

تمرین جلسه شش کارگاه برنامه‌نویسی پیشرفته

سوال اول:

انجام دهید: تعدادی از سناریوها را انتخاب کرده و کارت‌های CRC مربوطه و نحوه تعامل آن‌ها را استخراج کنید:

سناریوی اول:

۱. یک مشتری پنج صندلی مجاور را می‌خواهد. چطور سیستم صندلی‌های مجاور را پیدا می‌کند و پیشنهاد می‌دهد؟

۱. کارمند سینما ابتدا باید بر اساس نام و روز (Date) نمایش (Show) جستجو انجام داده و تمامی نمایش‌های موجود را ببیند. بنابراین کلاس سیستم رزرو بلیت سینما (CinemaTicketReservation) لیستی از نمایش‌ها داشته و مسئولیتی دارد که در آن به عنوان ورودی نام نمایش و روز مورد نظر را دریافت کرده و به عنوان خروجی: نام نمایش، روز نمایش (تاریخ نمایش)، ساعت و سالن نمایش را نشان می‌دهد. بنابراین کلاس رزرو بلیت سینما با کلاس نمایش در ارتباط است. از طرف دیگر کلاس نمایش مسئولیتی با عنوان چاپ جزئیات نمایش شامل: نام نمایش، روز نمایش (تاریخ نمایش)، ساعت و سالن نمایش دارد و همچنین با کلاس‌های سالن (Hall) و تاریخ (Date) نیز در ارتباط است.

۲. پس از دریافت لیست جزئیات نمایش‌های موجود با نام مشخص و در روز مشخص، کارمند سینما باید جستجو کند که در کدام یک از این زمان‌ها پنج صندلی خالی در کنار هم در سالن نمایش موجود است. بنابراین کلاس نمایش باید قابلیت ذخیره یک سالن را داشته باشد. هر سالن از چند ردیف صندلی تشکیل شده است. بنابراین کلاس سالن وظیفه دارد با گرفتن یک ورودی `int` مشخص کند که این تعداد صندلی خالی در کنار هم در کدام یک از ردیف‌ها وجود دارد. همچنین شماره‌ی مربوط به پنج صندلی خالی در هر ردیف را نیز اعلام کند.

۳. فرض کنیم دو ساعت مانند ۴:۰۰ و ۶:۰۰ به مشتری اعلام می‌شود که پنج صندلی خالی در کنار هم دارند. مشتری ساعت ۶:۰۰ را انتخاب می‌کند. کارمند سینما اعلام می‌کند در نمایش ساعت ۶:۰۰ ردیف‌های ۲ و ۳ و ۷ هستند که پنج صندلی خالی کنار هم دارند. مشتری ردیف دوم را انتخاب می‌کند. حال این پنج صندلی باید به حالت رزرو شده در بیایند. بنابراین در کلاس رزرو بلیت سینما باید مسئولیتی با عنوان انجام رزرو مشتری باشد. همچنین در سالن باید مسئولیتی با عنوان دریافت شماره‌ی صندلی‌ها و درآوردن حالت آن صندلی‌ها به حالت رزرو وجود داشته باشد. همچنین در کلاس صندلی نیز باید امکان ست کردن حالت صندلی به حالت رزرو باشد.

۴. باید امکان ذخیره‌ی این اطلاعات در سیستم وجود داشته باشد: مشتری با نام مشخص برای نمایش مشخص چند صندلی برای خود رزرو کرده‌است. بنابراین کلاس نمایش لیستی از مشتری‌ها باید داشته باشد و با این کلاس در ارتباط است. و کلاس مشتری نیز شماره‌ی صندلی‌هایی که رزرو کرده را در خود داشته و در صورت لزوم آن را چاپ کند. بنابراین کلاس مشتری مسئولیتی با عنوان تعیین شماره‌ی صندلی رزرو شده و همچنین چاپ شماره و ردیف صندلی خواهد داشت. بنابراین کلاس مشتری با کلاس صندلی در ارتباط است و باید در کلاس صندلی مسئولیت چاپ جزئیات مربوط به صندلی را داشته باشیم.

سناریوی دوم:

مشتری می‌گوید شماره‌ی صندلی‌هایی که رزرو کرده‌است را فراموش کرده‌است. سیستم چطور رزرو دیروز او را نشان می‌دهد؟

۱. در سناریوی قبل گفتیم که کلاس رزرو بلیت سینما لیستی از نمایش‌ها دارد و هر نمایش نیز لیستی از مشتری‌ها. بنابراین در کلاس نمایش که دارای لیستی از مشتری‌هاست باید این امکان فراهم باشد که این لیست بر اساس نام کامل مشتری جستجو شود. به واسطه‌ی این مسئولیت باید در کلاس مشتری نیز مسئولیت بررسی نام مشتری انجام شود (equals method).

۲. حال فرض می‌کنیم نام مشتری در لیست مشتری‌های نمایش مورد نظر وجود دارد، بنابراین هم اطلاعات مربوط به نمایش و هم اطلاعات مربوط به مشتری و صندلی رزرو شده ب استفاده از متدهای موجود در آن‌ها چاپ خواهد شد.

۳. حال اگر سیستم تمام نمایش‌های موجود، حتی نمایش‌هایی که تمام شده‌اند، را در خود ذخیره کند، در صورت عدم بررسی تاریخ ممکن است سیستم دچار اشتباه شود. بنابراین باید در کلاس رزرو بلیط سینما باید مسئولیتی باشد تا یک نمایش را گرفته و با بررسی تاریخ آن اگر تاریخ نمایش مربوط به گذشته باشد آن را از لیست نمایش‌ها حذف کند.

سناریوی سوم:

مشتری می‌خواهد رزرو خود را لغو کند. او نام خود و اطلاعات نمایش را دارد ولی شماره‌ی صندلی‌های خود را فراموش کرده‌است.

۱. وقتی مشتری اطلاعات مربوط به نمایش را می‌داند پس کارمند سینما باید بتواند با وارد کردن اطلاعات مربوط به نمایش به آن نمایش دسترسی داشته باشد. بنابراین کلاس رزرو بلیت سینما مسئولیتی با عنوان جستجوی لیست نمایش‌ها بر اساس اطلاعات مربوط به نمایش دارد.

۲. بعد از دسترسی به نمایش مورد نظر باید بتوان نام مشتری را در بین لیست مشتریان این نمایش جستجو کرد. از سناریوهای قبل این مسئولیت را در کلاس نمایش قرار دادیم تا با جستجوی نام مشتری در بین مشتری‌های نمایش اطلاعات مربوط به نمایش، مشتری و صندلی‌های رزرو شده را چاپ کند. حال باید مسئولیتی در کلاس نمایش داشته باشیم تا با دریافت نام مشتری در صورتیکه مشتری در لیست آن نمایش باشد، مشتری را از لیست حذف کند. در ادامه سالن مربوط به نمایش باید بتواند صندلی‌های رزرو شده‌ی این مشتری را به حالت خالی و رزرو نشده تبدیل کند. کلاس رزرو بلیت سینما باید مسئول اعلام کردن اتمام کار لغو رزرو باشد.

سناریوی چهارم:

نمایش لغو شده‌است و سینما می‌خواهد به همه‌ی مشتریان خود اعلام کند.

۱. در ابتدا کارمند سینما باید امکان دسترسی به نمایش مورد نظر را با استفاده از اطلاعات مربوط به نمایش داشته باشد که در سناریوهای قبلی این مسئولیت را در کلاس رزرو بلیت سینما قرار دادیم. حال باید در کلاس نمایش مسئولیتی قرار داده شود تا برای تمام مشتریان خود اطلاعات مشتری شامل نام و شماره‌ی مشتری را چاپ کند. بنابراین همچنین باید کلاس مشتری مسئولیتی برای ثبت شماره‌ی مشتری باشد. همچنین کلاس شماره تلفن باید ایجاد شده و همچنین مسئولیت این را داشته باشد که در زمان ثبت شماره معتبر بودن آن را چک کند.

۲. در سناریوهای قبلی برای کلاس رزرو بلیت سینما مسئولیتی قرار دادیم تا یک نمایش را بر اساس اطلاعات آن و اگر تاریخ آن گذشته باشد، از لیست نمایش‌ها حذف کند. حال در این سناریو کارمند سینما باید بتواند این نمایش را که تاریخ آن نیز به روز است از لیست نمایش‌های سیستم حذف کند. پس مسئولیتی با عنوان لغو نمایش را نیز باید به لیست مسئولیت‌های این کلاس اضافه کنیم.

۳. برای لغو شدن نمایش باید سالن مربوط به نمایش نیز برای زمان نمایش لغو شده خالی اعلام شود. بنابراین باید در کلاس سالن مسئولیتی با عنوان اعلام خالی یا پر بودن سالن و همچنین مسئولیت تغییر حالت سالن از پر به خالی و برعکس را قرار دهیم.

Class name: CinemaTicketReservation	Collaborators: <ul style="list-style-type: none"> • Show • Collection • Customer
Responsibilities: <ul style="list-style-type: none"> • Find <u>list</u> of Shows by title and day • Do the reservation for Customer • Remove a Show if we are done with the Show • Search Shows list by Show's information (name, Hall, Date, time) • Cancel the reservation for Customer • Cancel the Show 	

Class name: Show	Collaborators: <ul style="list-style-type: none"> • Hall • Date • Customer
Responsibilities: <ul style="list-style-type: none"> • Print Show information • Determine Hall for Show • Search Customers list by Customer's name • Remove a Customer from list if exists • Print information of all Customers in the list 	

Class name: Hall	Collaborators: <ul style="list-style-type: none"> • Seat
Responsibilities: <ul style="list-style-type: none"> • List number of rows that have n empty seats next to each other • Get Seat number and set the Seat as reserved Seat 	

<ul style="list-style-type: none"> • Get Seat number and set the Seat as not-reserved • Declare Hall state. Is it reserved for a Show or not? • Set Hall state as reserved or not-reserved for a Show 	
--	--

Class name: Seat	Collaborators: <ul style="list-style-type: none"> • Hall • Customer
Responsibilities: <ul style="list-style-type: none"> • Set the Seat as reserved Seat • Print Seat number and row number • Set the Seat state as not-reserved 	

Class name: Customer	Collaborators: <ul style="list-style-type: none"> • Seat • PhoneNumber
Responsibilities: <ul style="list-style-type: none"> • Set Customer's reserved Seat number(s) • Print Customer and reserved Seats information • Override equals method • Set Customer's phone number 	

Class name: PhoneNumber	Collaborators:
Responsibilities: <ul style="list-style-type: none"> • Check PhoneNumber validity. 	

سوال دوم: طراحی و پیاده‌سازی سیستم متمرکز سلامت

سیستم متمرکز سلامت: (اسم‌ها را با highlight و فعل‌ها را با underline مشخص می‌کنیم).

وزارت بهداشت خود قصد دارد سیستم متمرکز سلامت را در کشور راه‌اندازی کند. هدف از ایجاد این سیستم، نگهداری سابقه پزشکی افراد به منظور سهولت در دسترسی به آن‌ها در نقاط مختلف جغرافیایی کشور است. علاوه بر آن وزارت بهداشت قصد دارد این اطلاعات را برای انجام تحقیقات در اختیار دانشگاه‌های تحت نظارت خود قرار دهد. علاوه بر آن بیماران نیز می‌توانند به سابقه پزشکی خود دسترسی داشته باشند و مجوز استفاده از اطلاعات پزشکی خود را بدون ذکر نام صادر نمایند. در کنار این سیستم، با توجه به

نوع بیماری و داروهای تجویز شده توسط پزشک اطلاعات در اختیار پزشک داروخانه نیز قرار می‌گیرد تا بیمار بدون نیاز به استفاده از دفترچه داروی خود را دریافت کند.

اطلاعات بیمار شامل تاریخ تولد، جنسیت، نوع بیمه پایه، نوع بیمه تکمیلی، مدرک تحصیلی، شغل، محل زندگی و موقعیت جغرافیایی است.

سابقه پزشکی شامل نوع کلی بیماری، شرح حال بیمار، تشخیص پزشک، سطح امنیتی بیماری، داروی تجویز شده، تاریخ، و پزشک معالج است. نوع کلی بیماری شامل خانواده‌های اصلی بیماری‌ها می‌باشد. مانند ارتوپدی، قلب، کلیه، زنان، گوارش و غیره. سطح امنیتی بیمار نشان‌دهنده اهمیت بیماری از لحاظ افشا اطلاعات است. مثلاً بیماری آیدز دارای سطح امنیتی بالا ولی سرماخوردگی در سطح امنیتی پایین است (فرض کنید سه سطح امنیتی بالا، متوسط و پایین وجود دارد).

سابقه پزشکی بیمار فقط توسط پزشک مورد اعتماد او قابل ایجاد است. اولین پزشک مورد اعتماد توسط بیمار انتخاب می‌شود. از آن جایی که بیماری‌های مختلفی وجود دارد، پزشک مورد اعتماد فقط می‌تواند برای درمان یک بیماری خاص، بیمار را به پزشک جدید ارجاع دهد. در این حالت پزشک جدید به لیست پزشکان مورد اعتماد فرد افزوده می‌شود و به سابقه پزشکی فرد دسترسی پیدا خواهد کرد. این دسترسی می‌تواند به دو صورت تعریف شود: دسترسی به کل سوابق پزشکی و یا تنها دسترسی به سوابق مربوط به بیماری‌هایی که نوع کلی آن با بیماری فعلی یکسان است.

سناریوی اول:

۱. بیمار برای بیماری جدیدش به پزشک مورد اعتماد خود مراجعه و برای درمان بیماری خود اقدام می‌کند.

۱. در این سناریو بیمار پزشک مورد اعتماد خود را انتخاب کرده‌است. بنابراین یا باید کلاس بیمار (Patient) این مسئولیت را قبول کرده و پزشک مورد اعتماد خود را ثبت کند و یا کلاس پزشک (Doctor) لیستی از بیماران داشته‌باشد که پزشک را مورد اعتماد دانسته‌اند. روش اول را انتخاب می‌کنیم و برای هر بیمار لیستی از پزشک‌های مورد اعتماد قرار می‌دهیم. بنابراین کلاس بیمار باید این مسئولیت را قبول کند که با گرفتن یک پزشک آن را به لیست پزشکان مورد اعتماد خود اضافه کند.

۲. حال فرض کنیم پزشک بیماری بیمار را تشخیص داده و به اون نیز اعلام می‌کند. حال پزشک باید این بیماری را به سابقه‌ی پزشکی بیمار اضافه کند. بنابراین کلاس پزشک باید مسئولیتی با عنوان افزودن به سابقه پزشکی بیمار و کلاس بیمار مسئولیتی با عنوان اضافه شدن سابقه‌ی پزشکی (بیماری) جدید و همچنین کلاس سابقه‌ی پزشکی (MedicalRecord) مسئولیتی با عنوان اضافه کردن بیماری به سابقه پزشکی داشته باشد.

۳. حال قبل از ثبت سابقه‌ی پزشکی جدیدی که در مرحله‌ی قبل انجام شد پزشک معالج باید تجویز دارو انجام دهد. بنابراین مسئولیتی در کلاس پزشک با عنوان تجویز دارو برای بیمار قرار می‌گیرد (اطلاعات مربوط به داروهای تجویز شده نیز در مرحله‌ی قبل به سابقه‌ی پزشکی بیمار اضافه می‌شود).

سناریوی دوم:

۲. بیمار به پزشک معالج مراجعه کرده و نسخه‌ی پزشک نیز برای او ثبت شده‌است، حال می‌خواهد به داروخانه مراجعه کرده و داروهای خود را دریافت کند.

۱. بنابراین کلاسی با عنوان نسخه (Prescription) خواهیم داشت که اطلاعات مربوط به نسخه‌ی بیمار شامل پزشکی که نسخه را نوشته‌است، تاریخ وارد شدن نسخه و داروهای نوشته‌شده را نگهداری می‌کند. حال برای اینکه دکتر داروخانه بتواند نسخه‌ی بیمار را مشاهده کند باید بتواند در سیستم متمرکز سلامت (CentralizedHealthCareSystem) با جستجوی اطلاعات بیمار به بخشی از سابقه‌ی پزشکی او دسترسی داشته باشد. بنابراین در کلاس سیستم متمرکز سلامت مسئولیتی با عنوان جستجوی سابقه‌ی پزشکی بیمار با استفاده از اطلاعات بیمار باید قرار داشته باشد.

۲. حال باید در نظر داشت که این جستجو نباید همه‌ی اطلاعات و سابقه‌ی پزشکی بیمار را به نمایش بگذارد. بلکه پزشک داروخانه فقط باید به نسخه‌های بیمار دسترسی داشته باشد. بنابراین در کلاس سیستم متمرکز سلامت باید مسئولیتی با عنوان تعیین سطح دسترسی برای هر کاربر وجود داشته باشد. برای نمونه سطح دسترسی برای خود بیمار و پزشک‌های مورد اعتماد او کامل بوده و سطح دسترسی برای پزشک‌های داروخانه‌ها در حد نسخه‌های بیمار باشد.

۳. بنابراین اکنون سطح دسترسی برای پزشک داروخانه در حد نسخه‌های بیمار تعریف شده‌است و با جستجوی اطلاعات بیمار توسط پزشک داروخانه باید پزشک به نسخه‌ها دسترسی داشته باشد. بنابراین کلاس سابقه‌ی پزشکی باید مسئولیت برگرداندن نسخه‌های بیمار را داشته باشد.

۴. حال پزشک باید نسخه‌ی موردنظر بیمار را از بین نسخه‌های او داشته باشد. بنابراین برای کلاس سابقه‌ی پزشکی باید مسئولیتی با عنوان جستجوی نسخه بر اساس نام پزشک معالج و تاریخ وارد شدن نسخه تعریف کرد. همچنین در کلاس نسخه نیز باید مسئولیت بررسی تاریخ نسخه و نام پزشک معالج قرار بگیرد.

۵. حال فرض کنیم نسخه‌ی بیمار توسط پزشک داروخانه تحویل داده شد، باید این تحویل نسخه به گونه‌ای اعلام شود. بنابراین در کلاس نسخه باید مسئولیتی با عنوان درآوردن نسخه به حالت تحویل داده‌شده یا بررسی شده قرار داده‌شود.

سناریوی سوم:

۳. بیمار برای یک بیماری جدید به پزشک مورد اعتماد خود مراجعه می‌کند. پزشک برای درمان بیماری مورد نظر بیمار را به پزشک دیگری ارجاع می‌دهد.

۱. وقتی پزشک مورد اعتماد بیمار را به پزشک جدید ارجاع می‌دهد باید این پزشک جدید را به لیست پزشک‌های مورد اعتماد بیمار اضافه کند. بنابراین در کلاس پزشک باید مسئولیتی با عنوان اضافه کردن به لیست پزشکان مورد اعتماد بیمار و همچنین در کلاس بیمار نیز مسئولیتی با عنوان اضافه کردن به لیست پزشکان قرار بگیرد.

۲. حال که پزشک مورد اعتماد بیمار یک پزشک دیگر را به لیست پزشکان مورد اعتماد بیمار اضافه کرد، باید برای این پزشک سطح دسترسی تعیین کند. بنابراین در کلاس پزشک باید مسئولیت‌های تعیین سطح دسترسی برای یک پزشک دیگر و همچنین اعمال سطح دسترسی قرار بگیرد.

سناریوی چهارم:

۴. بیمار می‌خواهد وارد سیستم شده و از سابقه‌ی پزشکی خود مطلع شده و آن را بررسی کند.

۱. وقتی بیمار می‌خواهد وارد سیستم شود باید سطح دسترسی کامل به اطلاعات خود داشته باشد که این کار در کلاس سیستم متمرکز سلامت انجام می‌شود. حال برای اینکه اطلاعات بیمار حفظ شود و فرد دیگری قادر به وارد شدن به سامانه به جای بیمار را

نداشته باشد، بیمار باید علاوه بر اطلاعات ورودی مانند نام و شماره ملی یک رمز ورودی نیز داشته باشد. بنابراین در کلاس بیمار مسئولیتی با نام ثبت کلمه عبور و در کلاس سیستم متمرکز سلامت نیز مسئولیتی با عنوان تعیین رمز عبور قرار می‌دهیم.

۲. حال که بیمار در حال وارد شدن است ممکن است رمز ورودی درستی وارد نکرده باشد. برای جلوگیری از این مشکل باید در کلاس بیمار این رمز چک شود. پس در این کلاس مسئولیتی با عنوان چک کردن رمز ورودی قرار می‌دهیم که اعتبار آن را تایید کند. کلاس سیستم متمرکز سلامت نیز باید مسئولیتی داشته باشد تا معتبر بودن رمز بیمار را تایید کرده و ورود او به سیستم را اعلام کند.

۳. حال بیمار وارد سیستم شده است و باید بتواند نمایش مناسبی از سابقه‌ی پزشکی خود ببیند. پس در کلاس سیستم متمرکز سلامت مسئولیتی با عنوان نمایش سابقه‌ی پزشکی قرار می‌دهیم. در ابتدا باید اطلاعات خود بیمار چاپ شده و نمایش داده شود (اطلاعاتی مانند نام، جنسیت، تاریخ تولد، نوع بیمه و ...). به همین منظور باید در کلاس بیمار مسئولیتی با عنوان نمایش اطلاعات بیمار قرار داد.

۴. بعد از آن شاید مناسب باشد لیستی از تمام مراجعات بیمار به پزشک‌های مختلف که بر اساس تاریخ مرتب شده‌اند به نمایش درآید. برای این کار لازم است تا در هر مراجعه‌ی بیمار به پزشک این اطلاعات در کلاس سابقه پزشکی ثبت شود. بنابراین کلاسی با عنوان مراجعه به پزشک (DoctorVisit) خواهیم داشت که مسئولیت ذخیره‌ی تمام اطلاعات مربوط به مراجعه به پزشک را در خود نگه می‌دارد. بنابراین کلاس سابقه‌ی پزشکی لیستی از مراجعات به دکتر را نگه می‌دارد و باید مسئولیت نمایش جزئیات تمامی این مراجعات را داشته باشد. بنابراین در کلاس مراجعه به دکتر مسئولیتی با عنوان نمایش جزئیات مراجعه قرار می‌دهیم. حال به کلاس سابقه‌ی پزشکی مسئولیت نمایش تمام مراجعات موجود در لیست را اعمال می‌کنیم.

Class name: CentralizedHealthCareSystem	Collaborators: <ul style="list-style-type: none">• Patient• MedicalRecord• Password
Fields: <ul style="list-style-type: none">•	
Responsibilities: <ul style="list-style-type: none">• Search Patient's list for a Patient's MedicalRecord• Set access level for user• Set a password for user• For a valid password confirm login• Show Patient's MedicalRecord	

Class name: Patient	Collaborators: <ul style="list-style-type: none">• Collection• Doctor• GeographicalLocation
Fields: <ul style="list-style-type: none">• List of Patient's trustworthy Doctors	

Responsibilities: <ul style="list-style-type: none"> • Add Doctor to trustworthy Doctors list • Add new record(Disease) to MedicalRecord • Set Patient's password • Check password validity • Print Patient's info 	<ul style="list-style-type: none"> • MedicalRecord
--	---

Class name: Doctor	Collaborators: <ul style="list-style-type: none"> • Patient • Prescription • DoctorVisit
Responsibilities: <ul style="list-style-type: none"> • Add to Patient's MedicalRecord • Prescribe for Patient • Add Doctor to Patient's trustworthy Doctors list • Set Doctor's access level • Determine another Doctor's access level 	

Class name: Disease	Collaborators: <ul style="list-style-type: none"> •
Responsibilities: <ul style="list-style-type: none"> • 	

Class name: MedicalRecord	Collaborators: <ul style="list-style-type: none"> • Date • Disease • Prescription • DoctorVisit
Fields: <ul style="list-style-type: none"> • List of DoctorVisits 	
Responsibilities: <ul style="list-style-type: none"> • Add new Disease • Get Patient's Prescriptions • Search Prescriptions list based on Doctor name and Prescription Date • Print info of all DoctorVisits in the list 	

Class name: Prescription	Collaborators: <ul style="list-style-type: none"> • Doctor • DoctorVisit • MedicalRecord
Responsibilities: <ul style="list-style-type: none"> • Check Prescription based on Doctor's name and Prescription's Date • Set the Prescription as delivered 	

Class name: DoctorVisit	Collaborators: <ul style="list-style-type: none"> • Disease • Prescription • MedicalRecord
Responsibilities: <ul style="list-style-type: none"> • Print DoctorVisit info 	

Class name: HomeAddress	Collaborators: <ul style="list-style-type: none"> • Patinet
Responsibilities: <ul style="list-style-type: none"> • 	

Class name: GeographicalLocation	Collaborators: <ul style="list-style-type: none"> • Patient
Responsibilities: <ul style="list-style-type: none"> • Print info. 	

نکات قسمت پیاده‌سازی پروژه:

- برای پیاده‌سازی کلاس‌های بیمار و دکتر ابتدا یک کلاس انسان () پیاده‌سازی می‌کنیم تا این دو کلاس نیز از آن ارث‌بری کنند. چرا که امکان دارد بعدها مسئولیت‌های دیگری نیز به این سیستم افزوده شود.
- کلاس سابقه پزشکی مجموعه‌ای از کلاس‌های مراجعه به پزشک، نسخه و بیماری را در خود دارد. در مواقعی که نیاز به نمایش کل سابقه‌ی پزشکی باشد، مانند زمانی که خود بیمار وارد سیستم می‌شود یا یکی از دکترهای مورد اعتماد بیمار که دسترسی کامل دارد، تمام اعضای لیست مراجعه به پزشک باید نمایش داده شود. در موارد دیگر مانند ورود دکتر داروخانه به سیستم باید صرفاً نسخه‌ها را نمایش داد. اگر خلاصه‌ای از بیماری‌های کاربر مورد نیاز باشد می‌توان از لیست بیماری‌ها در این کلاس استفاده کرد.