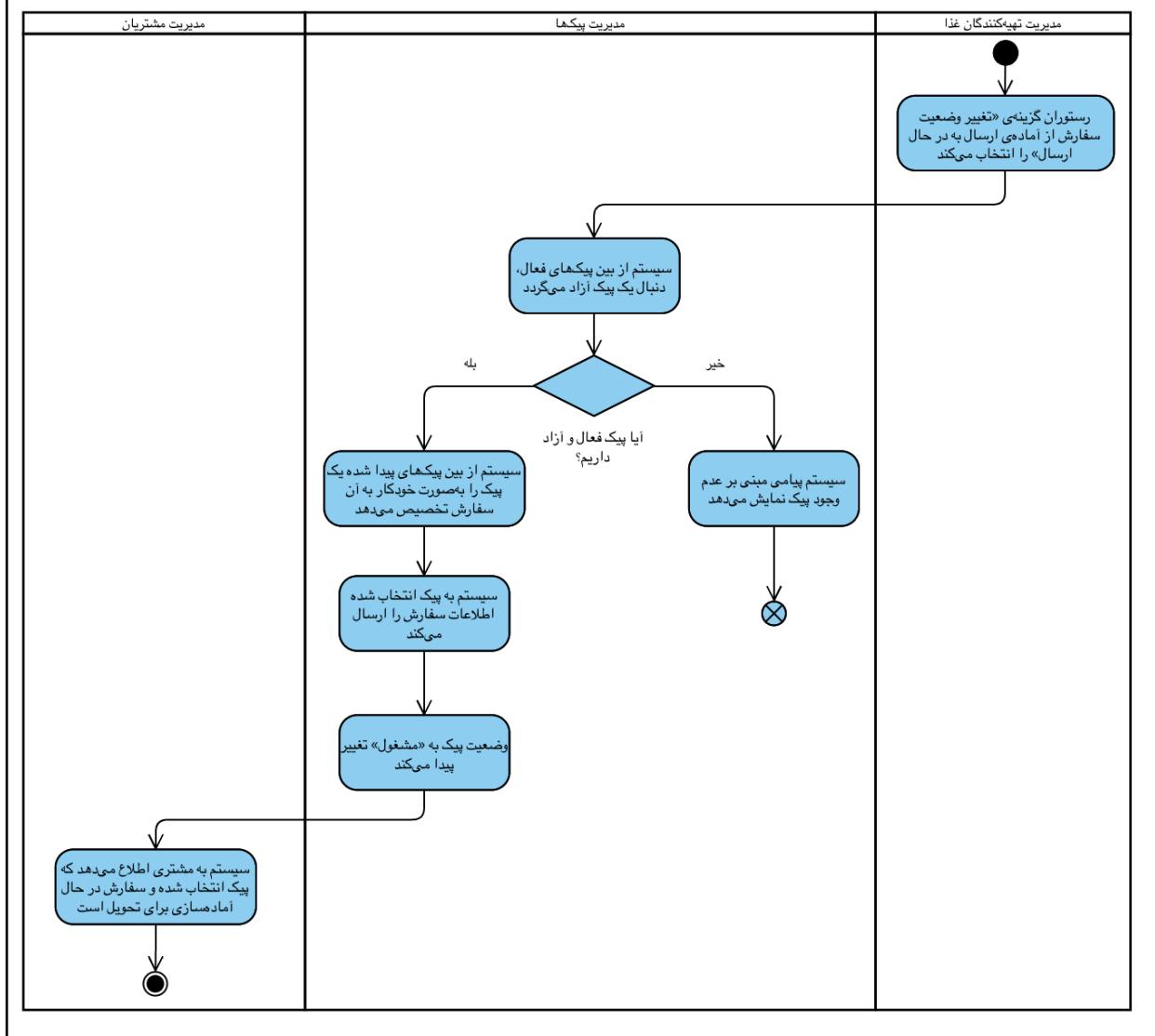
چاپ فاکتور	مورد کاربرد
۲۵	شناسه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
• فاکتور سفارش چاپ شده باشد.	پس‌نیازها



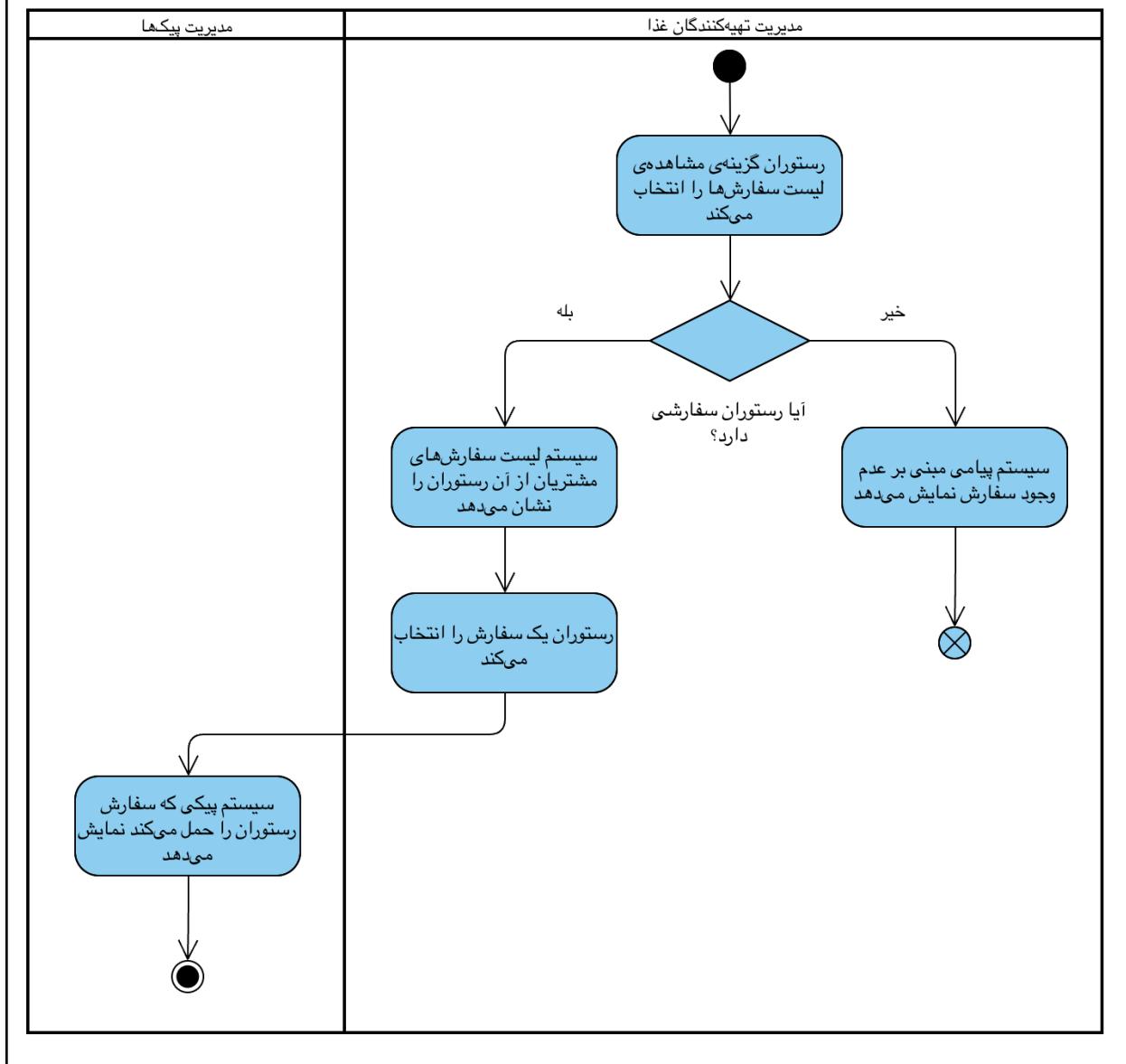
درخواست پیک	مورد کاربرد
۲۶	شناسه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
• وضعیت سفارش به «در حال ارسال» تغییر کرده باشد.	پس‌نیازها



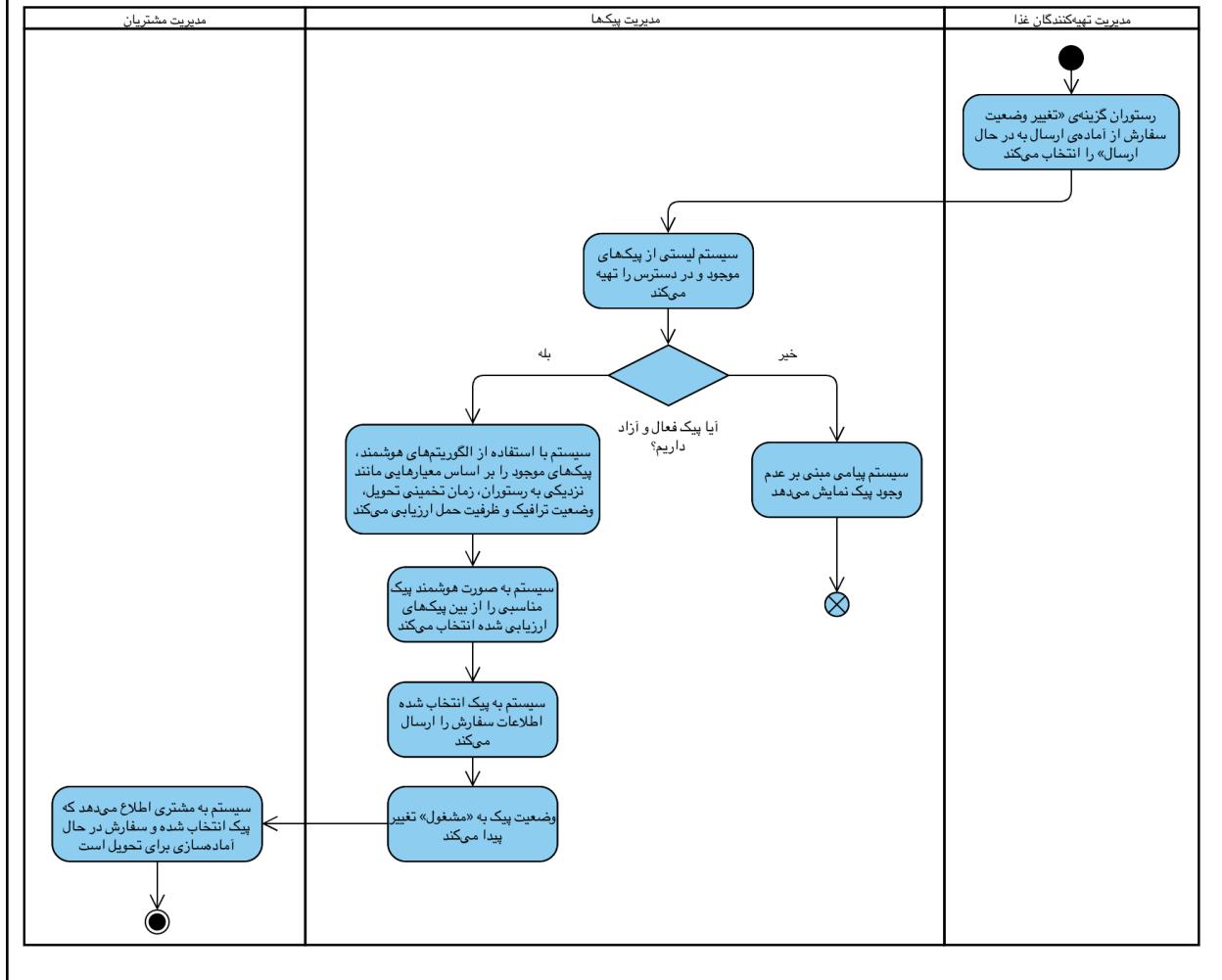
تعیین خودکار یک پیک برای هر سفارش	مورد کاربرد
۲۷	شناسه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
• برای سفارش ثبت شده یک پیک تخصیص داده شود.	پس‌نیازها



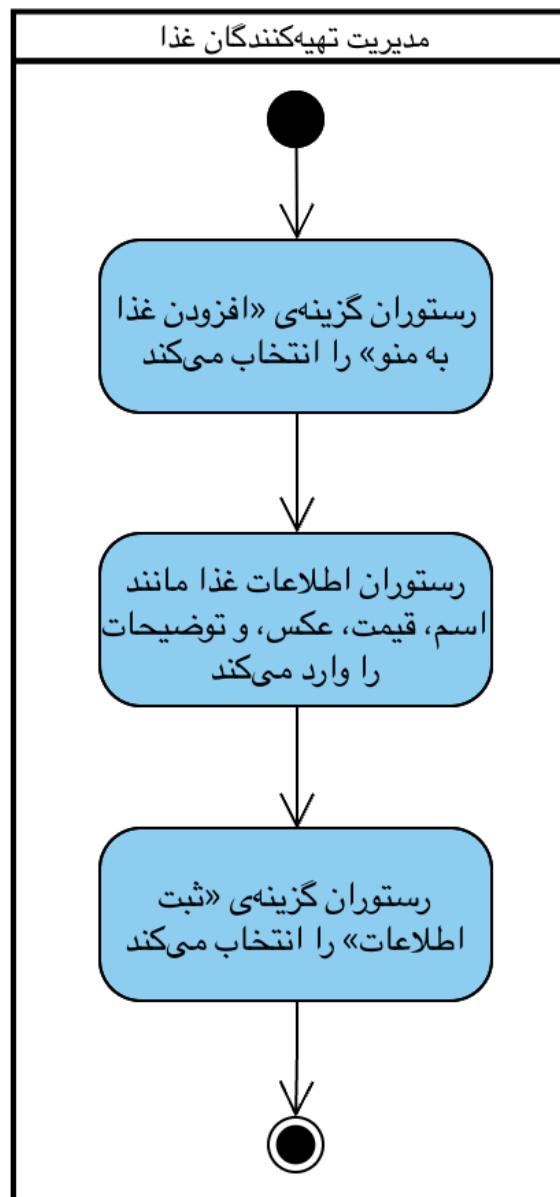
مشاهده‌ی پیک‌های حمل‌کننده‌ی سفارشات آن رستوران	مورد کاربرد
۲۸	شناسه
<ul style="list-style-type: none"> رستوران وارد سیستم شده باشد. یک یا چند سفارش باید «در حال ارسال» باشند. 	پیش‌نیازها
سیستم پیک در حال حمل سفارش را نمایش می‌دهد.	پس‌نیازها



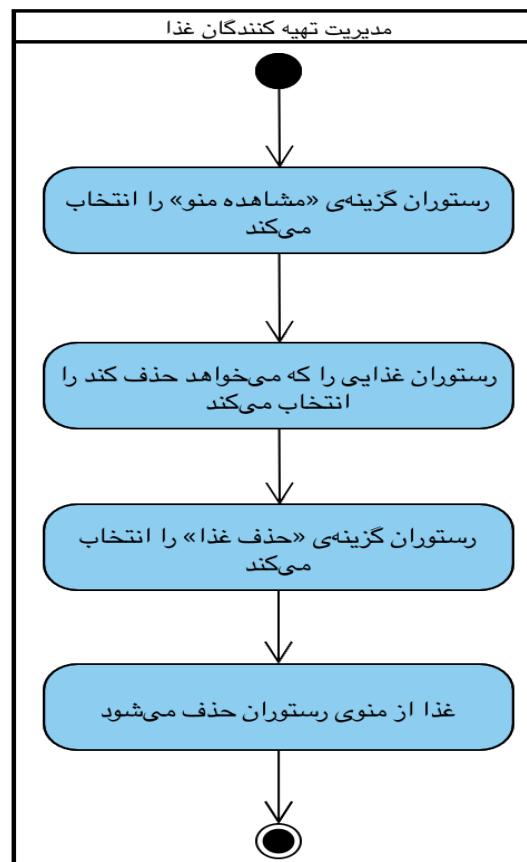
تعیین هوشمند یک پیک برای هر سفارش	مورد کاربرد
۲۹	شناسه
• رستوران وارد سیستم شده باشد.	پیش‌نیازها
• سیستم یک پیک را به صورت هوشمند برای تحویل سفارش انتخاب می‌کند.	پس‌نیازها



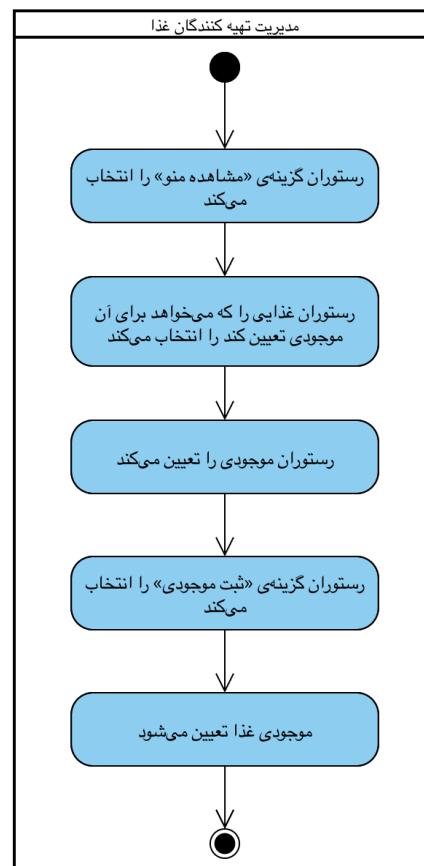
افزودن غذا به منو		مورد کاربرد
۳۰		شناسه
• رستوران وارد سیستم شده باشد.		پیش‌نیازها
• غذا به منوی رستوران افزوده می‌شود.		پس‌نیازها



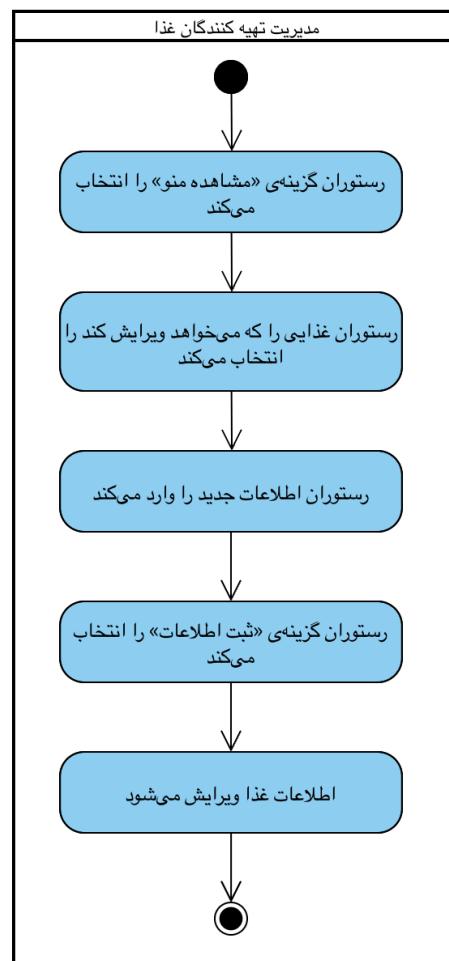
مورد کاربرد	حذف غذا از منو
شناسه	۳۱
پیش‌نیازها	<ul style="list-style-type: none"> ● رستوران وارد سیستم شده باشد. ● در منوی رستوران حداقل یک غذا وجود داشته باشد.
پس‌نیازها	<ul style="list-style-type: none"> ● غذا از منوی رستوران حذف می‌شود.



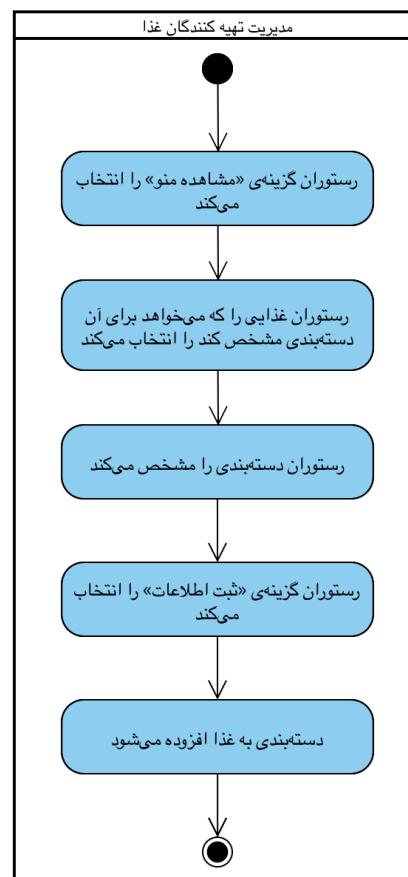
تعیین موجودی هر غذا	مورد کاربرد
۳۲	شناسه
<p>رستوران وارد سیستم شده باشد.</p> <ul style="list-style-type: none"> ● در منوی رستوران حداقل یک غذا وجود داشته باشد. 	پیش‌نیازها
• موجودی غذا تعیین می‌شود.	پس‌نیازها



مورد کاربرد	ویرایش غذا در منو
شناسه	۳۳
پیش‌نیازها	<ul style="list-style-type: none"> • رستوران وارد سیستم شده باشد. • در منوی رستوران حداقل یک غذا وجود داشته باشد.
پس‌نیازها	<ul style="list-style-type: none"> • اطلاعات غذا ویرایش می‌شود.

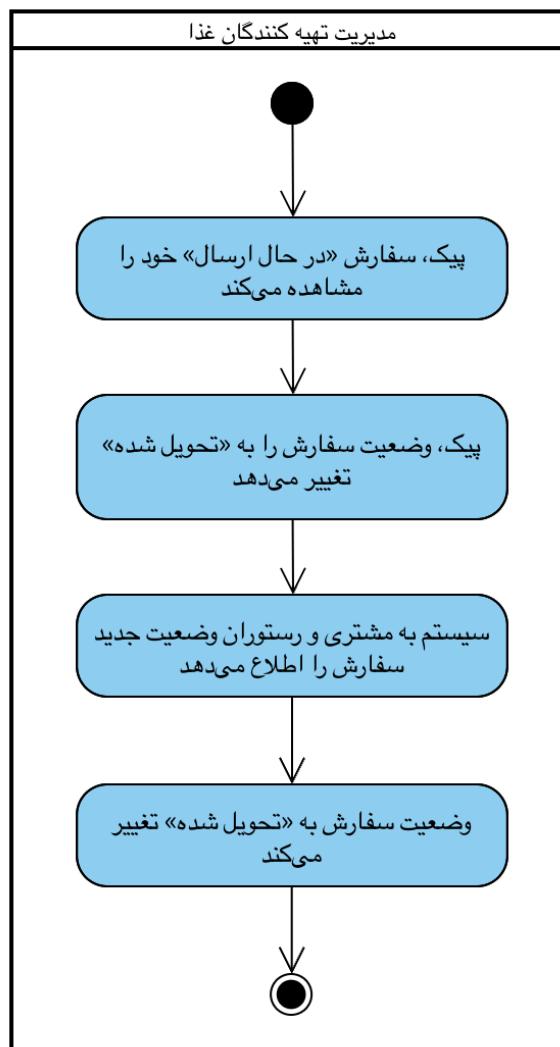


افزودن دسته‌بندی برای هر غذا		مورد کاربرد
۳۴		شناسه
• رستوران وارد سیستم شده باشد. • در منوی رستوران حداقل یک غذا وجود داشته باشد.		پیش‌نیازها
• دسته‌بندی به غذا افزوده می‌شود.		پس‌نیازها

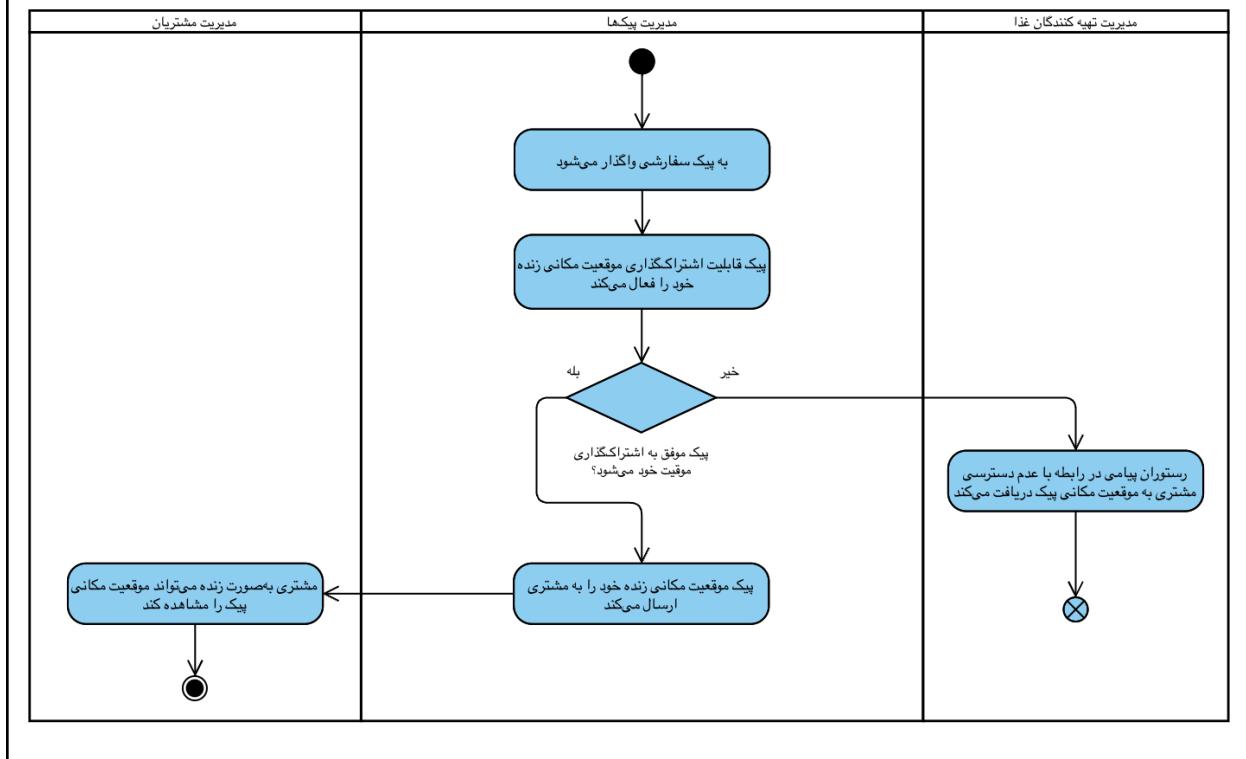


زیرسیستم مدیریت پیک‌ها

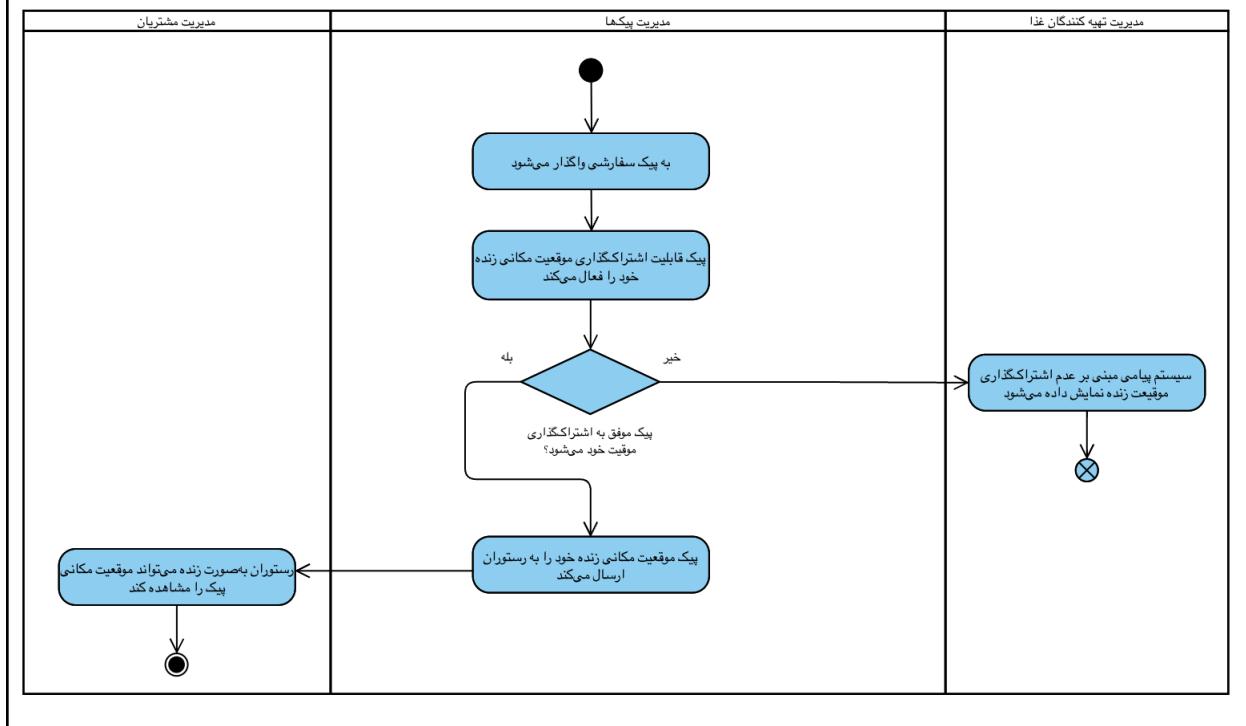
تغییر دادن وضعیت هر سفارش	مورد کاربرد
۳۵	شناسه
<ul style="list-style-type: none"> ● پیک وارد سیستم شده باشد. ● سفارشی باید به پیک تخصیص داده شده باشد. 	پیش‌نیازها
<ul style="list-style-type: none"> ● مشتری و رستوران بتوانند وضعیت جدید سفارش را مشاهده کنند. 	پس‌نیازها



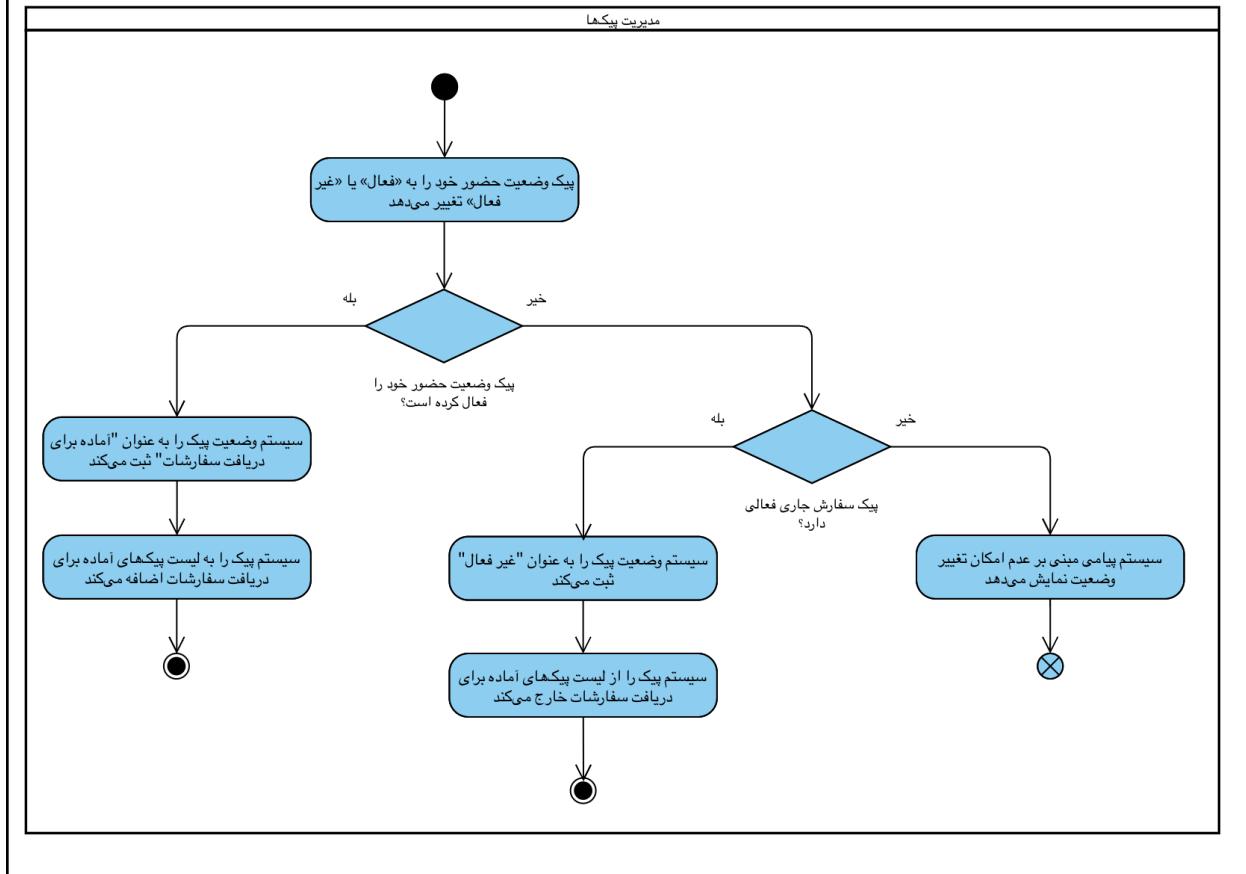
ارسال موقعیت به صورت زنده برای مشتری	مورد کاربرد
۳۶	شناسه
• سفارشی باید به پیک تخصیص داده شده باشد.	پیش‌نیازها
• مشتری باید امکان پیگیری موقعیت مکانی زنده پیک را در هر لحظه داشته باشد.	پس‌نیازها



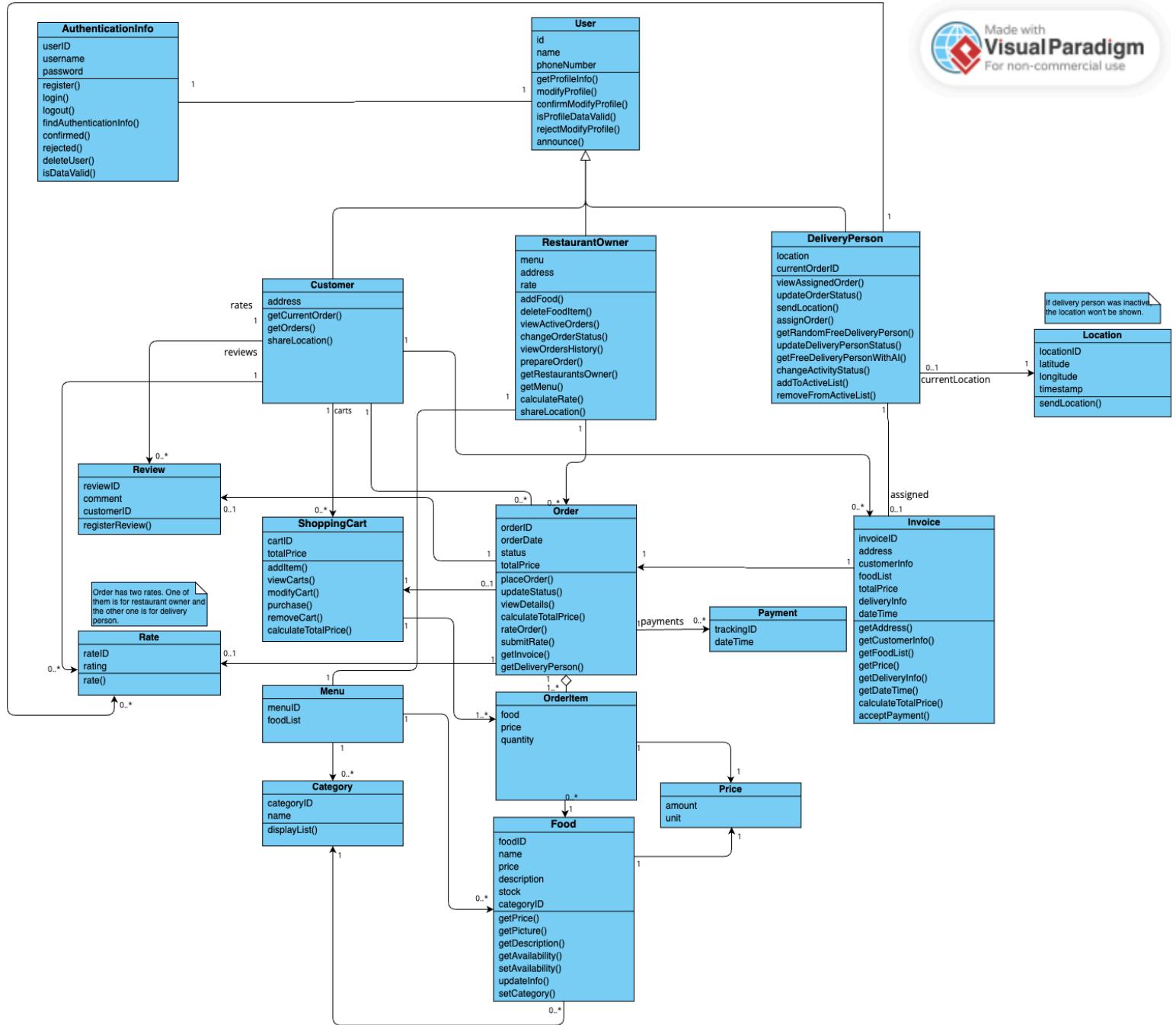
ارسال موقعیت به صورت زنده برای رستوران	مورد کاربرد
۳۷	شناسه
<ul style="list-style-type: none"> سفارشی باید به پیک تخصیص داده شده باشد. 	پیش‌نیازها
<ul style="list-style-type: none"> رستوران باید امکان پیگیری موقعیت مکانی زنده پیک را در هر لحظه داشته باشد. 	پس‌نیازها



مورد کاربرد	تغییر وضعیت بین فعال و غیرفعال
شناسه	۳۸
پیش‌نیازها	• پیک وارد سیستم شده است.
پس‌نیازها	• وضعیت فعال یا غیر فعال بودن پیک ویرایش شود.

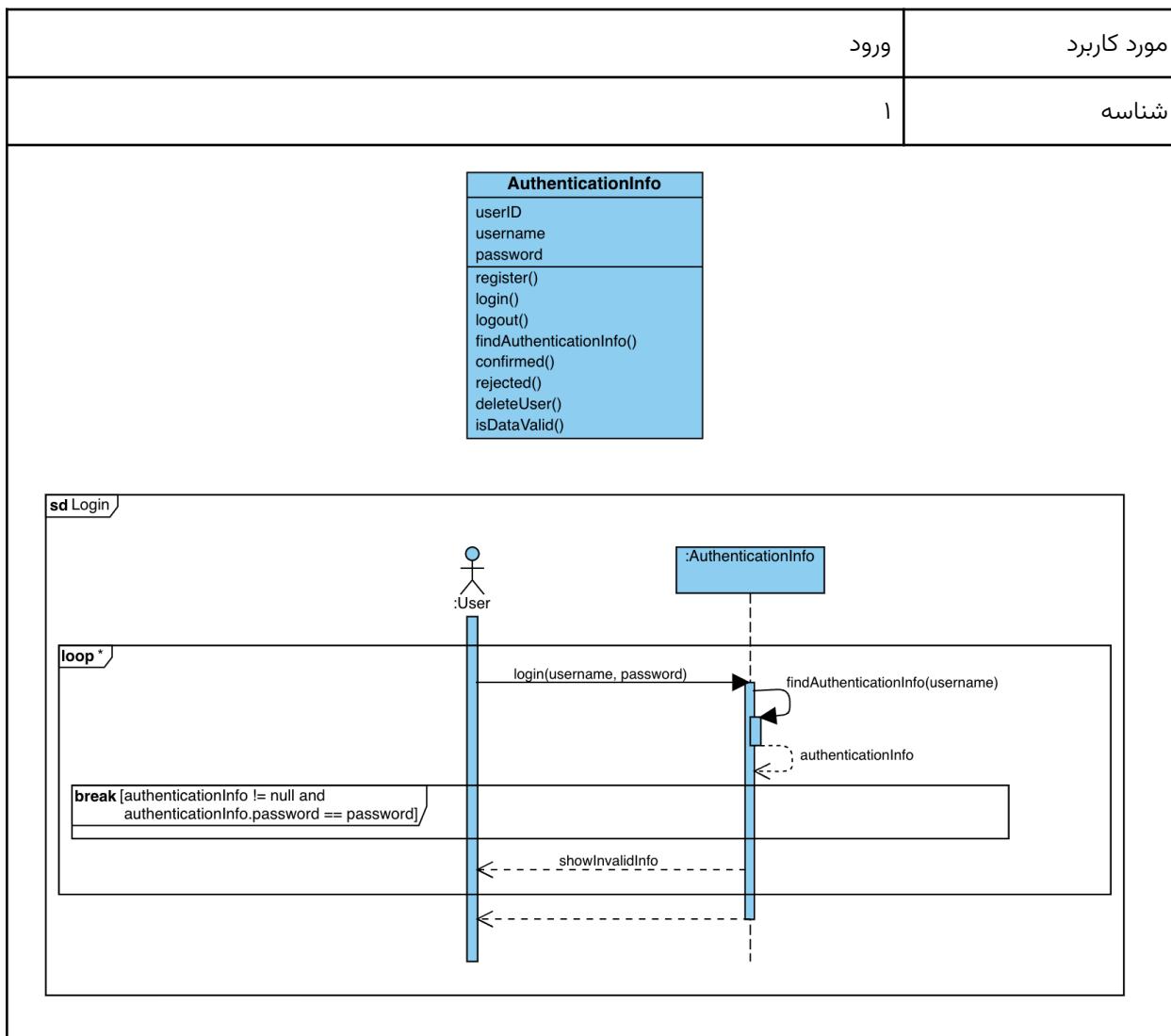


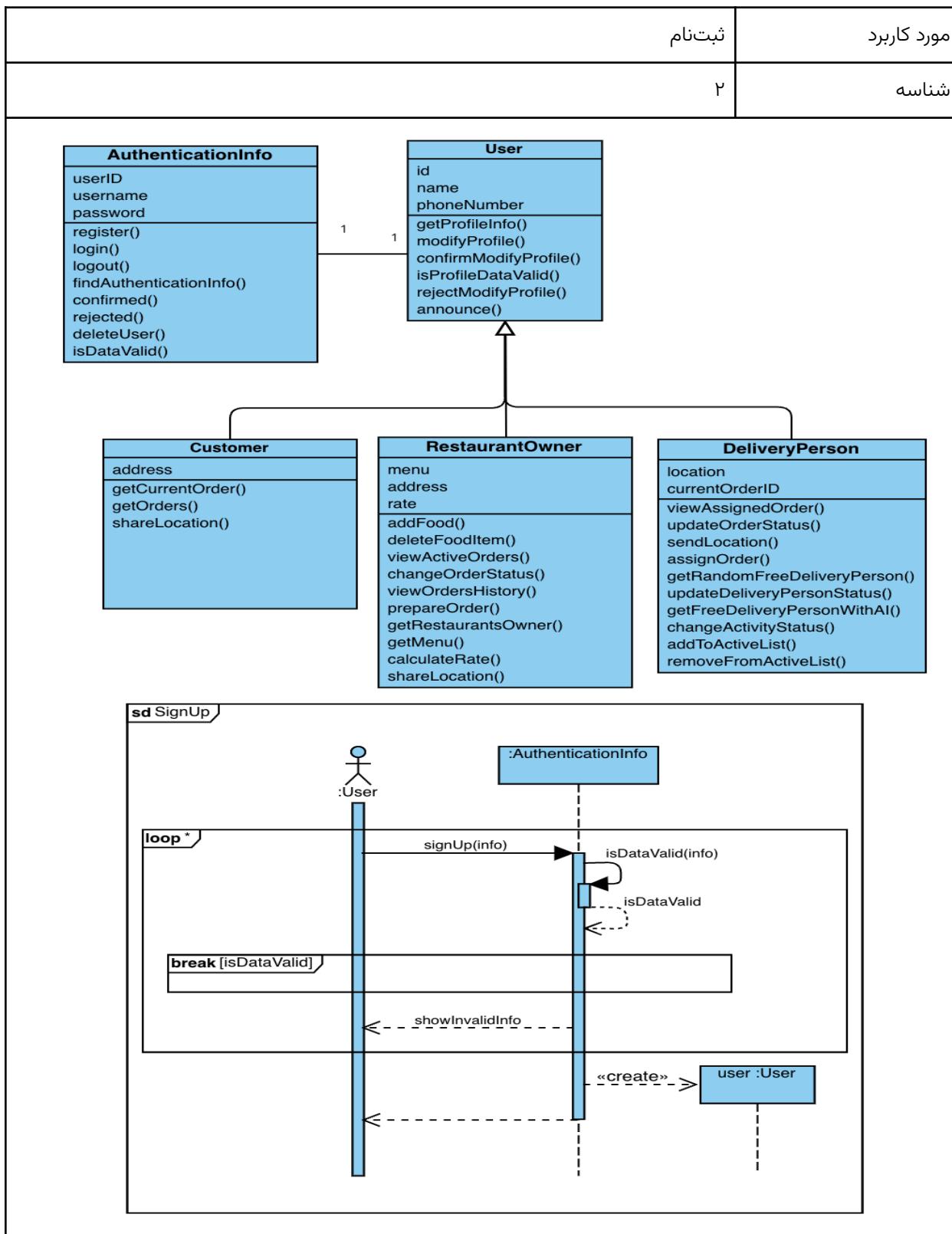
نمودار کلاس‌های تحلیل



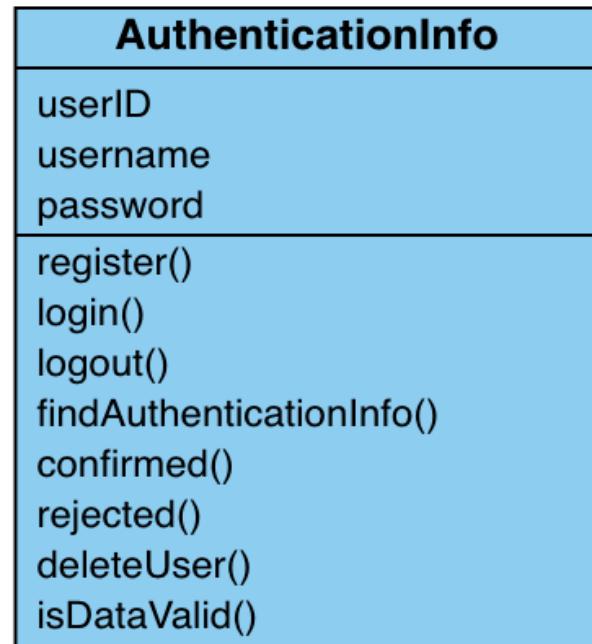
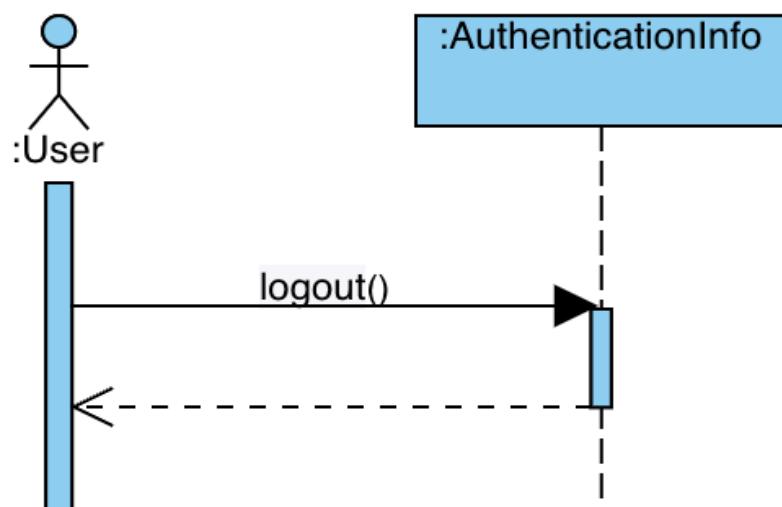
نمودارهای توالی تحلیل

زیرسیستم احراز هویت و دسترسی

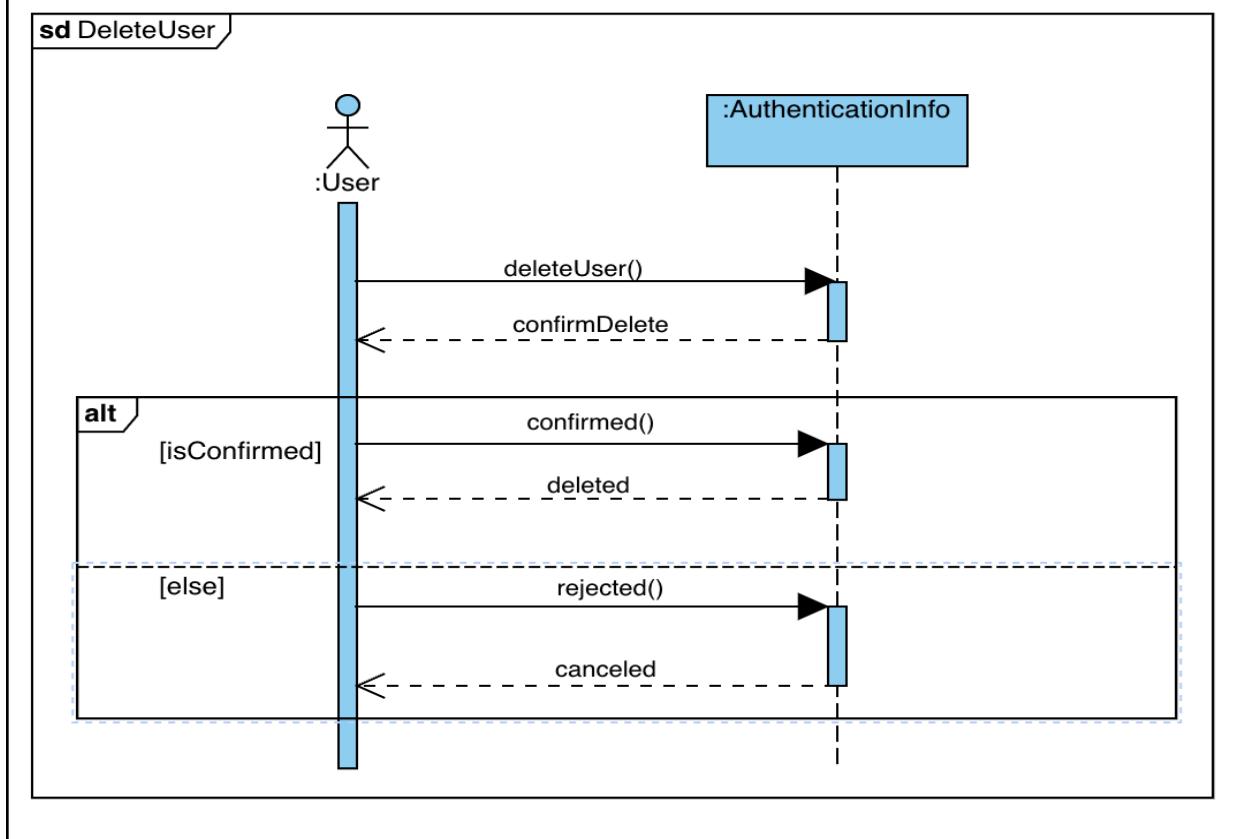
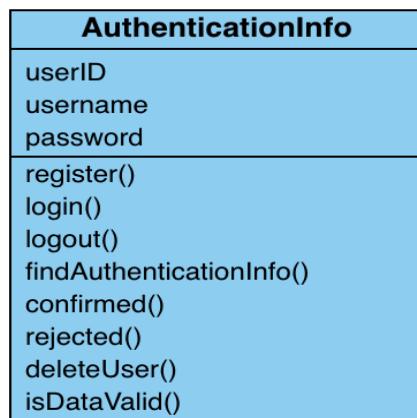




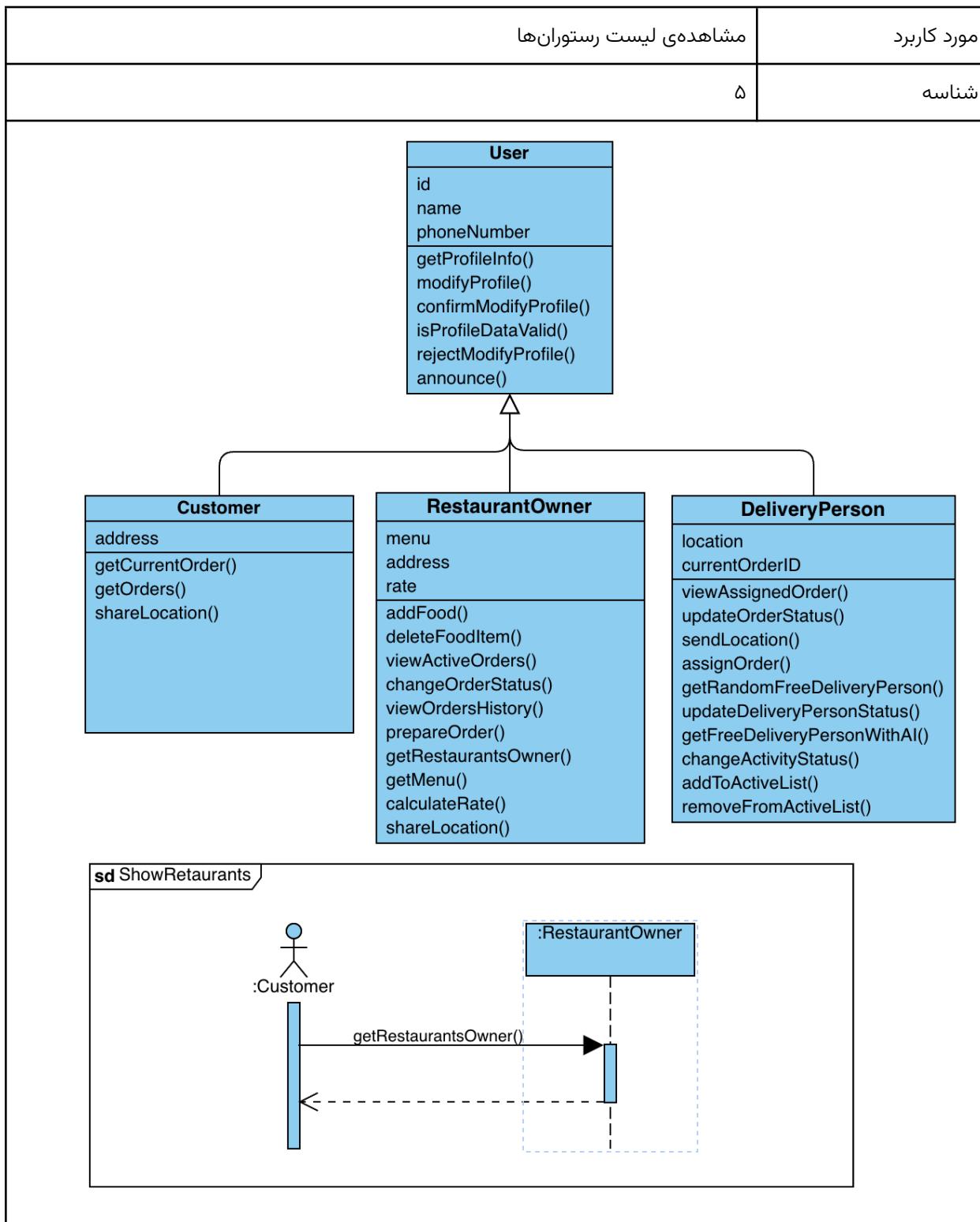
خروج	مورد کاربرد
۳	شناسه

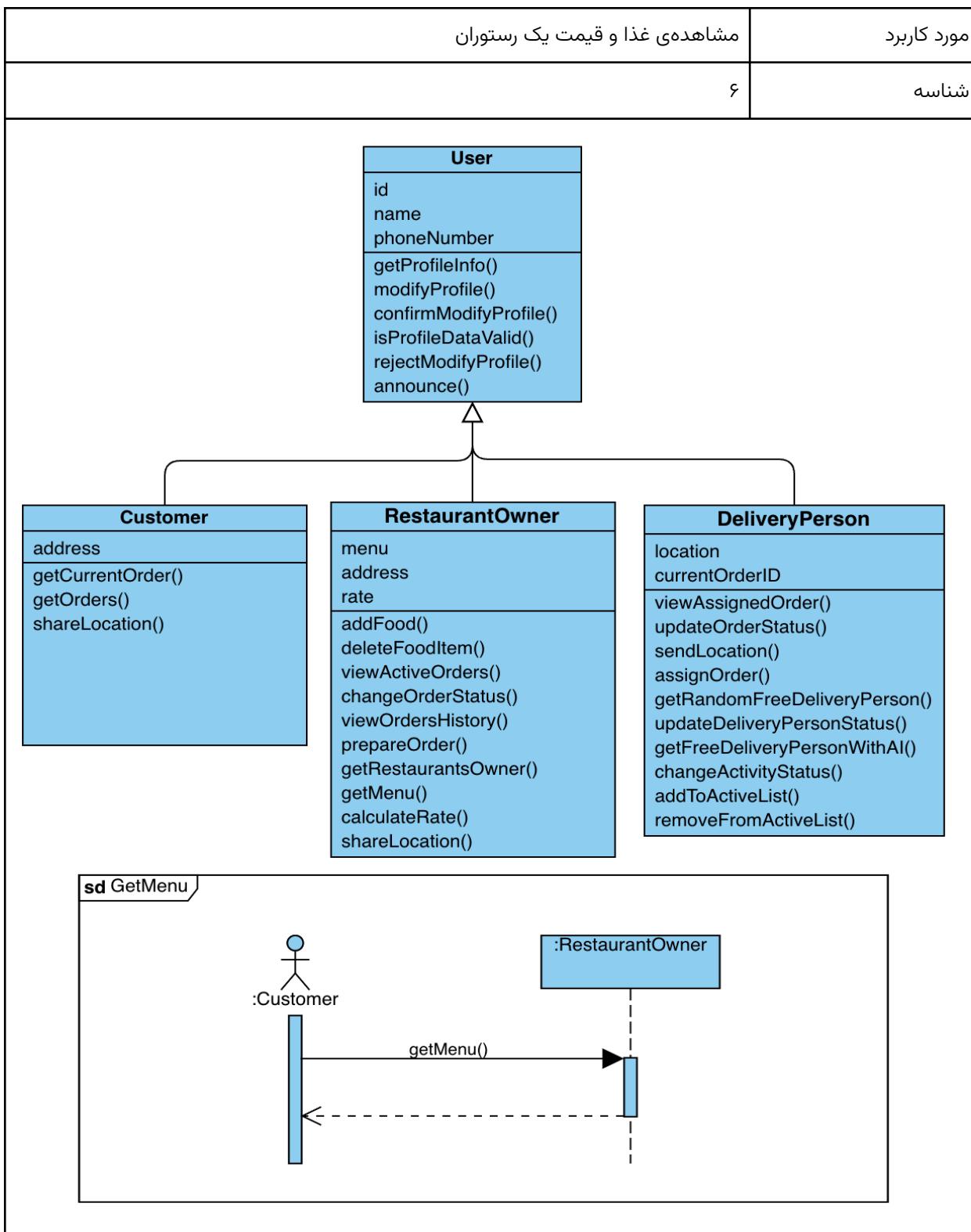
**sd Logout**

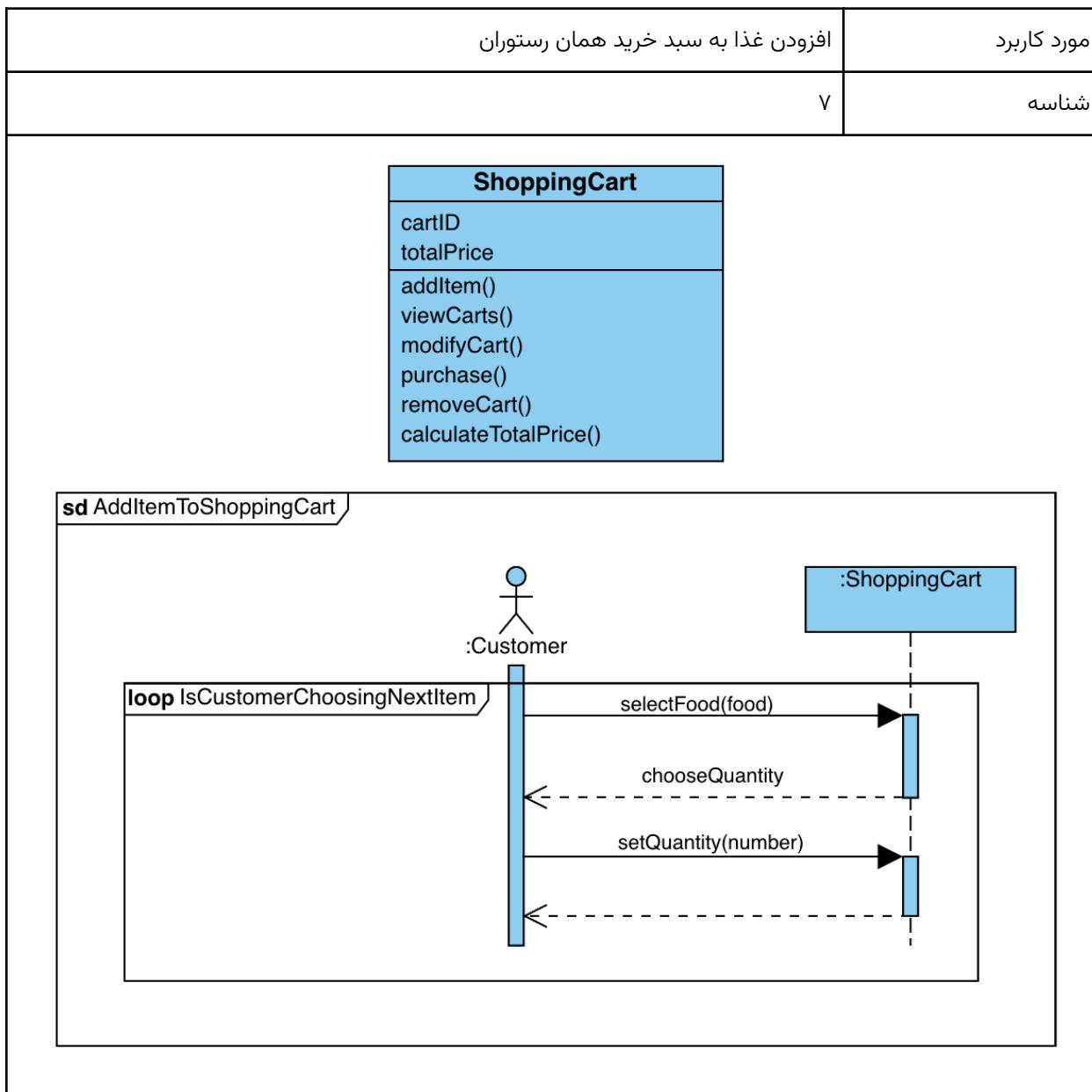
حذف حساب کاربری	مورد کاربرد
۴	شناسه

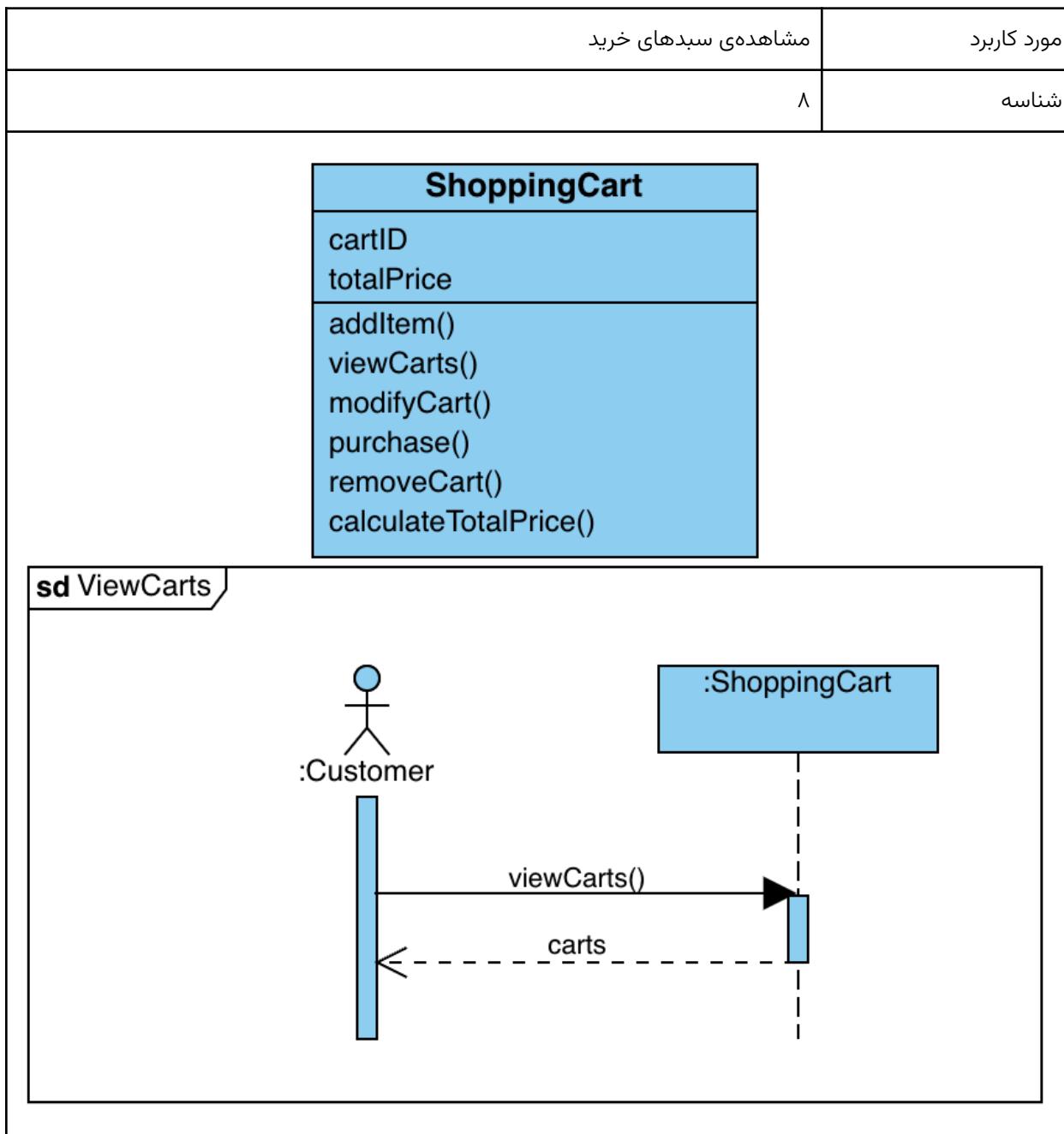


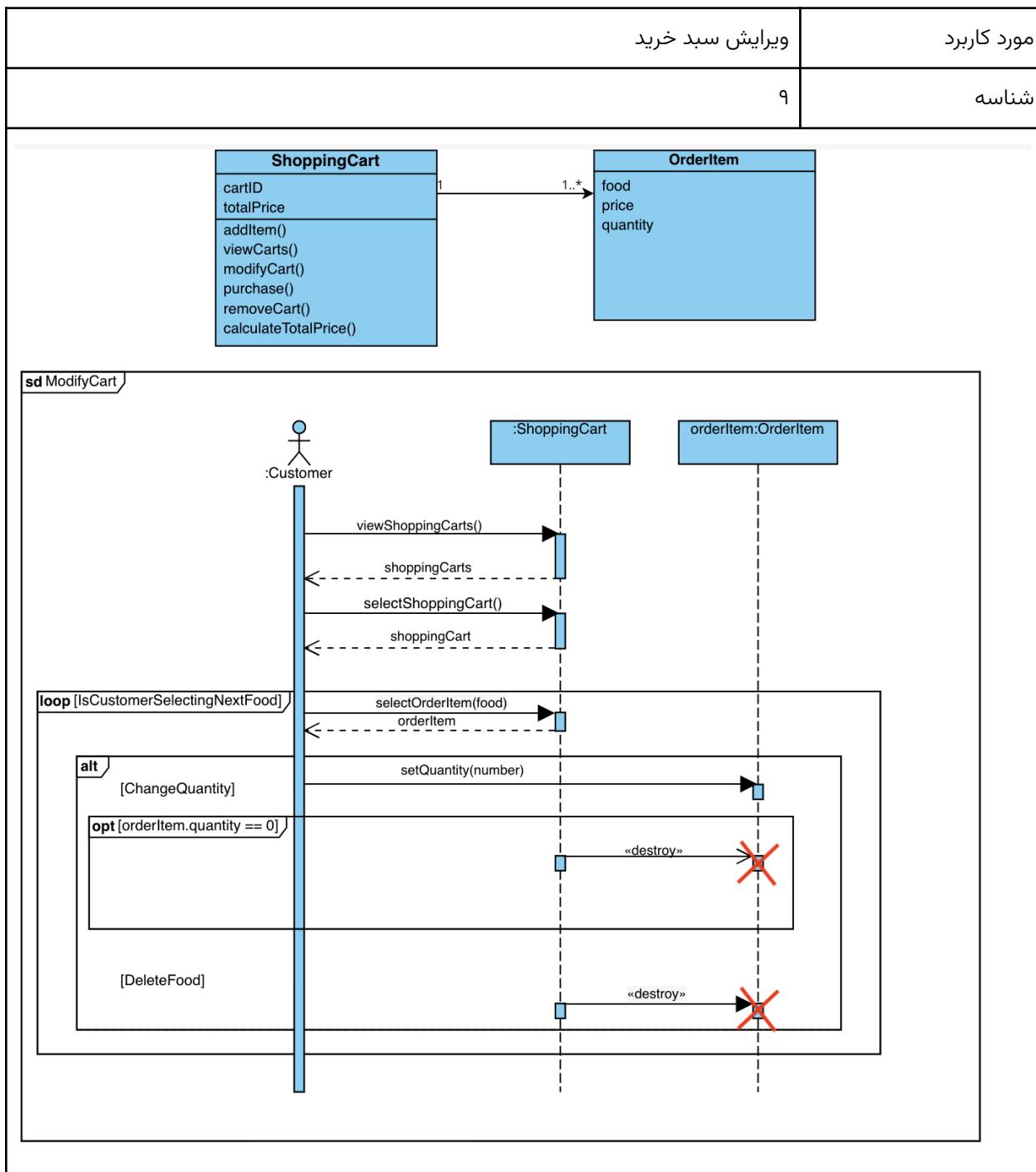
زیرسیستم مدیریت مشتریان



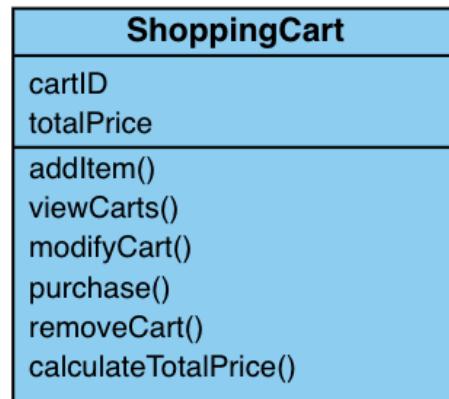




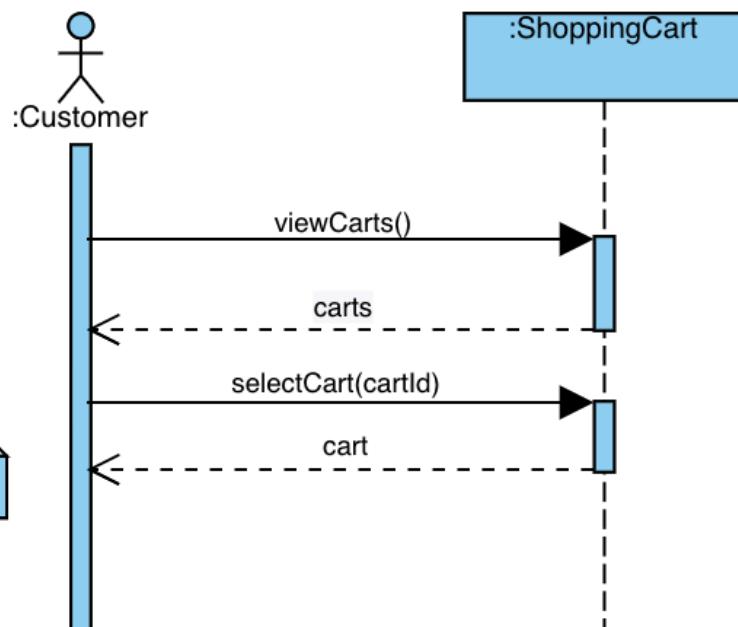


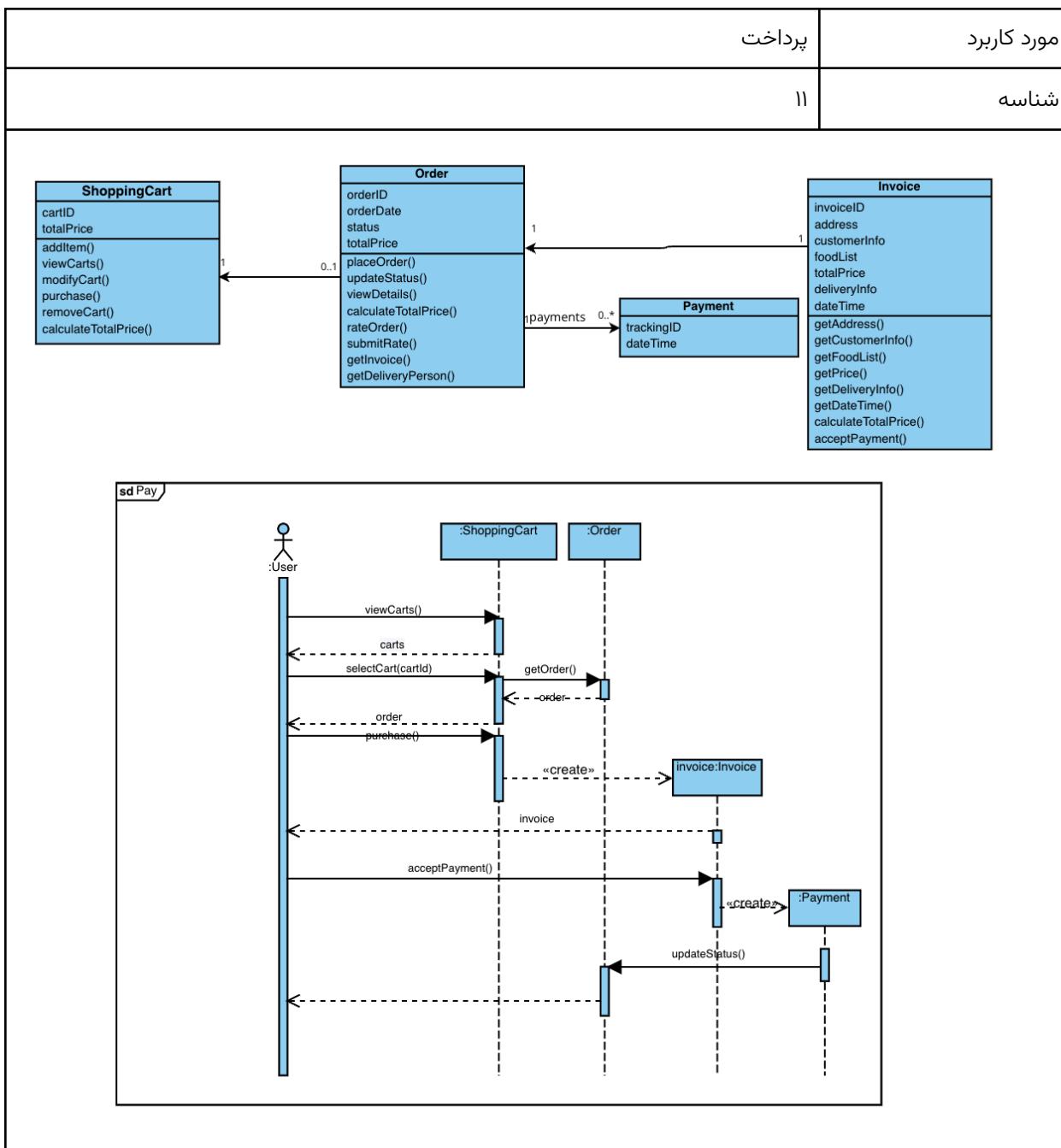


انتخاب و مشاهده یک سبد خرید	مورد کاربرد
۱۵	شناسه

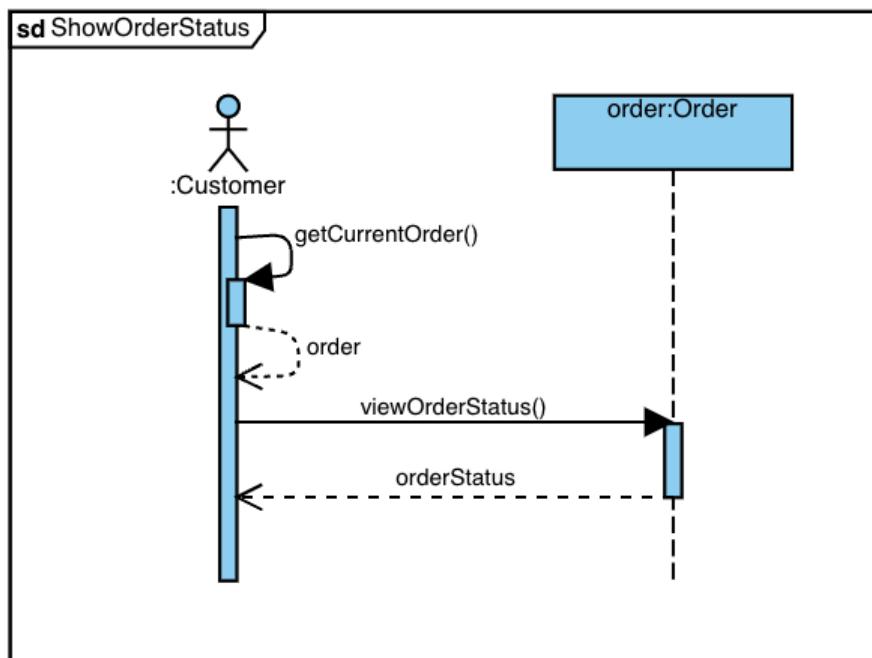


sd ShowCart

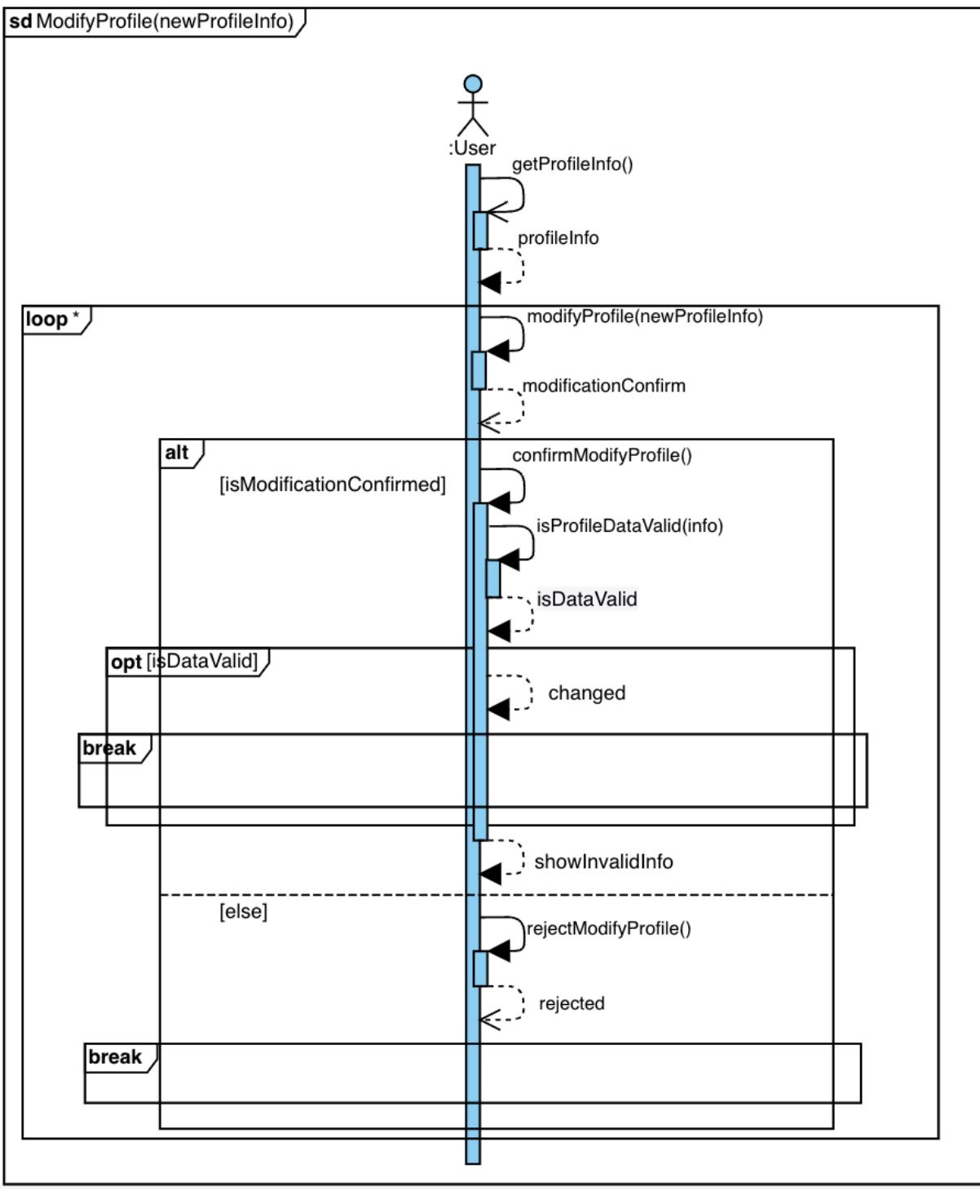
cart has
totalPrice, too.



مشاهده‌ی وضعیت فعلی سفارش	مورد کاربرد
۱۲	شناسه



امکان ویرایش اطلاعات	مورد کاربرد
۱۳	شناسه



مشاهده‌ی امتیاز هر رستوران	مورد کاربرد
	شناسه

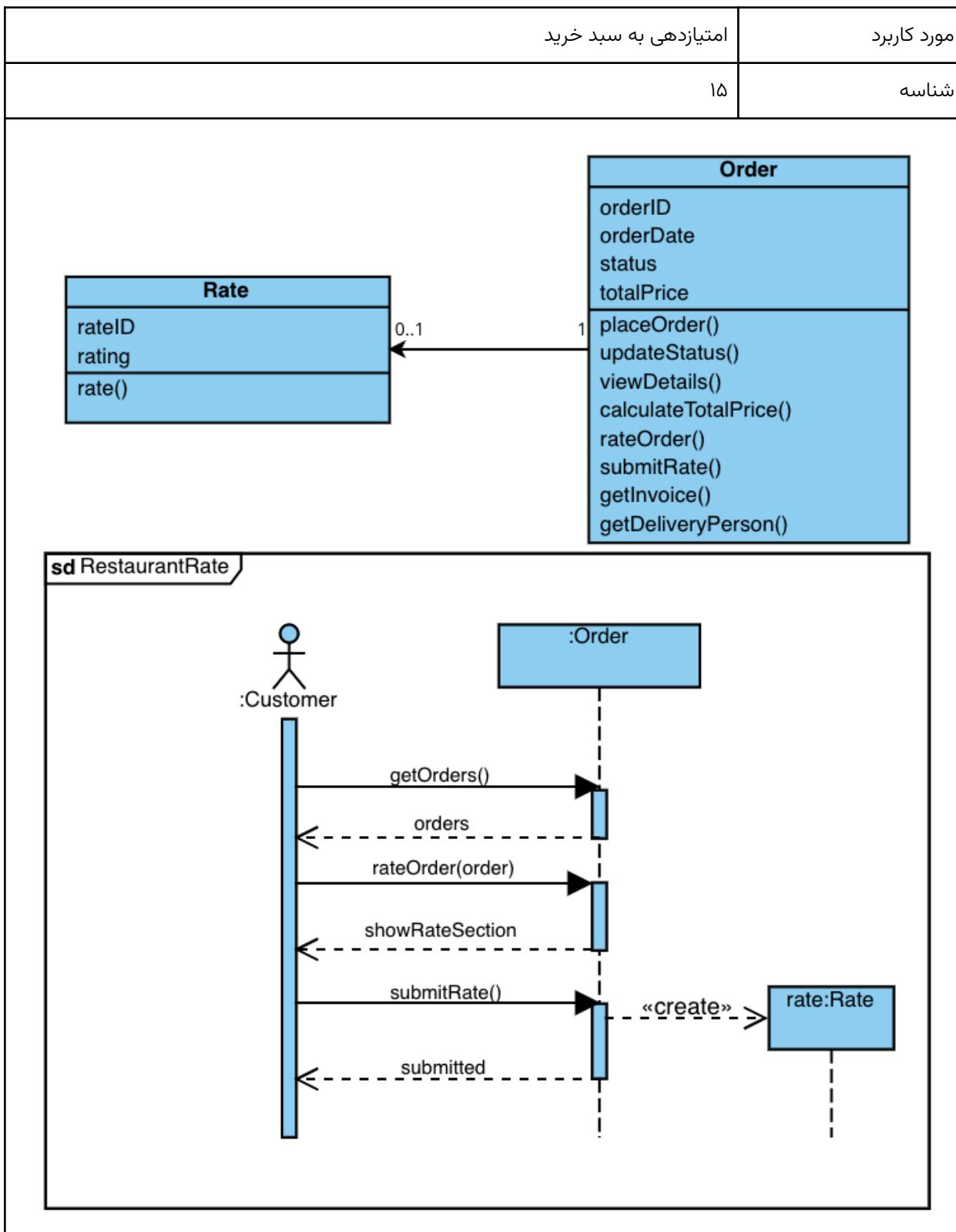

```

classDiagram
    class User {
        id
        name
        phoneNumber
        getProfileInfo()
        modifyProfile()
        confirmModifyProfile()
        isProfileDataValid()
        rejectModifyProfile()
        announce()
    }
    class Customer {
        address
        getCurrentOrder()
        getOrders()
        shareLocation()
    }
    class RestaurantOwner {
        menu
        address
        rate
        addFood()
        deleteFoodItem()
        viewActiveOrders()
        changeOrderStatus()
        viewOrdersHistory()
        prepareOrder()
        getRestaurantsOwner()
        getMenu()
        calculateRate()
        shareLocation()
    }
    class DeliveryPerson {
        location
        currentOrderId
        viewAssignedOrder()
        updateOrderStatus()
        sendLocation()
        assignOrder()
        getRandomFreeDeliveryPerson()
        updateDeliveryPersonStatus()
        getFreeDeliveryPersonWithAI()
        changeActivityStatus()
        addToActiveList()
        removeFromActiveList()
    }
    User <|-- Customer
    User <|-- RestaurantOwner
    User <|-- DeliveryPerson
  
```

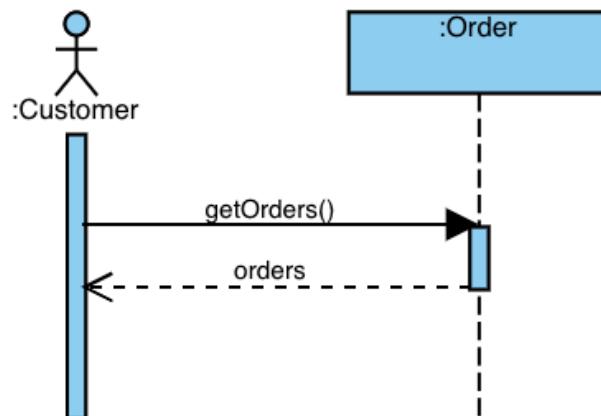


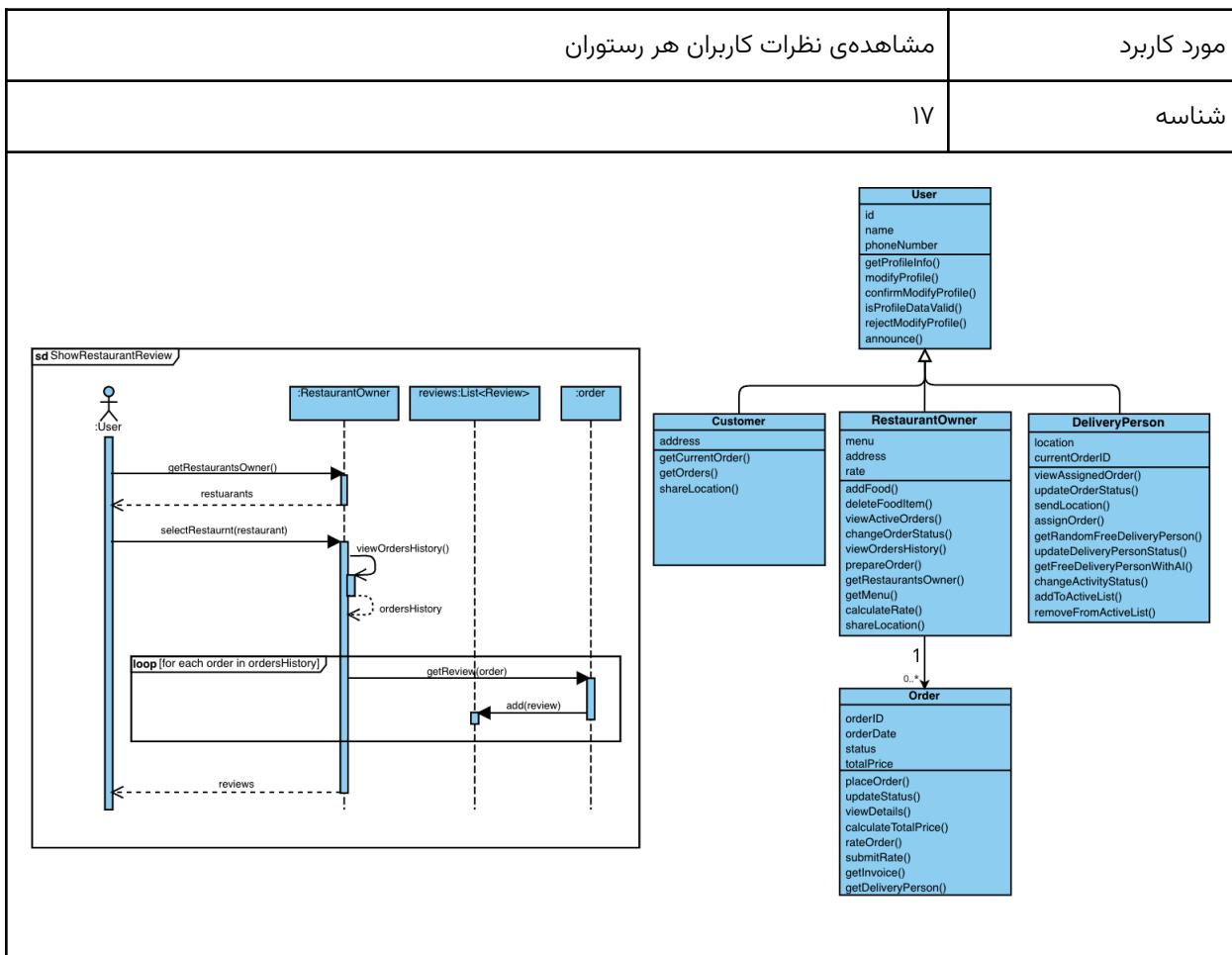
```

sequenceDiagram
    participant User
    participant RestaurantOwner
    User->>RestaurantOwner: getRestaurantsOwner()
    activate RestaurantOwner
    RestaurantOwner->>User: restaurants
    User->>RestaurantOwner: selectRestaurnt(restaurant)
    activate RestaurantOwner
    RestaurantOwner->>User: calculateRate()
    activate User
    User-->>User: restaurant has rate too.
    deactivate User
    deactivate RestaurantOwner
  
```

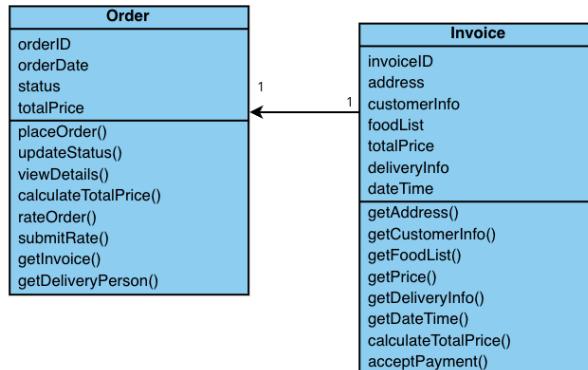
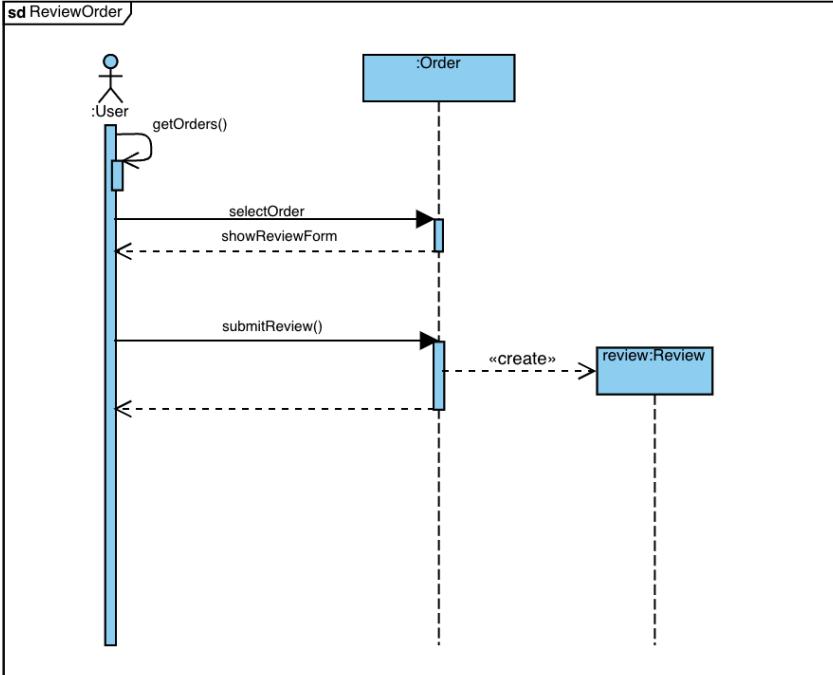


مشاهده‌ی تاریخچه‌ی سفارش‌ها	مورد کاربرد
۱۶	شناسه

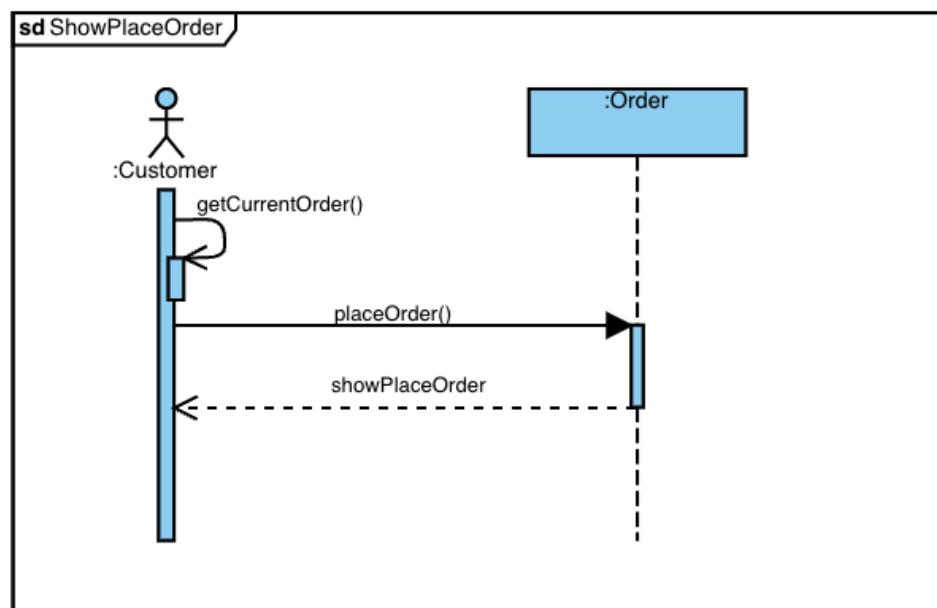
**sd ShowOrdersHistory**

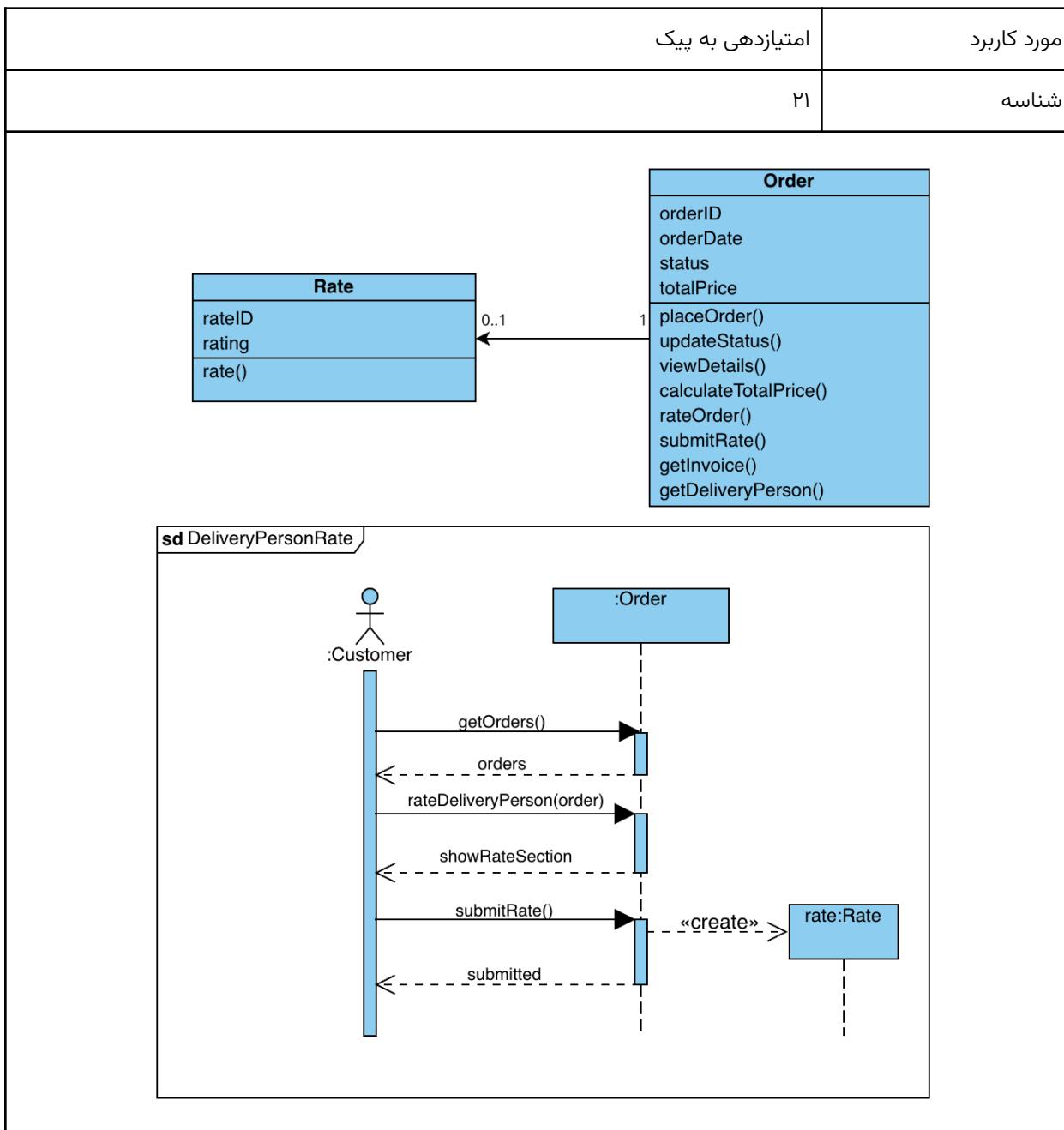


مشاهده‌ی توضیحات غذای یک رستوران	مورد کاربرد
۱۸	شناسه
<pre> sd ShowFoodDescription participant Customer participant RestaurantOwner participant Menu Note over Customer, RestaurantOwner, Menu : < -- User Customer->> RestaurantOwner : getRestaurantsOwner() activate RestaurantOwner RestaurantOwner-->> Customer : restaurants deactivate RestaurantOwner Customer->> Menu : getMenu(restaurant) activate Menu Menu-->> Customer : menu deactivate Menu Customer->> Menu : selectFood(food) activate Menu Menu-->> Customer : showDescription deactivate Menu </pre>	<p>The diagram illustrates the class structure and associations for the food distribution system:</p> <ul style="list-style-type: none"> User is the superclass, with Customer, RestaurantOwner, and DeliveryPerson as subclasses. Customer has attributes: address, menu, and rate. Methods include: getAddress(), getCurrentOrder(), getOrders(), shareLocation(), addFood(), deleteFoodItem(), viewActiveOrders(), changeOrderStatus(), viewOrdersHistory(), getRestaurantsOwner(), getMenu(), calculateRate(), and shareLocation(). RestaurantOwner has attributes: menu, address, and rate. Methods include: addFood(), deleteFoodItem(), viewActiveOrders(), changeOrderStatus(), viewOrdersHistory(), getRestaurantsOwner(), getMenu(), calculateRate(), and shareLocation(). DeliveryPerson has attributes: location and currentOrderID. Methods include: viewAssignedOrder(), updateOrderStatus(), sendLocation(), assignOrder(), getRandomFreeDeliveryPerson(), updateDeliveryPersonStatus(), getFreeDeliveryPersonWithAI(), changeActivityStatus(), addToActiveList(), and removeFromActiveList(). Menu has attributes: menuID and foodList. Methods include: getMenu() and calculateRate().

نظردهی به سفارش	مورد کاربرد
	شناسه
	۱۹
 <pre> classDiagram class Order { orderId orderDate status totalPrice placeOrder() updateStatus() viewDetails() calculateTotalPrice() rateOrder() submitRate() getInvoice() getDeliveryPerson() } class Invoice { invoiceID address customerInfo foodList totalPrice deliveryInfo date getAddress() getCustomerInfo() getFoodList() getPrice() getDeliveryInfo() getDate() calculateTotalPrice() acceptPayment() } Order "1" --> "1" Invoice </pre>	
 <pre> sequenceDiagram participant User participant Order participant Review User->>Order: getOrders() activate Order User->>Order: selectOrder activate Order User->>Order: showReviewForm deactivate Order User->>Review: submitReview() activate Review Note over Review: «create» deactivate Review </pre>	

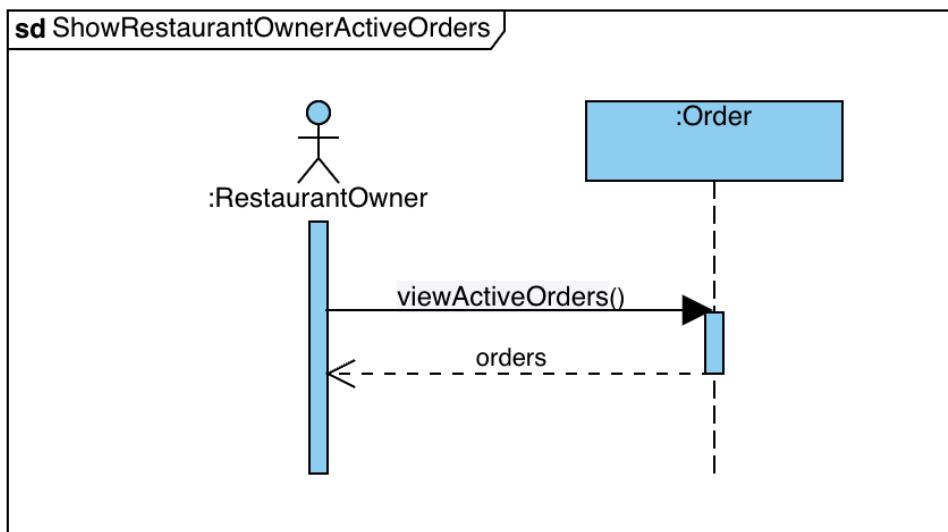
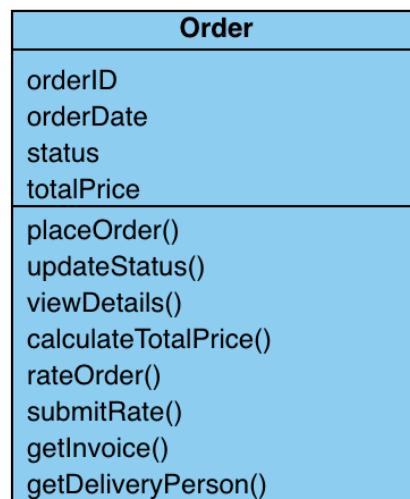
مشاهده‌ی محل فعلی سفارش	مورد کاربرد
۲۰	شناسه



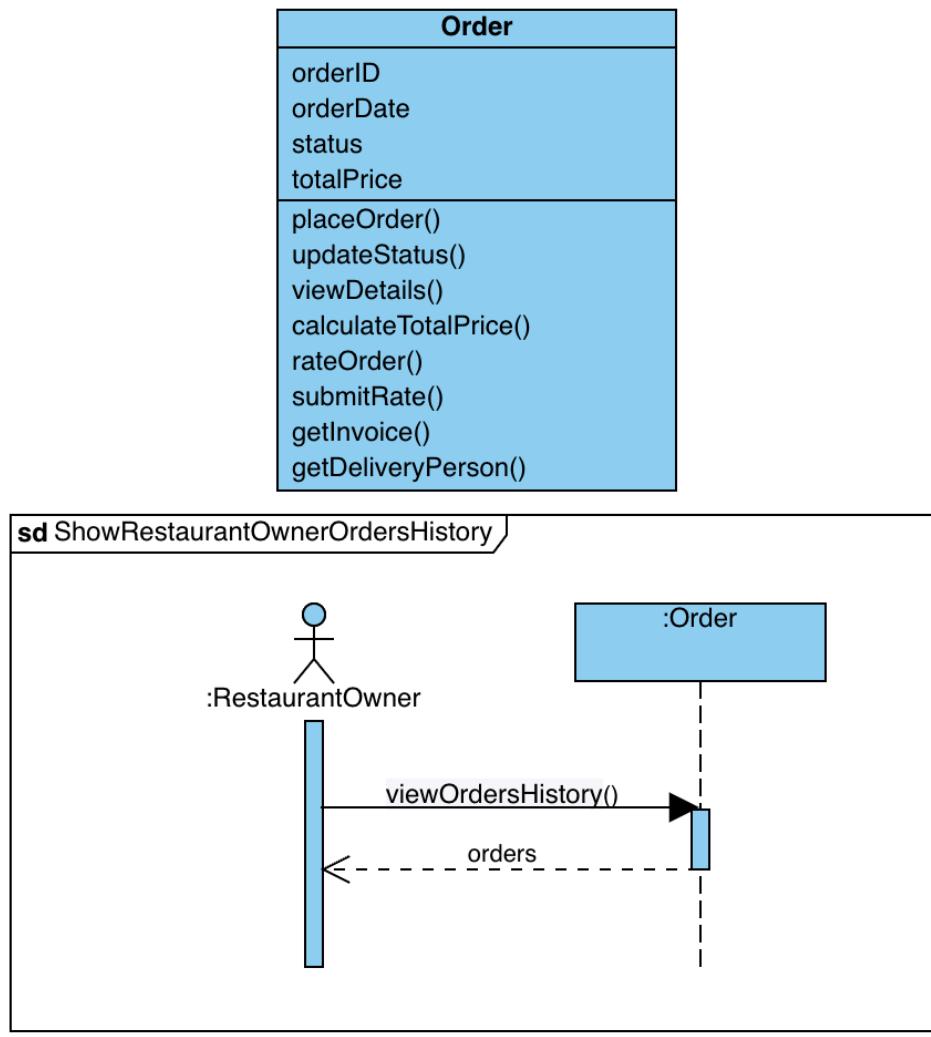


زیرسیستم مدیریت تهیه‌کنندگاه غذا

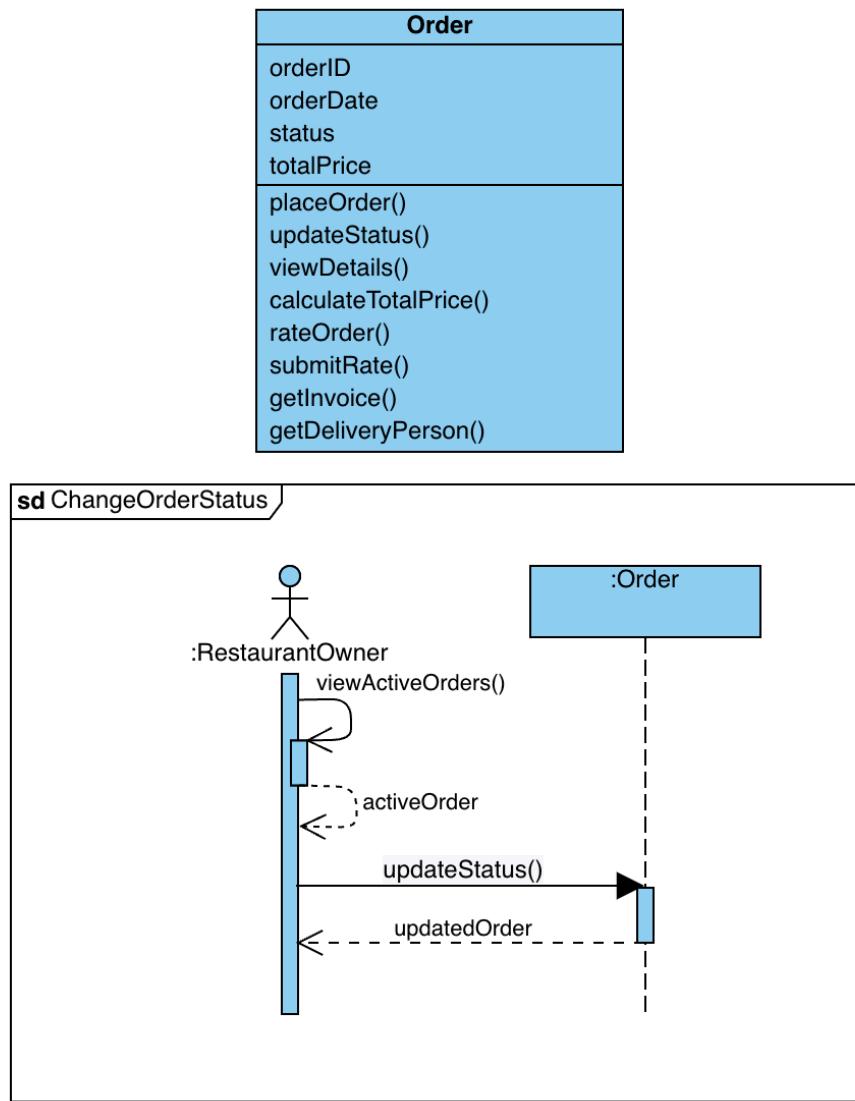
دیدن لیست سفارشات	مورد کاربرد
۲۲	شناسه



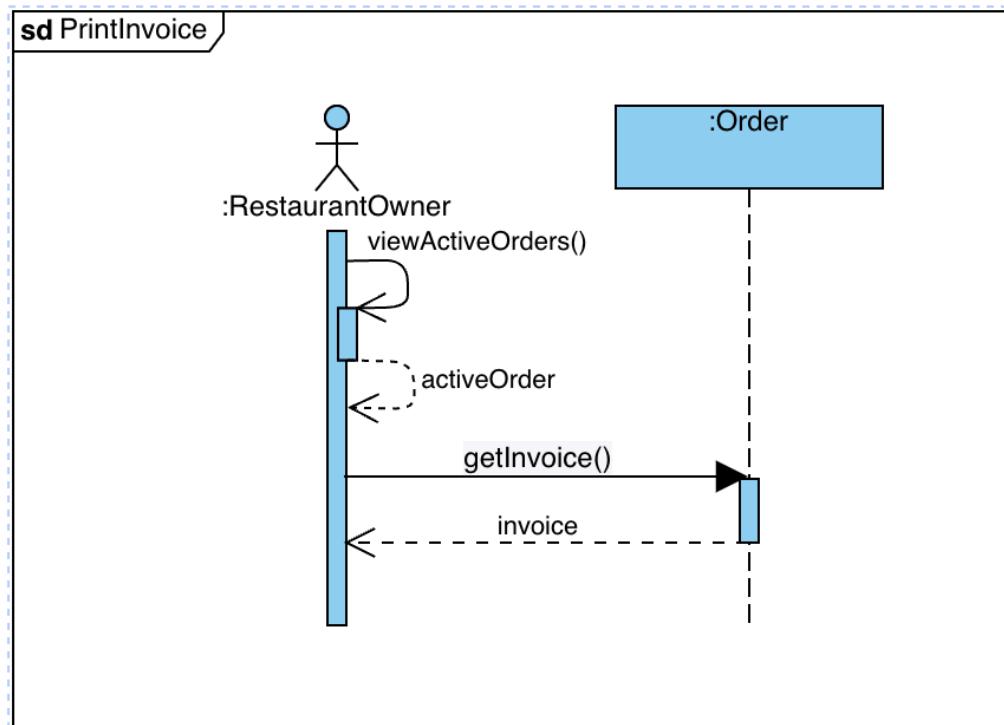
دیدن لیست تاریخچه‌ی سفارشات	مورد کاربرد
۲۳	شناسه



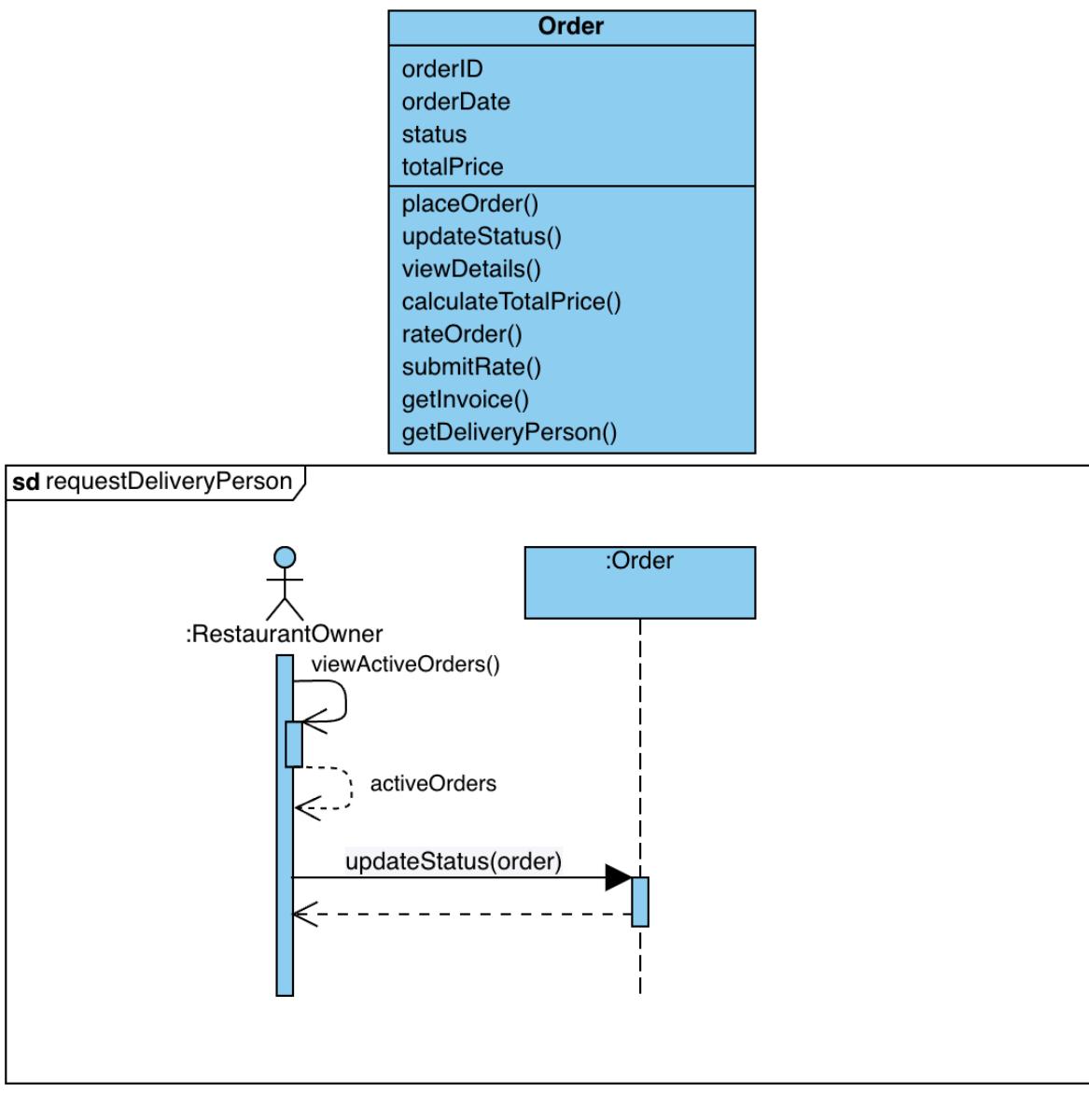
تغییر دادن وضعیت هر سفارش	مورد کاربرد
۲۴	شناسه



چاپ فاکتور	مورد کاربرد
۲۵	شناسه



درخواست پیک	مورد کاربرد
۲۶	شناسه



تعیین خودکار یک پیک برای هر سفارش	مورد کاربرد
	۲۷
<pre> sequenceDiagram participant RO as RestaurantOwner participant O as Order participant DP as DeliveryPerson participant U as User participant C as Customer participant RO2 as RestaurantOwner participant O2 as Order participant I as Invoice RO->>O: updateStatus() activate O O->>DP: assignOrder(order) activate DP DP->>U: getRandomFreeDeliveryPerson() activate U U-->>DP: freeDeliveryPerson DP->>O: assignOrder(order) activate O O-->>DP: updateDeliveryPersonStatus(freeDeliveryPerson) activate DP DP-->>O: statusUpdates activate O O-->>DP: deliveryPersonAssigned(order) activate DP DP-->>I: create activate I I-->>RO: assigned deactivate I deactivate DP deactivate O deactivate U deactivate DP deactivate O deactivate RO </pre> <p>The sequence diagram illustrates the process of assigning a delivery person to an order. It starts with a RestaurantOwner updating the order status. The order then triggers an assignment to a delivery person. The delivery person retrieves a random free delivery person from the user system. Once assigned, the delivery person updates the delivery person status and sends status updates back to the order. Finally, the order creates an invoice, which is then assigned to the delivery person.</p>	شناسه

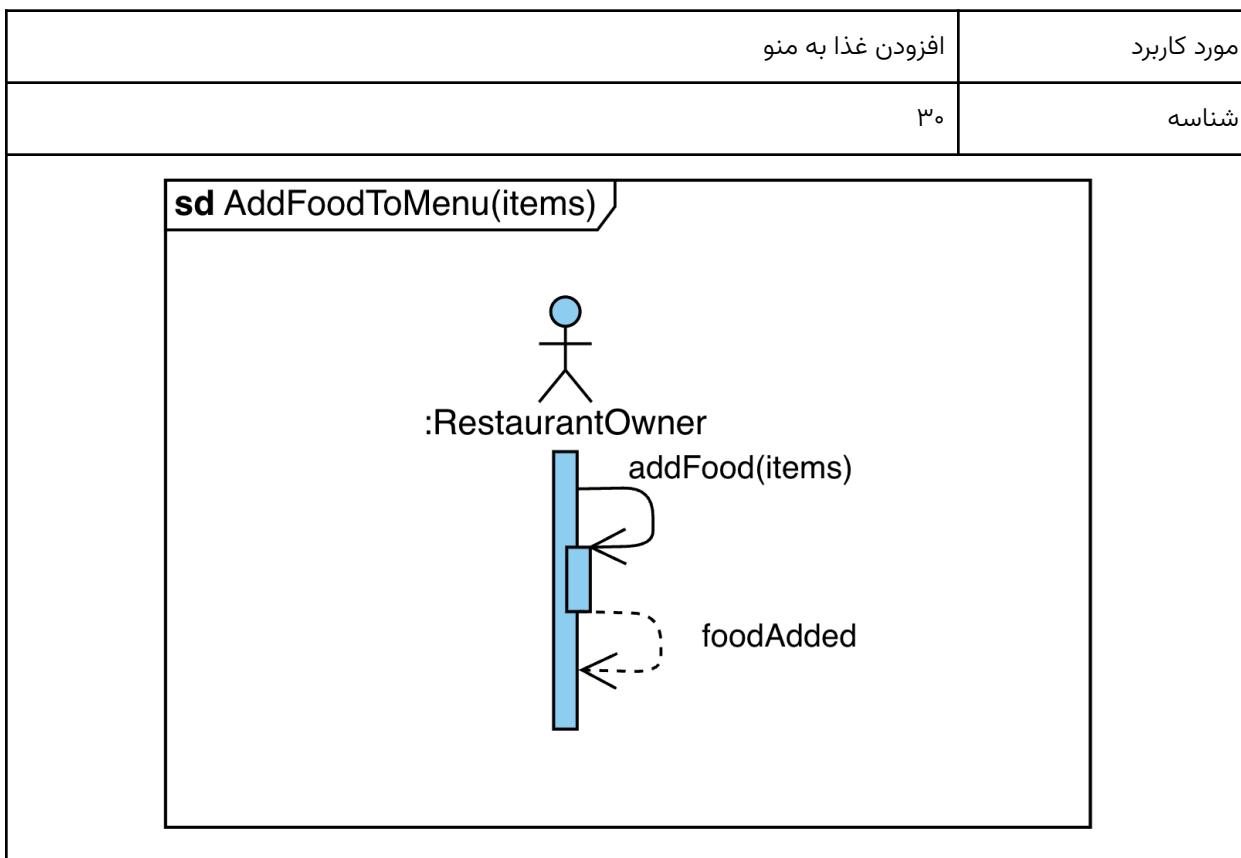
مشاهده‌ی پیکهای حمل‌کننده‌ی سفارشات آن رستوران	مورد کاربرد
	۲۸
<pre> classDiagram class Order { orderID orderDate status totalPrice placeOrder() updateStatus() viewDetails() calculateTotalPrice() rateOrder() submitRate() getInvoice() getDeliveryPerson() } class Invoice { invoiceID address customerInfo foodList totalPrice deliveryInfo dateTime getAddress() getCustomerInfo() getFoodList() getPrice() getDeliveryInfo() getDate() calculateTotalPrice() acceptPayment() } Order "1" --> "1" Invoice </pre>	شناسه

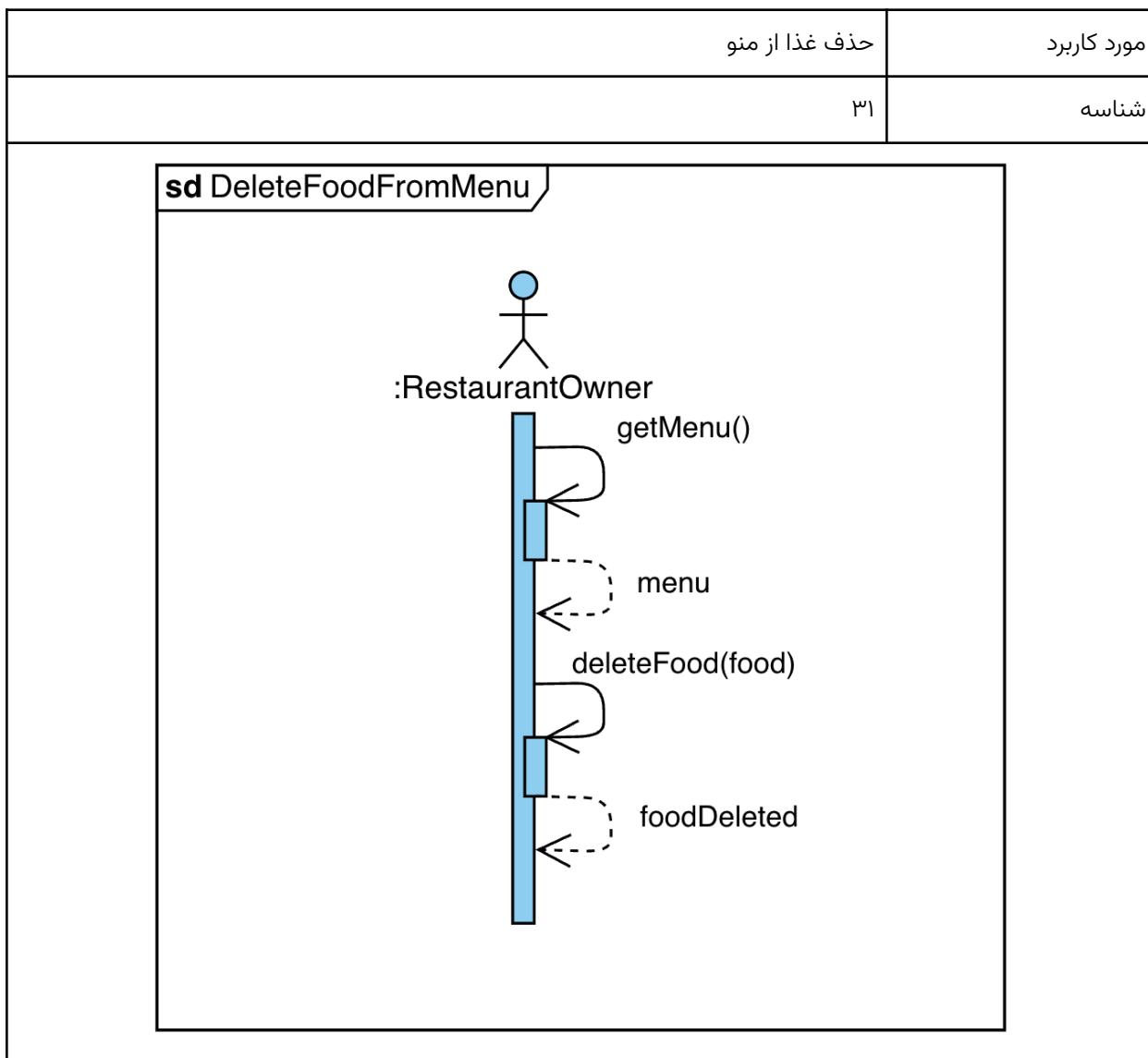
sd ShowAssignDeliveryPerson

```

sequenceDiagram
    participant RO as :RestaurantOwner
    participant O as :Order
    participant I as :Invoice
    RO->>O: viewActiveOrders()
    activate O
    O-->>RO: activeOrders
    deactivate O
    RO->>O: getDeliveryPerson(order)
    activate O
    O->>I: getDeliveryInfo()
    activate I
    I-->>O: deliveryInfo
    deactivate I
    deactivate O
  
```

تعیین هوشمند یک پیک برای هر سفارش	مورد کاربرد
۲۹	شناسه
<pre> sequenceDiagram participant RO as RestaurantOwner participant O as Order participant DP as DeliveryPerson participant U as User participant I as Invoice RO->>O: updateStatus() activate O O->>DP: assignOrder(order) activate DP Note over DP: The AI algorithm is currently unknown because it is a WIP use case. DP-->>O: freeDeliveryPerson activate O O-->>DP: assignOrder(order) activate DP DP-->>O: updateDeliveryPersonStatus(freedeliveryPerson) activate O O-->>DP: statusUpdates activate DP Note over DP: deliveryPersonAssigned(order) DP-->>I: create activate I I-->>RO: assigned deactivate I deactivate O deactivate DP </pre> <p>The diagram illustrates a sequence of interactions for assigning a delivery person to an order. It starts with a RestaurantOwner updating the order status. The order then triggers an assignment to a delivery person. A note indicates that the AI algorithm is currently unknown. The delivery person sends a 'freeDeliveryPerson' message back to the order. The order then sends an 'assignOrder(order)' message to the delivery person. The delivery person updates the delivery person's status and sends a note indicating the delivery person has been assigned. Finally, the delivery person creates an invoice.</p>	User id name phoneNumber getProfile() modifyProfile() isProfileDataValid() rejectModifyProfile() announce() Customer address getCurrentOrder() getOrders() shareLocation() RestaurantOwner menu address rate addFoodItem() deleteFoodItem() viewActiveOrders() changeOrderStatus() viewOrdersStatus() prepareOrder() getRestaurantsOwner() getMenu() calculateRate() shareLocation() DeliveryPerson location currentOrderID viewAssignedOrder() updateOrderStatus() shareLocation() assignOrder() getRandomFreeDeliveryPerson() updateDeliveryPersonStatus() getFreeDeliveryPersonWithAI() changeActivityStatus() addToActiveList() removeFromActiveList() Order orderID orderDate status totalPrice placeOrder() updateStatus() viewDetails() calculateTotalPrice() rateOrder() submitRate() getInvoice() getDeliveryPerson() Invoice invoiceID address customerInfo foodList totalPrice deliveryInfo dateTime getAddress() getCustomerInfo() getFoodList() getPrice() getDeliveryInfo() getDateTime() calculateTotalPrice() acceptPayment()





تعیین موجودی هر غذا	مورد کاربرد
۳۲	شناسه

Food

```

foodID
name
price
description
stock
categoryID
getPrice()
getPicture()
getDescription()
getAvailability()
setAvailability()
updateInfo()
setCategory()

```

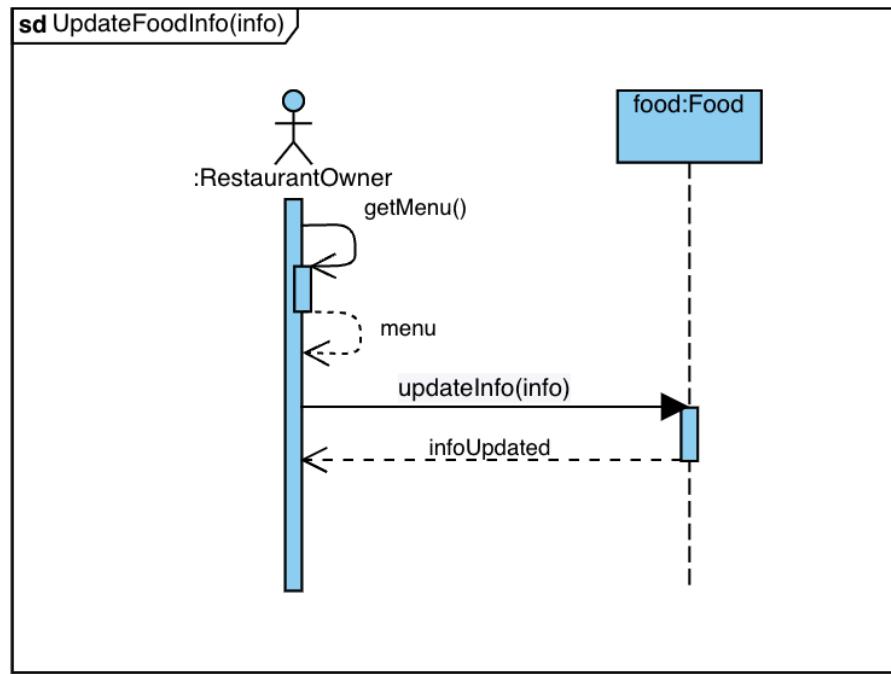
sd SetFoodAvailability

```

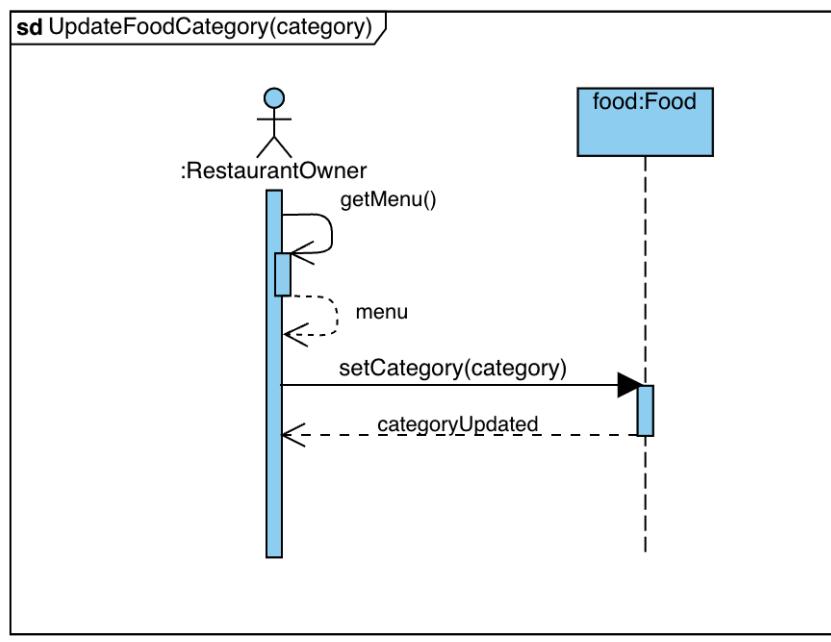
sequenceDiagram
    participant RO as :RestaurantOwner
    participant F as food:Food
    RO->>RO: getMenu()
    activate RO
    RO-->>F: setAvailability()
    deactivate RO
    F->>RO: availabilityUpdated

```

ویرایش غذا در منو	مورد کاربرد
۳۳	شناسه



افزودن دسته‌بندی برای هر غذا	مورد کاربرد
۳۴	شناسه



زیرسیستم مدیریت پیکها

تغییر دادن وضعیت هر سفارش	مورد کاربرد
۳۵	شناسه


```

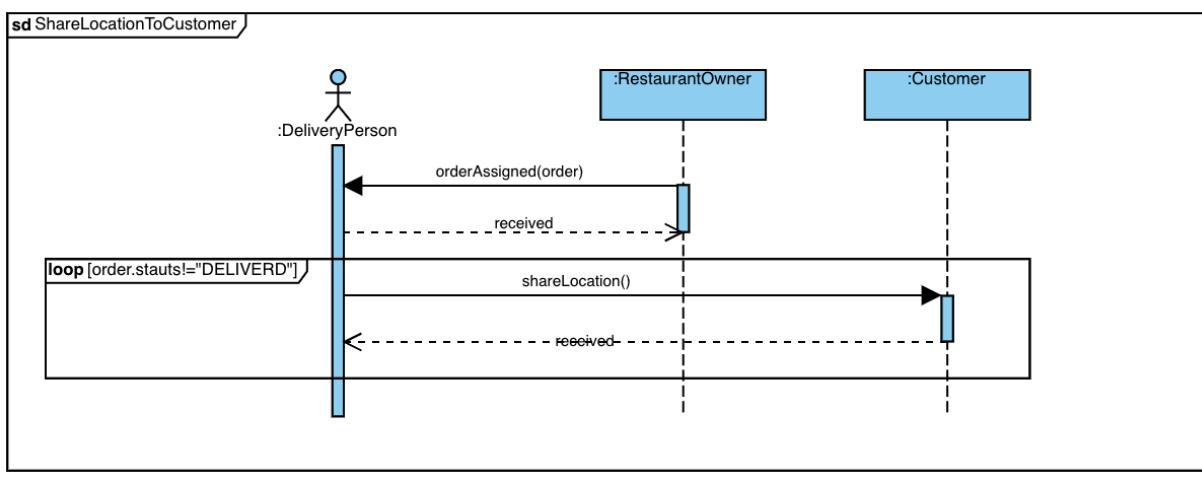
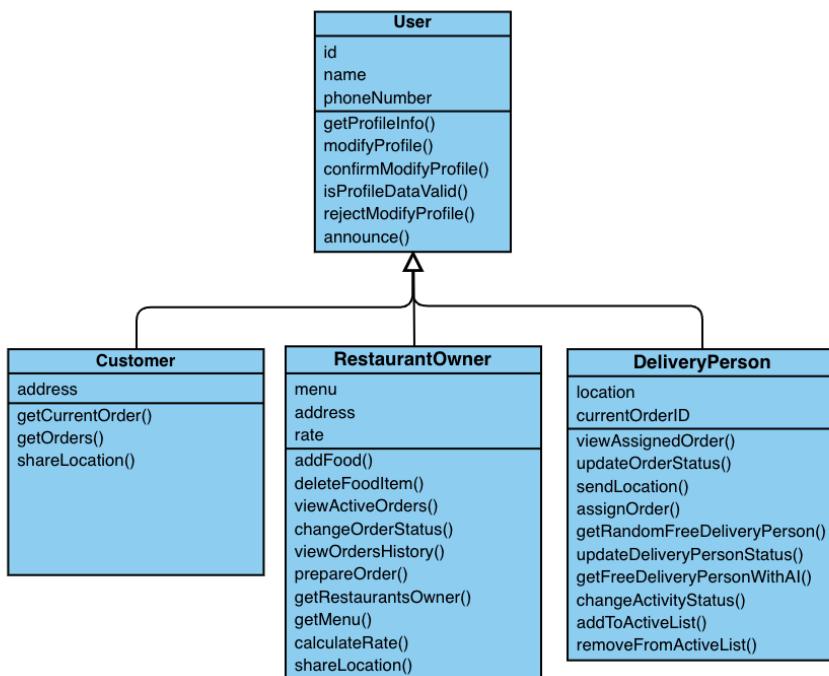
classDiagram
    class User {
        id
        name
        phoneNumber
        getProfileInfo()
        modifyProfile()
        confirmModifyProfile()
        isProfileDataValid()
        rejectModifyProfile()
        announce()
    }
    class Customer {
        address
        getCurrentOrder()
        getOrders()
        shareLocation()
    }
    class RestaurantOwner {
        menu
        address
        rate
        addFood()
        deleteFoodItem()
        viewActiveOrders()
        changeOrderStatus()
        viewOrdersHistory()
        prepareOrder()
        getRestaurantsOwner()
        getMenu()
        calculateRate()
        shareLocation()
    }
    class DeliveryPerson {
        location
        currentOrderId
        viewAssignedOrder()
        updateOrderStatus()
        sendLocation()
        assignOrder()
        getRandomFreeDeliveryPerson()
        updateDeliveryPersonStatus()
        getFreeDeliveryPersonWithAI()
        changeActivityStatus()
        addToActiveList()
        removeFromActiveList()
    }
    User <|-- Customer
    User <|-- RestaurantOwner
    User <|-- DeliveryPerson
  
```


sd ChangeStatusByDeliveryPerson

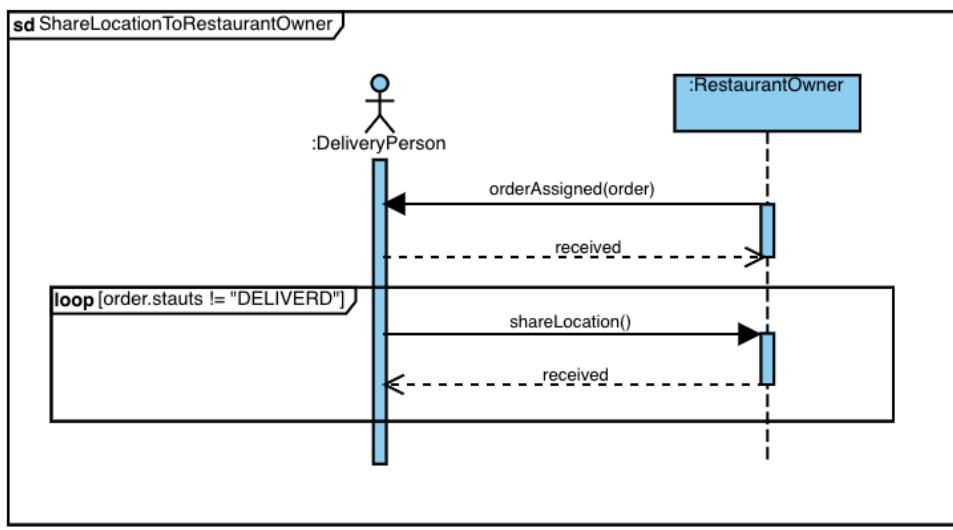
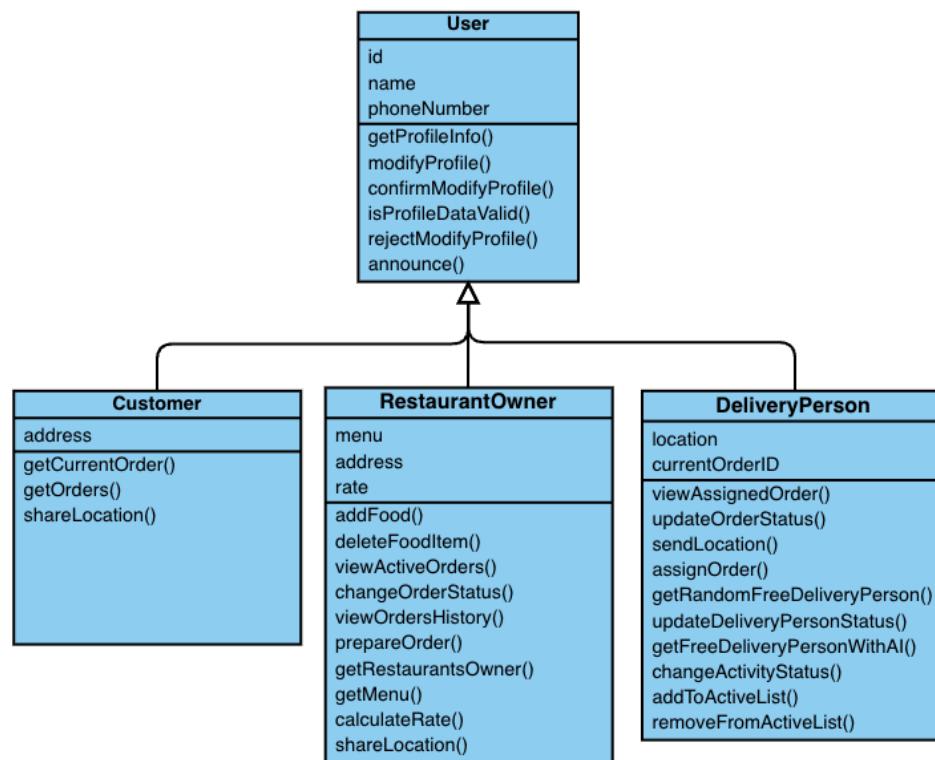
```

sequenceDiagram
    participant DP as :DeliveryPerson
    participant RO as :RestaurantOwner
    participant C as :Customer
    Note left of DP: 
    Note left of RO: 
    Note left of C: 
    DP->>RO: orderAssigned(order)
    activate RO
    RO-->>DP: received
    DP->>C: changeOrderStatus("DELIVERD")
    activate C
    C-->>DP: announce()
    activate DP
    DP-->>C: statusChanged
    deactivate C
    deactivate DP
  
```

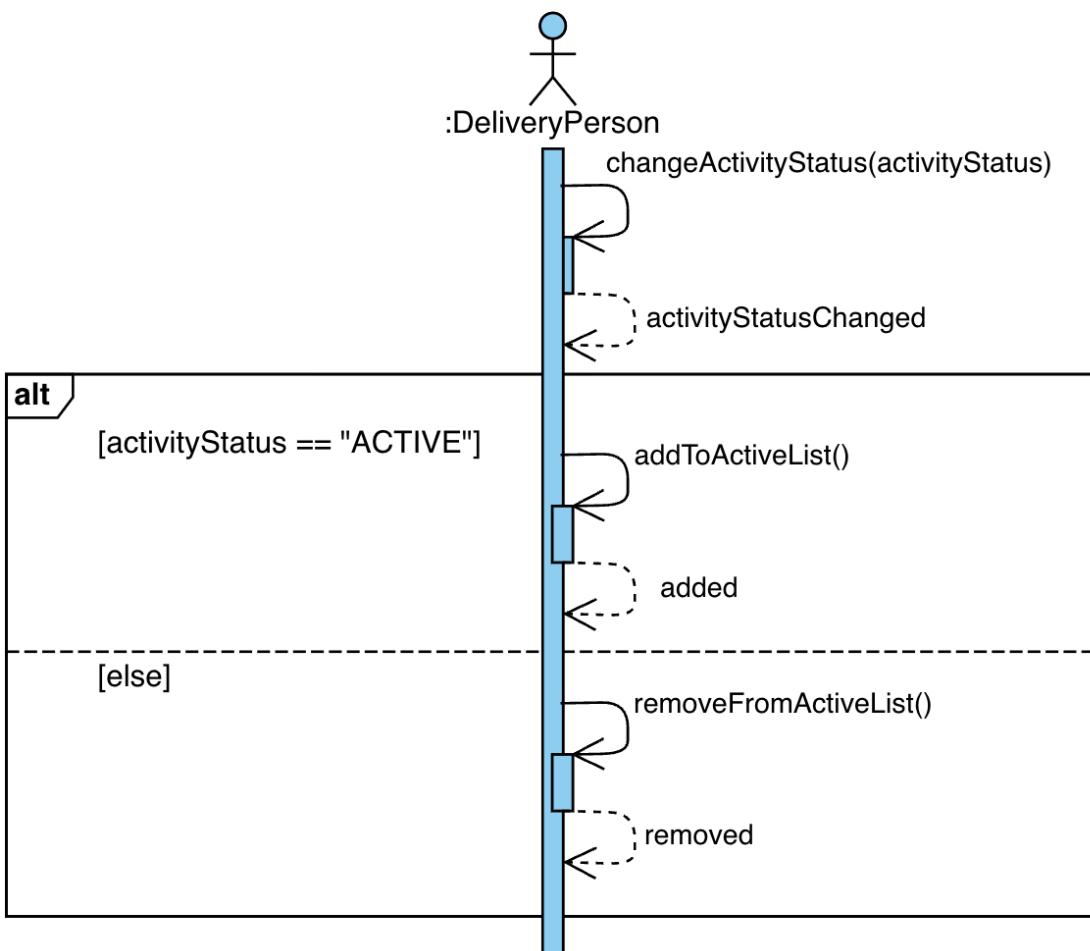
ارسال موقعیت به صورت زنده برای مشتری	مورد کاربرد
۳۶	شناسه



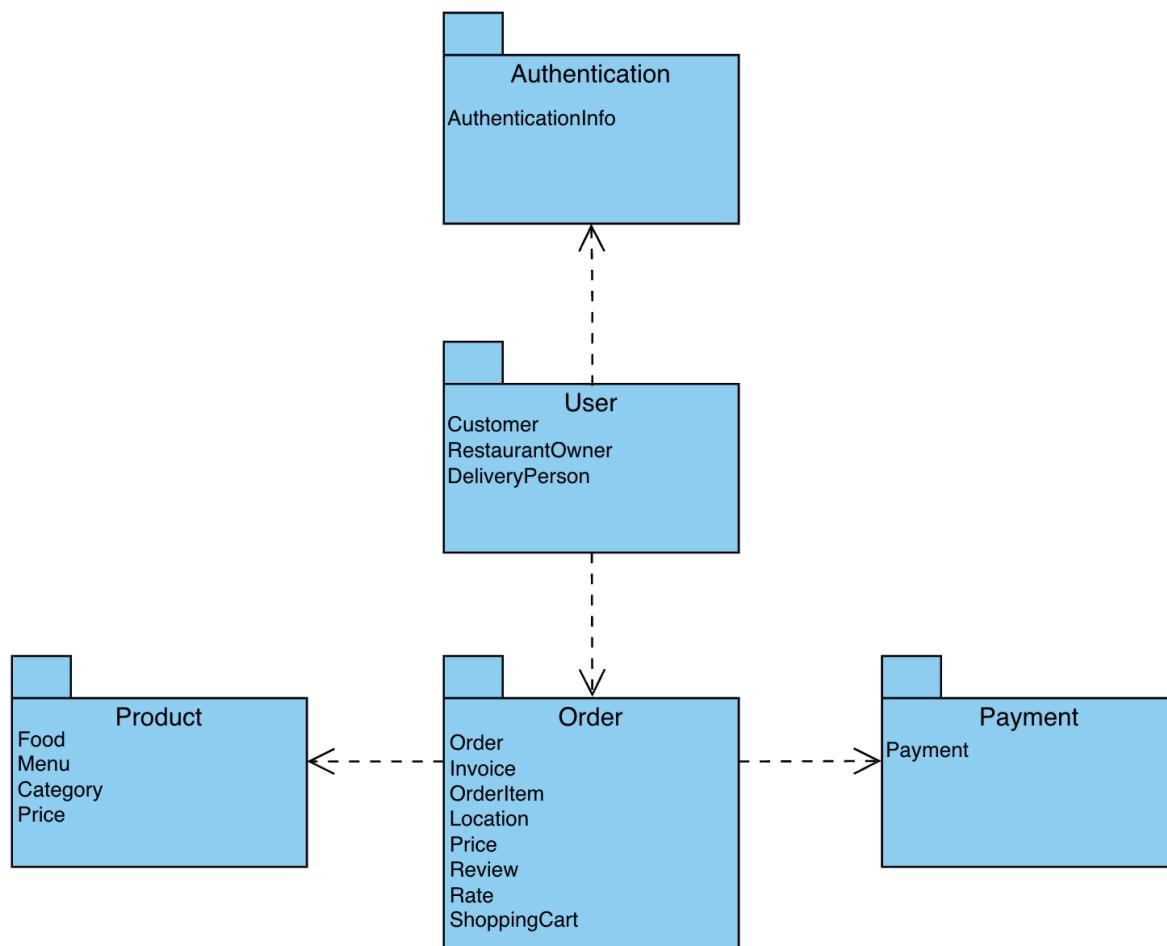
ارسال موقعیت به صورت زنده برای رستوران	مورد کاربرد
۳۷	شناسه



تغییر وضعیت بین فعال و غیرفعال	مورد کاربرد
۳۸	شناسه

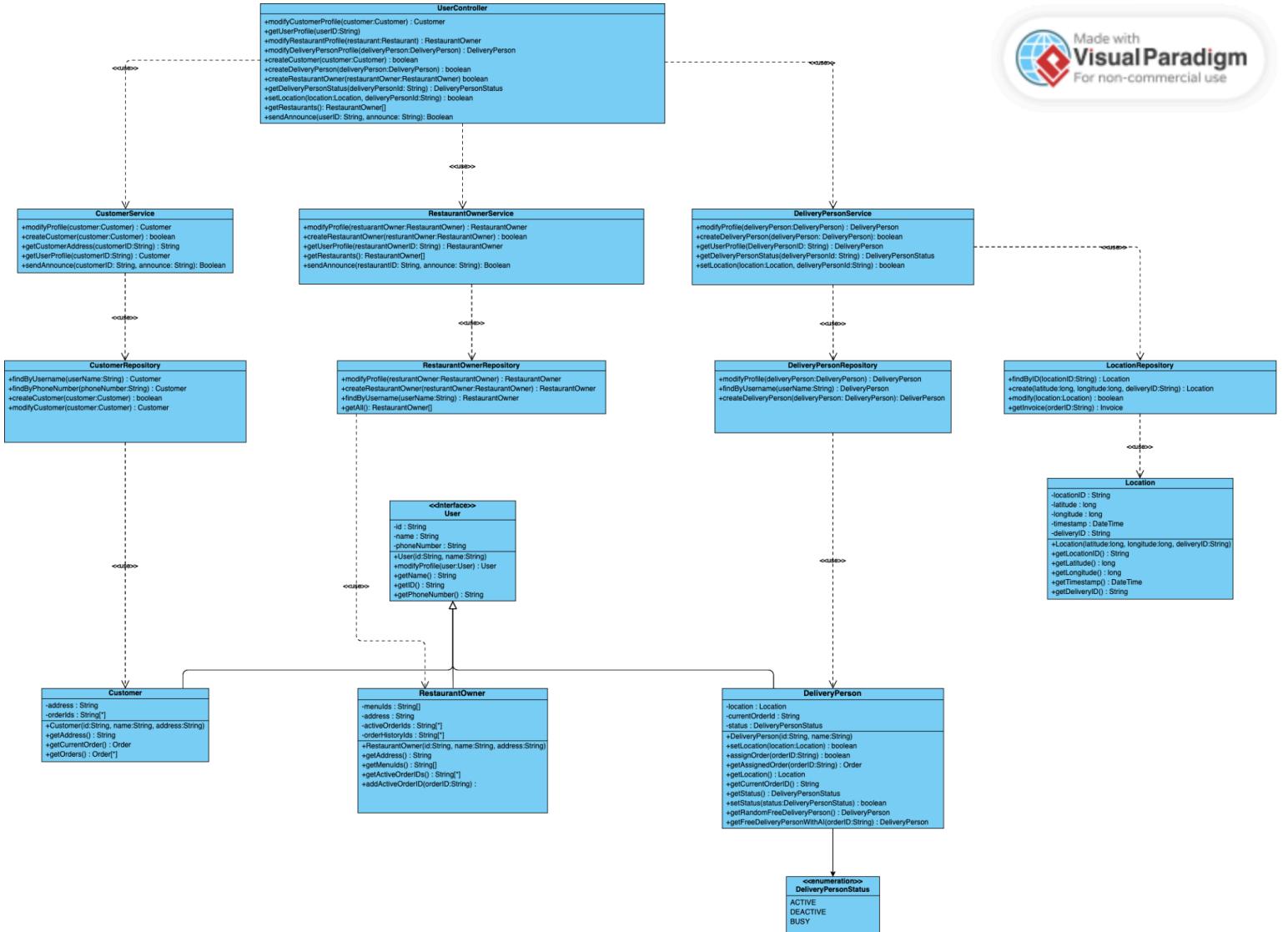
sd Changed(activityStatus)

نمودار بسته



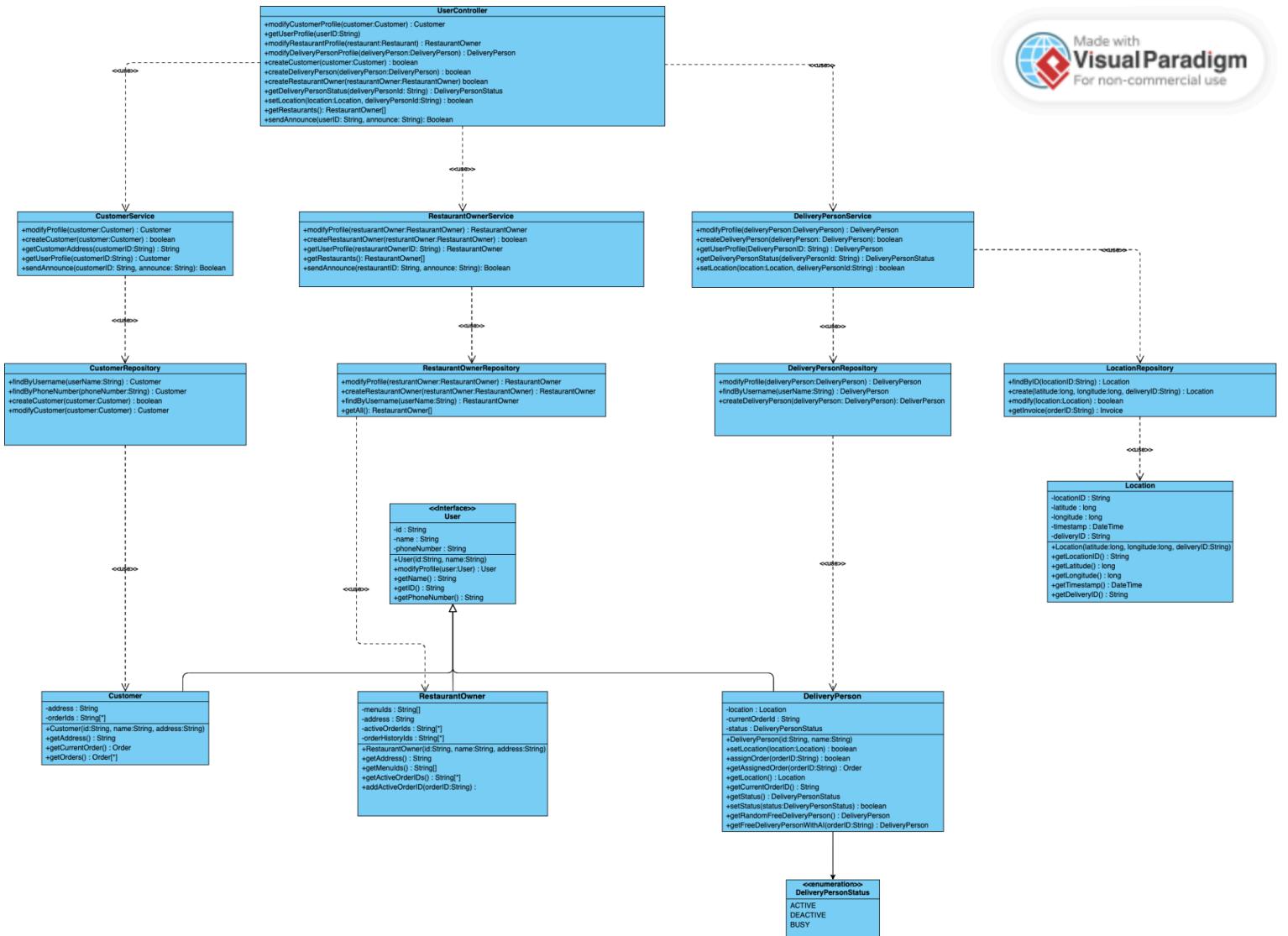
نمودار کلاس‌های طراحی

مولفه‌ی User

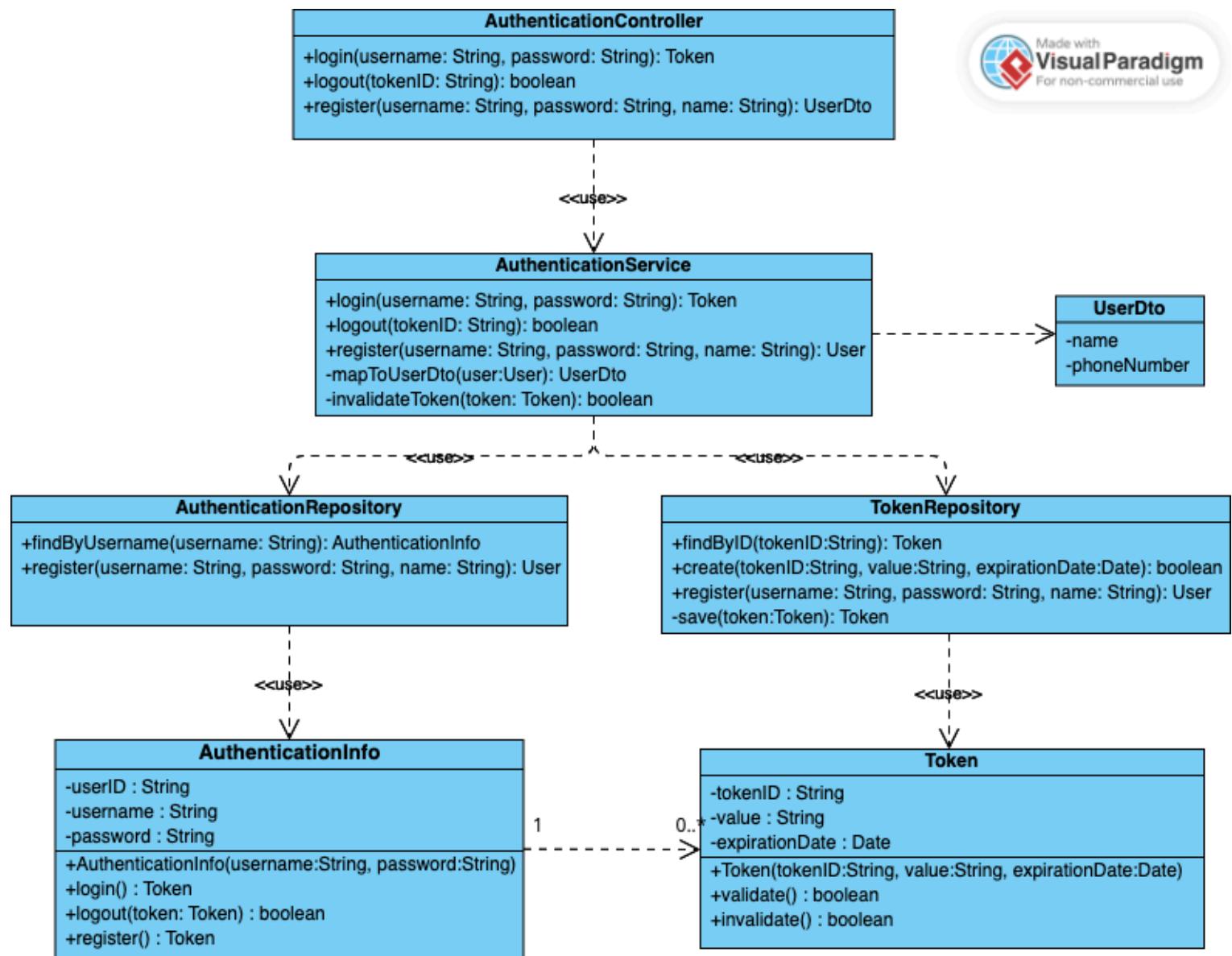


مولفه‌ی Order

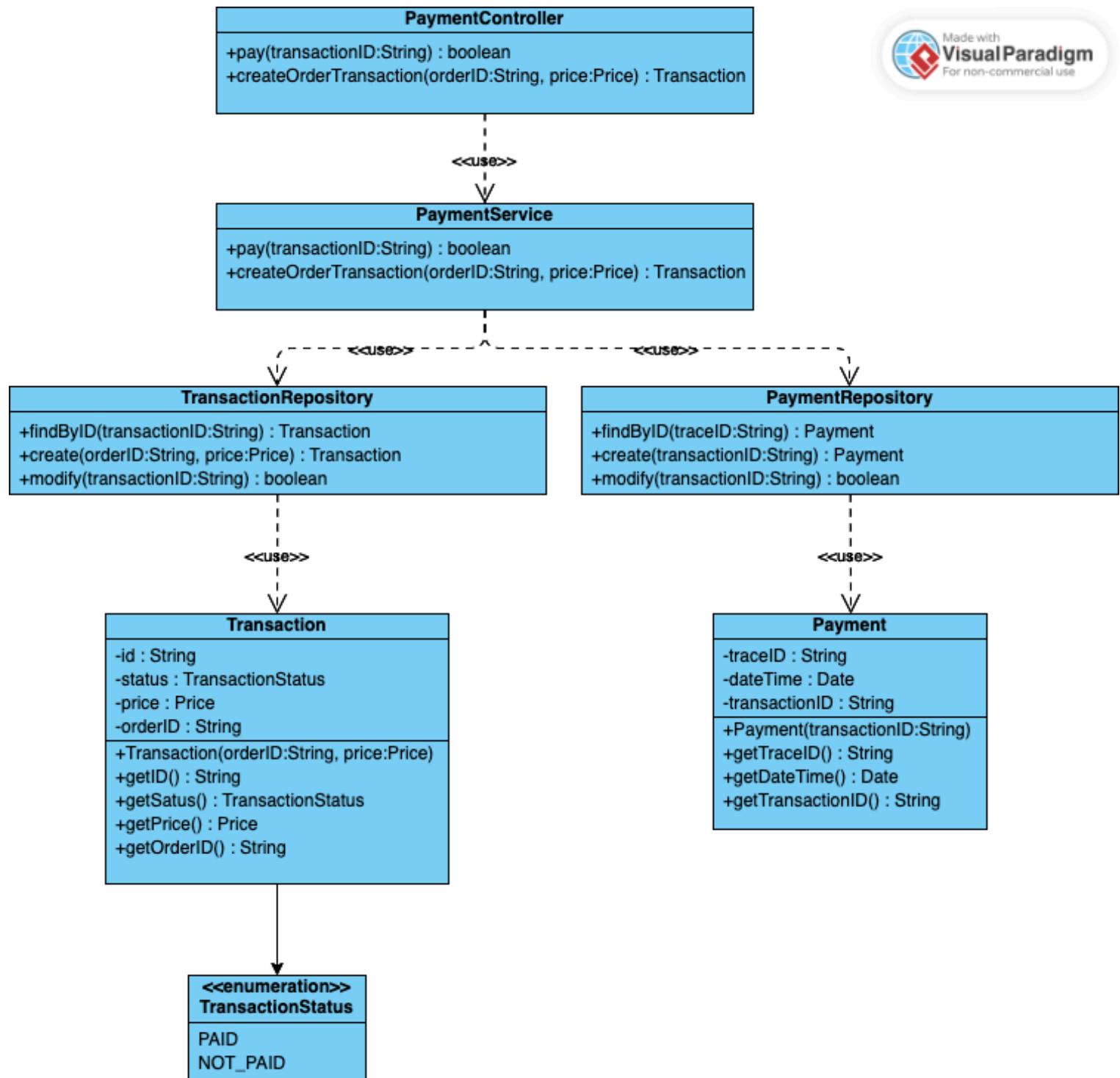

 Made with
Visual Paradigm
 For non-commercial use



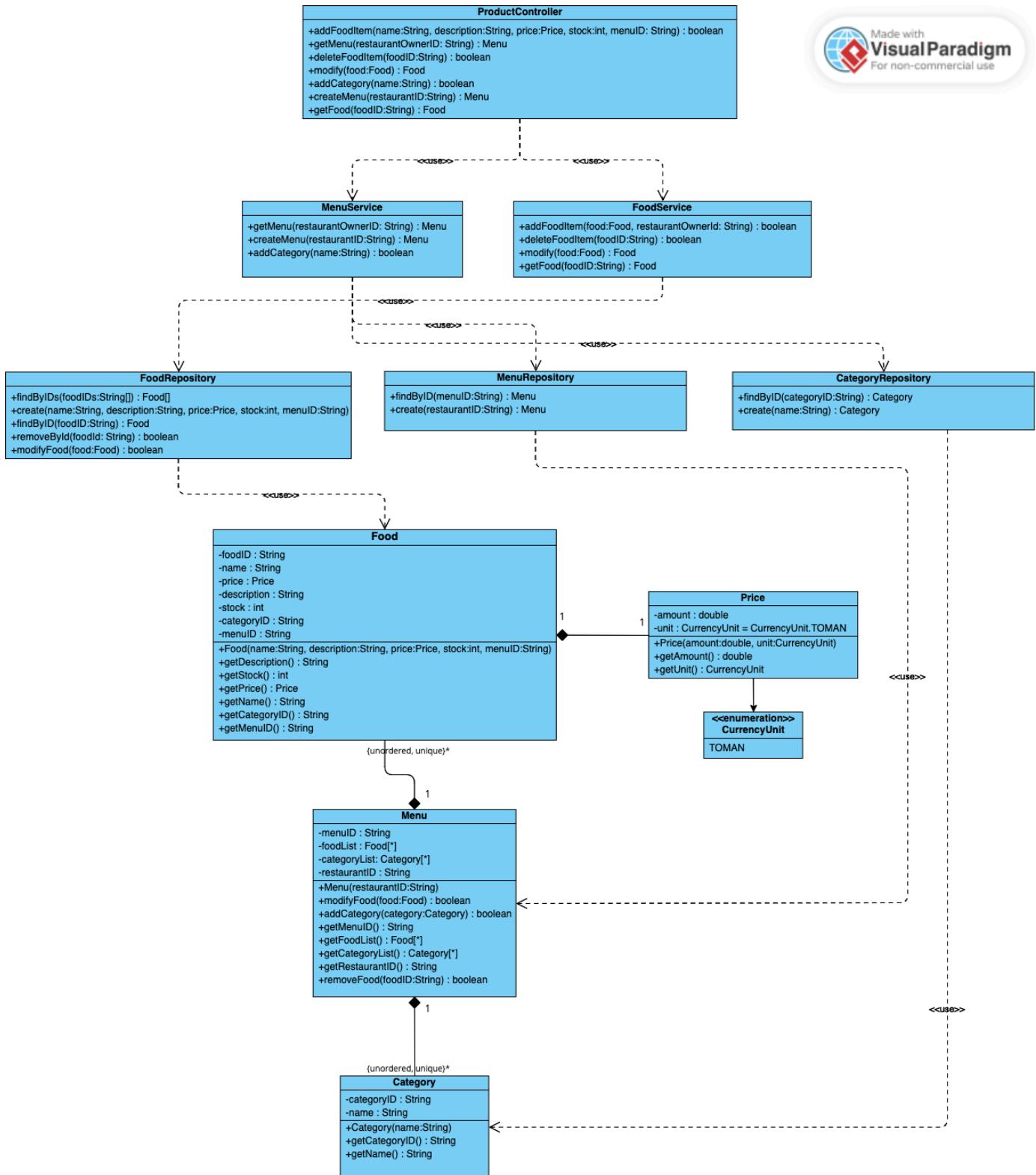
مولفه‌ی Authentication



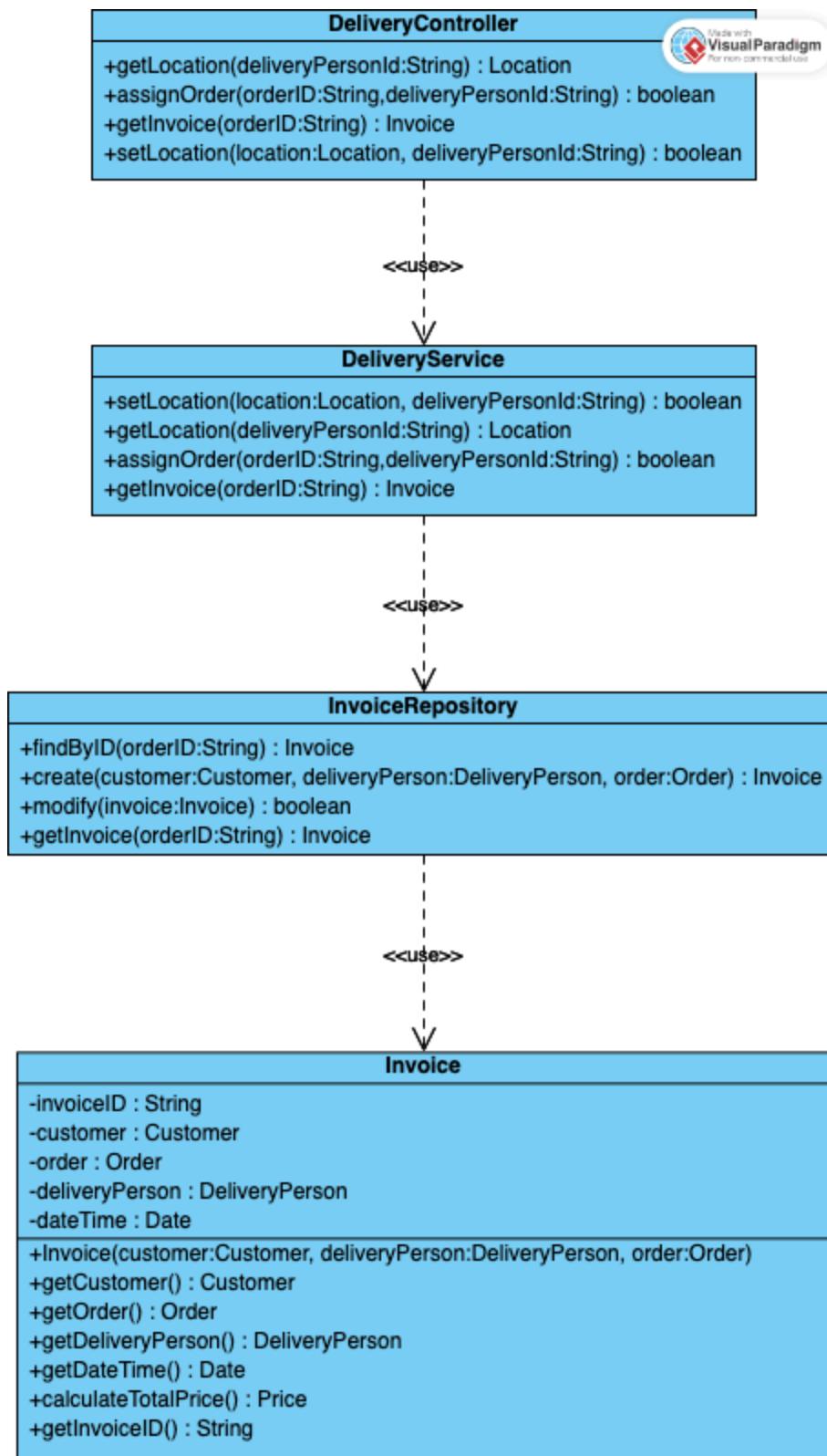
مولفه‌ی Payment



مولفه‌ی Product

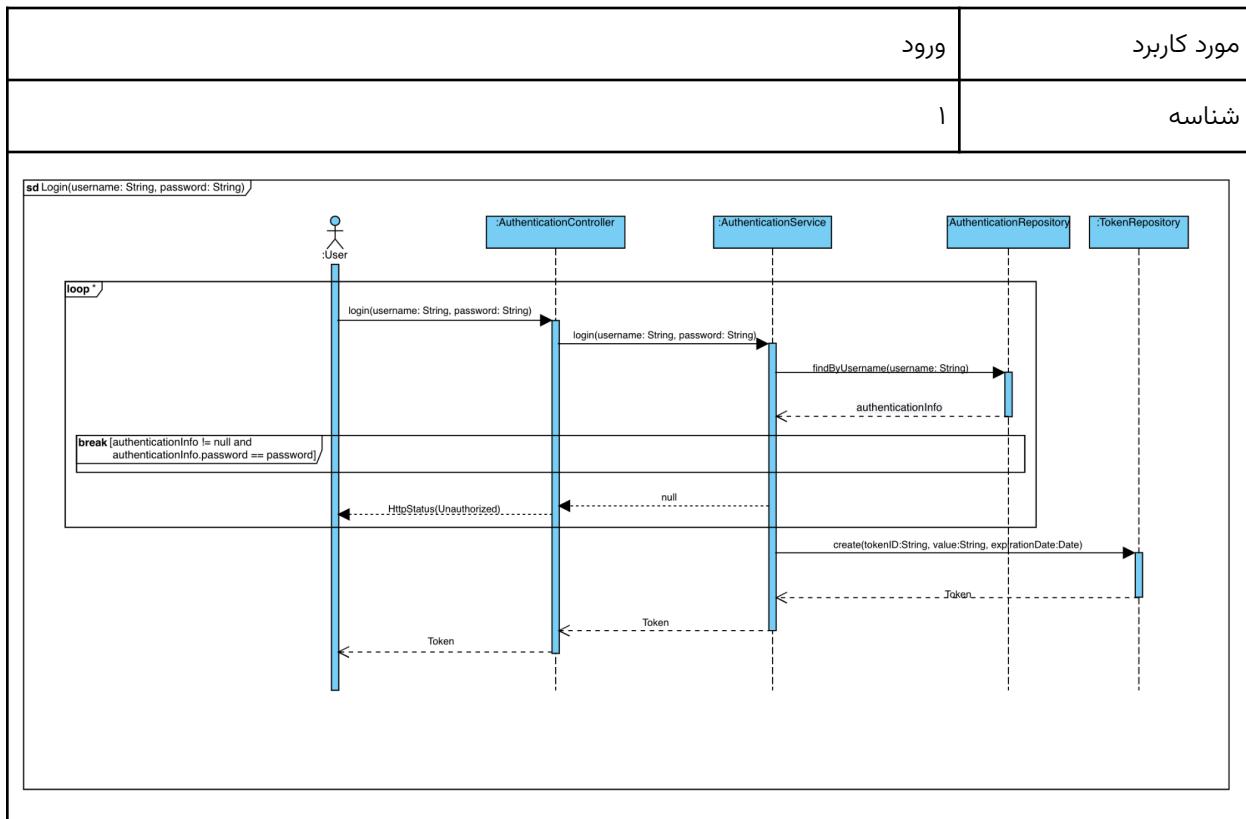


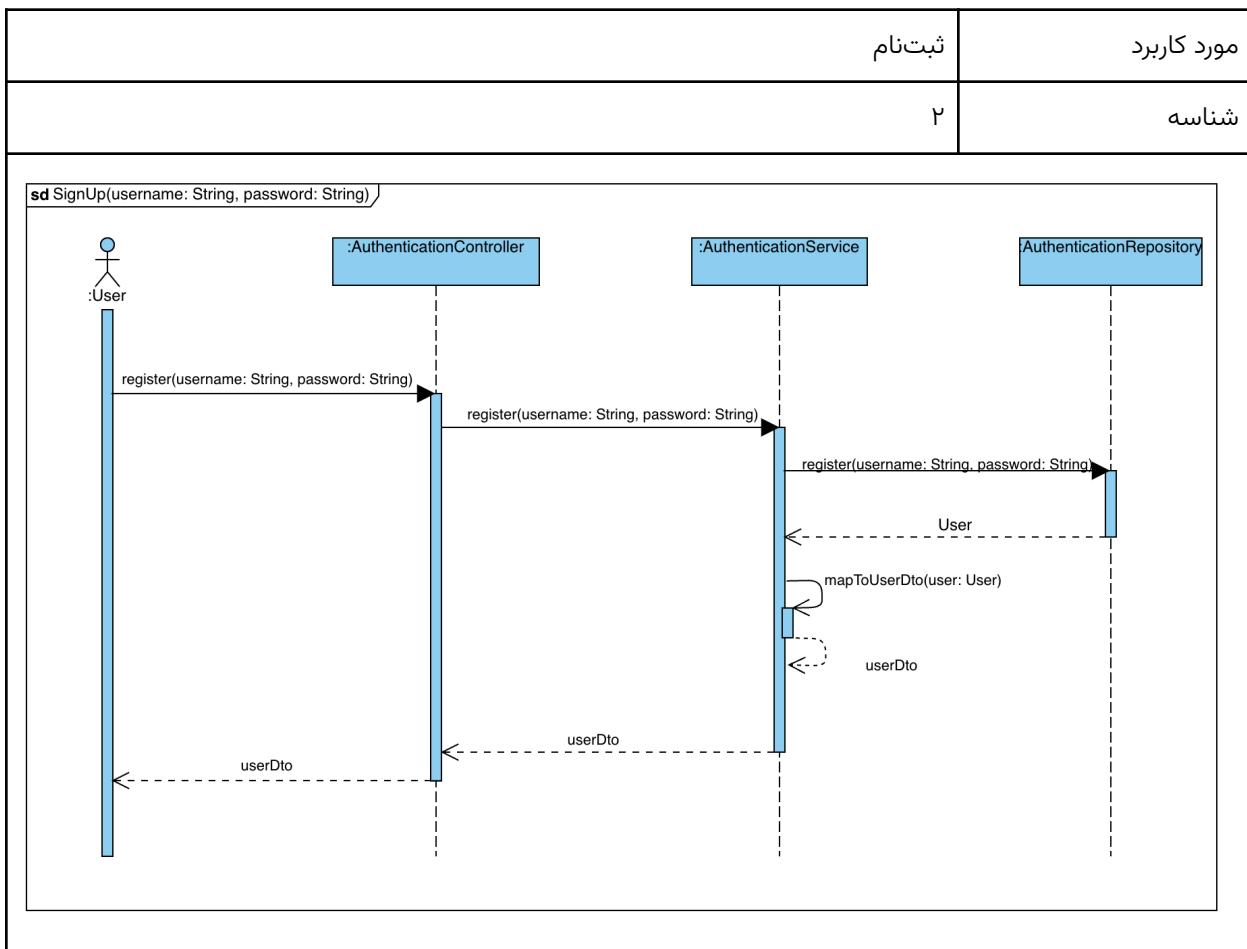
مولفه‌ی Delivery

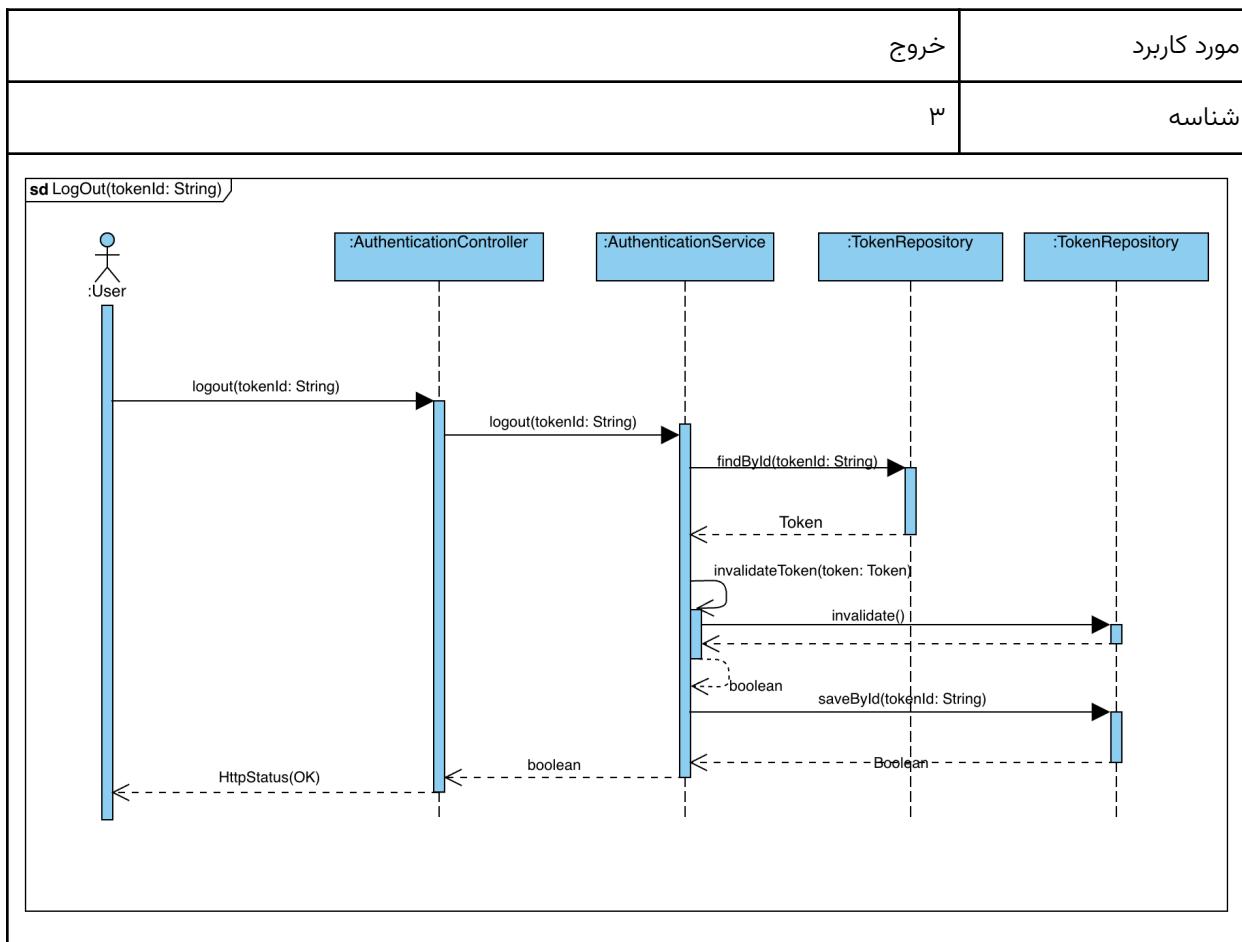


نمودارهای توالی طراحی

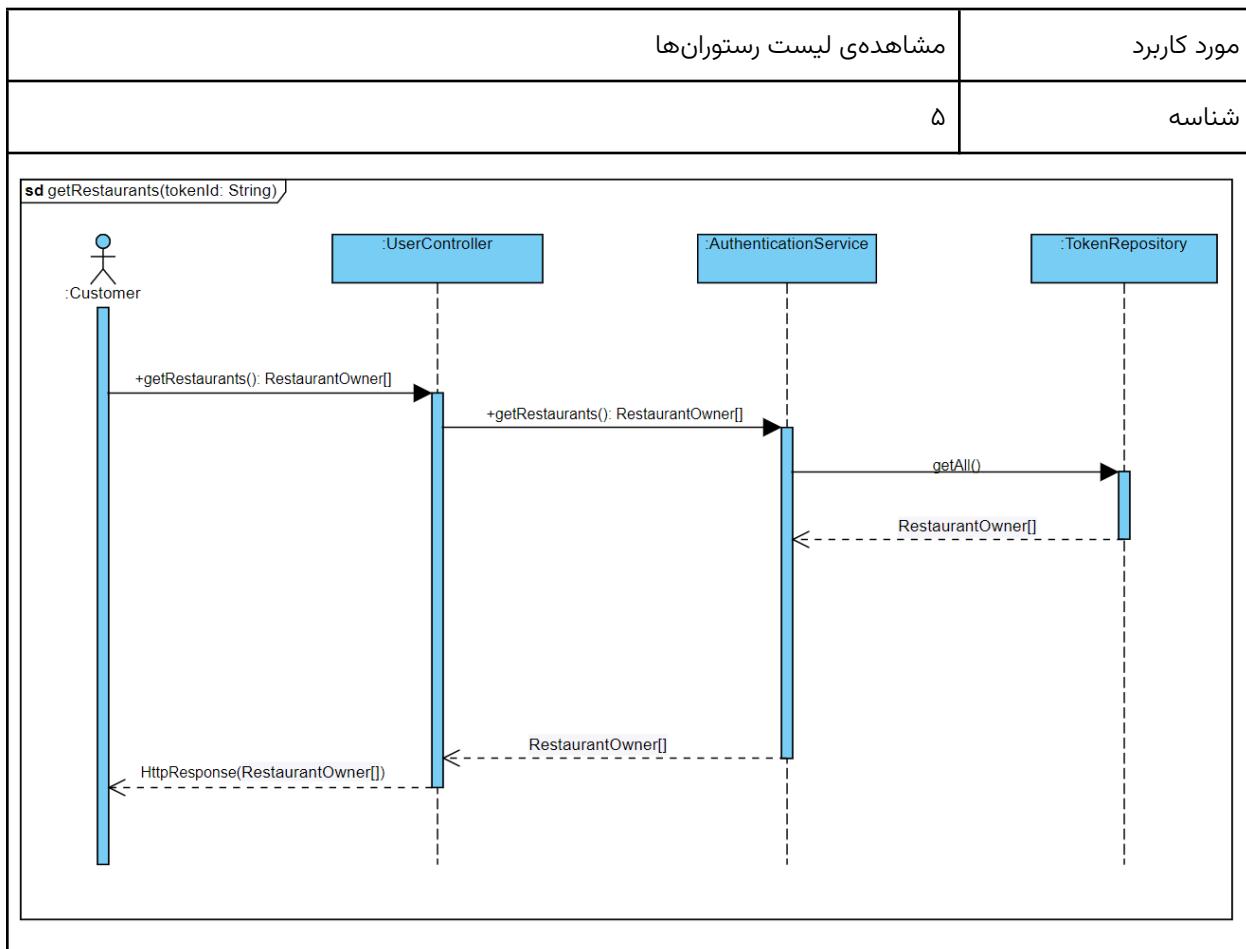
زیرسیستم احراز هویت و دسترسی

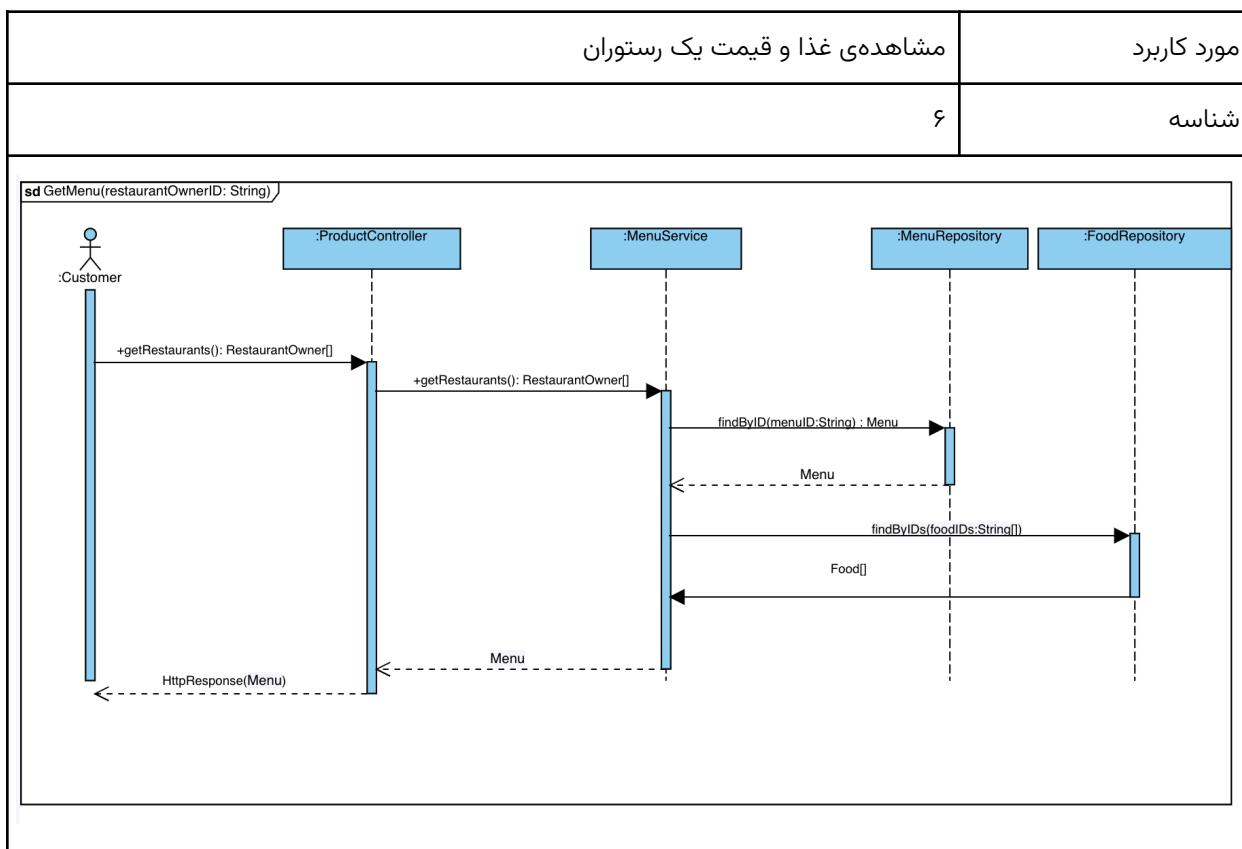


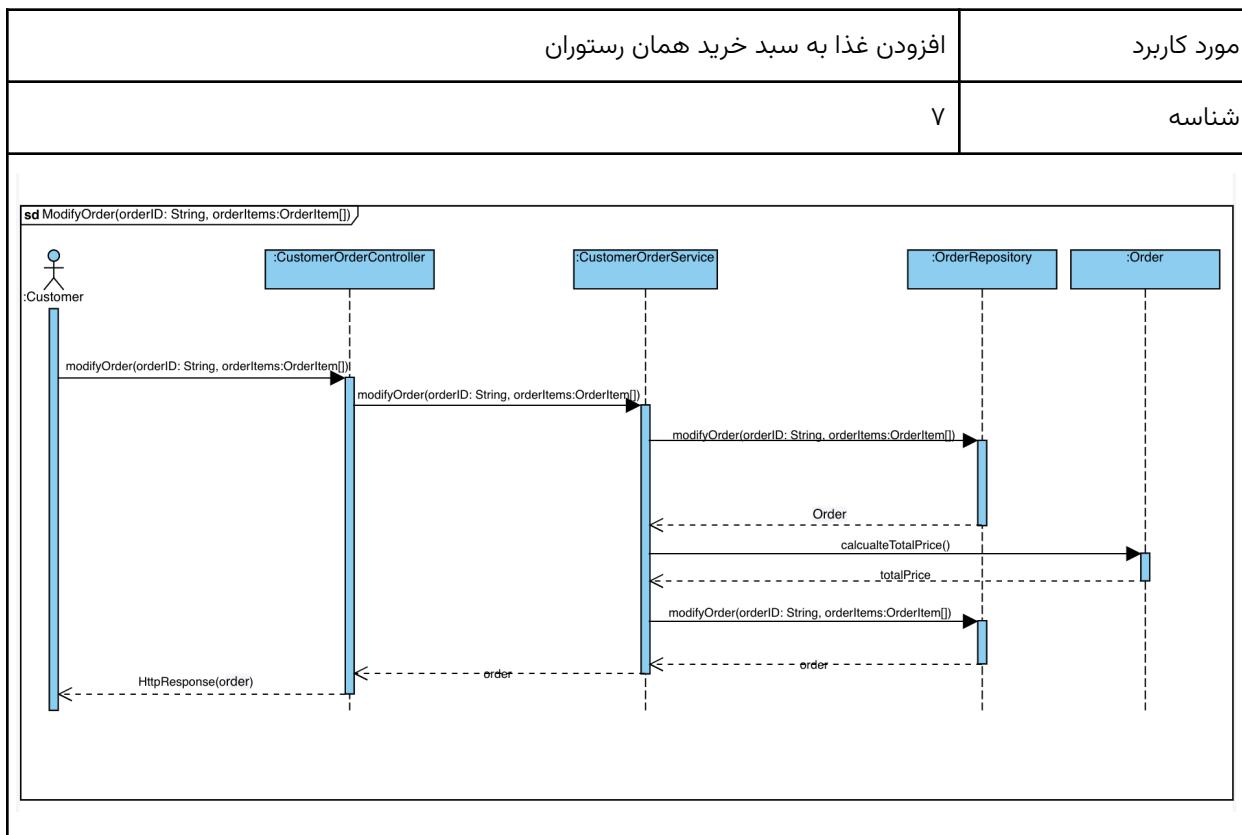


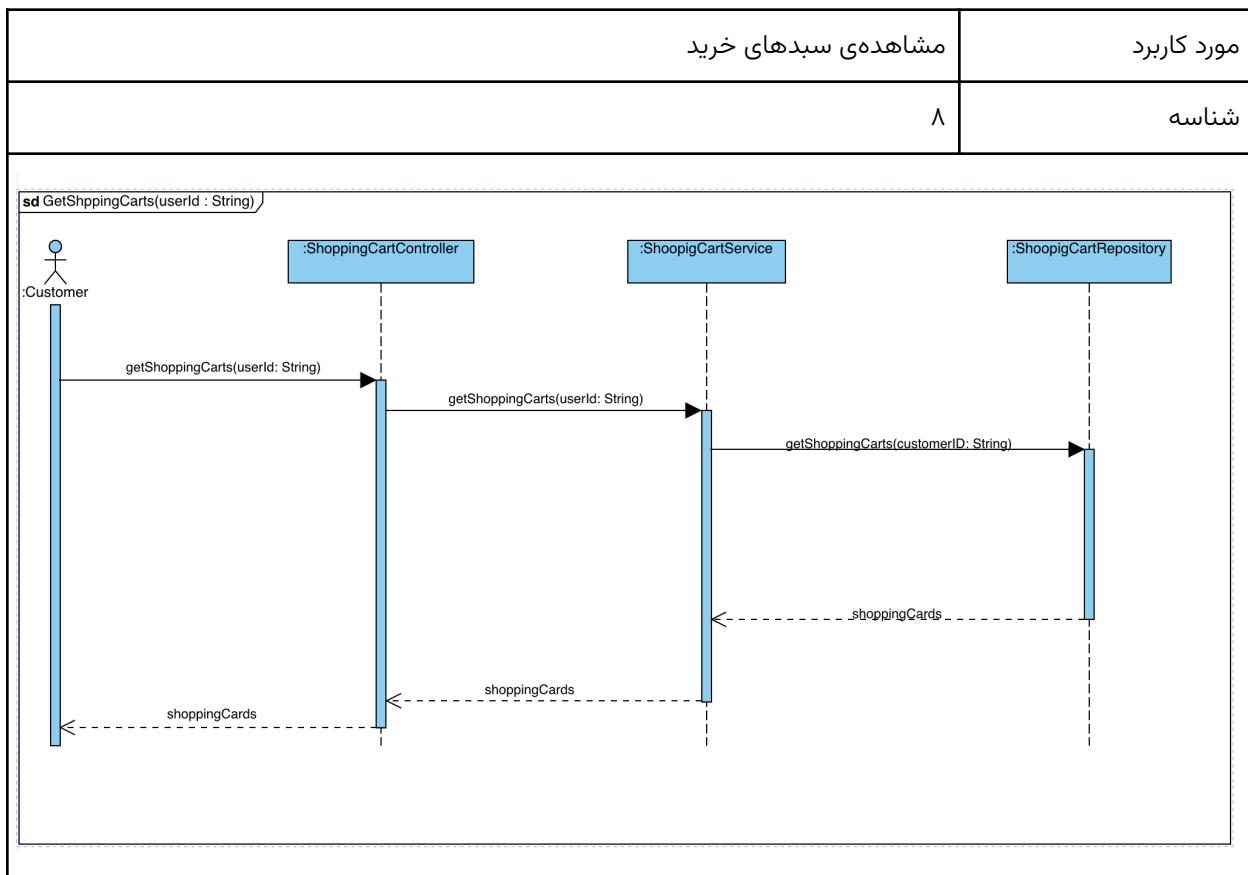


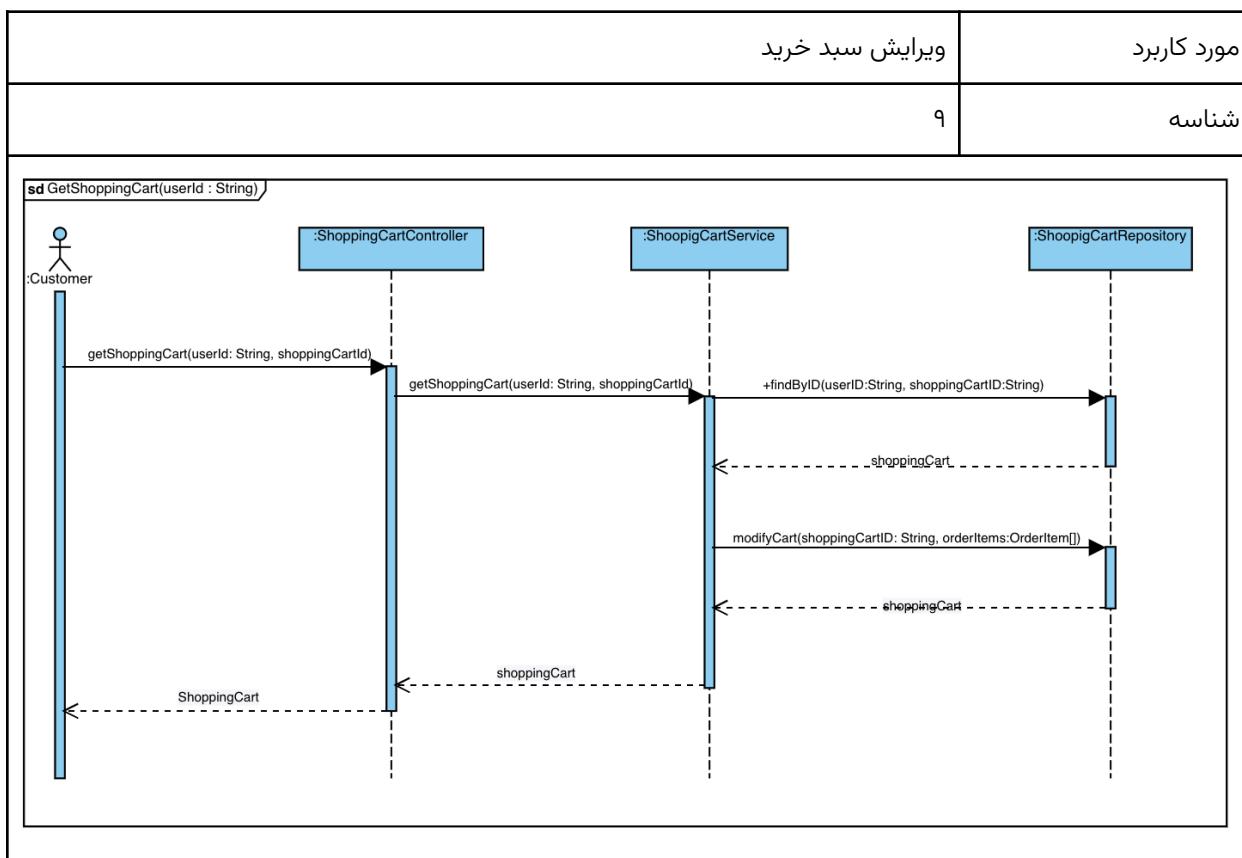
زیرسیستم مدیریت مشتریان

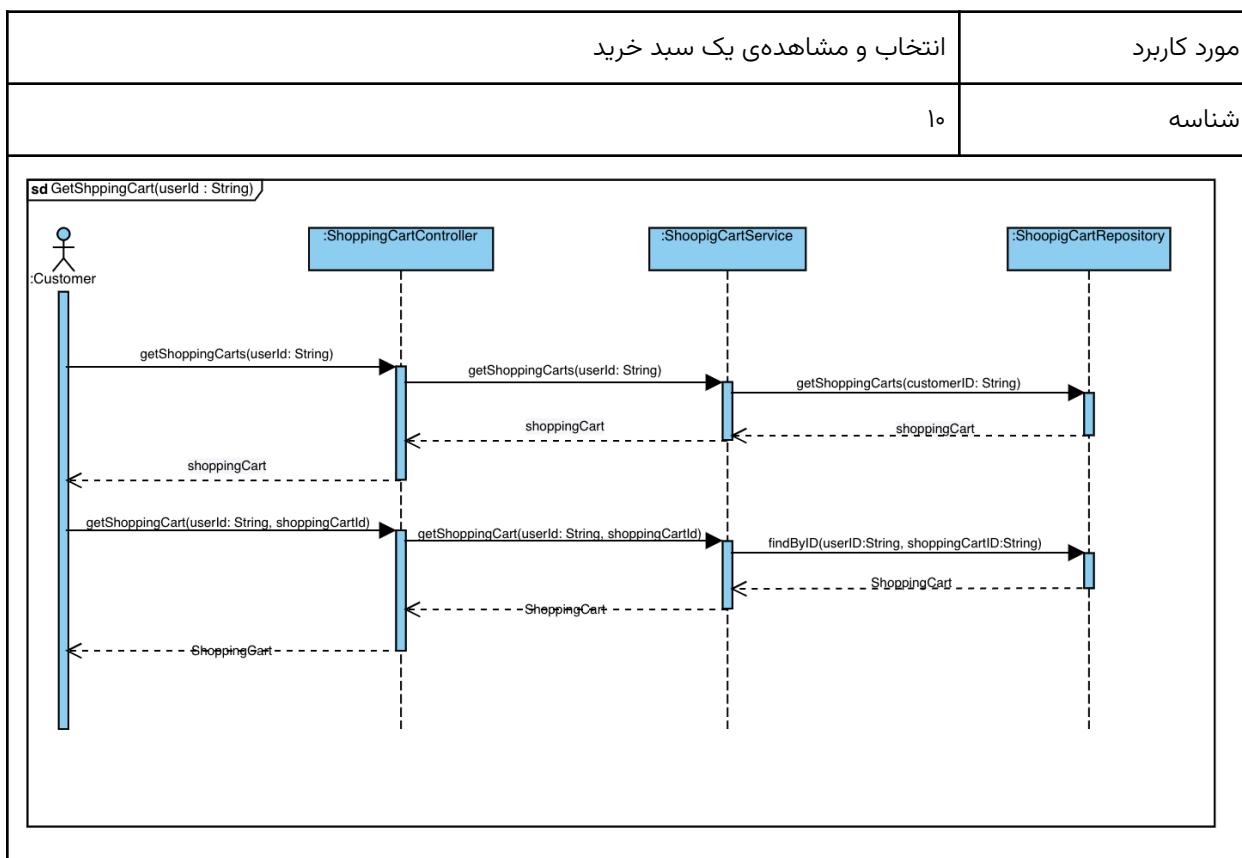


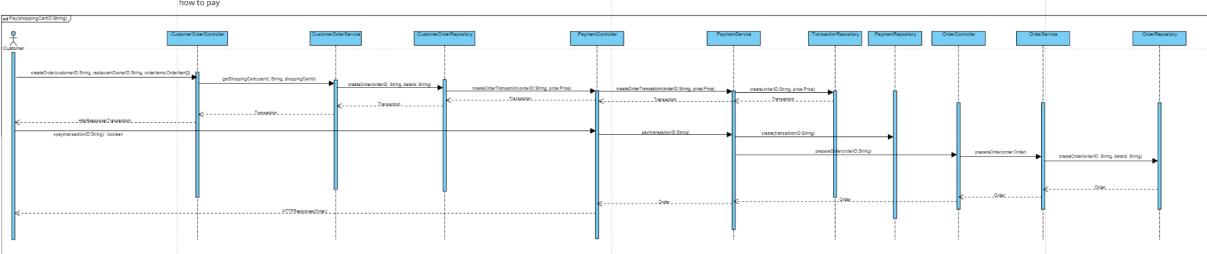


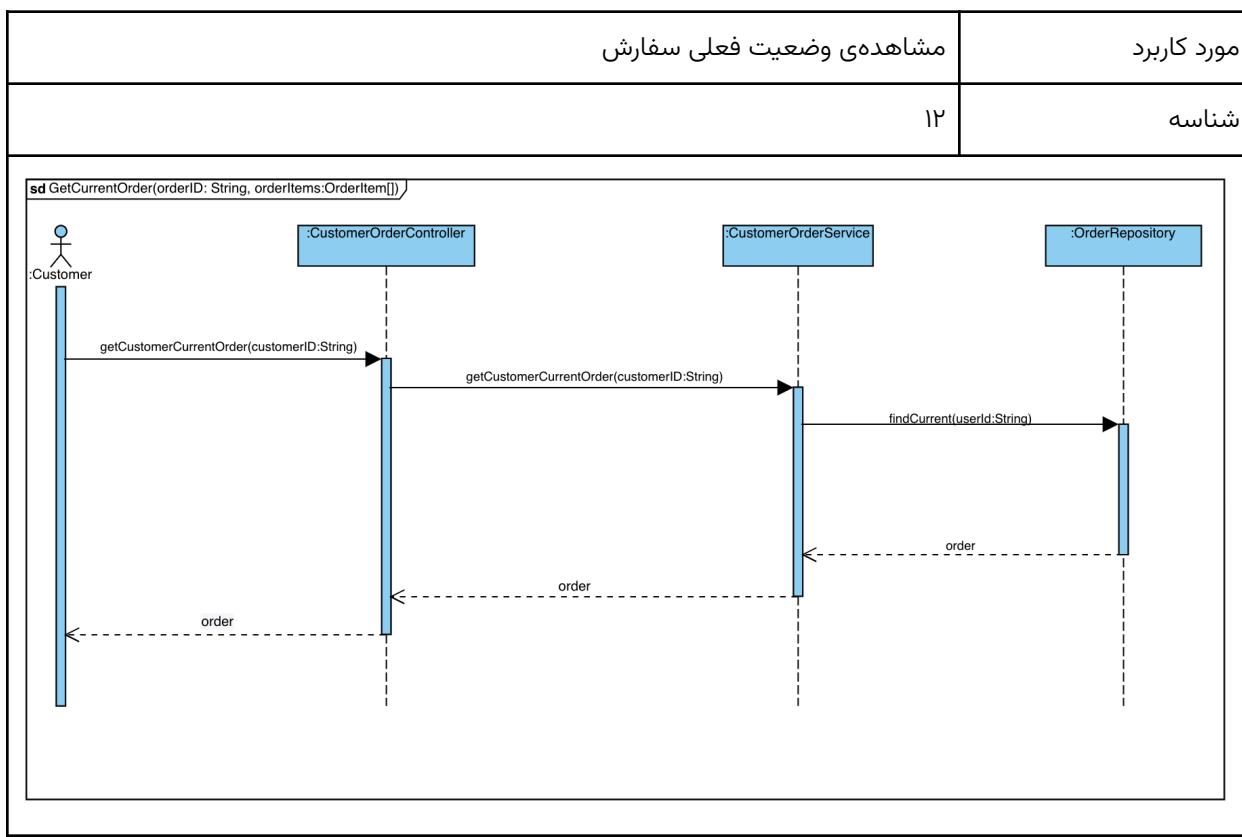








پرداخت	مورد کاربرد
	شناسه
<p>Based on that we have shown how to show the shopping cart, in this sequence we just show how to pay</p>  <pre> sequenceDiagram participant COC as CustomerOrderController participant COS as CustomerOrderService participant PC as PaymentController participant PS as PaymentService participant TR as TransactionRepository participant OC as OrderController participant OS as OrderService COC->>COS: createCustomerOrder(String orderId, int items) activate COS COS->>PC: getShoppingCart(String orderId) activate PC PC->>COS: checkCustomerOrder(String orderId) activate COS COC->>PS: payCustomerOrder(String orderId, String price) activate PS PS->>PC: createCustomerOrder(String orderId, String price) activate PC PC->>COS: updateCustomerOrder(String orderId, String price) activate COS COC->>TR: updateCustomerOrder(String orderId, String price) activate TR TR->>COS: updateCustomerOrder(String orderId, String price) activate COS COC->>OC: createOrder(String orderId) activate OC OC->>OS: createOrder(String orderId) activate OS OS->>OC: createOrder(String orderId) deactivate OC deactivate OS </pre>	



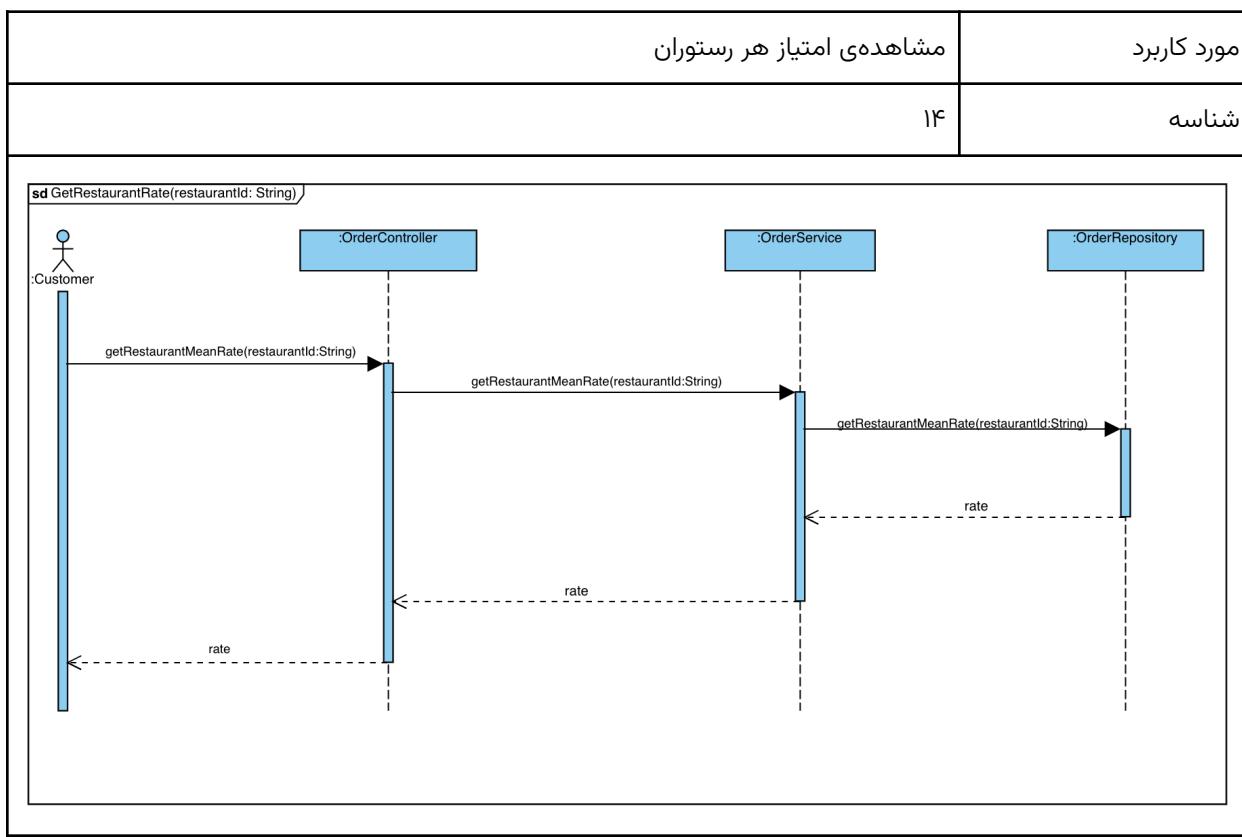
امکان ویرایش اطلاعات	مورد کاربرد
۱۳	شناسه


```

sd ModifyProfile(customer: Customer)
    participant Customer
    participant UserController
    participant CustomerService
    participant CustomerRepository

    Customer->>UserController: modifyCustomerProfile(customer: Customer)
    activate UserController
    UserController->>CustomerService: modifyCustomerProfile(customer: Customer)
    activate CustomerService
    CustomerService->>CustomerRepository: modifyCustomer(customer: Customer)
    activate CustomerRepository
    CustomerRepository-->>Customer: customer
    deactivate CustomerRepository
    deactivate CustomerService
    Customer-->>UserController: customer
    deactivate UserController
    UserController-->>Customer: HttpResponse(Customer)
    deactivate Customer

```



مورد کاربرد	امتیازدهی به سبد خرید
شناسه	۱۵
<pre> sequenceDiagram participant Customer participant CustomerOrderController participant CustomerOrderService participant OrderRepository participant RateController participant RateService participant RateRepository Customer->>CustomerOrderController: getCustomerOrders(customerId:String) activate CustomerOrderController CustomerOrderController->>CustomerOrderService: getCustomerOrders(customerId:String) activate CustomerOrderService CustomerOrderService->>OrderRepository: findByID(id:String) activate OrderRepository OrderRepository-->>RateController: order deactivate OrderRepository RateController->>RateService: setOrderRate(customerId:String, orderId:String, point:int) activate RateService RateService->>RateRepository: createRate(customerId:String, orderId:String, point:int) activate RateRepository RateRepository-->>Rate: <<create>> deactivate RateRepository Rate-->>RateController: rate deactivate Rate RateController-->>Customer: rateSubmitted deactivate RateController </pre>	امتیازدهی به سبد خرید شناسه ۱۵

مشاهده‌ی تاریخچه‌ی سفارش‌ها	مورد کاربرد
۱۶	شناسه
<pre>sd ShowOrdersHistory</pre> <pre> sequenceDiagram participant Customer as :Customer participant Controller as :CustomerOrderController participant Service as :CustomerOrderService participant Repository as :OrderRepository Customer->>Controller: getCustomerOrders(customerID:String) Controller->>Service: getCustomerOrders(customerID:String) activate Service Service->>Repository: findByID(id:String) Repository-->>Service: order deactivate Service activate Controller Controller-->>Customer: orders deactivate Controller </pre> <p>This sequence diagram illustrates the process of retrieving a customer's order history. It starts with a customer sending a 'getCustomerOrders(customerID:String)' message to a controller. The controller then sends a similar message to a service. The service performs a database query ('findByID(id:String)') on a repository to retrieve the specific order. Finally, the service returns the order information ('order') to the controller, which then sends a 'orders' message back to the customer.</p>	

مشاهده نظرات کاربران هر رستوران	مورد کاربرد
IV	شناسه

sd ShowRestaurantReview(restaurantId:String)

```

sequenceDiagram
    participant Customer as :Customer
    participant OrderController as :OrderController
    participant OrderService as :OrderService
    participant OrderRepository as :OrderRepository
    Customer->>OrderController: getRestaurantReviews(restaurantId:String)
    OrderController->>OrderService: getRestaurantReviews(restaurantId:String)
    OrderService->>OrderRepository: getRestaurantReviews(restaurantId:String)
    OrderRepository-->>Customer: reviews
  
```

The diagram illustrates a sequence of interactions between four components: Customer, OrderController, OrderService, and OrderRepository. It begins with the Customer sending a synchronous message 'getRestaurantReviews(restaurantId:String)' to the OrderController. The OrderController then sends a synchronous message 'getRestaurantReviews(restaurantId:String)' to the OrderService. Finally, the OrderService sends a synchronous message 'getRestaurantReviews(restaurantId:String)' to the OrderRepository. The OrderRepository returns a return message 'reviews' to the Customer.

مشاهده‌ی توضیحات غذای یک رستوران	مورد کاربرد
۱۸	شناسه
<pre>sd ShowFoodDescription(restaurantOwnerId: String)</pre> <p>The diagram illustrates the sequence of operations for displaying food descriptions. It starts with a Customer object sending a 'getMenu(restaurantOwnerId: String)' message to a ProductController. The ProductController then sends a 'getMenu(restaurantOwnerId: String)' message to a MenuService. The MenuService sends a 'findByID(menuID: String)' message to a MenuRepository. The MenuRepository returns a 'menu' object to the MenuService. The MenuService then sends a 'getFood(foodID: String)' message to a FoodService. The FoodService sends a 'findByID(foodID: String)' message to a FoodRepository. The FoodRepository returns a 'food' object to the FoodService. Finally, the FoodService sends a 'getFood(foodID: String)' message back to the Customer, which is then returned to the Customer.</p> <pre> sequenceDiagram participant Customer participant ProductController participant MenuService participant MenuRepository participant FoodService participant FoodRepository Customer->>ProductController: getMenu(restaurantOwnerId: String) activate ProductController ProductController->>MenuService: getMenu(restaurantOwnerId: String) activate MenuService MenuService->>MenuRepository: findByID(menuID: String) activate MenuRepository Note over MenuRepository: menu MenuRepository-->>MenuService: menu deactivate MenuRepository MenuService->>FoodService: getFood(foodID: String) activate FoodService FoodService->>FoodRepository: findByID(foodID: String) activate FoodRepository Note over FoodRepository: food FoodRepository-->>FoodService: food deactivate FoodRepository FoodService-->>Customer: getFood(foodID: String) deactivate FoodService deactivate Customer </pre>	

نظردهی به سفارش	مورد کاربرد
۱۹	شناسه
<pre> sd ReviewOrder +Customer +CustomerOrderController +CustomerOrderService +OrderRepository +.ReviewController +.ReviewService +.ReviewRepository Customer->CustomerOrderController: getCustomerOrders(customerID:String) CustomerOrderController->CustomerOrderService: getCustomerOrders(customerID:String) CustomerOrderService->OrderRepository: findByID(id:String) OrderRepository-->Order: order Order->CustomerOrderController: <<orders>> CustomerOrderController-->Customer: <<setOrderReview(customerID:String, orderId:String, comment:String)>> Customer-->ReviewController: reviewSubmitted ReviewController->ReviewService: setOrderReview(customerID:String, orderId:String, comment:String) ReviewService->ReviewRepository: createRate(customerID:String, orderId:String, point:Int) ReviewRepository-->Review: <<review>> Review-->>Customer: <<createReview>> </pre>	

مشاهده‌ی محل فعلی سفارش	مورد کاربرد
<pre>sd ShowPlaceOrder :Customer->CustomerOrderController: getCustomerCurrentOrder(customerID:String) -> Order CustomerOrderController->CustomerOrderService: getCustomerOrders(customerID:String) CustomerOrderService->OrderRepository: findByID(id:String) OrderRepository->Order: order Order->DeliveryController: getInvoice(orderID:String) DeliveryController->DeliveryService: getInvoice(orderID:String) DeliveryService->InvoiceRepository: invoice InvoiceRepository->DeliveryPerson: deliveryPerson:DeliveryPerson loop order.status != "DELIVERED!" getInvoice() deliveryPerson->Customer: location</pre>	شناسه

امتیازدهی به پیک	مورد کاربرد
۲۱	شناسه

(ad DeliveryPersonRate)

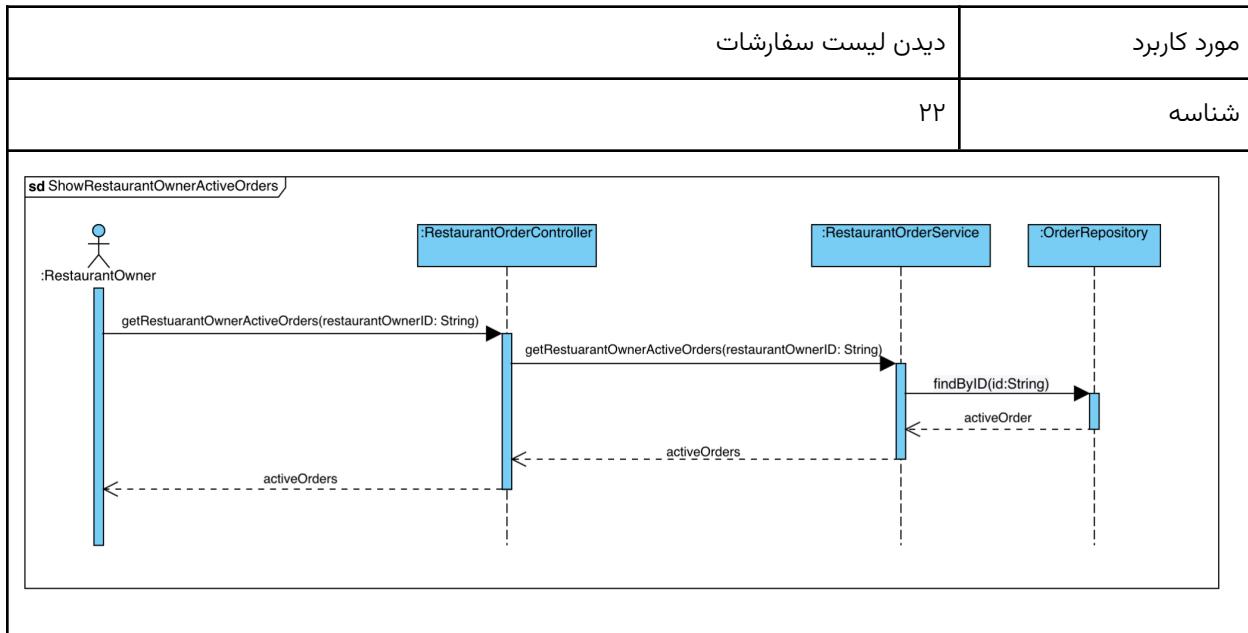
```

sequenceDiagram
    participant Customer
    participant CustomerOrderController
    participant CustomerOrderService
    participant OrderRepository
    participant RateController
    participant RateService
    participant RateRepository

    Customer->>CustomerOrderController: getCustomerOrders(customerID:String)
    activate CustomerOrderController
    CustomerOrderController->>CustomerOrderService: getCustomerOrders(customerID:String)
    activate CustomerOrderService
    CustomerOrderService->>OrderRepository: findByID(id:String)
    activate OrderRepository
    OrderRepository-->>CustomerOrderService: order
    deactivate OrderRepository
    CustomerOrderService-->>CustomerOrderController: orders
    deactivate CustomerOrderService
    CustomerOrderController-->>RateController: setDeliveryRate(customerID:String, orderID:String, point:int)
    activate RateController
    RateController-->>RateService: setDeliveryRate(customerID:String, orderID:String, point:int)
    activate RateService
    RateService-->>RateRepository: createRate(customerID:String, orderID:String, point:int)
    activate RateRepository
    RateRepository-->>RateService: rate
    deactivate RateRepository
    RateService-->>RateController: rateSubmitted
    deactivate RateService
    RateController-->>Customer: rateSubmitted
    deactivate RateController

```

زیرسیستم مدیریت تهیه‌کنندگاه غذا



دیدن لیست تاریخچه‌ی سفارشات	مورد کاربرد
۲۳	شناسه

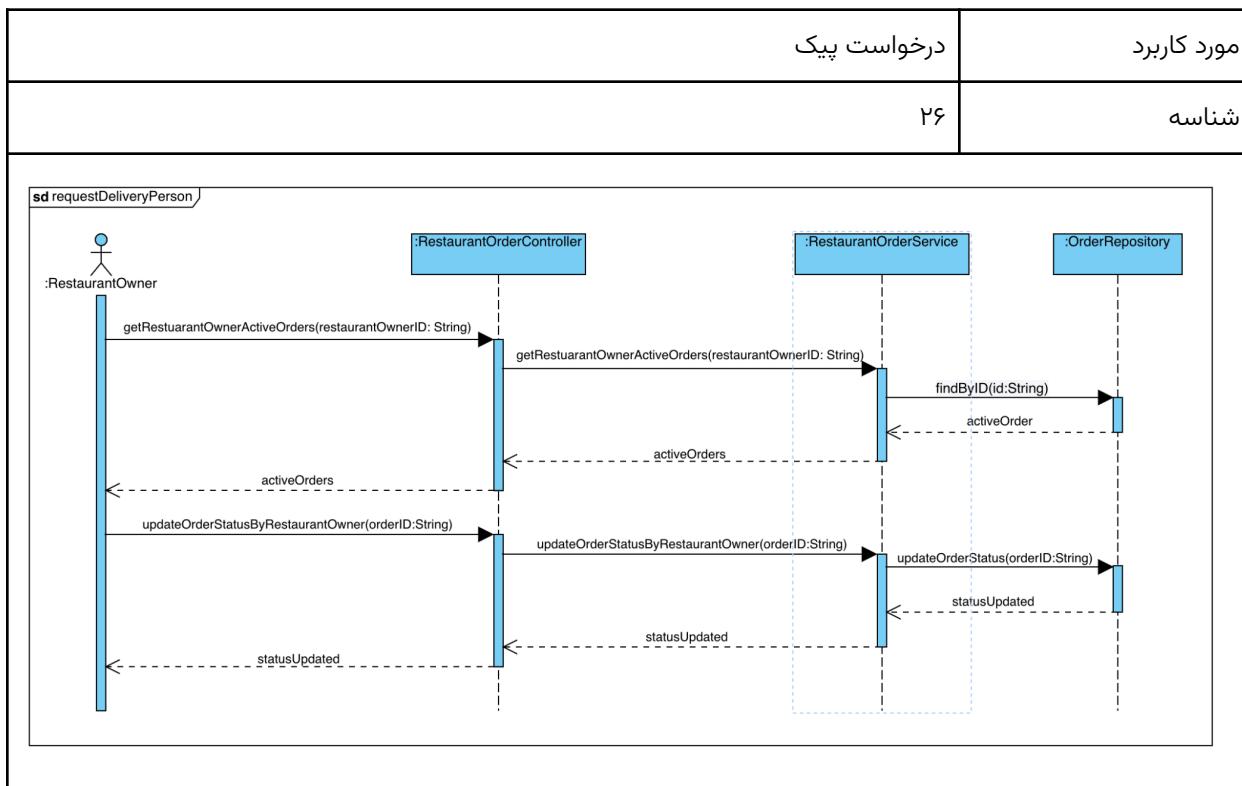
sd ShowRestaurantOwnerOrdersHistory

```

sequenceDiagram
    actor RestaurantOwner
    participant Controller as :RestaurantOrderController
    participant Service as :RestaurantOrderService
    participant Repository as :OrderRepository
    RestaurantOwner->>Controller: getRestaurantOwnerOrdersHistory(restaurantOwnerId: String)
    Controller->>Service: getRestaurantOwnerOrdersHistory(restaurantOwnerId: String)
    Service->>Repository: findById(id: String)
    Repository-->>Service: order
    Service-->>Controller: orders
    Controller-->>RestaurantOwner: orders
  
```

تغییر دادن وضعیت هر سفارش	مورد کاربرد
۲۴	شناسه
<pre> sd ChangeOrderStatus +:RestaurantOwner +:RestaurantOrderController +:RestaurantOrderService +:OrderRepository RestaurantOwner->RestaurantOrderController: getRestuarantOwnerActiveOrders(restaurantOwnerId: String) RestaurantOrderController->RestaurantOrderService: getRestuarantOwnerActiveOrders(restaurantOwnerId: String) RestaurantOrderService->OrderRepository: findByID(id:String) OrderRepository-->RestaurantOrderService: activeOrder OrderRepository-->RestaurantOrderController: activeOrders RestaurantOrderController-->RestaurantOwner: activeOrders RestaurantOwner->RestaurantOrderController: updateOrderStatusByRestaurantOwner(orderId:String) RestaurantOrderController->RestaurantOrderService: updateOrderStatusByRestaurantOwner(orderId:String) RestaurantOrderService->OrderRepository: updateOrderStatus(orderId:String) OrderRepository-->RestaurantOrderService: statusUpdated OrderRepository-->RestaurantOrderController: statusUpdated RestaurantOrderController-->RestaurantOwner: statusUpdated </pre>	

چاپ فاکتور	مورد کاربرد
۲۵	شناسه
<pre> sd PrintInvoice +-----+ RestaurantOwner +-----+ CustomerOrderController CustomerOrderService OrderRepository DeliveryController DeliveryService InvoiceRepository +-----+ </pre> <p>The sequence diagram illustrates the process of printing an invoice. It starts with a synchronous call from <code>RestaurantOwner</code> to <code>CustomerOrderController</code> with the message <code>getCustomerCurrentOrder(customerID:String) : Order</code>. <code>CustomerOrderController</code> then sends a synchronous call to <code>CustomerOrderService</code> with <code>getCustomerOrders(customerID:String)</code>. <code>CustomerOrderService</code> returns an <code>order</code> object to <code>CustomerOrderController</code>. <code>CustomerOrderController</code> sends a synchronous call to <code>OrderRepository</code> with <code>findById(id:String)</code>, receiving an <code>order</code> object in return. Finally, <code>CustomerOrderController</code> sends a synchronous call to <code>DeliveryController</code> with <code>getInvoice(orderID:String)</code>, which then calls <code>DeliveryService</code> with the same message. <code>DeliveryService</code> returns an <code>invoice</code> object to <code>DeliveryController</code>, who in turn returns it to <code>CustomerOrderController</code>.</p>	



تعیین خودکار یک پیک برای هر سفارش	مورد کاربرد
	۲۷
<pre>sd AssignDeliveryPerson(orderId:String)</pre> <pre> sequenceDiagram participant RO as RestaurantOwner participant ROC as RestaurantOrderController participant ROS as RestaurantOrderService participant OR as OrderRepository participant DP as DeliveryPerson RO->>ROC: updateOrderStatusByRestaurantOwner(orderId:String) activate ROC ROC->>ROS: updateOrderStatusByRestaurantOwner(orderId:String) activate ROS ROS->>OR: updateOrderStatus(orderId:String) activate OR OR-->>DP: statusUpdated deactivate OR deactivate ROS DP->>RO: statusUpdated deactivate DP </pre> <p>The sequence diagram illustrates the process of assigning a delivery person to an order. It starts with a RestaurantOwner sending a message to a RestaurantOrderController. The controller then sends a message to a RestaurantOrderService. The service updates the order status in an OrderRepository and sends a 'statusUpdated' event back to the controller. Finally, the controller sends a 'statusUpdated' event back to the RestaurantOwner.</p>	شناسه

مشاهده‌ی پیکهای حمل‌کننده‌ی سفارشات آن رستوران	مورد کاربرد
۲۸	شناسه
<pre>sd ShowAssignDeliveryPerson(restaurantOwnerId: String)</pre> <p>The sequence diagram illustrates the flow of assigning a delivery person. It starts with a RestaurantOwner sending a message to the RestaurantOrderController. The RestaurantOrderController then interacts with the DeliveryController, which in turn sends a message to the RestaurantOrderService. The RestaurantOrderService performs a query on the OrderRepository to find an order by ID. Finally, the DeliveryController sends back the invoice and location details to the RestaurantOwner.</p> <pre> sequenceDiagram participant RO as :RestaurantOwner participant ROC as :RestaurantOrderController participant DC as :DeliveryController participant ROS as :RestaurantOrderService participant OR as :OrderRepository RO->>ROC: getRestaurantOwnerActiveOrders(restaurantOwnerId: String) activate ROC ROC->>DC: getRestaurantOwnerActiveOrders(restaurantOwnerId: String) activate DC DC->>ROS: findByID(id: String) activate ROS ROS-->>order: order deactivate ROS ROS-->>DC: invoice deactivate DC DC-->>RO: invoice DC-->>RO: location deactivate DC deactivate ROC </pre>	

تعیین هوشمند یک پیک برای هر سفارش	مورد کاربرد
۲۹	شناسه

sd AssignDeliveryPersonWithAI(orderID:String)

```

sequenceDiagram
    participant RO as RestaurantOwner
    participant ROC as RestaurantOrderController
    participant ROS as RestaurantOrderService
    participant OR as OrderRepository
    participant DP as DeliveryPerson
    RO->>ROC: updateOrderStatusByRestaurantOwner(orderID:String)
    activate ROC
    ROC->>ROS: updateOrderStatusByRestaurantOwner(orderID:String)
    activate ROS
    ROS->>OR: updateOrderStatus(orderID:String)
    activate OR
    OR-->>DP: statusUpdated
    deactivate OR
    deactivate ROS
    DP-->>RO: statusUpdated
    deactivate DP
    
```

The diagram illustrates the sequence of events for assigning a delivery person to an order. It starts with a RestaurantOwner sending a message to a RestaurantOrderController. The controller then sends a message to a RestaurantOrderService. The service updates the order status in a repository and sends a 'statusUpdated' event back to the controller. Finally, the controller sends a 'statusUpdated' event back to the RestaurantOwner.

افزودن غذا به منو	مورد کاربرد
۳۰	شناسه


```
sd AddFoodToMenu(name:String, description:String, price:Price, stock:int, menuID: String)
```

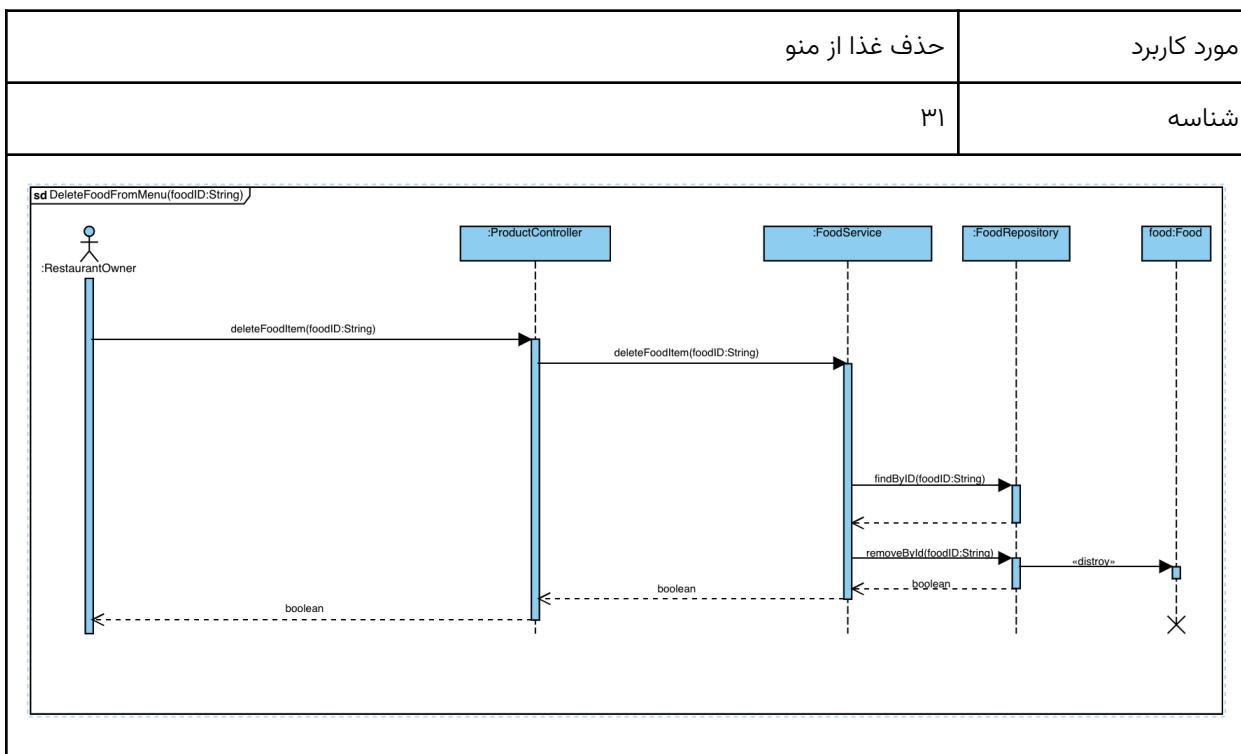
```

sequenceDiagram
    participant RO as :RestaurantOwner
    participant PC as :ProductController
    participant FS as :FoodService
    participant ROS as :RestaurantOwnerService
    participant FR as :FoodRepository

    RO->>PC: addFoodItem(name:String, description:String, price:Price, stock:int, menuID: String)
    activate PC
    PC->>FS: addFoodItem(food:Food, restaurantOwnerId: String)
    activate FS
    FS->>ROS: findByUsername(userName:String)
    activate ROS
    ROS-->>FS: restaurantOwner
    ROS->>FR: create(name:String, description:String, price:Price, stock:int, menuID:String)
    activate FR
    FR-->>ROS: food
    ROS-->>PC: boolean
    deactivate PC
    deactivate FS
    deactivate ROS
    deactivate FR

```

The diagram illustrates the sequence of operations for adding a food item to a menu. It starts with a RestaurantOwner sending an 'addFoodItem' message to a ProductController. The ProductController then sends an 'addFoodItem' message to a FoodService, passing a Food object and the RestaurantOwner's ID. The FoodService performs a search ('findByUsername') and creates a new food item ('create') in a FoodRepository, returning it to the FoodService. Finally, the FoodService sends a 'boolean' confirmation back to the ProductController.



تعیین موجودی هر غذا	مورد کاربرد
۳۲	شناسه


```
sd SetFoodAvailability(restaurantOwnerId : String)
    participant RestaurantOwner
    participant ProductController
    participant MenuService
    participant FoodService
    participant MenuRepository
    participant FoodRepository

    RestaurantOwner->>ProductController: getMenu(restaurantOwnerId : String)
    activate ProductController
    ProductController->>MenuService: getMenu(restaurantOwnerId : String)
    activate MenuService
    MenuService->>FoodService: findByID(menuId : String)
    activate FoodService
    FoodService->>MenuRepository: menu
    activate MenuRepository
    MenuRepository->>FoodRepository: food
    activate FoodRepository
    FoodRepository-->>FoodService: modify(food : Food)
    deactivate FoodRepository
    deactivate FoodService
    FoodService-->>MenuRepository: boolean
    deactivate FoodService
    deactivate MenuRepository
    deactivate ProductController
    deactivate RestaurantOwner
```

ویرایش غذا در منو	مورد کاربرد
۳۳	شناسه

```
sd UpdateFoodInfo(restaurantOwnerId : String)
```

The sequence diagram illustrates the process flow for updating food information:

- Participants:** RestaurantOwner (Actor), ProductController, MenuService, FoodService, MenuRepository, and FoodRepository.
- Sequence:**
 - RestaurantOwner sends `getMenu(restaurantOwnerId: String)` to ProductController.
 - ProductController sends `getMenu(restaurantOwnerId: String)` to MenuService.
 - MenuService sends `findByID(menuId: String)` to MenuRepository.
 - MenuRepository returns `menu` to MenuService.
 - ProductController sends `modify(food: Food)` to FoodService.
 - FoodService sends `modify(food: Food)` to FoodRepository.
 - FoodRepository returns `boolean` to FoodService.
 - FoodService returns `boolean` to ProductController.
 - ProductController returns `boolean` to RestaurantOwner.

افزودن دسته‌بندی برای هر غذا	مورد کاربرد
۳۴	شناسه


```
sd UpdateFoodCategory(restaurantOwnerId : String)
```

The sequence diagram illustrates the process of updating a food category. It starts with a **:RestaurantOwner** object sending a **getMenu(restaurantOwnerId: String)** message to a **:ProductController**. The **:ProductController** then sends a **getMenu(restaurantOwnerId: String)** message to a **:MenuService**. The **:MenuService** sends a **findById(menuId: String)** message to a **:MenuRepository**, which returns a **menu** object. The **:ProductController** sends a **modify(food: Food)** message to the **:FoodService**. The **:FoodService** sends a **modify(food: Food)** message to the **:FoodRepository**, which returns a **boolean** value. Finally, the **:FoodService** sends a **modify(food: Food)** message back to the **:ProductController**, which returns a **boolean** value.

```

sequenceDiagram
    participant RO as :RestaurantOwner
    participant PC as :ProductController
    participant MS as :MenuService
    participant FS as :FoodService
    participant MR as :MenuRepository
    participant FR as :FoodRepository

    RO->>PC: getMenu(restaurantOwnerId: String)
    activate PC
    PC->>MS: getMenu(restaurantOwnerId: String)
    activate MS
    MS->>MR: findById(menuId: String)
    activate MR
    MR-->>PC: menu
    deactivate MR
    PC->>FS: modify(food: Food)
    activate FS
    FS->>FR: modify(food: Food)
    activate FR
    FR-->>FS: boolean
    deactivate FR
    FS->>PC: modify(food: Food)
    deactivate FS
    PC-->>RO: boolean
    deactivate PC
  
```

زیرسیستم مدیریت پیکها

تغییر دادن وضعیت هر سفارش	مورد کاربرد
۳۵	شناسه


```

sd ChangeStatusByDeliveryPerson(deliveryPersonId: String)
+-----+
| :RestaurantOwner |
+-----+
| :DeliveryPersonOrderController |
+-----+
| :DeliveryPersonOrderService |
| :OrderRepository |
| :RestaurantOwnerService |
| :CustomerService |
+-----+

```

Sequence diagram illustrating the process of changing the status of an order assigned to a delivery person:

- Participants:** RestaurantOwner, DeliveryPersonOrderController, DeliveryPersonOrderService, OrderRepository, RestaurantOwnerService, CustomerService.
- Initial Message:** RestaurantOwner sends `getDeliveryPersonCurrentOrder(deliveryPersonId: String)` to DeliveryPersonOrderController.
- DeliveryPersonOrderController Response:** DeliveryPersonOrderController sends `getDeliveryPersonCurrentOrder(deliveryPersonId: String)` to DeliveryPersonOrderService.
- DeliveryPersonOrderService Interaction:** DeliveryPersonOrderService sends `findById(id: String)` to OrderRepository, which returns an `order`.
- DeliveryPersonOrderController Update:** DeliveryPersonOrderController sends `updateOrderStatusByDeliveryPerson(orderID: String)` to DeliveryPersonOrderService.
- DeliveryPersonOrderService Response:** DeliveryPersonOrderService sends `sendAnnounce(restaurantID: String, announce: String)` and `sendAnnounce(customerID: String, announce: String)` to CustomerService, both returning `Boolean`.
- DeliveryPersonOrderController Confirmation:** DeliveryPersonOrderController receives `Boolean` from CustomerService and returns `Boolean` to RestaurantOwner.

ارسال موقعیت به صورت زنده برای مشتری	مورد کاربرد
۳۶	شناسه

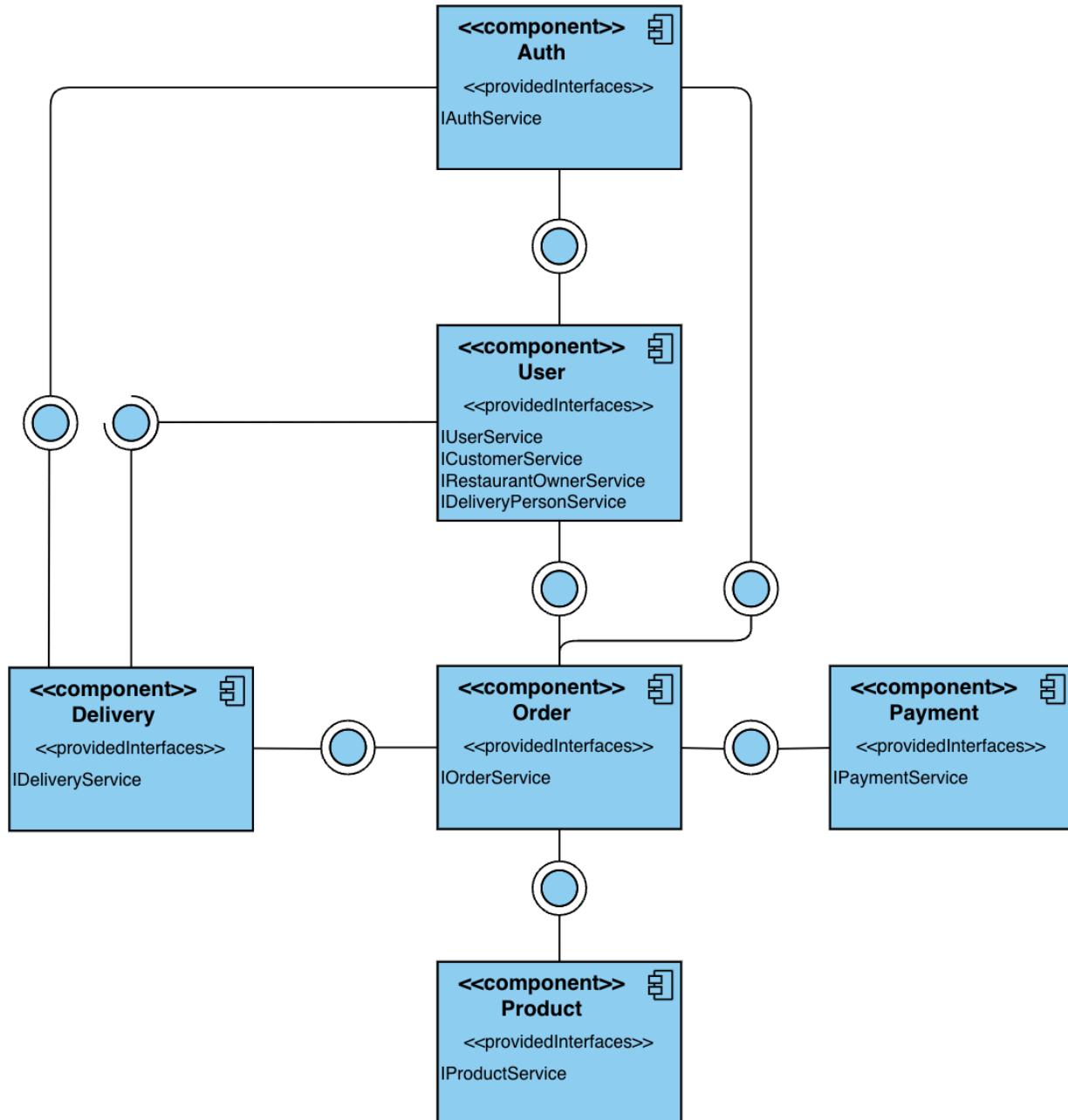

```
sd ShareLocationToCustomer(orderId: String, deliveryPersonId: String)
```

The diagram illustrates the sequence of operations for sharing location information. It starts with a RestaurantOwner sending an 'assignOrder' message to the DeliveryPersonOrderController. The controller then sends an 'assignOrder' message to the DeliveryPersonOrderService. Both the controller and service return Boolean values. A loop condition 'loop [order.status!="DELIVERD"]' is shown, with the controller sending a 'setLocation' message to the service, which returns a Boolean value.

```

sequenceDiagram
    participant RO as :RestaurantOwner
    participant DPOC as :DeliveryPersonOrderController
    participant DPOS as :DeliveryPersonOrderService
    RO->>DPOC: assignOrder(orderId:String,deliveryPersonId:String)
    activate DPOC
    DPOC->>DPOS: assignOrder(orderId:String,deliveryPersonId:String)
    deactivate DPOC
    activate DPOS
    DPOS-->>RO: Boolean
    deactivate DPOS
    activate DPOC
    DPOC-->>DPOS: setLocation(location:Location, deliveryPersonId:String)
    deactivate DPOC
    activate DPOS
    DPOS-->>RO: Boolean
    deactivate DPOS
    note over DPOC: loop [order.status!="DELIVERD"]
  
```

نمودار مولفه



سند ریسک‌ها

انواع ریسک‌ها

ریسک‌های فنی

- **نفوذ و هک:** ریسک‌های مرتبط با نفوذ به سیستم و سرقت اطلاعات حساس مشتریان و جزئیات پرداخت.
- **حفظ حریم خصوصی:** اطمینان از رعایت استانداردهای امنیتی برای حفاظت از اطلاعات شخصی کاربران.
- **خرابی سیستم:** احتمال خرابی و توقف سیستم که می‌تواند موجب اختلال در روند پیگیری و تحويل سفارشات شود.
- **بار ترافیک بالا:** عدم توانایی سیستم در مدیریت افزایش تعداد کاربران و سفارشات در زمان پیک.
- **مشکل در یکپارچه‌سازی با سیستم‌های موجود:** احتمال وجود مشکلات در هماهنگ‌سازی سیستم جدید با سیستم‌های موجود تأمین‌کنندگان غذا و مشتریان.
- **همگام‌سازی داده‌ها:** چالش‌های مرتبط با همگام‌سازی لحظه‌ای داده‌ها از منابع مختلف.

ریسک‌های مدیریتی

- **تغییر نیازمندی‌ها:** تغییرات مکرر در نیازمندی‌ها که می‌تواند باعث افزایش دامنه پروژه و تأخیر در زمان‌بندی شود.
- **عدم تعریف دقیق نیازمندی‌ها:** عدم تعریف دقیق نیازمندی‌ها از ابتدا که ممکن است باعث سوء تفاهem و مشکلات بعدی شود.
- **عدم تحقق زمان‌بندی:** عدم تحقق زمان‌بندی مقرر که می‌تواند پروژه را با تأخیر مواجه کند.
- **كمبود منابع انسانی و مالی:** تخصیص ناکافی منابع که می‌تواند منجر به تأخیر در اجرای پروژه شود.
- **هزینه‌های پیش‌بینی نشده:** بروز هزینه‌های غیرمنتظره که می‌تواند بودجه پروژه را تحت تأثیر قرار دهد.
- **تست ناکافی:** فرآیندهای ناکافی تست و تضمین کیفیت که می‌تواند منجر به تولید محصول نهایی معیوب شود.

ریسک‌های سازمانی

- **عدم هماهنگی بین ذینفعان:** مشکلات هماهنگی و ارتباطی بین ذینفعان مختلف که می‌تواند باعث بروز تعارضات و تأخیر در پروژه شود.
- **مقاومت در برابر تغییر:** مقاومت کارکنان و کارکنان در برابر تغییرات و پذیرش سیستم جدید.

ریسک‌های خارجی

- **قوانين و مقررات:** نیاز به تطابق با قوانین و مقررات محلی مربوط به حریم خصوصی داده‌ها و تراکنش‌های آنلاین.
- **تغییرات بازار:** تغییرات در شرایط بازار یا اقدامات رقبا که می‌تواند بر موفقیت پروژه تأثیر بگذارد.

ریسک‌های اضافی

- **تغییر انتظارات مشتری:** در طول انجام پروژه، ممکن است انتظارات مشتری و به تبع از آن، نیازمندی‌های پروژه و گستره‌ی آن‌ها تغییر کنند. برای مدیریت این ریسک، تیم برنامه‌ریزی خود را بر اساس قاعده‌ی MoSCoW انجام داده و از تخمین بیش از حد و برنامه‌ریزی جزئی خودداری می‌کند.
- **عدم آشنایی کافی با برخی از تکنولوژی‌ها و چارچوب‌ها:** ممکن است برخی از اعضای تیم با معماری انتخاب شده برای پروژه (به طور خاص تکنولوژی‌ها و چارچوب‌ها) آشنایی کافی نداشته باشند. برای حل این ریسک قبل از انتخاب آنها با تمامی افراد، صحبت‌های لازم انجام شده و جلسات گروهی‌ای برگزار شده است. همچنین از تکنولوژی‌ها و چارچوب‌هایی استفاده شده که افراد در آنها تخصص بیشتری دارند.
- **از بین رفتن قابلیت نگهداری و ایجاد:** در طول انجام پروژه به دلیل وقت محدود و همچنین انجام دادن موارد متعددی maintenance، ممکن است که پروژه رفته تحلیل برود و به نقطه‌ای برسد که دیگر قابل نگهداری نباشد. برای مقابله با این مشکل باید در طول انجام پروژه test به صورت مداوم داشته باشیم. همچنین idiom و رعایت coding style، refactoring، رعایت style و رعایت idiom زبان، ما را در پیشگیری از رویه‌رو شدن با این ریسک کمک می‌کند.
- **ایمنی سامانه و داده‌ها:** باید مطمئن بود داده‌ها ایمن هستند به این معنی که به علت اتفاقی مانند خرابشدن سرورها، داده‌های کاربران حذف نشود یا مورد تغییر ناخواسته‌ای قرار نگیرد. به این منظور راهکار پشتیبان‌گیری دوره‌ای و استفاده از سیستم‌های اصالحت‌سنجی برای جلوگیری از تغییرات ناخواسته در نظر گرفته شده است.

اولویت‌بندی ریسک‌ها

در این بخش ریسک‌های گفته شده در بخش قبل را در سه دسته‌ی «پر اهمیت»، «مهم» و «کم اهمیت» اولویت‌بندی می‌کنیم.

پر اهمیت

- نفوذ و هک
- حفظ حریم خصوصی
- خرابی سیستم
- تغییر نیازمندی‌ها
- عدم تعریف دقیق نیازمندی‌ها
- عدم تحقق زمان‌بندی
- عدم هماهنگی بین ذینفعان
- قوانین و مقررات
- تغییر انتظارات مشتری
- از بین رفتن قابلیت نگهداری و ایجاد
- ایمنی سامانه و داده‌ها

مهم

- بار ترافیک بالا
- مشکل در یکپارچه‌سازی با سیستم‌های موجود
- کمبود منابع انسانی و مالی
- هزینه‌های پیش‌بینی نشده
- مقاومت در برابر تغییر
- تغییرات بازار

کم اهمیت

- همگام‌سازی داده‌ها
- تست ناکافی
- عدم آشنایی کافی با برخی از تکنولوژی‌ها و چارچوب‌ها

تحلیل دنیای واقع

Customer	RestaurantOwner	DeliveryPerson	ShoppingCart
Price	Invoice	Restaurant	Menu
Rate	Review	Delivery	ShoppingCartStatus
Address	Payment	Food	Category
Location	ProfileInfo		

تحليل اسم/ فعل

Noun	Related Nouns/Verbs	Noun	Related Nouns/Verbs
Customer (مشتری)	<ul style="list-style-type: none"> • Register • Login • Logout • ModifyProfile • ViewRestaurants • AddToShoppingCart • Checkout • ViewOrderStatus • ViewOrderHistory • RateOrder • ReviewShoppingCart • ProfileInfo 	RestaurantOwner (مالک رستوران)	<ul style="list-style-type: none"> • ModifyRestaurant • ViewMenu • ModifyMenu • ViewOrders • ViewReviews • ViewRatings • ViewOrdersHistory • ChangeOrderStatus • ProfileInfo
DeliveryPerson (پیکر)	<ul style="list-style-type: none"> • ViewAssignedOrder • UpdateOrderStatus • ChangeOrderStatus • SendingLocation • ProfileInfo 	ShoppingCart (سبد خرید)	<ul style="list-style-type: none"> • AddToCart • ViewCart • ModifyCart • Payment • Address • CustomerInfo • List of Food • UpdateDeliveryStatus
Price (قیمت)	<ul style="list-style-type: none"> • Price 	Invoice (فاکتور)	<ul style="list-style-type: none"> • Address • CustomerInfo • List of Food • Price • DeliveryPerson • DateTime

Restaurant (رسوران)	<ul style="list-style-type: none"> ModifyProfile AddFood DeleteFood PrintInvoice ProfileInfo 	Menu (منو)	<ul style="list-style-type: none"> List of Food
Rate (امتیاز دادن)	<ul style="list-style-type: none"> Rate Customer ShoppingCart DeliveryPerson 	Review (نظر دادن)	<ul style="list-style-type: none"> Text Customer ShoppingCart
Delivery (تحویل دادن)	<ul style="list-style-type: none"> AssignDelivery TrackDelivery 	ShoppingCartStatus (وضعیت سبد خرید)	<ul style="list-style-type: none"> ShoppingCartStatus
Address (آدرس)	<ul style="list-style-type: none"> Street1 Street2 Alley UnitNumber Plate PostNumber 	Payment (پرداخت)	<ul style="list-style-type: none"> Price Status Gateway DateTime
Food (غذا)	<ul style="list-style-type: none"> Price Picture Description Count 	Category (دسته‌بندی)	<ul style="list-style-type: none"> List of Categories
Location (موقعیت مکانی)	<ul style="list-style-type: none"> Location 	ProfileInfo (اطلاعات پروفایل)	<ul style="list-style-type: none"> Name Phone Number Address

CRC کارت‌های

Customer	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> • Order (سفارش) • AuthenticationService (سرویس احراز هویت) 	<ul style="list-style-type: none"> • ویرایش پروفایل • مشاهده تاریخچه سفارش‌ها • امتیازدهی به سفارش

AuthenticationInfo	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> • Customer • RestaurantOwner • DeliveryPerson 	<ul style="list-style-type: none"> • ثبت‌نام • ورود • خروج • دریافت اطلاعات هویتی کاربر

RestaurantOwner	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> • Menu • Order • AuthenticationService 	<ul style="list-style-type: none"> • ویرایش پروفایل • اضافه کردن آیتم غذایی • حذف آیتم غذایی • مشاهده سفارش‌ها • تغییر وضعیت سفارش • دریافت لیست رستوران‌ها

DeliveryPerson	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> Order Location AuthenticationService 	<ul style="list-style-type: none"> مشاهده سفارش اختصاص یافته بهروزرسانی وضعیت سفارش ارسال موقعیت آزاد یا مشغول

ShoppingCart	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> Customer Order 	<ul style="list-style-type: none"> اضافه کردن آیتم به سبد خرید مشاهده سبد خرید ویرایش سبد خرید پرداخت نهایی محاسبه مجموع خالی کردن سبد خرید

Order	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> Customer ShoppingCart DeliveryPerson RestaurantOwner 	<ul style="list-style-type: none"> ثبت سفارش بهروزرسانی وضعیت مشاهده جزئیات محاسبه قیمت کل

Price	
همکاران	مسئولیت‌ها
• Food	• محاسبه قیمت

Invoice	
همکاران	مسئولیت‌ها
• Order • Customer	• نمایش آدرس • نمایش اطلاعات مشتری • نمایش لیست غذاها • نمایش قیمت • نمایش اطلاعات تحويل‌دهنده • نمایش تاریخ و زمان

Menu	
همکاران	مسئولیت‌ها
• Food • Restaurant • Category	• نمایش لیست غذاها

Food	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> • Category • ShoppingCart 	<ul style="list-style-type: none"> • نمایش قیمت • نمایش تصویر • نمایش توضیحات • بهروزرسانی موجودی • نمایش دسته‌بندی

Rate	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> • Customer • Order 	<ul style="list-style-type: none"> • امتیازدهی

Review	
همکاران	مسئولیت‌ها
<ul style="list-style-type: none"> • Customer • Order 	<ul style="list-style-type: none"> • ثبت نظر

Payment	
همکاران	مسئولیت‌ها
• Invoice	<ul style="list-style-type: none"> نمایش قیمت نمایش وضعیت پرداخت نمایش درگاه پرداخت نمایش تاریخ و زمان

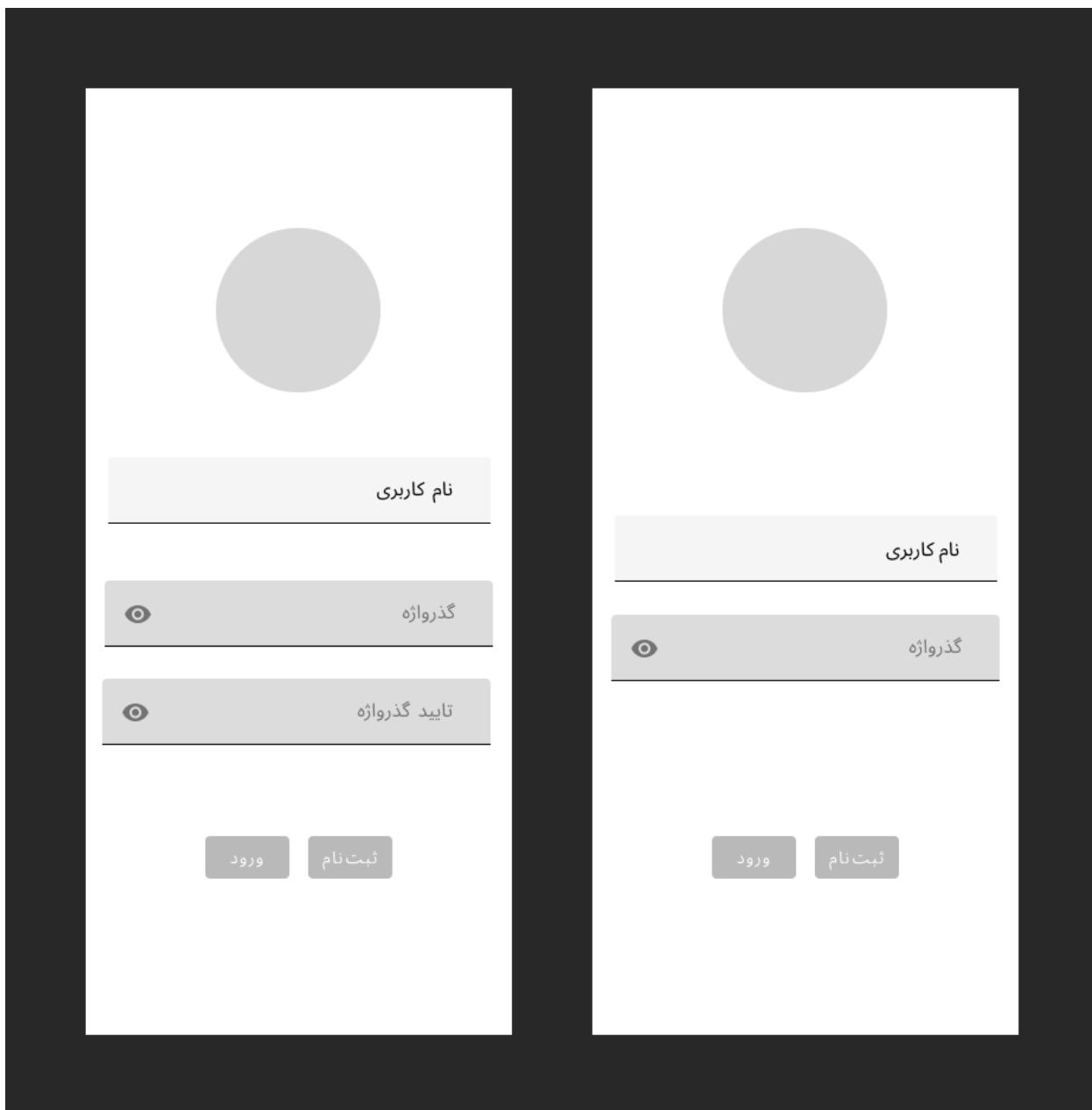
Category	
همکاران	مسئولیت‌ها
• Food	<ul style="list-style-type: none"> نمایش لیست دسته‌بندی‌ها

Location	
همکاران	مسئولیت‌ها
	<ul style="list-style-type: none"> ارسال موقعیت

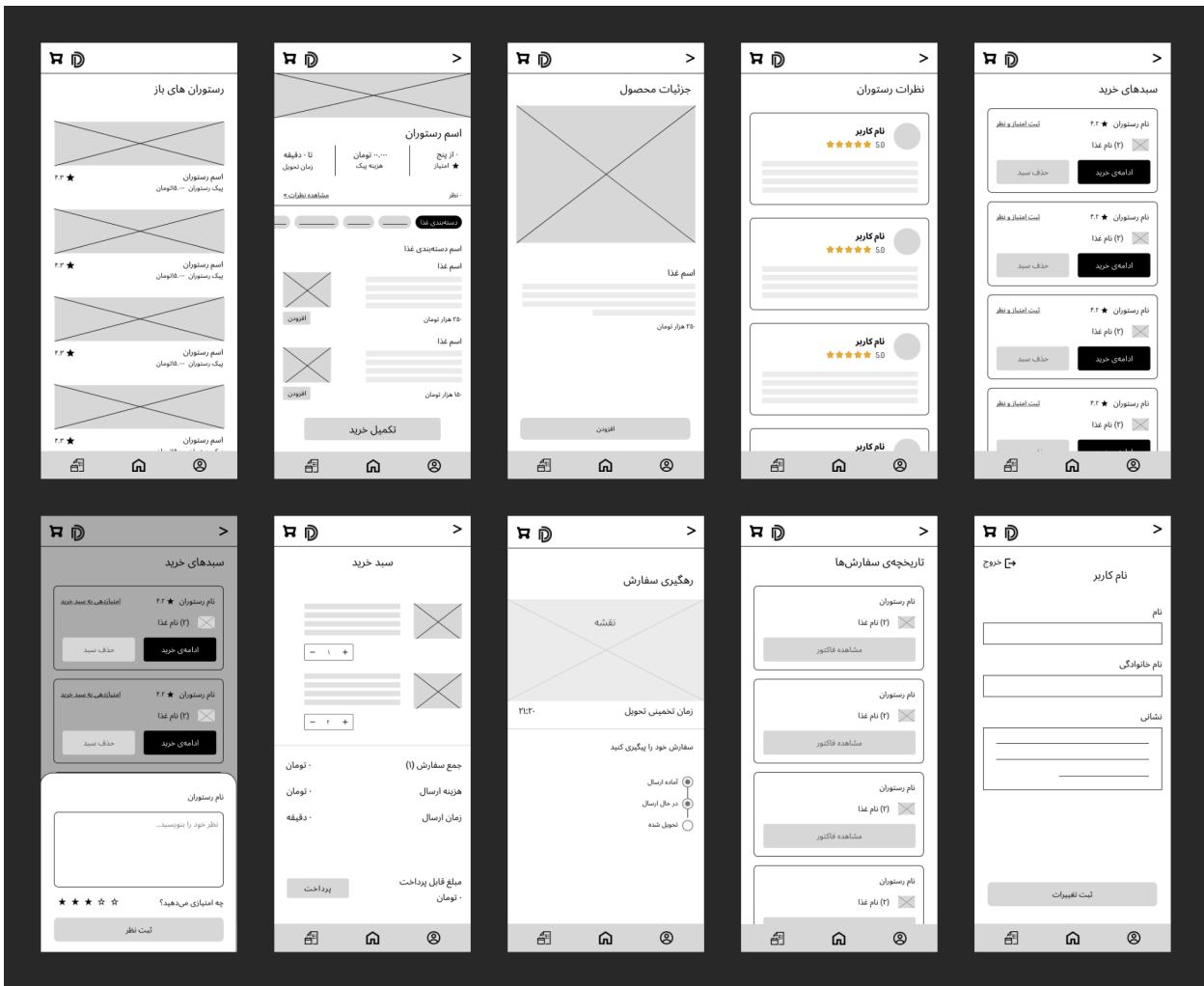
نمونه‌ی اولیه‌ی واسط کاربری قابل اجرا

صفحات زیر نمونه‌ی اولیه‌ی واسط کاربری هستند که در Figma پیاده‌سازی شده‌اند و پروتوتاپ شده‌اند. برای مشاهده‌ی نمونه‌ی اولیه‌ی واسط کاربری قابل اجرا، به این [لینک](#) مراجعه کنید و فیلم را تماشا کنید. همچنین برای مشاهده آن به طور مستقیم، می‌توانید به این [لینک](#) در Figma مراجعه کنید. (برای صرفه‌جویی در وقت فیلم تهیه شده است)

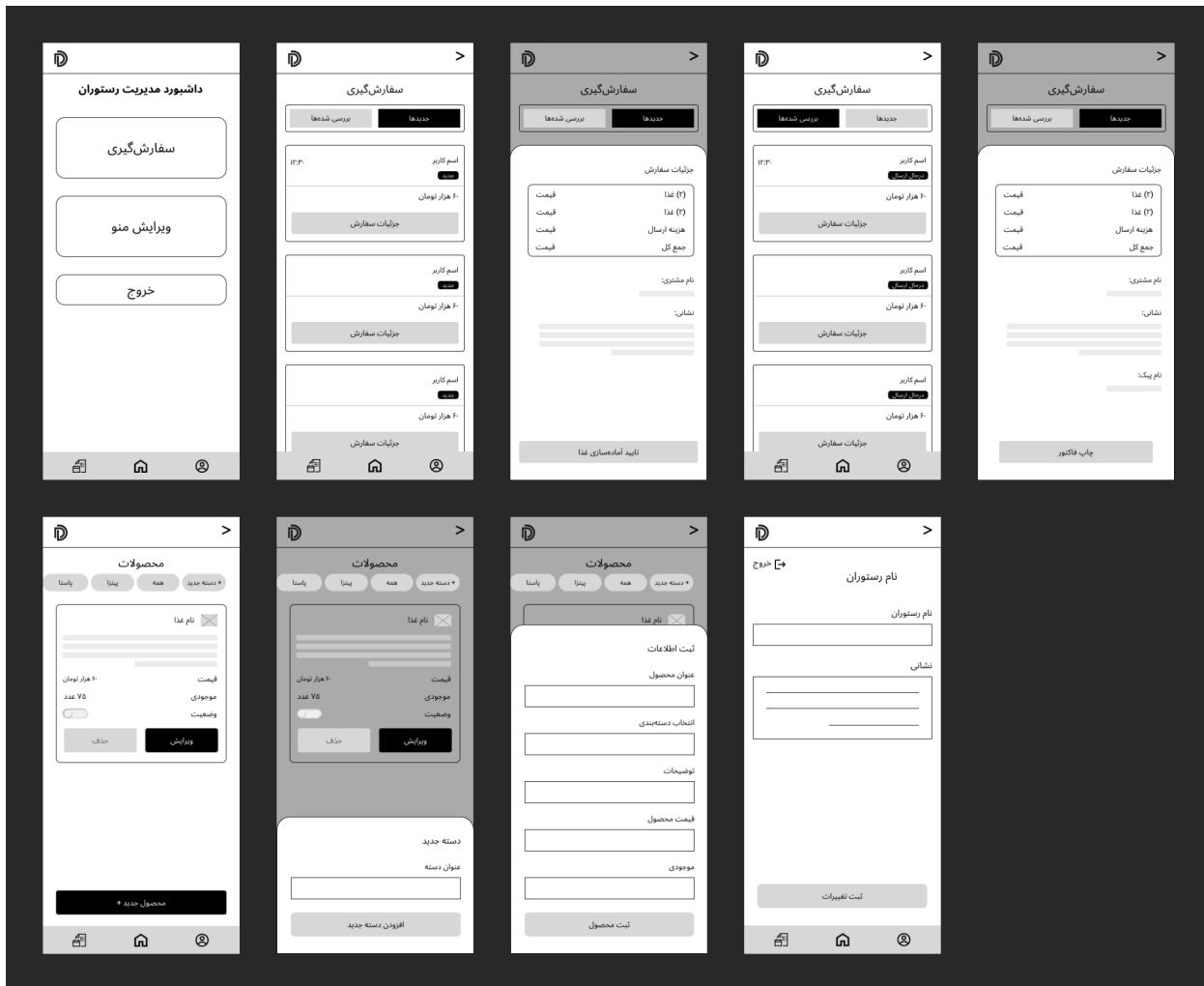
زیرسیستم احراز هویت و دسترسی



زیرسیستم مدیریت مشتریان



زیرسیستم مدیریت تهیه‌کنندگان غذا

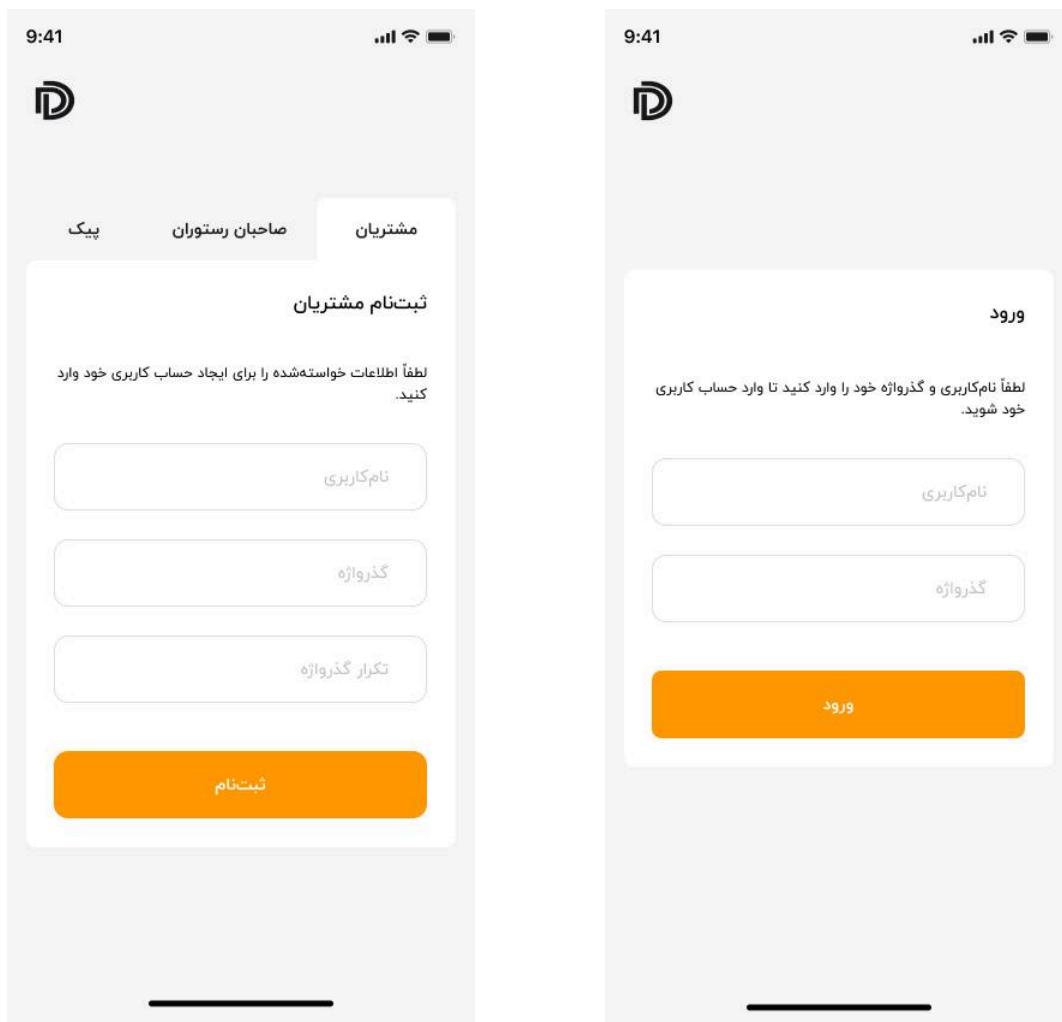


زیرسیستم مدیریت پیکها



DishDash کاربری سامانه‌ی تصاویر رابط

زیرسیستم احراز هویت و دسترسی



زیرسیستم مدیریت تهیه‌کنندگان غذا

The screenshots demonstrate the following features:

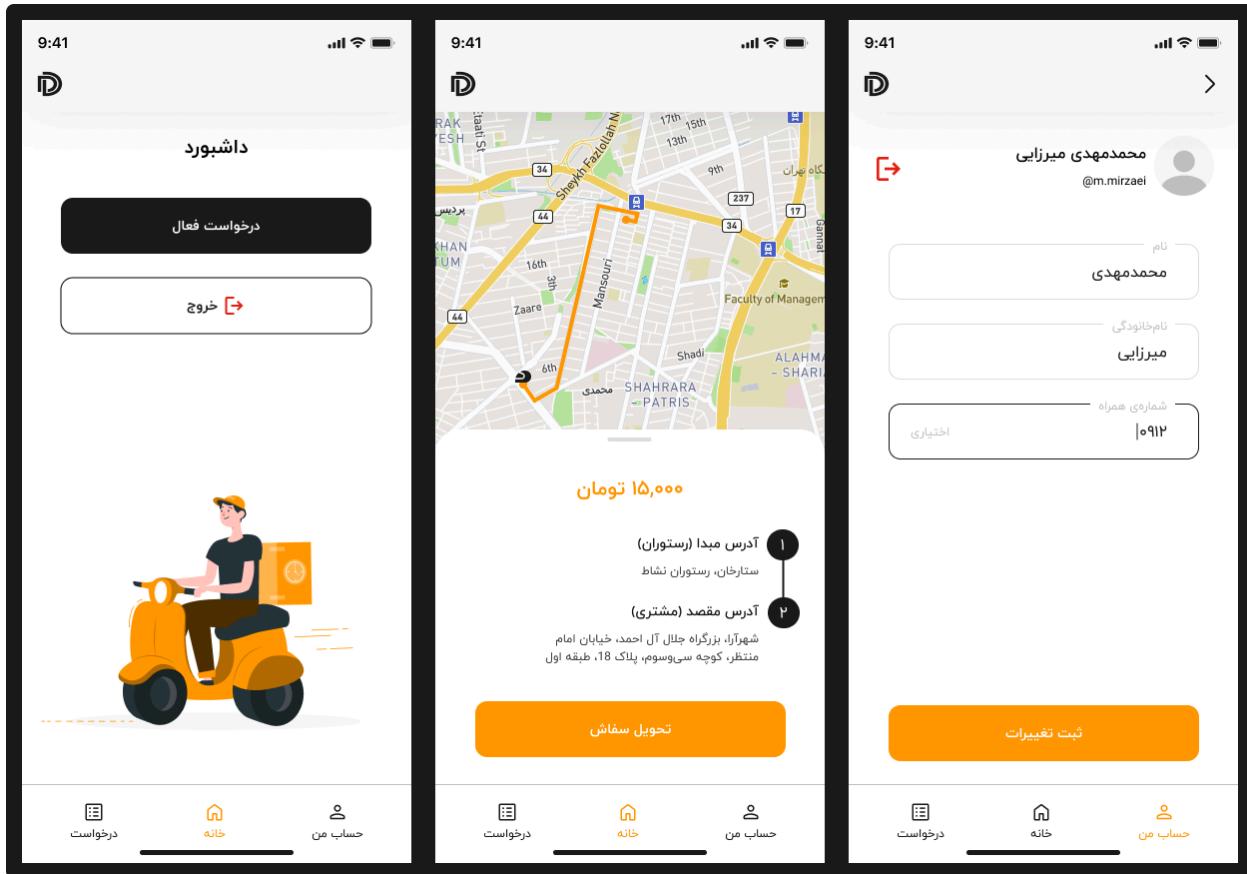
- Dashboard (داشبورد):** Shows a cooking illustration, a delivery icon, a delivery note icon, and a log-out button.
- Order Status (سفارشات آماده شده):** Displays three completed orders with details like name, time, price, and a 'View Details' button.
- Order Status (سفارشات فعال):** Shows a summary of active orders with a total value of 1,170,480.
- Customer Information (جزئیات سفارش):** Provides details for a customer named Amirreza Fakhimi, including address, contact information, and a 'Print' button.
- Product Catalog (محصولات):** Lists various dishes with names, descriptions, prices, and a 'Edit Product' button.
- New Product Creation (محصول جدید):** A form for adding new products with fields for name, description, price, and availability status.

زیرسیستم مدیریت مشتریان

The screenshots illustrate the user interface of a system for managing restaurant customers (Customers Management System). The interface includes:

- Top Left Screen:** Shows a list of restaurants with their names, ratings, and descriptions. It includes a search bar and navigation icons.
- Top Middle Screen:** Displays a detailed view of a restaurant's menu items, including names, descriptions, prices, and images. It also shows a summary of the total amount and a 'Continue Shopping' button.
- Top Right Screen:** Shows a shopping cart with items, including a pasta dish, its details, and a summary of the total amount.
- Bottom Left Screen:** A map showing the location of a restaurant with a highlighted route. Below it, there are sections for delivery status (در حال آماده سازی) and delivery history (تاریخچه سفارش ها).
- Bottom Middle Screen:** A list of delivery history entries, each with a thumbnail, name, rating, and a 'View Details' button.
- Bottom Right Screen:** A profile screen for a customer named 'محمدامین لطفی'. It shows basic information like name, address, and a large orange 'Edit Profile' button.

زیرسیستم مدیریت پیکها



گزارش استفاده از الگوهای طراحی در بک‌اند

پروژه با استفاده از معماری میکروسرویس‌ها ایجاد شده و توسعه یافته و از الگوهای طراحی متعددی برای بهبود قابلیت‌های توسعه، نگهداری، و مقیاس‌پذیری استفاده شده است. در این گزارش، به بررسی و توضیح الگوهای طراحی به کار رفته در پروژه، دلایل استفاده از هر یک، ارتباط آن‌ها با الگوهای GOF و محل استفاده‌ی آن‌ها در کد پروژه می‌پردازیم.

۱. معماری میکروسرویس‌ها

توضیحات: معماری میکروسرویس‌ها برنامه را به سرویس‌های کوچک‌تر و مستقل تقسیم می‌کند که هر کدام به صورت جداگانه قابل توسعه و استقرار هستند. این معماری مزایایی مانند توسعه و استقرار مستقل، ایزوله کردن خطاهای و مقیاس‌پذیری را به ارمغان می‌آورد.

۲. الگوی API Gateway

توضیحات: در این پروژه، از یک دروازه API به عنوان نقطه‌ی ورودی برای همه‌ی درخواست‌های کلاینت استفاده شده است. دروازه API وظایفی مانند مسیریابی درخواست‌ها، مدیریت احراز هویت، پایان‌دهی SSL و توازن بار را انجام می‌دهد.

چرا دروازه API بهترین انتخاب است؟ دروازه API نقطه‌ی مرکزی برای مدیریت درخواست‌ها و مسائل مقطعی فراهم می‌کند، که امنیت و کارایی را بهبود می‌بخشد.
ارتباط با GOF: مشابه الگوی Proxy دسترسی به سرویس‌ها را کنترل می‌کند.

۳. الگوی کشف سرویس سمت کلاینت

توضیحات: استفاده از ابزارهایی مانند Netflix Eureka به سرویس‌ها امکان می‌دهد تا به صورت دینامیک یکدیگر را کشف و با هم ارتباط برقرار کنند.

چرا بهترین انتخاب است؟ این الگو به سرویس‌ها اجازه می‌دهد تا بدون نیاز به نقاط انتهایی ثابت، یکدیگر را پیدا کنند، که برای مقیاس‌پذیری و انعطاف‌پذیری مهم است.
ارتباط با GOF: مشابه الگوی Observer، زیرا سرویس‌ها به صورت دینامیک ثبت و کشف می‌شوند.

۴. الگوی Feign Client

توضیحات: استفاده از Feign به عنوان یک کلاینت وب اعلانی، پیچیدگی درخواست‌های HTTP را کاهش می‌دهد.

چرا بهترین انتخاب است؟ Feign با استفاده از انوتیشن‌ها، کدهای کلاینت را ساده‌تر و قابل نگهداری‌تر می‌کند.
ارتباط با GOF: مشابه الگوی Proxy، زیرا یک رابط ساده برای عملکردۀای پیچیده یا راه دور فراهم می‌کند.

۵. الگوی پیکربندی خارجی

توضیحات: پیکربندی خارجی، تنظیمات را از کد جدا می‌کند.

چرا بهترین انتخاب است؟ امکان تغییر پیکربندی بدون نیاز به تغییر کد را فراهم می‌کند.

ارتباط با GoF: مشابه الگوی Strategy، زیرا استراتژی‌های پیکربندی مختلف را انتزاع می‌کند.

محل استفاده در پروژه:

- فایل‌های application.properties برای هر سرویس مانند authentication، gateway، order

۶. الگوی Singleton

توضیحات: الگوی Singleton تضمین می‌کند که یک کلاس تنها یک نمونه دارد و نقطه دسترسی جهانی به آن فراهم می‌کند.

چرا بهترین انتخاب است؟ بهینه‌سازی مصرف منابع در معماری میکروسرویس‌ها با حفظ سازگاری.

چرا دیگر الگوها مناسب نیستند؟ الگوهای Prototype یا دیگر الگوهای ساختاری می‌توانند منجر به مصرف غیرضروری منابع و عدم سازگاری در مدیریت منابع مشترک شوند.

محل استفاده در پروژه:

- استفاده پیش‌فرض از Singleton در تمام بین‌های Spring مانند AuthenticationService در authentication/src/main/java/com/dish_dash/authentication/application/service/Authe ntificationService.java

۷. الگوی (Spring Bean) (در Singleton)

توضیحات: Spring Bean‌ها به طور پیش‌فرض به صورت Singleton هستند، به این معنی که هر بین فقط یک بار برای هر کانتینر IoC ساخته می‌شود.

چرا بهترین انتخاب است؟ منابع گران‌قیمت مانند اتصالات پایگاه داده یا کلاس‌های پیکربندی یک بار ایجاد و استفاده می‌شوند.

ارتباط با GoF: مستقیماً با الگوی Singleton از GoF ارتباط دارد.

محل استفاده در پروژه:

- استفاده پیش‌فرض از RateService در تمام Bean‌ها Mانند RateService در order/src/main/java/com/dish_dash/order/application/service/RateService.java

۸. طراحی دامنه‌گرا (DDD)

توضیحات: این پروژه با استفاده از اصول طراحی دامنه‌گرا، منطق تجاری پیچیده را مدل‌سازی می‌کند. چرا بهترین انتخاب است؟ تضمین می‌کند که منطق تجاری به صورت دقیق و کامل مدل‌سازی شده و ارتباطات واضحی بین توسعه‌دهندگان و کارشناسان دامنه برقرار می‌شود.

محل استفاده در پروژه:

- مدل‌های دامنه مانند Order و Rate در .order/src/main/java/com/dish_dash/order/domain/model
- مخازن مانند RateRepository در .order/src/main/java/com/dish_dash/order/domain/repository/RateRepository.java

۹. ترکیب API

توضیحات: کنترلرهای API دروازه، داده‌ها را از چندین سرویس میکرو تجمیع کرده و یک پاسخ یکپارچه به کلاینت ارائه می‌دهند.

چرا بهترین انتخاب است؟ ترکیب API باعث کاهش تعداد تماس‌ها و بهبود کارایی می‌شود.
ارتباط با GoF: مشابه الگوی Facade، زیرا پیچیدگی‌های پشت صحنه را پنهان می‌کند و یک رابط ساده به کلاینت ارائه می‌دهد.

محل استفاده در پروژه:

- کنترلرهای API مانند RateController در .gateway/src/main/java/com/dish_dash/gateway/adapters/RateController.java

۱۰. الگوی مخزن (Repository)

توضیحات: استفاده از مخزن‌ها، لایه دسترسی به داده را انتزاع می‌کند.
چرا بهترین انتخاب است؟ جداسازی منطق تجاری از دسترسی به داده‌ها و تسهیل تست و نگهداری.
ارتباط با GoF: مشابه الگوی Strategy، زیرا استراتژی‌های دسترسی به داده را انتزاع می‌کند.

محل استفاده در پروژه:

- مخازن مانند RateRepository در .order/src/main/java/com/dish_dash/order/domain/repository/RateRepository.java

۱۱. الگوی سرویس

توضیحات: سرویس‌ها، منطق تجاری را از کنترلرهای مخزن‌ها جدا می‌کنند.
چرا بهترین انتخاب است؟ منطق تجاری را در یک مکان مرکزی قرار می‌دهد، که باعث مدولار بودن و مقیاس‌پذیری می‌شود.

ارتباط با GoF: مشابه الگوی Facade، زیرا پیچیدگی‌های منطق تجاری را مخفی می‌کند و یک رابط ساده به دیگر اجزا ارائه می‌دهد.

محل استفاده در پروژه:

- کلاس‌های سرویس مانند RateService در `.order/src/main/java/com/dish_dash/order/application/service/RateService.java`

۱۲. الگوی انتقال داده (DTO)

توضیحات: DTO‌ها داده‌ها را برای انتقال بین لایه‌ها یا سرویس‌ها تجمیع می‌کنند. چرا بهترین انتخاب است؟ تجمیع داده‌ها باعث کاهش تعداد تماس‌ها و بهبود کارایی شبکه می‌شود. ارتباط با GoF: مشابه الگوی Composite، زیرا داده‌های متعدد را در یک ساختار واحد تجمیع می‌کند.

محل استفاده در پروژه:

- DTO‌های مانند RateDto در `.common/src/main/java/com/dishDash/common/dto/RateDto.java`

۱۳. الگوی Adapter

توضیحات: الگوی Adapter به کلاس‌ها امکان می‌دهد تا رابطه‌ای ناسازگار را با هم سازگار کنند. چرا بهترین انتخاب است؟ در سیستم‌هایی با خدمات و API‌های متنوع خارجی، الگوی Adapter به یکپارچه‌سازی بدون تغییر در منطق اصلی سرویس کمک می‌کند.

ارتباط با GoF: مستقیم با الگوی Adapter از GoF ارتباط دارد.

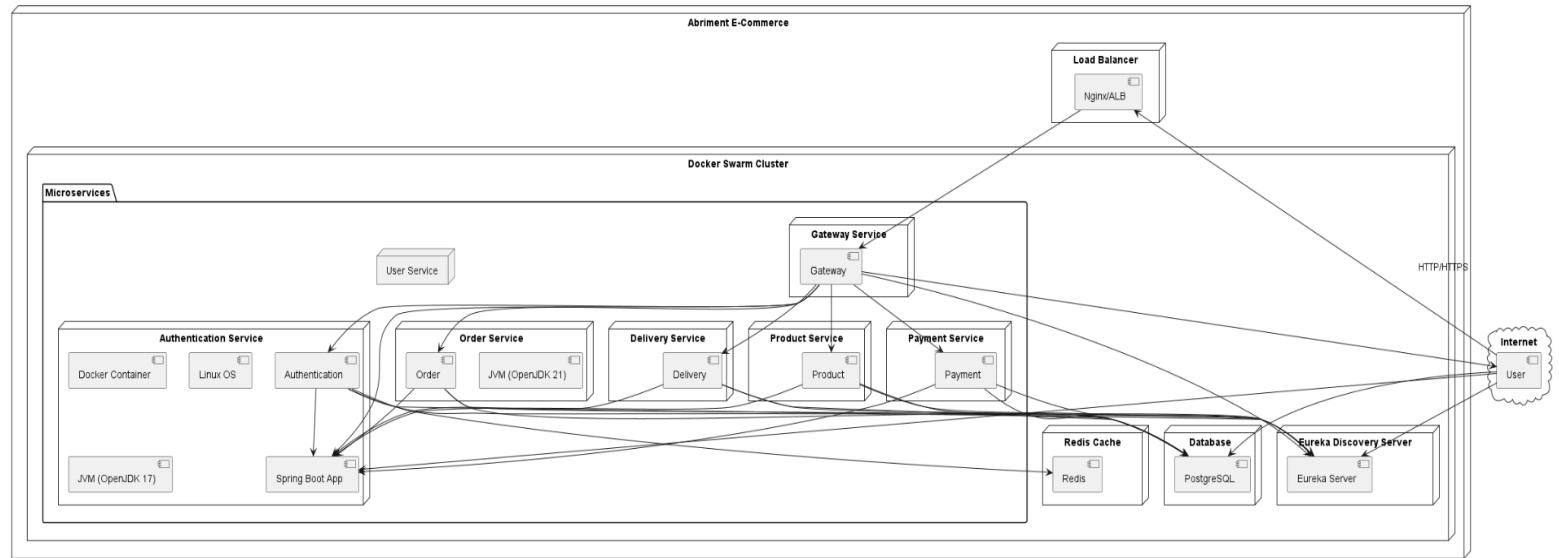
محل استفاده در پروژه:

- Mapper‌هایی مانند OrderMapper

شماتی پایگاه داده



نمودار استقرار



فهرست بررسی فاز تدقیق

یک خط مبنای معماري اجرائي² مقاوم و قوي ایجاد شده است.

تحویلدادنی: خط مبنای معماري اجرائي

خط مبنای معماري اجرائي نشان مى دهد که ريسک های مهم شناسايي و رفع شده اند.

تحویلدادنی ها:

مدل استاتيك UML

مدل ديناميک UML

مدل موارد استفاده UML

چشم انداز محصول ثبتیت شده است.

تحویلدادنی: سند چشم انداز

ارزیابی ريسک بازنگری شده است.

تحویلدادنی: ارزیابی ريسک به روز شده

یک برنامه پروژه به اندازه کافی دقیق ایجاد شده است.

تحویلدادنی: برنامه پروژه به روز شده

² Executable architectural baseline

فهرست بررسی جزئی‌تر فاز تدقیق

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
				✓	تحقیق یافتن موارد کاربرد
				✓	سازگاری نمودارها در طول فازها
				✓	روابط سازگار میان موارد کاربرد، نمودارهای فعالیت و کلاس‌های تحلیل
به این مورد در این تکرار به صورت جزئی پرداخته شده است و قرار است که طبق برنامه‌ی زمان‌بندی در تکرارهای آینده دقیق‌تر شود.		✓			وجود cohesion بالا بین کلاس‌ها
به این مورد در این تکرار به صورت جزئی پرداخته شده است و قرار است که طبق برنامه‌ی زمان‌بندی در تکرارهای آینده دقیق‌تر شود.		✓			وجود کم coupling بین کلاس‌ها
				✓	رفع و سنجش ریسک‌ها به وسیله‌ی EAB
				✓	مرتبه بودن کلاس‌های تحلیل با همکاران آن
				✓	مرتبه بودن کلاس‌های تحلیل با

مسئولیت‌های آن					
				<input checked="" type="checkbox"/>	روابط سازگار میان تحلیل‌های دنیای واقع، اسم/ فعل و کارت‌های CRC
به این مورد در این تکرار به صورت جزئی پرداخته شده است و قرار است که طبق برنامه‌ی زمان‌بندی در تکرارهای آینده دقیق‌تر شود.	<input checked="" type="checkbox"/>				استفاده‌ی صحیح و دقیق از gen/spec در موارد کاربرد
				<input checked="" type="checkbox"/>	تهیه و تنظیم سند چشم‌انداز
				<input checked="" type="checkbox"/>	برنامه‌ی زمان‌بندی در جهت انجام دقیق و کامل تمام بخش‌ها

فهرست بررسی نمودارهای فعالیت

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
				✓	تحقیق موارد کاربرد
				✓	سازگاری نمودارها در طول فازهای تدقیق
			✓		وجود cohesion بالا
			✓		وجود coupling کم
				✓	استفاده صحیح از الگوهای مناسب
				✓	روابط سازگار میان نمودارهای فعل و انفعال

فهرست بررسی نمودار کلاس‌های تحلیل

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
				✓	عدم وجود God Class
				✓	عدم وجود دور
			✓		توجه به اولویت روابط Aggregation و Composition
				✓	استفاده صحیح از Gen/Spec
				✓	اولویت روابط یکسویه نسب به دوسویه
				✓	نامگذاری صحیح و گویا
				✓	وجود cohesion بالا بین کلاس‌ها
				✓	وجود coupling کم بین کلاس‌ها

فهرست بررسی نمودارهای توالی تحلیل

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
				<input checked="" type="checkbox"/>	تحقیق موارد کاربرد
			<input checked="" type="checkbox"/>		سازگاری نمودارهای توالی با کلاس تحلیل
				<input checked="" type="checkbox"/>	استفاده از الگوهای
			<input checked="" type="checkbox"/>		ردیابی نمودارهای توالی از تحلیل به طراحی
			<input checked="" type="checkbox"/>		سازگاری نمودارها در طول فازهای تدقیق

فهرست بررسی نمودار بسته

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
			<input checked="" type="checkbox"/>		کمینه بودن رابطه‌ی use
			<input checked="" type="checkbox"/>		افراز کلاس‌ها
				<input checked="" type="checkbox"/>	عدم وجود دور
				<input checked="" type="checkbox"/>	کمینه کردن عناصر عمومی (public)
				<input checked="" type="checkbox"/>	بیشینه کردن عناصر خصوصی (private)

فهرست بررسی نمودار کلاس‌های طراحی

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
				✓	نمایش خروجی متدها
				✓	تمکیل بودن ورودی متدها و کلاس‌شان
				✓	واضح بودن چندی رابطه
				✓	استفاده صحیح از Gen/Spec
				✓	استفاده از الگوهای
				✓	عدم وجود God Class
				✓	عدم وجود دور
			✓		توجه به اولویت روابط Aggregation و Composition
				✓	اولویت روابط یک‌سویه نسب به دوسویه
				✓	نامگذاری صحیح و گویا
			✓		وجود cohesion بالا بین کلاس‌ها
			✓		وجود کم بین coupling کلاس‌ها

فهرست بررسی نمودارهای توالی طراحی

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
				<input checked="" type="checkbox"/>	تحقیق موارد کاربرد
				<input checked="" type="checkbox"/>	سازگاری نمودارهای توالی با کلاس طراحی
				<input checked="" type="checkbox"/>	استفاده از الگوها
			<input checked="" type="checkbox"/>		ردیابی نمودارهای توالی از طراحی به تحلیل
			<input checked="" type="checkbox"/>		سازگاری نمودارها در طول فازهای تدقیق

فهرست بررسی نمودار مولفه

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
			<input checked="" type="checkbox"/>		وجود cohesion بالا بین مولفه‌ها
			<input checked="" type="checkbox"/>		وجود coupling کم بین مولفه‌ها
				<input checked="" type="checkbox"/>	واضح بودن روابط
		<input checked="" type="checkbox"/>			استفاده صحیح از -connector‌ها
				<input checked="" type="checkbox"/>	سازگاری با نمودار کلاس‌های طراحی

فهرست بررسی پیاده‌سازی

معیار	عالی	خوب	متوسط	ضعیف	توضیحات
سازگاری با نمودارهای کلاس طراحی	✓				
خوانایی کد	✓				
واسطه‌های صحیح	✓				
بدون حالت بودن سرور	✓				
زیرسیستم‌های مستقل	✓				
پکیج‌های مناسب برای کد	✓				

فهرست بررسی شمای پایگاه داده

معیار	عالی	خوب	متوسط	ضعیف	توضیحات
تعیین الزام یکتایی مقدارهای ستون‌ها	✓				
تعیین کلیدهای فرعی	✓				
تعیین کلیدهای اصلی	✓				
انطباق با نمودار کلاس‌های طراحی	✓				
تعیین انواع ارتباط	✓				
تعیین امکان خالی بودن ستون‌ها	✓				
تعیین نوع ستون‌ها	✓				

فهرست بررسی الگوها

معیار	عالی	خوب	متوسط	ضعیف	توضیحات
قابل استفاده مجدد ^۳ بودن الگوها	✓				
قابل انعطاف بودن الگوها	✓				
قابل درک بودن الگوها	✓				
قابل آزمون بودن الگوها	✓				
قابل اعتماد بودن الگوها	✓				
قابل ردیابی بودن با کد پیاده‌سازی شده	✓				
سازگاری الگوها با نمودارهای مختلف	✓				
واضح بودن الگوهای استفاده شده	✓				
تحقیق نیازمندی‌ها توسط الگوها	✓				
مشخص کردن مسئله، راه حل، و زمینه‌ی مسئله برای الگوهای بالا	✓				

^۳ Reusable

فهرست بررسی محصولات

توضیحات	ضعیف	متوسط	خوب	عالی	معیار
				<input checked="" type="checkbox"/>	تحقیق یافتن تمامی موارد کاربرد
				<input checked="" type="checkbox"/>	استفاده‌ای درست از الگوهای
				<input checked="" type="checkbox"/>	سازگار بودن نمودارها در فازهای مختلف
				<input checked="" type="checkbox"/>	وجود cohesion بالا در بین اجزای درون کلاس‌ها و بسته‌ها
				<input checked="" type="checkbox"/>	وجود coupling کم میان کلاس‌ها و بسته‌ها
				<input checked="" type="checkbox"/>	قابلیت گسترش مولفه‌ها و سازگاری آن‌ها با واسطه‌ها
				<input checked="" type="checkbox"/>	شناسایی و اولویت‌بندی درست نیازمندی‌ها
				<input checked="" type="checkbox"/>	شناسایی همه‌ی نیازمندی‌های مدنظر کاربر
				<input checked="" type="checkbox"/>	رابط کاربری فوق العاده
				<input checked="" type="checkbox"/>	تجربه کاربری خوب
				<input checked="" type="checkbox"/>	قابلیت ردیابی نمودار تحلیل به طراحی
				<input checked="" type="checkbox"/>	قابلیت ردیابی از مولفه‌ها در پیاده‌سازی به بسته‌های طراحی
			<input checked="" type="checkbox"/>		کاهش ریسک‌های معماري با تحقیق یافتن Architecture baseline

برنامه‌ی زمانبندی پروژه

پایان	آغاز	انجام دهنده	مدت (۴MD ^۴)	نام تسک	شناسه
فاز آغاز					
۱۴۰۳/۰۲/۲۷	۱۴۰۳/۰۲/۲۶	امیرمحمد فخیمی	۰.۵	طراحی قالب برای سند	۱
۱۴۰۳/۰۲/۲۹	۱۴۰۳/۰۲/۲۸	کل تیم	۱	استخراج نیازمندی‌های وظیفه‌ای	۲
۱۴۰۳/۰۲/۳۰	۱۴۰۳/۰۲/۳۰	محمدامین لطفی	۰.۵	استخراج نیازمندی‌های غیروظیفه‌ای	۳
۱۴۰۳/۰۳/۰۱	۱۴۰۳/۰۲/۳۰	امیرحسین عربزاده	۲	استخراج ریسک‌ها	۴
۱۴۰۳/۰۳/۰۲	۱۴۰۳/۰۳/۰۱	محمدامین لطفی	۱	معماری سیستم	۵
۱۴۰۳/۰۳/۰۵	۱۴۰۳/۰۳/۰۲	کل تیم	۳	توصیف موارد کاربرد	۶
۱۴۰۳/۰۳/۰۵	۱۴۰۳/۰۳/۰۴	کل تیم	۱	رسم نمودارهای موارد کاربرد	۷
۱۴۰۳/۰۳/۰۵	۱۴۰۳/۰۳/۰۵	امیرحسین عربزاده	۰.۵	واژه‌نامه	۸
۱۴۰۳/۰۳/۰۵	۱۴۰۳/۰۳/۰۵	کل تیم	۰.۵	ویرایش نهایی سند فاز آغاز	۹
فاز تدقیق - تکرار اول					
۱۴۰۳/۰۳/۰۶	۱۴۰۳/۰۳/۰۵	کل تیم	۱	استخراج و به روزرسانی نیازمندی‌های وظیفه‌ای	۱۰
۱۴۰۳/۰۳/۰۶	۱۴۰۳/۰۳/۰۶	کل تیم	۰.۵	استخراج و به روزرسانی نیازمندی‌های غیروظیفه‌ای	۱۱
۱۴۰۳/۰۳/۰۶	۱۴۰۳/۰۳/۰۶	کل تیم	۰.۵	به روزرسانی ریسک‌ها	۱۲
۱۴۰۳/۰۳/۰۸	۱۴۰۳/۰۳/۰۷	کل تیم	۱	به روزرسانی توصیف موارد کاربرد	۱۳
۱۴۰۳/۰۳/۰۸	۱۴۰۳/۰۳/۰۸	محمدامین لطفی	۰.۵	به روزرسانی نمودارهای موارد کاربرد	۱۴
۱۴۰۳/۰۳/۰۸	۱۴۰۳/۰۳/۰۸	محمدمهدی میرزایی	۰.۵	به روزرسانی واژه‌نامه	۱۵
۱۴۰۳/۰۳/۱۰	۱۴۰۳/۰۳/۰۹	کل تیم	۱	تحقیق‌سازی موارد کاربرد	۱۶
۱۴۰۳/۰۳/۱۱	۱۴۰۳/۰۳/۱۰	امیرحسین عربزاده	۰.۵	تحلیل دنیای واقع	۱۷

^۴ Man-day / نفر-روز

		امیرمحمد فخیمی، محمدمهدی میرزایی			
۱۴۰۳/۰۳/۱۱	۱۴۰۳/۰۳/۱۰	امیرحسین عربزاده، امیرمحمد فخیمی، محمدمهدی میرزایی	۰.۵	تحلیل اسم/ فعل	۱۸
۱۴۰۳/۰۳/۱۲	۱۴۰۳/۰۳/۱۱	کل تیم	۱	CRC کارت‌های	۱۹
۱۴۰۳/۰۳/۱۶	۱۴۰۳/۰۳/۱۲	کل تیم	۴	نمودارهای فعالیت	۲۰
۱۴۰۳/۰۳/۱۶	۱۴۰۳/۰۳/۱۶	کل تیم	۰.۵	فهرست بررسی فاز تدقیق	۲۱
۱۴۰۳/۰۳/۱۸	۱۴۰۳/۰۳/۱۶	محمدامین لطفی، محمدمهدی میرزایی	۱	Executable پیاده‌سازی Architectural Baseline	۲۲
۱۴۰۳/۰۳/۱۶	۱۴۰۳/۰۳/۱۶	محمدامین لطفی	۰.۵	زمان‌بندی فاز تدقیق	۲۳
۱۴۰۳/۰۳/۱۶	۱۴۰۳/۰۳/۱۳	محمدامین لطفی	۳	نمونه‌ی اولیه‌ی واسط کاربری قابل اجرا	۲۴

فاز تدقیق - تکرار دوم

۱۴۰۳/۰۴/۰۹	۱۴۰۳/۰۴/۰۵	کل تیم	۴	افزودن خطوط شنا به نمودارهای فعالیت	۲۵
۱۴۰۳/۰۴/۱۱	۱۴۰۳/۰۴/۰۹	محمدامین لطفی، امیرحسین عربزاده	۲	نمودار کلاس‌های تحلیل	۲۶
۱۴۰۳/۰۴/۱۶	۱۴۰۳/۰۴/۱۱	امیرمحمد فخیمی، محمدمهدی میرزایی	۵	نمودارهای توالی تحلیل	۲۷
۱۴۰۳/۰۴/۱۷	۱۴۰۳/۰۴/۱۶	محمدامین لطفی، امیرحسین عربزاده	۱	نمودار بسته	۲۸
۱۴۰۳/۰۴/۲۵	۱۴۰۳/۰۴/۲۵	محمدامین لطفی	۰.۵	به روزرسانی واژه‌نامه به ترتیب حروف الفبا	۲۹
۱۴۰۳/۰۴/۲۰	۱۴۰۳/۰۴/۱۶	محمدامین لطفی، امیرحسین عربزاده	۴	نمودار کلاس‌های طراحی	۳۰
۱۴۰۳/۰۴/۲۲	۱۴۰۳/۰۴/۱۶	امیرحسین عربزاده، امیرمحمد فخیمی، محمدمهدی میرزایی	۷	نمودارهای توالی طراحی	۳۱

۱۴۰۳/۰۴/۱۹	۱۴۰۳/۰۴/۱۸	محمدامین لطفی، امیرحسین عربزاده	۱	نمودار مولفه	۳۲
۱۴۰۳/۰۴/۲۲	۱۴۰۳/۰۴/۱۶	محمدامین لطفی	۷	طراحی رابط کاربری در فیگما	۳۳
۱۴۰۳/۰۴/۲۲	۱۴۰۳/۰۴/۱۹	امیرحسین عربزاده	۳	تمکیل Executable Architectural Baseline	۳۴
۱۴۰۳/۰۳/۲۸	۱۴۰۳/۰۳/۲۸	کل تیم	۰.۵	بهروزرسانی ریسکها	۳۵
۱۴۰۳/۰۴/۲۲	۱۴۰۳/۰۴/۲۲	محمدامین لطفی	۰.۵	افزودن تصاویر رابط کاربری	۳۶
۱۴۰۳/۰۴/۲۰	۱۴۰۳/۰۴/۱۹	امیرمحمد فخیمی	۱	چک لیست	۳۷
۱۴۰۳/۰۳/۲۶	۱۴۰۳/۰۳/۲۶	محمدمهدی میرزاپی	۰.۵	بهروزرسانی اولویت نیازمندی‌های وظیفه‌ای	۳۹
۱۴۰۳/۰۴/۲۲	۱۴۰۳/۰۴/۲۲	محمدامین لطفی	۱	زمان‌بندی فاز تدقیق	۴۰

فاز ساخت - تکرار اول

۱۴۰۳/۰۴/۲۲	۱۴۰۳/۰۴/۲۲	محمدامین لطفی	۰.۵	بهروزرسانی اولویت نیازمندی‌های وظیفه‌ای	۴۱
۱۴۰۳/۰۴/۲۳	۱۴۰۳/۰۴/۲۳	امیرمحمد فخیمی	۰.۵	بهروزرسانی اولویت ریسکها	۴۲
۱۴۰۳/۰۴/۲۴	۱۴۰۳/۰۴/۲۳	کل تیم	۱	بازنگری نمودارهای کلاس تحلیل	۴۳
۱۴۰۳/۰۴/۲۵	۱۴۰۳/۰۴/۲۴	کل تیم	۱	بازنگری نمودارهای توالی تحلیل	۴۴
۱۴۰۳/۰۴/۲۶	۱۴۰۳/۰۴/۲۵	کل تیم	۱	بازنگری نمودارهای کلاس طراحی	۴۵
۱۴۰۳/۰۴/۲۷	۱۴۰۳/۰۴/۲۶	کل تیم	۱	بازنگری نمودارهای توالی طراحی	۴۶
۱۴۰۳/۰۴/۲۸	۱۴۰۳/۰۴/۲۷	امیرحسین عربزاده	۱	شرح الگوهای اعمال شده	۴۷
۱۴۰۳/۰۴/۳۰	۱۴۰۳/۰۴/۲۳	کل تیم	۸	نمونه‌ی اولیه‌ی تکامل یافته سیستم	۴۸
۱۴۰۳/۰۴/۳۰	۱۴۰۳/۰۴/۲۹	محمدامین لطفی	۱	چک لیست	۴۹
۱۴۰۳/۰۴/۲۹	۱۴۰۳/۰۴/۲۸	محمدمهدی میرزاپی	۱	استقرار کد روی سرور	۵۰
۱۴۰۳/۰۴/۲۴	۱۴۰۳/۰۴/۲۳	محمدامین لطفی	۱	تمکیل رابط کاربری در فیگما	۵۱

فاز ساخت - تکرار دوم

۱۴۰۳/۰۴/۲۷	۱۴۰۳/۰۴/۲۷	محمد مهدی میرزایی	۰.۵	بازنگری نیازمندی‌ها	۵۲
۱۴۰۳/۰۴/۳۰	۱۴۰۳/۰۴/۳۰	امیر محمد فخیمی	۰.۵	بازنگری ریسک‌ها	۵۳
۱۴۰۳/۰۵/۱۲	۱۴۰۳/۰۵/۱۲	محمد امین لطفی، امیر محمد فخیمی	۲	مستند نحوه استفاده	۵۴
۱۴۰۳/۰۵/۱۲	۱۴۰۳/۰۵/۱۲	امیرحسین عربزاده	۱	مستند نصب	۵۴
۱۴۰۳/۰۵/۰۹	۱۴۰۳/۰۵/۰۲	کل تیم	۸	نسخه‌ی نهایی سامانه	۵۵
۱۴۰۳/۰۵/۰۱	۱۴۰۳/۰۵/۰۱	محمد امین لطفی	۰.۵	زمان‌بندی فاز ساخت - تکرار دوم	۵۶
۱۴۰۳/۰۵/۱۲	۱۴۰۳/۰۵/۱۱	محمد امین لطفی	۱	چک لیست	۵۷
۱۴۰۳/۰۵/۱۲	۱۴۰۳/۰۵/۱۱	امیرحسین عربزاده	۲	نمودار استقرار	۵۸
۱۴۰۳/۰۵/۱۲	۱۴۰۳/۰۵/۱۰	کل تیم	۲	آزمون سامانه	۵۹
فاز گذار					
۱۴۰۳/۰۵/۱۳	۱۴۰۳/۰۵/۱۳	محمد امین لطفی	۰.۵	زمان‌بندی فاز گذار	۶۰
۱۴۰۳/۰۵/۱۳	۱۴۰۳/۰۵/۱۳	کل تیم	۱	تکمیل نسخه نهایی طبق نتایج آزمون‌ها	۶۱

واژه‌نامه

هم‌معنی‌ها	تعریف	واژه
الف		
-	فرآیندی که طی آن سیستم هویت کنشگر را با بررسی نام کاربری و رمز عبور تأیید می‌کند.	احراز هویت
-	فرآیندی که طی آن مشتری می‌تواند به کیفیت خدمات سفارش یا پیک امتیاز دهد.	امتیازدهی
پ		
-	فرآیندی که طی آن مشتری هزینه‌ی سفارش خود را پرداخت می‌کند و سفارش ثبت می‌شود.	پرداخت
راننده‌ی پیک، تحویل‌دهنده	فردی که مسئول دریافت و تحویل سفارشات، به روزرسانی وضعیت سفارش و ارائهٔ موقعیت مکانی زنده با سیستم تعامل دارد.	پیک
-	پیکی که در زمان تخصیص پیک برای هر سفارش، آزاد باشد و مشغول تحویل سفارش دیگری نباشد.	پیک آزاد
-	پیکی که در زمان تخصیص پیک برای هر سفارش، غیر فعال (مثلاً خواب) باشد.	پیک غیر فعال
-	پیکی که در زمان تخصیص پیک برای هر سفارش، فعال و آماده‌ی به کار (مثلاً خواب نباشد.) باشد.	پیک فعال

-	پیک که در زمان تخصیص پیک برای هر سفارش، مشغول تحویل سفارش دیگری باشد.	پیک مشغول
ث		
رجیستر	فرآیندی که طی آن کنشگر یک حساب کاربری جدید در سیستم ایجاد می‌کند.	ثبت نام
ح		
-	فرآیندی که طی آن کنشگر حساب کاربری خود را به طور دائمی از سیستم حذف می‌کند.	حذف حساب کاربری
خ		
-	فرآیندی که طی آن کنشگر از حساب کاربری خود خارج می‌شود.	خروج
ر		
-	استفاده از تکنیک‌های رمزگاری برای حفاظت از اطلاعات حساس در حین انتقال و ذخیره‌سازی.	رمزگذاری داده
-	نهادی که برای مدیریت منو، مشاهده و آماده‌سازی سفارشات و بهروزرسانی وضعیت سفارشات با سیستم تعامل دارد.	رستوران
س		
سیستم مدیریت غذا	سیستم نرم‌افزاری که برای مدیریت سفارشات غذا، تحویل و پرداختها استفاده می‌شود.	سامانه مدیریت توزیع غذا (DishDash)

-	مجموعه‌ای از اقلام غذایی که مشتری انتخاب کرده و قصد خرید آنها را دارد.	سبد خرید
-	مجموعه‌ای از اجزا و مؤلفه‌های نرم‌افزاری و سخت‌افزاری که با هم همکاری می‌کنند تا سامانه‌ی مدیریت توزیع غذا (DishDash) را اجرا کنند و رفتار آن توسط موارد کاربرد توصیف می‌شود.	سیستم
-	زیرسیستمی که مسئولیت پردازش پرداخت‌های مشتریان را بر عهده دارد.	سیستم پرداخت

غ

	شامل تمام محصولاتی می‌شود که رستوران درون فهرست غذای خود قرار می‌دهد که شامل انواع خوراکی‌ها از جمله پیش‌غذا و نوشیدنی‌ها می‌شود.	غذا
--	---	-----

ک

-	فردی که از سامانه‌ی مدیریت توزیع غذا استفاده می‌کند، شامل مشتری، رستوران و پیک می‌شود.	کاربر
-	هر موجودی (انسان یا سیستم) که با سامانه تعامل دارد.	کنشگر

م

صرف‌کننده	فردی که برای ثبت سفارش، مشاهده‌ی اطلاعات رستوران‌ها و مدیریت حساب خود با سیستم تعامل دارد.	مشتری
	شامل تمام مواردی می‌شود که رستوران به مشتریان ارائه می‌دهد.	منو
-	توصیف تعاملات بین کنشگرها و سیستم برای دستیابی به هدف خاصی در قالب یک سناریو.	موارد کاربرد ⁵

⁵ Use Case

-	ارسال لحظه‌ای موقعیت مکانی پیک به مشتری و رستوران برای پیگیری وضعیت تحويل سفارش.	موقعیت مکانی زنده
ن		
-	فرآیندی که طی آن مشتری می‌تواند نظر خود را درباره‌ی سبد خرید ثبت کند.	نظردهی
۹		
لاغین	فرآیندی که طی آن کنشگر با وارد کردن نام کاربری و رمز عبور به سیستم دسترسی پیدا می‌کند.	ورود
-	وضعیت فعلی سفارش که می‌تواند شامل مراحل مختلفی مانند «در حال آماده‌سازی»، «در حال ارسال» و «تحویل شده» باشد.	وضعیت سفارش

لینک‌ها

- [مخزن مستندات](#)
- [مخزن فرانت‌اوند](#)
- [مخزن بکاند](#)
- [فیگما](#)
- [کانیان](#)
- [فیلم توضیح فرانت‌اوند - فاز ساخت - تکرار اول](#)
- [فیلم توضیح بکاند - فاز ساخت - تکرار اول](#)
- [مستند نحوه‌ی استفاده از سامانه](#)
- [مستند نصب](#)