

پروژه اول: سیستم رزرو بلیط سینما

توجه: پروژه به صورت کامل و همراه با Command User Interface پیاده‌سازی شده است.

کارت‌های CRC با سناریوهای توضیح داده شده قبلی:

البته به نظر من بهتر است کلاس سیستم رزرو بلیط، به جای ارتباط مستقیم با کلاس Show، با کلاس سالن یا همان Theater در ارتباط باشد؛ کلاس Theater نیز مجموعه‌ای از نمایش‌هایی که در آن سالن به نمایش در می‌آیند را نگهداری می‌کند.

همچنین از آن جا که صندلی‌های موجود در هر سالن، معیار خوبی برای رزرو نیستند (چون ممکن است یک صندلی را اشخاص مختلفی برای نمایش‌های مختلفی رزرو کنند)، ردیف‌های صندلی را در کلاس Show نگهداری می‌کنیم. در حقیقت هر نمایشی، یک مجموعه ردیف خالی دارد که مشتری می‌تواند آن‌ها را خریداری کند.

ResevationSystem	<ul style="list-style-type: none">TheaterCollection
<ul style="list-style-type: none">توانایی یافتن نمایش‌ها بر اساس عنوان و روزنمایش جزئیات یک نمایشقبول رزرو صندلی از مشتری	

Show	<ul style="list-style-type: none">RowCollection
<ul style="list-style-type: none">ارائه جزئیات سالن و ظرفیتتوانایی رزرو صندلیتوانایی ذخیره سالن	

Theater	<ul style="list-style-type: none">ShowCollection
<ul style="list-style-type: none">قبول درخواست رزرو (با دیدن ردیف یا کل صندلی‌ها)ذخیره ردیف	

Row	<ul style="list-style-type: none"> • Seat • Collection
<ul style="list-style-type: none"> • ذخیره مجموعه‌ای از صندلی‌ها • دریافت درخواست رزرو صندلی • یافتن صندلی با شماره 	

Seat	
<ul style="list-style-type: none"> • دریافت رزرو • نگهداری وضعیت صندلی 	

سناریو جدید ۱-

مشتری می‌گوید شماره صندلی‌هایی که رزرو کرده است را فراموش کرده است. سیستم چطور رزرو دیروز او را نشان می‌دهد؟

یک کلاس مشتری ایجاد می‌کنیم؛ در کلاس مشتری باید کالکشنی از نمایش‌های رزرو شده و تعداد صندلی در هر نمایش نگهداری شود. کلاس سیستم رزرو بلیت سینما باید بتواند رزروهای هر مشتری را با گرفتن نام وی نشان دهد.

ResevationSystem	<ul style="list-style-type: none"> • Theater • Collection
<ul style="list-style-type: none"> • توانایی یافتن نمایش‌ها بر اساس عنوان و روز • نمایش جزئیات یک نمایش • قبول رزرو صندلی از مشتری • نمایش رزروهای هر مشتری با گرفتن نام وی 	

Customer	<ul style="list-style-type: none"> • Show
----------	--

• نگهداری صندلی نمایش‌های رزرو شده	• Collection
------------------------------------	--------------

سناریو جدید ۲-

مشتری می‌خواهد رزرو خود را لغو کند. او نام خود و اطلاعات نمایش را دارد ولی شماره صندلی‌های خود را فراموش کرده است!

کلاس سیستم رزرو بلیت سینما باید بتواند با گرفتن نام مشتری و اطلاعات نمایش، شماره صندلی‌های رزرو شده این مشتری را نشان دهد. همچنین باید بتواند صندلی رزرو شده را حذف کند.

ResevationSystem	<ul style="list-style-type: none"> Theater Collection
<ul style="list-style-type: none"> توانایی یافتن نمایش‌ها بر اساس عنوان و روز نمایش جزئیات یک نمایش قبول رزرو صندلی از مشتری نمایش رزروهای هر مشتری با گرفتن نام وی نشان دادن صندلی‌های رزرو شده با گرفتن نام مشتری و اطلاعات نمایش حذف رزرو با گرفتن شماره صندلی 	

سناریو جدید ۳-

یک فیلم جدید قرار است برای دو هفته هر روز در سه نوبت نمایش داده شود. اطلاعات نمایش باید در سیستم ثبت شود.

با گرفتن نمایش، آن را باید به لیست نمایش‌های سینما در کلاس سیستم رزرو بلیت سینما اضافه کنیم. علاوه بر این برای آن که بتوان روزهای هفته و سانس‌های مختلف در یک روز را نمایش داد، از دو enum با نام‌های Day و Period استفاده می‌کنیم.

ResevationSystem	<ul style="list-style-type: none"> • Theater • Collection
<ul style="list-style-type: none"> • توانایی یافتن نمایش‌ها بر اساس عنوان و روز • نمایش جزئیات یک نمایش • قبول رزرو صندلی از مشتری • نمایش رزروهای هر مشتری با گرفتن نام وی • نشان دادن صندلی‌های رزرو شده با گرفتن نام مشتری و اطلاعات نمایش • حذف رزرو با گرفتن شماره صندلی • گرفتن و اضافه کردن نمایش 	

Day enum	
<ul style="list-style-type: none"> • نگهداری روزهای هفته 	

Period enum	
<ul style="list-style-type: none"> • نگهداری سانس‌های یک روز 	

سناریو جدید ۴-

می‌خواهیم تعدادی صندلی به یکی از سالن‌ها اضافه کنیم.

کلاس Theater باید با گرفتن تعداد ردیف‌ها و اطلاعات سالن، تعداد ردیف‌ها را آپدیت کند و برای نمایش‌هایی که در این سالن وجود دارند، صندلی اضافه کند. همچنین باید در کلاس سیستم رزرو بلیت سینما، امکانی برای اضافه کردن صندلی به Theater فراهم کنیم.

Theater	<ul style="list-style-type: none"> Show Collection
<ul style="list-style-type: none"> قبول درخواست رزرو (با دیدن ردیف یا کل صندلی‌ها) ذخیره ردیف اضافه کردن تعدادی ردیف 	

ResevationSystem	<ul style="list-style-type: none"> Theater Collection
<ul style="list-style-type: none"> توانایی یافتن نمایش‌ها بر اساس عنوان و روز نمایش جزئیات یک نمایش قبول رزرو صندلی از مشتری نمایش رزروهای هر مشتری با گرفتن نام وی نشان دادن صندلی‌های رزرو شده با گرفتن نام مشتری و اطلاعات نمایش حذف رزرو با گرفتن شماره صندلی گرفتن و اضافه کردن نمایش امکان اضافه کردن صندلی به Theater 	

پروژه دوم: سیستم متمرکز سلامت

سناریو ۱-

می‌خواهیم سابقه پزشکی یک بیمار را در سیستم ثبت کنیم.

برای این کار باید یک کلاس HealthSystem در نظر بگیریم که امکان ثبت نام بیمار دارد. همچنین باید در کلاس Patient مجموعه‌ای از سوابق پزشکی وی را نگهداری کنیم. همچنین باید یک کلاس User بسازیم؛ یوزر می‌تواند بیمار، پزشک (پزشک اصلی یا داروخانه)، پژوهشگر دانشگاهی و .. باشد. به هر یک از این افراد یک نام کاربری و یک رمز اختصاص می‌یابد که با استفاده از آن می‌توانند وارد سیستم شوند. در کلاس Doctor نیز باید امکانی برای اضافه کردن رکورد ایجاد کنیم.

HealthSystem	<ul style="list-style-type: none"> User Collection
<ul style="list-style-type: none"> ثبت نام یوزر 	

Patient	<ul style="list-style-type: none"> Record Collection
<ul style="list-style-type: none"> اضافه کردن سابقه پزشکی به مجموعه سوابق پزشکی 	

Record	
<ul style="list-style-type: none"> نگهداری اطلاعات یک سابقه پزشکی 	

User	
<ul style="list-style-type: none"> نگهداری نام کاربری و رمز 	

Doctor	<ul style="list-style-type: none"> Patient
--------	---

• امکان اضافه کردن رکورد برای بیمار	• Collection
-------------------------------------	--------------

سناریو ۲-

بیمار نزد پزشک مورد اعتماد (اولین پزشک) می‌رود و پزشک باید بتواند با وارد کردن username بیمار در سیستم، اطلاعات تمامی رکوردهای پزشکی وی را مشاهده کند.

باید یک کلاس Doctor بسازیم. پزشک پس از ورود به سیستم، بعد از آن که کد یک بیمار را وارد می‌کند، در صورتی که اجازه دسترسی به اطلاعات این بیمار را داشته باشد، رکوردهای مربوطه به نمایش در می‌آید.

Doctor	<ul style="list-style-type: none"> • Patient • Collection
<ul style="list-style-type: none"> • امکان اضافه کردن رکورد برای بیمار • نگه داشتن اطلاعات پزشک • نگه داشتن کالکشنی از بیماران پزشک 	

سناریو ۳-

بیمار می‌خواهد اجازه دهد پژوهشگران دانشگاهی، بدون دسترسی به نام، به رکوردهای پزشکی وی دسترسی داشته باشند.

باید یک کلاس Researcher ایجاد کنیم. تمامی پژوهشگران باید بتوانند از یک کالکشن مشترک که رکوردهای پزشکی بیماران است که اجازه دسترسی به اطلاعات را صادر کرده‌اند، استفاده کنند. همچنین در کلاس HealthSystem باید امکانی برای انجام این کار اضافه کنیم.

Researcher	<ul style="list-style-type: none"> • Record
------------	--

<ul style="list-style-type: none"> • نگه داشتن کالشکنی از رکوردهای مجاز برای همه پژوهشگران به صورت مشترک 	<ul style="list-style-type: none"> • Collection
---	--

HealthSystem	<ul style="list-style-type: none"> • User • Collection
<ul style="list-style-type: none"> • ثبت نام یوزر • دسترسی دادن بیمار به پژوهشگران 	

سناریو ۴-

بیمار نزد پزشک مورد اعتماد خود می‌رود؛ پزشک دارویی برای آن می‌نویسد و آن رکورد خاص را با پزشک داروخانه به اشتراک می‌گذارد. همچنین برای پیگیری‌های جدی‌تر، تمامی رکوردهای بیمار که مربوط به مشکلات قلبی هستند را با پزشک متخصص به اشتراک می‌گذارد.

ما برای آن که بتوانیم دسترسی پزشکان به رکوردها را مدیریت کنیم، رکوردها را در کلاس Patient نگهداری نمی‌کنیم؛ به جای آن یک HashMap در کلاس Doctor تعریف می‌کنیم که هر بیمار را به رکوردهایش نزد آن پزشک، نگاشت می‌کند. پزشک باید بتواند رکوردهای خاص یا همه رکوردها را با کاربران دیگر به اشتراک بگذارد.

Doctor	<ul style="list-style-type: none"> • Patient • Collection
<ul style="list-style-type: none"> • امکان اضافه کردن رکورد برای بیمار • نگه داشتن اطلاعات پزشک • نگه داشتن کالکشنی از بیماران پزشک • امکان به اشتراک‌گذاری رکوردهای خاص یا همه رکوردهای یک بیمار با یک پزشک دیگر سیستم 	

علاوه بر این چندین enum با نام‌های زیر برای نگهداری مقادیر خاص می‌سازیم:

• Insurance

Degree •

DiseaseType •

Triage •

Region •