

# سیستم فروش بلیط سینما



Program :

**Soheil Nematollahi & mohammadJavad Khaje**

**2025/04/07**

Github :

**SoheilNematollahi & Mohammadcode20**

<b>صفده مقدمه</b>	.....:3
<b>صفده ویژگی های سیستم فروش بلیط سینما</b>	.....:4,5
<b>صفده تخمین زمانی ساخت پروژه</b>	.....:6
<b>( use case ) توضیحات و نمودار UML</b>	.....: 9-7
<b>Activity Diagram</b>	.....:13-10
<b>Class Diagram</b>	.....:15-14
<b>Object Diagram</b>	.....:17-16
<b>Sequence</b>	.....:19-18
<b>Component Diagram</b>	.....:20-21
<b>Deployment Diagram</b>	.....:24-22
<b>Package Diagram</b>	.....: 25-26
<b>Composite Structure Diagram</b>	.....:28-27
<b>State Machine Diagram</b>	.....:30-29
<b>Communication Diagram</b>	.....:33-31
<b>Timing Diagram</b>	.....:35-34

## مقدمه

### معرفی سیستم فروش بلیط سینما

با گسترش فناوری اطلاعات و رشد استفاده از اینترنت در زندگی روزمره، نیاز به ارائه خدمات آنلاین در حوزه‌های مختلف بیش از پیش احساس می‌شود. صنعت سینما نیز از این قاعده مستثنی نیست. یکی از نیازهای اساسی مخاطبان سینما، دسترسی آسان و سریع به خرید بلیط فیلم‌ها بدون مراجعه حضوری به گیشه‌ها است.

سیستم فروش آنلاین بلیط سینما، یک سامانه‌ی تحت وب است که با هدف سهولت خرید بلیط، افزایش سرعت خدمات دهی، کاهش ازدحام و مدیریت بهینه صندلی‌ها طراحی می‌شود. این سیستم امکان مشاهده لیست فیلم‌ها، بررسی سانس‌های موجود، انتخاب صندلی، و پرداخت آنلاین بلیط را برای کاربران فراهم می‌کند. همچنین، ادمین سیستم می‌تواند اطلاعات مربوط به فیلم‌ها، سالن‌ها و سانس‌های نمایش را مدیریت نماید.

با بهره‌گیری از این سیستم، مشتریان قادر خواهند بود با چند کلیک ساده، بدون نیاز به ایستادن در صفحه‌ای طولانی، بلیط مورد نظر خود را رزرو یا خریداری کرده و در زمان مورد نظر در سالن سینما حضور یابند. این امر نه تنها تجربه‌ی کاربری بهتری برای مخاطبان فراهم می‌کند، بلکه باعث بهینه‌سازی فرآیندهای داخلی سینما نیز خواهد شد.

## ویژگی‌های سیستم فروش بلیط سینما

برای کاربران (مشتریان) :

ثبت‌نام و ورود به سیستم

ثبت‌نام با ایمیل یا شماره تماس

ورود امن با رمز عبور یا پیامک/ایمیل تأیید

نمایش لیست فیلم‌ها

به همراه اطلاعات کامل: ژانر، مدت، پوستر، خلاصه داستان

مشاهده سانس‌های موجود

امکان مشاهده سانس‌ها بر اساس تاریخ، سینما، زبان و نوع فیلم (سه‌بعدی، دوبله و...)

انتخاب سالن و صندلی

نقشه گرافیکی سالن برای انتخاب دقیق صندلی

نمایش صندلی‌های خالی و رزرو شده

## پرداخت آنلاین ایمن

اتصال به درگاههای بانکی معتبر

ذخیره سوابق تراکنش‌ها

## دریافت بلیط دیجیتال (PDF) یا QR Code)

امکان دریافت بلیط از طریق ایمیل یا دانلود مستقیم

لغو رزرو یا تغییر سانس (در صورت مجاز بودن)

طبق قوانین هر سینما

## پروفایل کاربری و مشاهده سوابق خرید

برای مدیر سیستم (ادمین) :

مدیریت فیلم‌ها

افزودن، ویرایش، حذف اطلاعات فیلم

مدیریت سانس‌های نمایش

تعریف زمان، سالن، تعداد صندلی، زبان و نوع نمایش

مدیریت سالن‌ها و نقشه‌ی صندلی‌ها

بررسی گزارش‌ها و آمار فروش

مشاهده بلیت‌های فروخته شده، سانس‌های شلوغ و...

## مدیریت کاربران

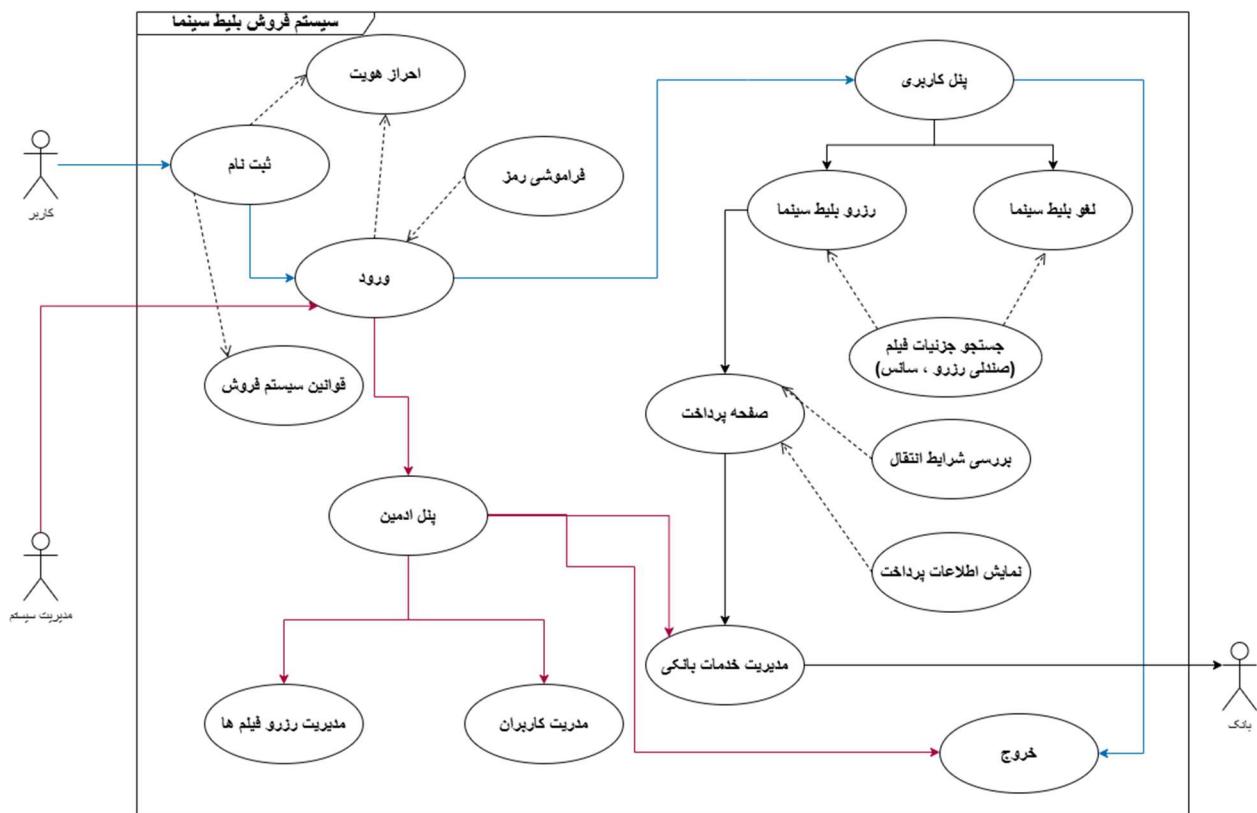
### مسوده‌سازی یا تأیید حساب‌ها در صورت نیاز

### تخمین زمانی ساخت سیستم

فاز پروژه	توضیحات	مدت زمان تقریبی
1. تحلیل نیازمندی‌ها	بررسی کاربران، امکانات، مدل‌سازی اولیه (Activity , class , use case ,...)	3 تا 5 روز
2. طراحی دیتابیس	طراحی جداول، ارتباطات و ساختار اولیه اطلاعات	3 تا 4 روز
3. طراحی رابط کاربری (UI)	طراحی صفحات وب (css / html / react )	4 تا 6 روز
4. پیاده‌سازی Backend	توسعه api ها / کنترلر / مدل ها ( Laravel / node.js/Django )	8 تا 12 روز
5. پیاده‌سازی Frontend	ارتباط با API ، منطق صفحات، نمایش داده‌ها	6 تا 10 روز
6. اتصال درگاه پرداخت	استفاده از زرین‌پال، آیدی‌پی یا سایر درگاه‌های بانکی	1 تا 2 روز
7. تست و رفع باگ	بررسی عملکرد، تست امنیت، تجربه کاربری	3 تا 5 روز
8. استقرار روی سرور	Deploy روی هاست یا پلتفرم‌هایی مثل Vercel، Heroku یا VPS	1 تا 2 روز

جمع کل : 28 تا 50 روز کاری

# UML



## نقش‌ها / بازیگران : Actors

### **کاربر :**

نقش اصلی برای استفاده از امکانات فروش بلیط سینما مثل ثبت نام، ورود، رزرو بلیط و... تعامل با سیستم از سمت کاربران عادی.

### **مدیریت سیستم :**

مسئول مدیریت رزروها، کاربران و محتوا (فیلم‌ها و سانس‌ها).  
دسترسی به پنل ادمین دارد.

### **بانک (Bank) :**

واسطه برای پرداخت آنلاین بلیط‌ها.  
دخلات در بخش تأیید و انتقال وجه از طریق "مدیریت خدمات بانکی".

### **Use Case ها موارد کاربردی :**

#### **برای کاربر :**

**ثبت نام**  
ایجاد حساب کاربری برای ورود اولیه.

#### **ورود**

ورود به سیستم با احراز هویت.

**فراموشی رمز / احراز هویت**  
سناریوهای امنیتی در فرآیند ورود.

**قوانین سیستم فروش**  
دیدن قوانین و مقررات خرید بلیط.

**پنل کاربری**  
داشبورد کاربر برای مدیریت کارها.

## **رزو بليط سينما**

انتخاب فilm، سانس، صندلی و رزو.

## **لغو بليط سينما**

امکان لغو بليط های رزو شده در پنل (سفارش های من) و نمایش زمان و نام بليط film .

## **جستجو جزئيات film (مثلاً ژانر، سانس،...)**

امکان جستجو در بین فilmها و سانسها.

## **بررسی شرایط انتقال / نمایش اطلاعات پرداخت**

نمایش پیش نیاز ها یا داده های مالی مرتبط با رزو.

## **صفحه پرداخت**

ورود به مرحله پرداخت هزینه بليط.

## **مدیریت خدمات بانکی**

بخشی که بین سیستم و بانک برای پرداخت ارتباط برقرار می کند.

## **خروج**

امکان خروج کاربر از پنل سیستم فروش بليط سينما .

## **برای مدیر سیستم :**

### **ورود → پنل ادمین**

ورود به سیستم و ورود به بخش مدیریتی.

## **مدیریت کاربران**

افزودن، حذف یا تغییر اطلاعات کاربران.

## **مدیریت رزو فilmها**

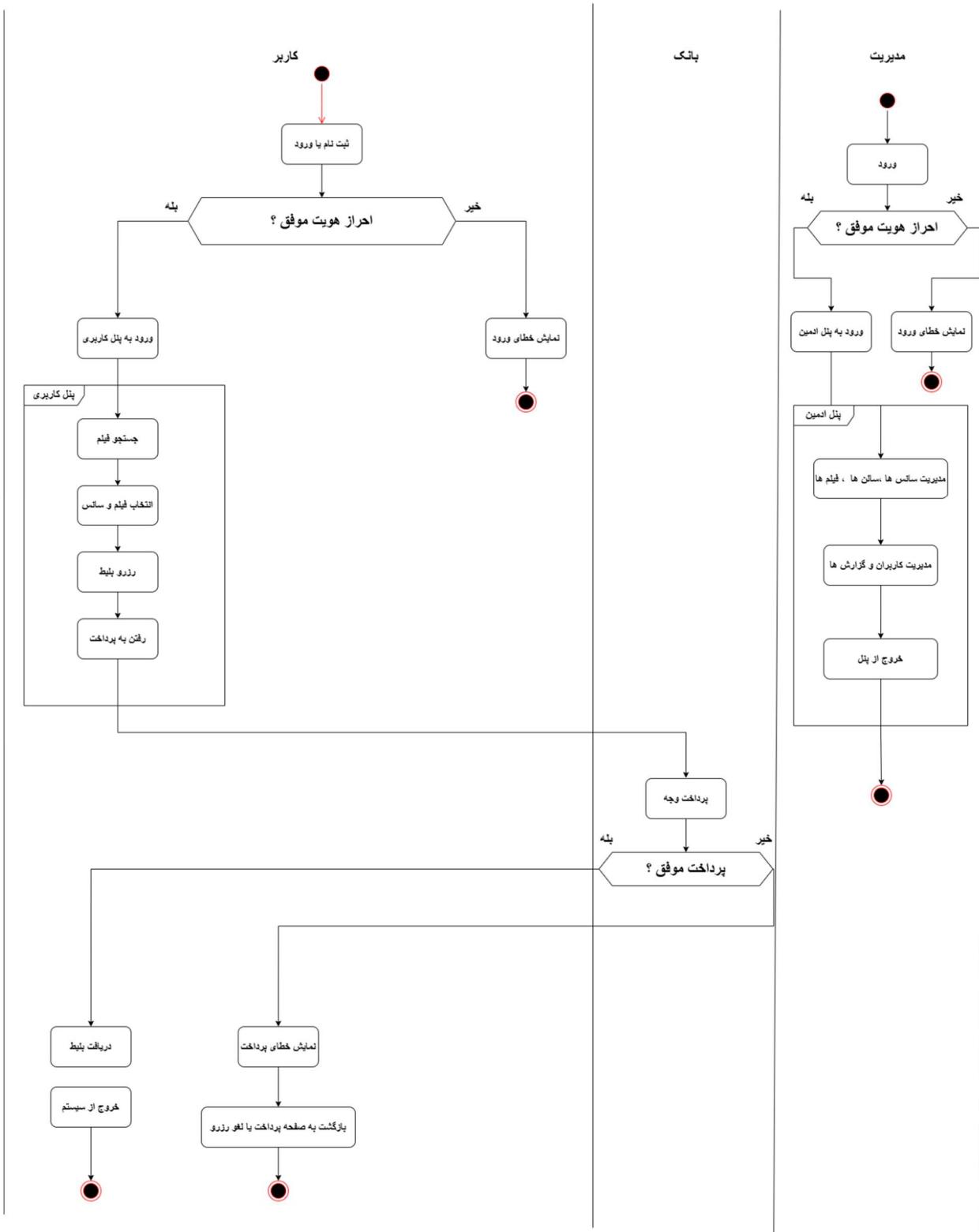
مدیریت سانس ها، filmها و رزو ها.

## **برای بانک :**

### **مدیریت خدمات بانکی / پرداخت**

انتقال وجه، تأیید تراکنش، گزارش پرداخت.

# Activity Diagram



### **ثبت نام : Use Case**

1. کاربر وارد صفحه ثبت‌نام می‌شود.
2. فرم اطلاعات شامل نام، ایمیل، شماره موبایل، رمز عبور را پر می‌کند.
3. سیستم صحت اطلاعات را بررسی می‌کند.
4. اگر اطلاعات ناقص یا تکراری باشد، پیام خطای نمایش داده می‌شود.
5. اگر اطلاعات معتبر باشد، حساب کاربری ایجاد می‌شود.
6. کاربر پیام موفقیت دریافت می‌کند و به صفحه ورود هدایت می‌شود.

### **ورود : Use Case**

1. کاربر نام کاربری و رمز را وارد می‌کند.
2. سیستم بررسی می‌کند اطلاعات با دیتابیس مطابقت دارد یا نه.
3. اگر درست باشد → وارد سیستم می‌شود.
4. اگر نادرست باشد → پیام خطای نمایش داده می‌شود.
5. در صورت سه بار خطای متوالی → تأخیر یا کپچا اعمال می‌شود.

### **مشاهده لیست فیلم ها : Use Case**

1. پس از ورود، کاربر وارد پنل کاربری می‌شود.
2. سیستم لیست فیلم‌های فعال را با اطلاعات سانس، سالن و ظرفیت نمایش می‌دهد.
3. کاربر می‌تواند لیست را براساس دسته‌بندی فیلتر کند.

### **جستجوی فیلم : Use Case**

1. کاربر در باکس جستجو نام فیلم یا ژانر را وارد می‌کند.
2. سیستم نتایج مرتبط را نمایش می‌دهد.
3. در صورت عدم تطابق → پیام "موردی یافت نشد" نمایش داده می‌شود.

### **بررسی جزئیات فیلم : Use Case**

1. کاربر روی فیلم موردنظر کلیک می‌کند.
2. سیستم اطلاعات کامل فیلم (خلاصه، بازیگران، سالن، سانس‌ها) را نمایش می‌دهد.
3. کاربر گزینه رزرو را مشاهده می‌کند.

### **رزرو بلیت سینما : Use Case**

1. کاربر سانس و سالن را انتخاب می‌کند.
2. نقشه صندلی‌ها باز می‌شود.
3. کاربر یک یا چند صندلی را انتخاب می‌کند.
4. سیستم در لحظه بررسی می‌کند که صندلی‌ها در دسترس‌اند یا نه.
5. اگر در دسترس نباشد → پیام خطا.
6. اگر در دسترس باشد → به مرحله پرداخت منتقل می‌شود.

### **صفحه پرداخت : Use Case**

1. کاربر اطلاعات بانکی) شماره کارت، CVV2، رمز دوم، تاریخ انقضا (را وارد می‌کند.
2. سیستم اطلاعات را به بانک ارسال می‌کند.
3. بانک صحت اطلاعات را بررسی می‌کند.
4. در صورت موفقیت → پیام پرداخت موفق و صدور بلیت.
5. در صورت شکست → بازگشت به صفحه پرداخت با پیام خطا.

### **لغو بلیط : Use Case**

1. کاربر وارد پنل کاربری و بخش "رزروهای من" می‌شود.
2. بلیت قابل لغو را انتخاب می‌کند.
3. روی گزینه لغو کلیک می‌کند.
4. سیستم بررسی می‌کند که زمان لغو هنوز باقی‌ست یا نه.
5. اگر مجاز به لغو باشد → بلیت حذف و مبلغ برگشت داده می‌شود.

6. اگر نه → پیام "لغو ممکن نیست."

#### **Use Case مدیریت کاربران (ویژه ادمین) :**

1. مدیر وارد پنل مدیریت می‌شود.
2. لیست کاربران فعال نمایش داده می‌شود.
3. امکان فیلتر، فعال/غیرفعال کردن یا حذف کاربران وجود دارد.

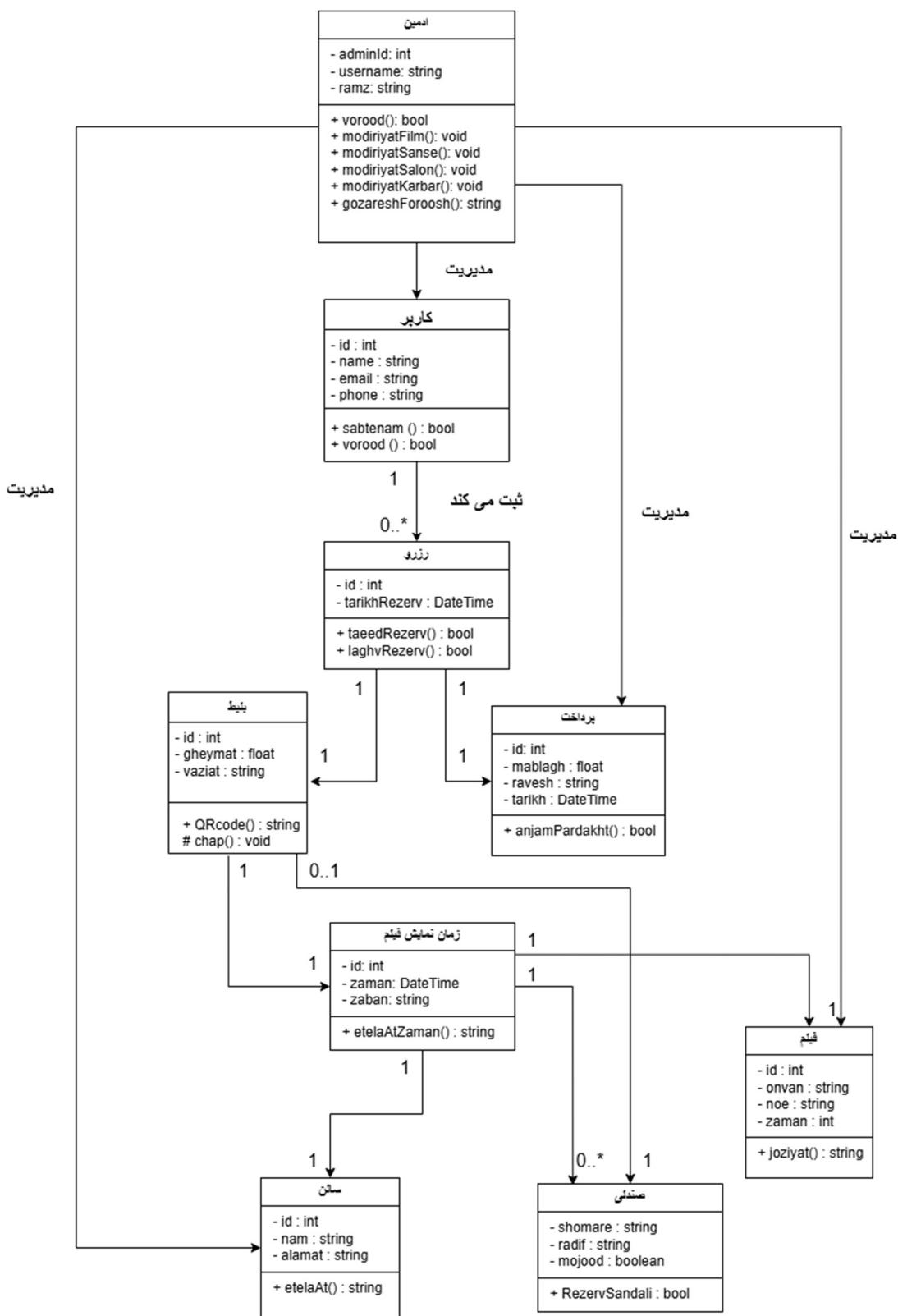
#### **Use Case مدیریت رزرو فیلم‌ها (ویژه ادمین) :**

1. مدیر به بخش مدیریت فیلم می‌رود.
2. می‌تواند فیلم جدید اضافه کند، فیلم قبلی را حذف یا ویرایش کند.
3. می‌تواند سانس‌ها را مدیریت کرده و ظرفیت تنظیم کند.

#### **Use Case مدیریت خدمات بانکی :**

1. سیستم پس از موفقیت پرداخت، جزئیات را ثبت می‌کند.
2. در صورت لغو بلیت → درخواست برگشت وجه به بانک ارسال می‌شود.
3. سیستم نتیجه عملیات بانکی را ذخیره می‌کند و به کاربر اعلام می‌کند.

# Class Diagram



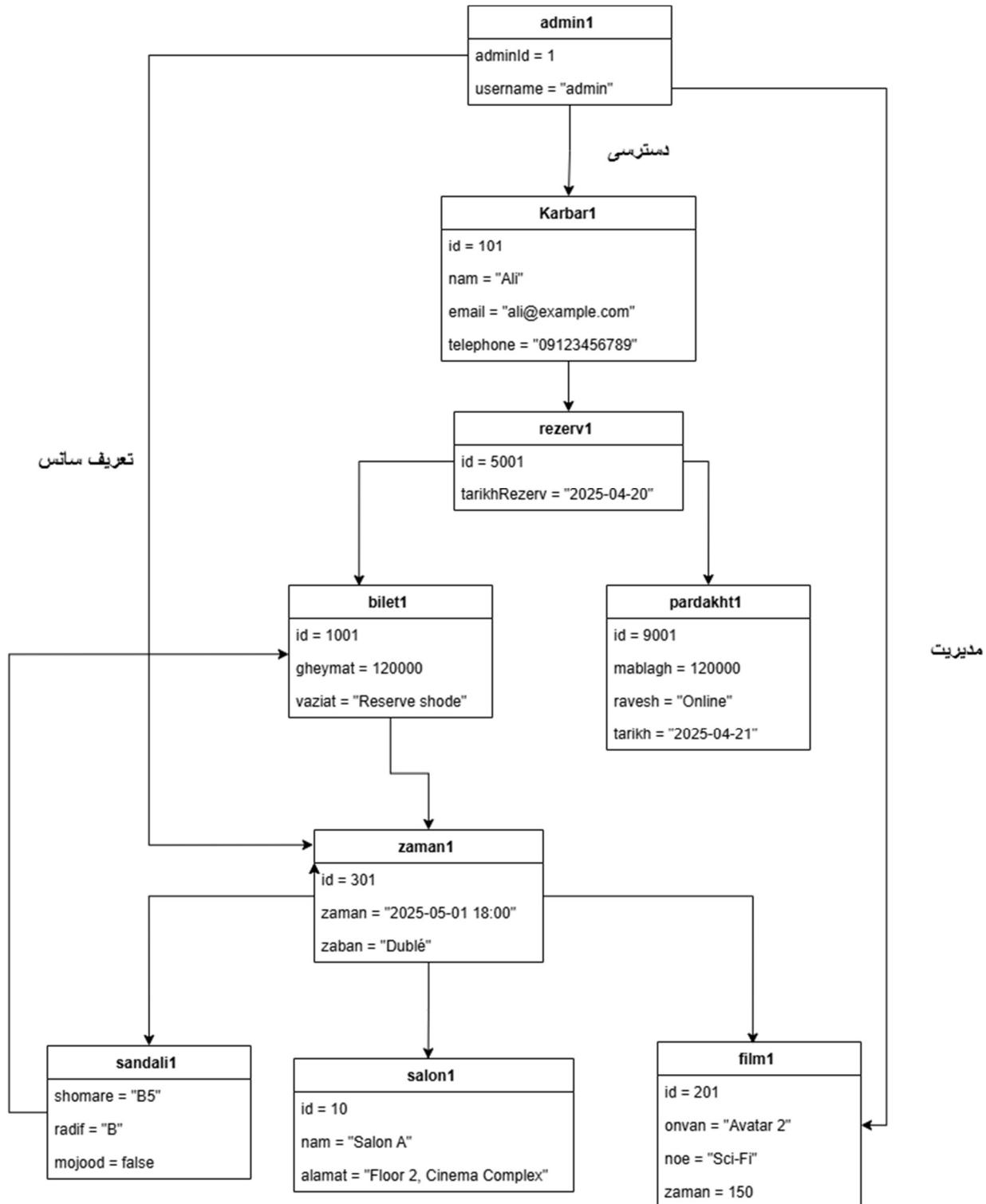
## توضیحات

سیستم فروش بلیت سینما ساختار ایستا و منطقی اجزای اصلی سیستم را نمایش می‌دهد و شامل کلاس‌هایی است که موجودیت‌ها و عملکردهای سیستم را مدل می‌کنند. کلاس Karbar نمایانگر کاربران عادی سیستم است که دارای ویژگی‌هایی مانند شناسه، نام، ایمیل و شماره تماس است و متدهایی برای ثبت‌نام و ورود دارد. کلاس Admin نیز نماینده مدیر سیستم است که می‌تواند با استفاده از متدهایی مانند modiriyatFilm، modiriyatSanse، gozareshForoosh و Film‌ها و سانس‌ها را مدیریت کرده و گزارش تهیه کند.

کلاس Film شامل اطلاعاتی درباره فیلم‌ها مانند عنوان، نوع و زمان نمایش است. کلاس Salon اطلاعات سالن‌های سینما را شامل می‌شود و کلاس ZamanNamayesh زمانبندی نمایش فیلم‌ها را با ویژگی‌هایی مثل زمان و زبان در بر دارد. کلاس Sandali برای مدیریت صندلی‌ها تعریف شده و امکان رزرو صندلی را فراهم می‌سازد. کلاس Bilet نمایانگر بلیت‌های صادر شده است که شامل قیمت، وضعیت و متدهایی برای چاپ و تولید QR کد است. کلاس Rezerv فرایند رزرو بلیت را مدل می‌کند و به بلیت، صندلی و پرداخت متصل است. کلاس Pardakht عملیات مالی را با ویژگی‌هایی مانند مبلغ، روش پرداخت و تاریخ پرداخت مدیریت می‌کند.

روابط بین کلاس‌ها نیز مشخص شده‌اند، به عنوان مثال هر Karbar می‌تواند چندین Rezerv داشته باشد، هر Rezerv به یک Bilet مربوط است، هر Bilet به یک ZamanNamayesh وابسته است، که آن هم به یک Film و Salon مربوط می‌شود. همچنین Admin دسترسی مدیریت به کلاس‌های مختلف دارد. این دیاگرام کلیه اجزای ساختاری سیستم را همراه با ویژگی‌ها، متدها و ارتباطات بین آن‌ها به صورت دقیق نمایش می‌دهد و پایه‌گذار پیاده‌سازی سیستم است.

# Object Diagram

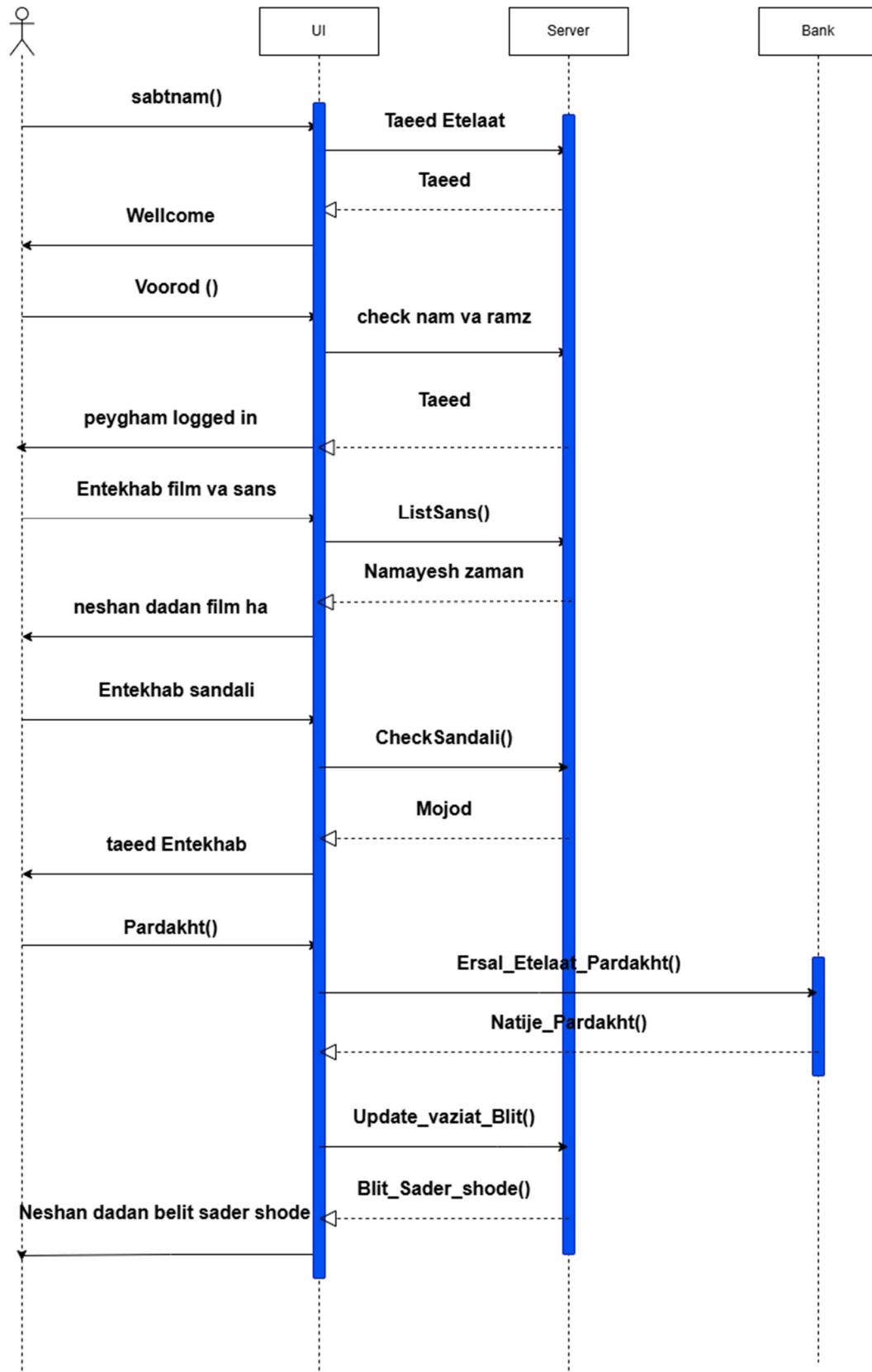


## توضیحات

در زمان اجرا را نمایش می‌دهد و روابط بین آن‌ها را نشان می‌دهد. در این دیاگرام، یک کاربر به نام karbar1 تعریف شده که یک رزرو (rezerv1) انجام داده است. این رزرو شامل یک بلیت (bilet1) است که برای یک زمان نمایش مشخص (zaman1) مربوط به یک فیلم (film1) و سالن (salon1) می‌باشد. صندلی مورد نظر (sandali1) در این سانس رزرو شده و وضعیت آن در بلیت مشخص است. همچنین پرداخت این رزرو توسط آبجکت pardakht1 انجام شده است. از طرفی، نقش مدیریت نیز با آبجکت admin1 نشان داده شده که به فیلم، سانس و اطلاعات کاربر دسترسی مدیریتی دارد. این دیاگرام نمایی ایستا از ارتباط اشیاء واقعی سیستم را در یک سناریوی مشخص، مانند خرید و صدور بلیت، به نمایش می‌گذارد.

# Sequence

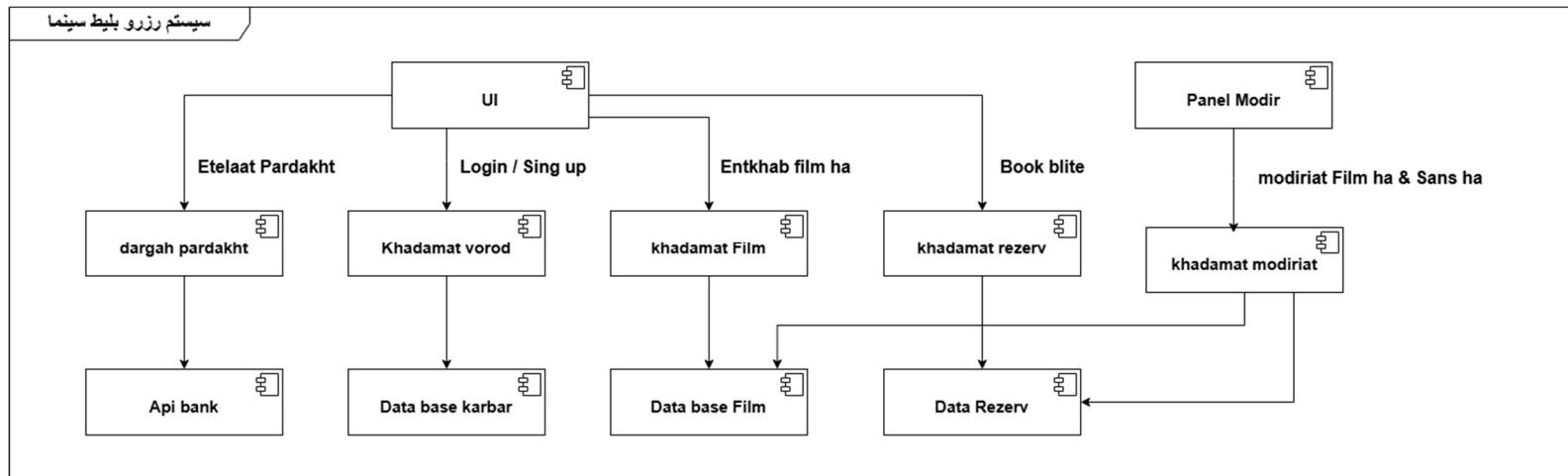
Karbar



## توضیحات

ابتدا کاربر عملیات ثبت نام را آغاز می‌کند و اطلاعات وارد شده برای بررسی به سرور ارسال می‌شود؛ در صورت تأیید، پیام خوش‌آمدگویی دریافت کرده و وارد سیستم می‌شود. پس از آن، کاربر فیلم و سانس مورد نظر خود را انتخاب می‌کند و رابط کاربری از سرور لیست سانس‌ها را دریافت و به کاربر نمایش می‌دهد. سپس کاربر صندلی را انتخاب کرده و موجود بودن آن توسط سرور بررسی می‌شود. در مرحله بعدی، اطلاعات پرداخت به بانک ارسال می‌شود و بانک نتیجه پرداخت را به سیستم بازمی‌گرداند. اگر پرداخت موفق باشد، سرور وضعیت بلیت را به «صادر شده» تغییر می‌دهد و بلیت نهایی به کاربر نمایش داده می‌شود. این دیاگرام توالی دقیق پیام‌ها و تعامل بین اجزای سیستم را در یک سناریوی کامل خرید بلیت به تصویر می‌کشد.

# Component Diagram



## توضیحات

دیاگرام کامپوننت (Component Diagram) ساختار کلی سیستم فروش آنلاین بلیت سینما را با نمایش اجزای اصلی (کامپوننت‌ها) و نحوه ارتباط بین آن‌ها مدل می‌کند. این دیاگرام نشان می‌دهد که چگونه مأژول‌های مختلف سیستم با هم تعامل دارند و کدام سرویس‌ها وظیفه‌ی ارائه‌ی خدمات به کاربر و مدیر سیستم را بر عهده دارند.

کامپوننت RabetKarbari مستقیماً با چهار سرویس اصلی در ارتباط است:

برای انجام عملیات ثبت‌نام و ورود کاربران KhadamateVorood

برای مرور و مشاهده فیلم‌ها و سانس‌های موجود KhadamateFilm

برای انجام عملیات رزرو و ثبت صندلی‌ها KhadamateRezerv

برای ارسال اطلاعات پرداخت به سیستم بانکی DargahPardakht

مدیر سیستم از طریق KhadamateModiriat دسترسی دارد تا فیلم‌ها، سانس‌ها و کاربران را مدیریت کند.

هر سرویس به پایگاه داده مخصوص خود متصل است:

DatabaseKarbar به KhadamateVorood

DatabaseFilm به KhadamateFilm

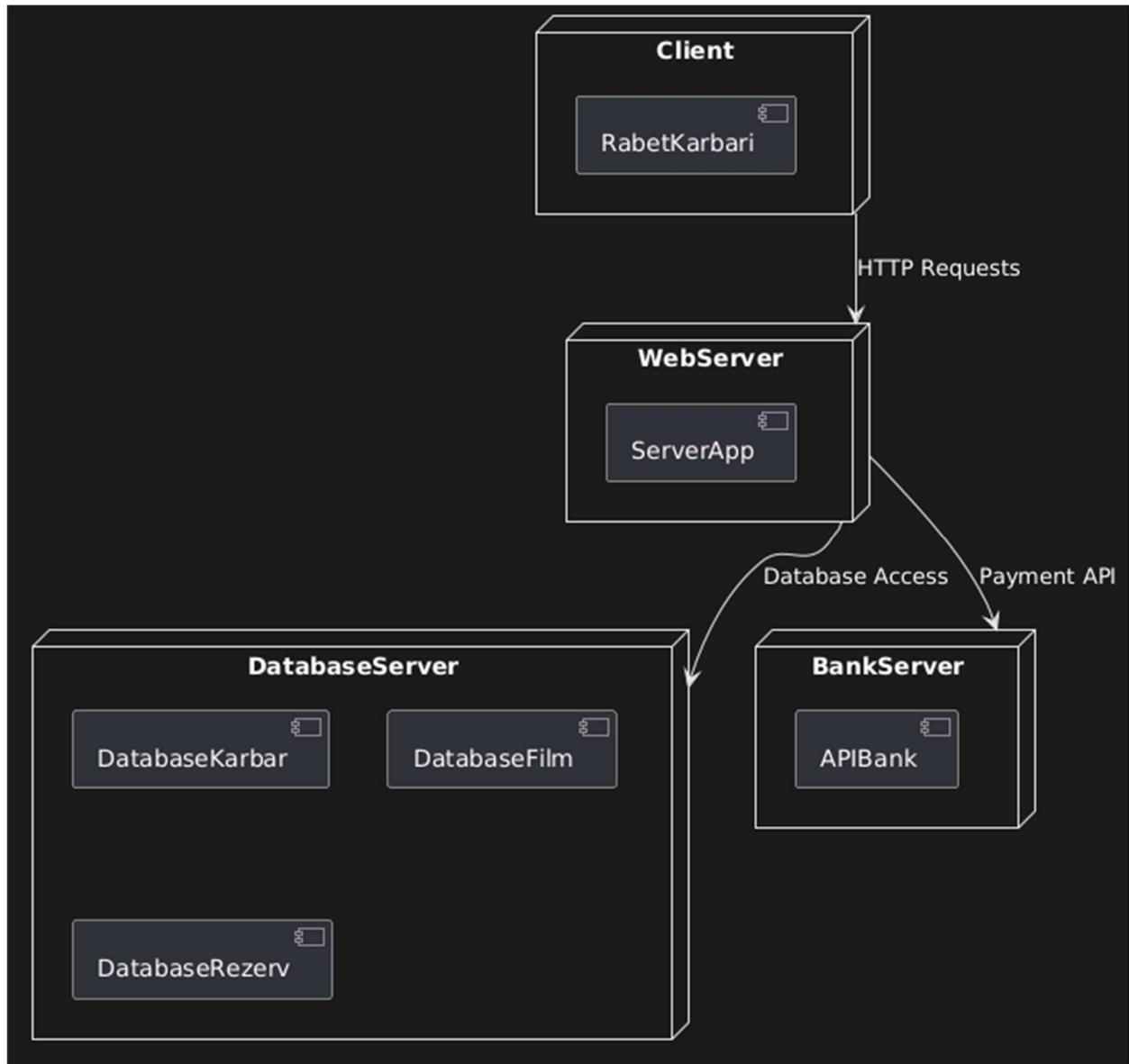
DatabaseRezerv به KhadamateRezerv

APIBank برای انجام عملیات پرداخت به DargahPardakht

همچنین در DatabaseRezerv و DatabaseFilm نیز با KhadamateModiriat ارتباط است تا اطلاعات مدیریت شده را ذخیره یا به روزرسانی کند.

این ساختار باعث می‌شود هر بخش از سیستم مسئولیت مشخصی داشته باشد و مازولار بودن سیستم باعث افزایش امنیت، کارایی و نگهداری ساده‌تر شود.

# Deployment Diagram



## توضیحات

Deployment Diagram معماری فیزیکی سیستم فروش آنلاین بلیت سینما را نمایش می‌دهد و چگونگی توزیع اجزای نرمافزاری روی سخت‌افزارها و سرورها را نشان می‌دهد.

در این سیستم، چهار گرهی اصلی (Node) در نظر گرفته شده است:

**کاربر**: این گره شامل کامپوننت RabetKarbari است که رابط گرافیکی کاربر روی مرورگر یا اپلیکیشن موبایل اجرا می‌شود.

**سرور وب**: این گره کامپوننت ServerApp را میزبانی می‌کند که وظیفه‌ی مدیریت منطق تجاری سیستم، پردازش درخواست‌ها، احراز هویت، رزرو و ارتباط با دیتابیس و درگاه بانکی را بر عهده دارد.

**سرور پایگاه داده**: شامل سه پایگاه داده‌ی مجزا برای مدیریت اطلاعات کاربران (DatabaseFilm)، فیلم‌ها (DatabaseKarbar) و رزروها (DatabaseRezerv) می‌باشد.

**سرور بانکی**: میزبانی کننده‌ی APIBank است که ارتباط مستقیم با درگاه‌های پرداخت بانکی برای انجام عملیات پرداخت کاربران برقرار می‌کند.

ارتباطات بین گره‌ها به این صورت است:

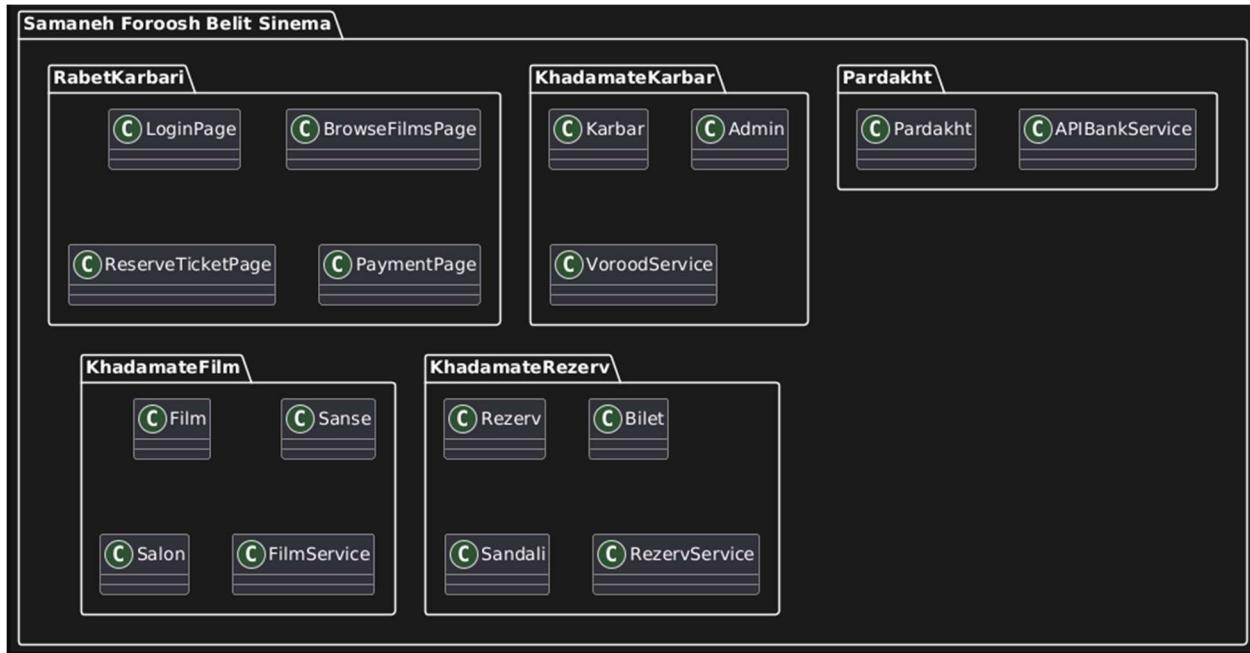
کاربر از طریق پروتکل HTTP با WebServer ارتباط برقرار می‌کند.

WebServer برای ذخیره‌سازی و بازیابی اطلاعات به DatabaseServer متصل می‌شود.

در مرحله‌ی پرداخت، BankServer از طریق API به WebServer متصل شده و نتیجه‌ی پرداخت را دریافت می‌کند.

این معماری توزیع شده به سیستم اجازه می‌دهد که به صورت امن، مقیاسپذیر و پایدار به کاربران خدمات ارائه دهد و فرآیند رزرو و خرید بلیت را به صورت کارآمد مدیریت کند.

## Package Diagram



دیاگرام پکیج ساختار مژولار سیستم فروش بلیت سینما را نمایش می‌دهد و نشان می‌دهد که کلاس‌ها و اجزای مرتبط در چه گروه‌هایی سازماندهی شده‌اند. این نوع دیاگرام به درک بهتر ساختار پروژه و نگهداری آسان‌تر کد کمک می‌کند، به خصوص در پروژه‌های بزرگ و چندبخشی.

این پکیج شامل کلاس‌هایی مانند **LoginPage**، **RabetKarbari** است که صفحات **PaymentPage**، **ReserveTicketPage**، **BrowseFilmsPage** مختلف رابط گرافیکی کاربر را نمایش می‌دهند.

: شامل کلاس‌های Admin و Karbar است که کاربران و مدیران سیستم را مدل می‌کنند، به همراه VoroodService برای مدیریت ورود و ثبت‌نام.

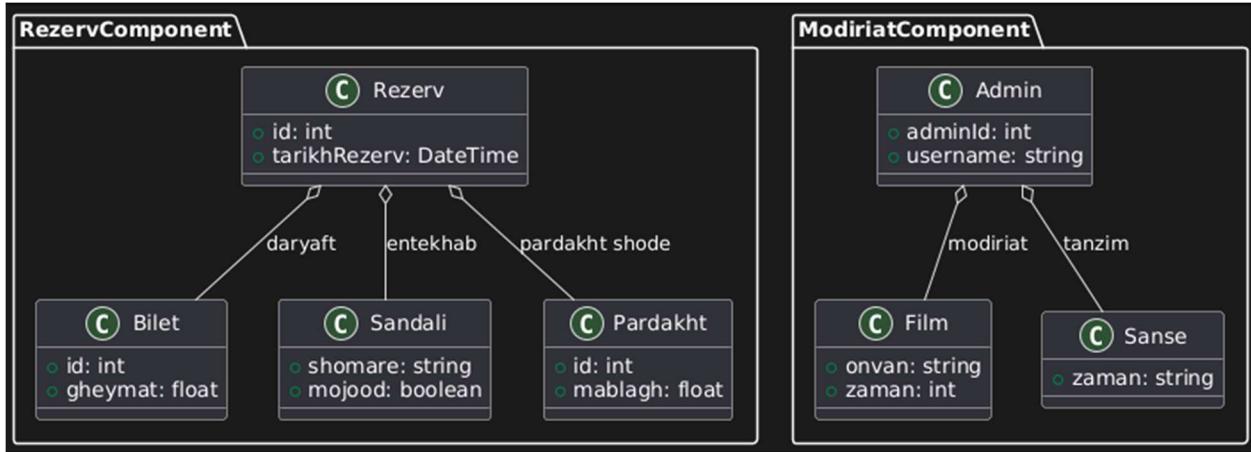
: مربوط به اطلاعات فیلم‌ها، سانس‌ها و سالن‌هاست، با کلاس‌هایی مانند KhadamateFilm . FilmService و سرویس مرتبط Salon ، Sanse ، Film

: این پکیج مسئول مدیریت فرآیند رزرو بلیت است و شامل کلاس‌های RezervService و Sandali ، Bilet ، Rezerv می‌باشد.

: در این پکیج عملیات مالی انجام می‌شود که شامل کلاس Pardakht و کلاس ارتباطی با سیستم بانکی APIBankService است.

این ساختار مازوکلار باعث جداسازی وظایف، افزایش خوانایی، توسعه‌پذیری، و تست‌پذیری بهتر در پروژه می‌شود.

# Composite Structure Diagram



دیاگرام ساختار ترکیبی برای سامانه فروش بلیت سینما، نمایی از ساختار داخلی اجزای پیچیده سیستم و روابط بین زیرسیستم‌ها را ارائه می‌دهد. این دیاگرام نشان می‌دهد که یک کامپوننت چگونه از کلاس‌ها و مؤلفه‌های داخلی تشکیل شده و چگونه این اجزا با یکدیگر در تعامل هستند.

## RezervComponent

نمایانگر ساختار داخلی فرایند رزرو بلیت است و شامل کلاس‌های زیر می‌باشد:

Rezerv : کلاس اصلی که رزرو بلیت را مدل می‌کند.

Bilet : بلیت رزرو شده با اطلاعات قیمت و شناسه.

Sandali : صندلی انتخاب شده برای آن رزرو.

Pardakht : اطلاعات پرداخت مربوط به آن رزرو.

ارتباطات از نوع ترکیبی (Composition) هستند، به این معنا که رزرو بدون بلیت، صندلی و پرداخت بی معناست و این مؤلفه‌ها وابسته به موجودیت رزرو هستند.

### ModiriatComponent

نمایانگر بخش مدیریتی سیستم است و شامل کلاس‌های زیر می‌باشد:

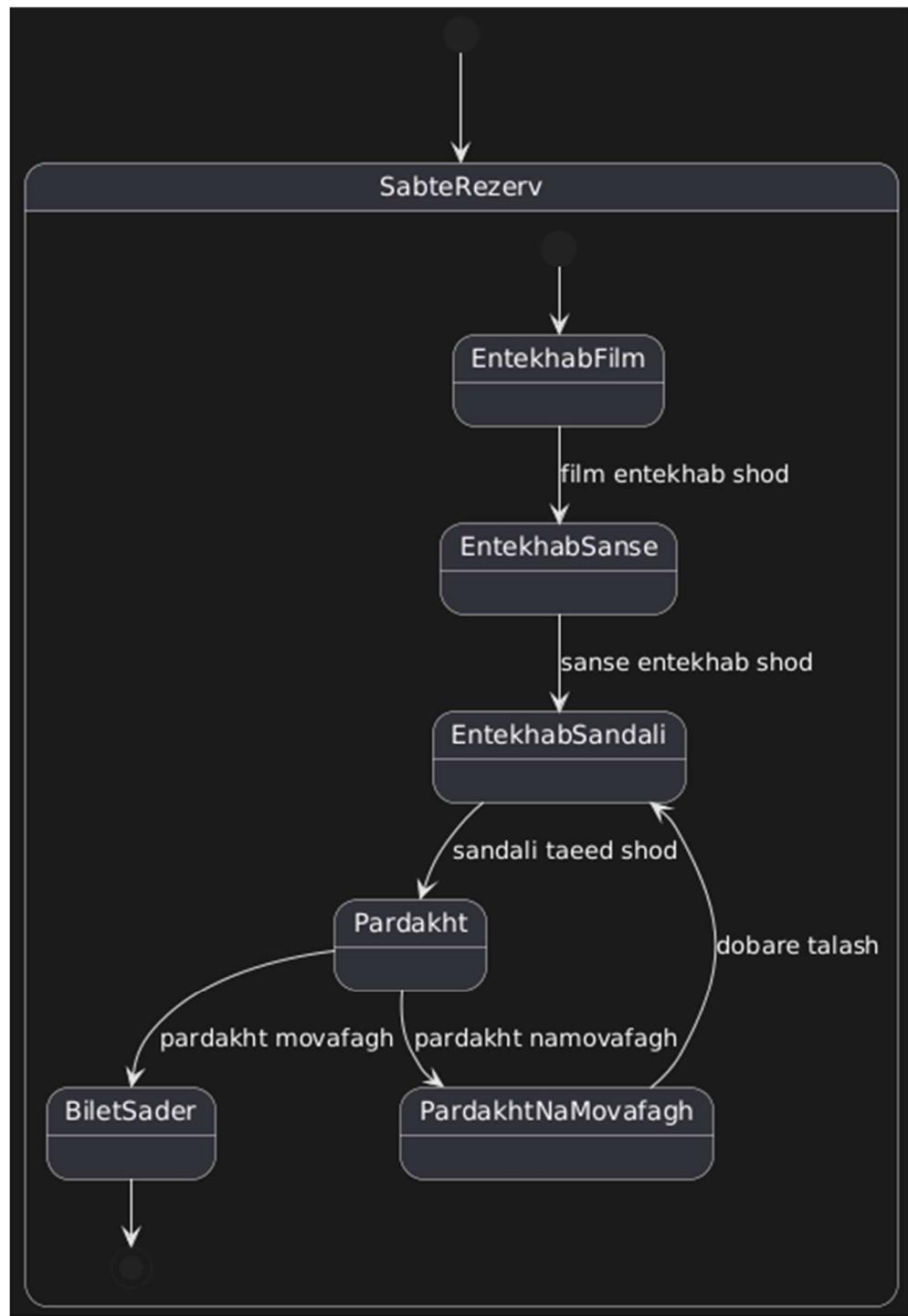
Admin : مدیر سیستم که مجوز دسترسی به اطلاعات فیلم‌ها و سانس‌ها دارد.

Film : نماینده فیلم‌های در حال اکران.

Sanse : زمان‌بندی‌های مربوط به نمایش فیلم‌ها.

در این بخش، کلاس Admin به صورت ترکیبی با Film و Sanse در ارتباط است و وظیفه تنظیم و مدیریت آن‌ها را بر عهده دارد.

## State Machine Diagram



دیاگرام ماشین حالات یکی از دیاگرام‌های رفتاری در UML است که تغییر وضعیت یک موجودیت در طول چرخه حیاتش را بر اساس رویدادها نمایش می‌دهد. در این دیاگرام، روند کامل رزرو بليت توسط کاربر مدل‌سازی شده است.

شروع فرآيند :

کاربر وارد روند رزرو می‌شود.

انتخاب فيلم:

کاربر یکی از فيلم هاي موجود را انتخاب می‌کند.

انتخاب سانس:

پس از انتخاب فيلم، سانس مناسب برای نمایش انتخاب می‌شود.

انتخاب صندلی:

صندلی دلخواه توسط کاربر مشخص و تأیيد می‌شود.

پرداخت:

سيستم اطلاعات پرداخت را دریافت کرده و با بانک در ارتباط قرار می‌گيرد.

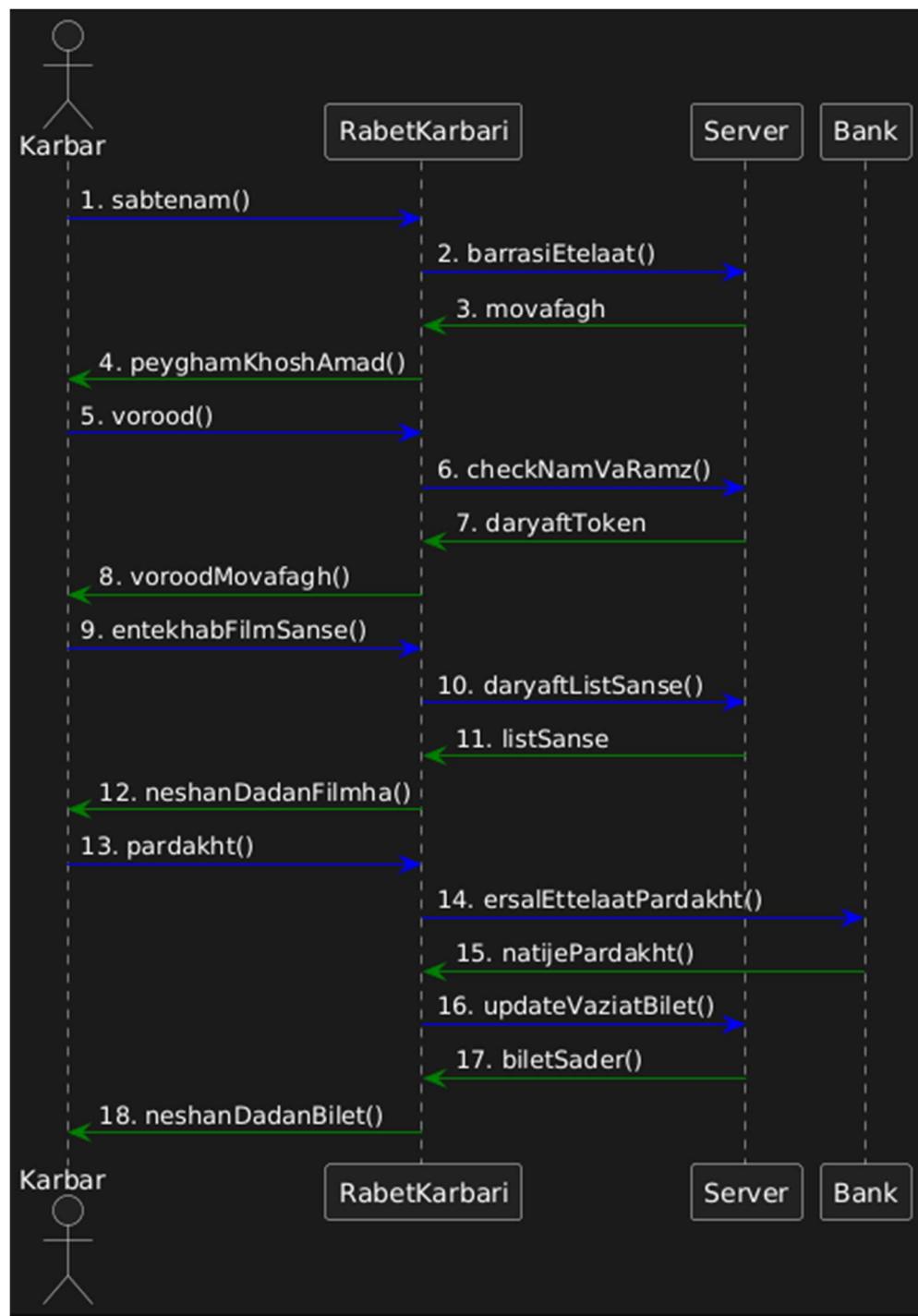
بليت صادر شده:

اگر پرداخت موفقیت‌آمیز باشد، بليت صادر می‌شود و فرآيند پایان می‌يابد.

پرداخت نا موفق:

اگر عملیات پرداخت با شکست مواجه شود، کاربر دوباره به مرحله انتخاب صندلی بازگردانده می‌شود تا مجدد تلاش کند یا فرآيند را لغو کند.

# Communication Diagram



**Communication Diagram** (نمودار ارتباطات) یکی از دیاگرام‌های رفتاری در UML است که نحوه تبادل پیام‌ها بین اشیاء یا اجزای سیستم را در قالب یک سناریو مشخص نمایش می‌دهد. برخلاف **Sequence Diagram** که بر ترتیب زمانی تمرکز دارد، این دیاگرام تمرکز بیشتری روی ساختار و مسیر ارتباطات بین اجزا دارد.

این دیاگرام سناریوی کامل «رزرو و خرید بلیت توسط کاربر» را مدل کرده و شامل چهار عنصر اصلی است:

کاربر (Karbar): آغازگر فرآیند تعامل با سیستم

رابط گرافیکی (RabetKarbari): رابط بین کاربر و سرور

سرور مرکزی (Server): پردازش درخواست‌ها و مدیریت اطلاعات

بانک (Bank): انجام و تأیید عملیات پرداخت

کاربر ابتدا از طریق `(() sabtenam()` ثبت‌نام می‌کند و اطلاعات به سرور برای بررسی ارسال می‌شود.

پیام تأیید موفق از سرور باز می‌گردد و خوش‌آمدگویی به کاربر نمایش داده می‌شود.  
کاربر وارد سیستم می‌شود (`((vorood()`، و توکن دسترسی از سرور دریافت می‌کند.

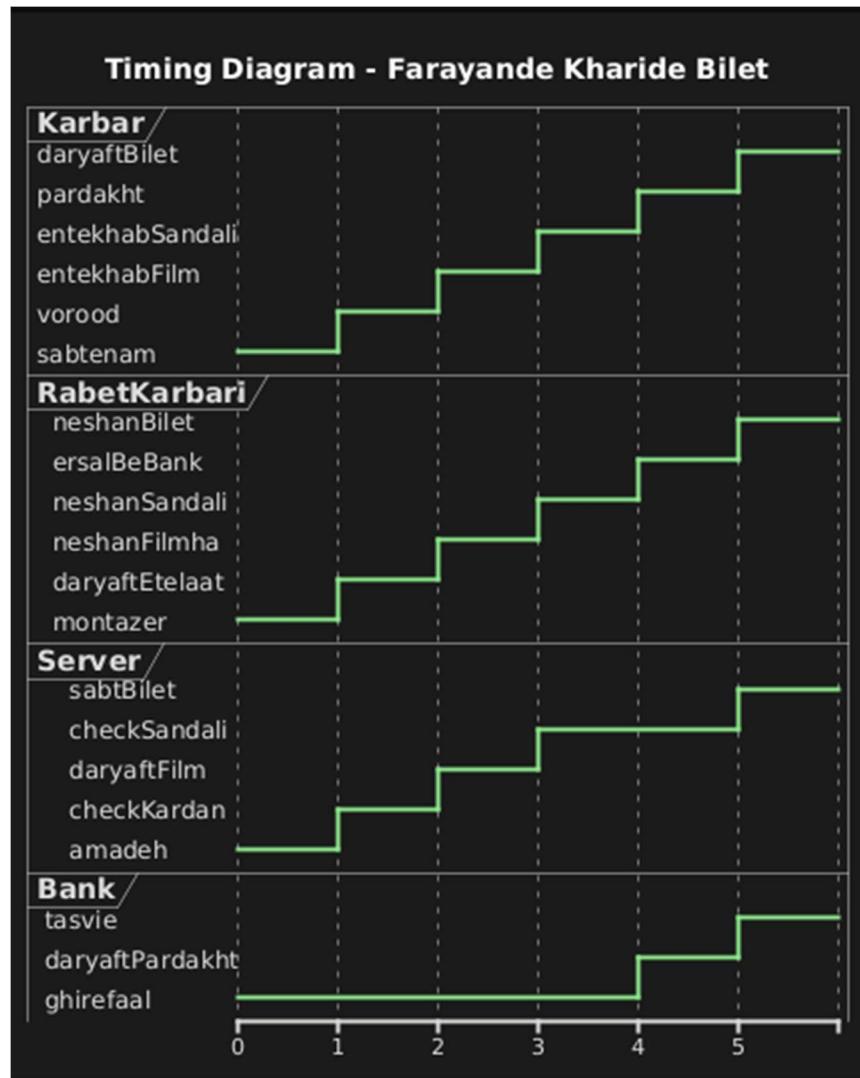
پس از ورود موفق، کاربر فیلم و سانس مورد نظر را انتخاب می‌کند  
. (`entekhabFilmSanse()`)

سیستم لیست سانس‌ها را از سرور دریافت و به کاربر نمایش می‌دهد.

در مرحله پرداخت، اطلاعات به بانک ارسال شده و نتیجه باز می‌گردد.

در نهایت، وضعیت بلیت به حالت «صادر شده» تغییر کرده و بلیت به کاربر نمایش داده می‌شود.

## Timing Diagram



Timing Diagram یا نمودار زمانبندی، یکی از دیاگرام‌های رفتاری در UML است که تغییر وضعیت اجزای مختلف سیستم را نسبت به زمان نمایش می‌دهد. این دیاگرام بسیار کاربردی است برای نمایش تعاملات و زمانبندی بین اشیاء در سناریوهایی که ترتیب و تأخیر زمانی اهمیت دارد.

در این دیاگرام، وضعیت چهار عنصر اصلی سیستم در طول زمان بررسی شده است:

1. کاربر : وضعیت‌های مختلف مانند **sabtenam**, **vorood**, **entekhabFilm**,

و **daryuftBilet** و **pardakht** را طی می‌کند.

2. رابط گرافیکی **RabetKarbari** : در هر مرحله مطابق با درخواست کاربر، وظایفی

مانند **montazer**, **daryaftEtelaat**, **neshanFilmha**, **ersalBeBank**,

و **neshanBilet** را انجام می‌دهد.

3. سرور: مسئول بررسی اطلاعات، دریافت داده‌ها، بررسی صندلی، و ثبت

بلیت می‌باشد.

4. بانک: در ابتداء غیرفعال است ولی در مرحله پرداخت فعال می‌شود

. (tasvie) و در نهایت وضعیت پرداخت را تسویه می‌کند (**daryaftPardakht**)