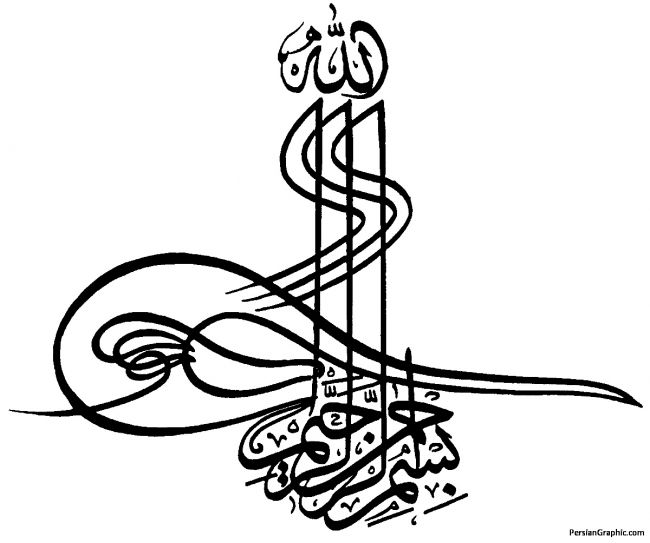
**پروژه درس**

عنوان پروژه:

**تحلیل سیستم مطب**

**نام پژوهشگر:   
 شماره دانشجوئي:**

**نام استاد راهنمای پروژه:**



**فهرست مطالب**

**چکیده** 5

**مقدمه** 7

**هدف سیستم جاری** 7

**زبان تحلیل سیستم** 7

**تاریخچة UML** 8

**ویژگیهای UML** 11

**فصل اول: Use Case Diagrams** 14

**1-1- Use case چیست؟** 15

**1-2- Actor چیست؟** 16

**1-3- دیاگرام use case چیست؟** 16

**1-4- ارتباطات بین use case ها چگونه است؟** 16

**1-5- شناسائی Actor های سیستم** 17

**1-5-1- Actor کاربر** 17

**1-5-2- Actor پزشک** 18

**1-5-3- Actor بیمار** 18

**1-6- شناسائی Use Case ها** 20

**1-6-1- Sing Up Use Case** 21

**1-6-1-1- سناریوی Sing Up Use Case** 22

**1-6-2- Sing In Use Case** 23

**1-6-2-1- سناریوی Sing In Use Case** 24

**1-6-3- Use Case پذیرش منشی** 25

**1-6-3-1- سناریوی Use Case پذیرش منشی** 26

**1-6-4- Use Case تجویزات و معاینات** 27

**1-6-4-1- سناریوی Use Case معاینات و تجویزات** 28

**1-6-5- Use Case برنامه زمانبندی** 29

**1-6-5-1- سناریوی Use Case برنامه زمانبندی** 30

**فصل دوم: Class Diagram** 31

**2-1- نمودار کلاس سیستم مطب** 32

**2-1- 1-نمودار کلاس سیستم مطب (بخش کاربران)** 33

**2-1- 1-1- توضیحات نمودار کلاس سیستم مطب (بخش کاربران)** 34

**2-1- 2-نمودار کلاس سیستم مطب (بخش معاینات و تجویزات)** 35

**فصل سوم: Sequence Diagram** 36

**3-1- نمودار توالی 1 – ثبت نام کاربر جدید** 37

**3-2- نمودار توالی 2 - ورود به سیستم** 38

**3-3- نمودار توالی 3 - عملیات برنامه ریزی پذیرش بیمار جدید** 39

**3-4- نمودار توالی 4 - عملیات ثبت پذیرش بیمار ویزیت شده** 40

**3-5- نمودار توالی 5 - عملیات استخدام منشی** 41

**3-6- نمودار توالی 6 - عملیات درخواست تجویزات انجام شده** 42

**فصل چهارم: Activity Diagram** 43

**4-1-** **نمودار فعالیت 1 – ثبت نام و ورد به سیستم** 44

**4-2-** **نمودار فعالیت 2 – ویرایش اطلاعات بیماران** 45

**4-3-** **نمودار فعالیت 3 - صندوق پستی** 46

**4-4-** **نمودار فعالیت 4- برنامه زمانبندی پذیرش بیماران** 47

**فصل پنجم: Domain Diagram** 48

**5-1- نمودار Domain سیستم مطب** 49

**فصل ششم: Collaboration Diagrams** 50

**6-1- نمودار همکاری – ثبت نام کاربر جدید** 51

**6-2- نمودار همکاری 2 - ورود به سیستم** 52

**6-3- نمودار همکاری - عملیات برنامه ریزی پذیرش بیمار جدید** 53

**6-4- نمودار همکاری - عملیات ثبت پذیرش بیمار ویزیت شده** 54

**6-5- نمودار همکاری - عملیات استخدام منشی** 55

**6-6- نمودار همکاری - عملیات درخواست تجویزات انجام شده** 56

**6-7- نمودار همکاری - عملیات ورود به سیستم** 57

**فصل هفتم: Entity Relation Diagram (ER)** 58

**7-1- نمودار ER منطقی سیستم مطب** 59

**7-2- نمودار ER فیزیکی سیستم مطب** 60

**فصل هشتم: Data Flow Diagrams** 61

**8-1- نمودار DFD سطح صفر** 62

**8-2- نمودار DFD سطح یک** 63

**8-3- نمودار DFD سطح یک (جداسازی مجموعه مورد نظر)** 64

**8-4- نمودار DFD سطح دو (پردازش پرداخت)** 65

**8-5- نمودار DFD سطح دو (پذیرش بیمار)** 66

**فصل نهم: State Chart Diagrams** 67

**9-1- نمودار وضعیت – ثبت نام کاربر جدید** 68

**9-2- نمودار وضعیت - ورود به سیستم** 69

**9-3- نمودار وضعیت - عملیات برنامه ریزی پذیرش بیمار جدید** 70

**9-4- نمودار وضعیت - عملیات ثبت پذیرش بیمار ویزیت شده** 71

**فصل دهم: زبان برنامه نویسی** 72

**ویژگی‌ها** 77

**سیستم یکپارچه شده** 79

**انواع داده** 80

Boxing و EnBoxing 81

**فصل یازدهم: Source Code های سیستم مطب** 82

**فصل دوازدهم: SQL Scripts** 98

**منابع و مواخذ** 121

**الف) کتب** 121

**ب) آدرسهای اینترنتی** 121

**چکیده**

با پیشرفت تکنولوژیهای سیستمهای اطلاع رسانی ، سمت و سوئی که این گونه سیستمهای اطلاعاتی پیدا نموده اند به علت حجم انبوه اطلاعات بیشتر به سمت سیستمهای پویایی بوده که مبتنی بر پایگاه های داده ای قدرتمند می باشند.

در این میان نباید نقش طراحان و تحلیل گران این گونه سیستمها را نادیده گرفت. چرا که در صورت نبود یک تحلیل مناسب از یک سیستم ، نمی توان از آن سیستم به نحو اساسی استفاده نمود.

در سیستم جاری (سیستم نرم افزاری مطب) سعی برآن شده که با روش تحلیل UML به بررسی سیستم با دید نرم افزاری پرداخته شود.

**مراحل مختلف طراحی و تحلیل به شرح ذیل انجام خواهد شد:**

* درفصل اول پروژه به بررسی Use Case ها و سناریوی Actor ها که در حقیقت وظائف آنها و روالهایی است که انجام می شود می پردازیم.
* در فصل دوم به بررسی Class Diagram ها (نمودار کلاس) خواهیم پرداخت. این مرحله مرحله بسیار مهمی می باشد.
* در فصل سوم به بررسی Sequence Diagram یا نموداهای توالی خواهیم پرداخت.
* در فصل چهارم به بررسی Activity Diagram ها (نمودارهای فعالیت) خواهیم پرداخت.
* در فصل پنجم به بررسی Domain Diagram خواهیم پرداخت.
* در فصل ششم به بررسی نمودار های Collaboration خواهیم پرداخت.
* در فصل هفتم به بررسی ER Diagram یا نمودار Entity Relations پرداخته خواهد شد.
* در فصل هشتم به بررسی و رسم نمودارهای DFD (نمودارهای جریان داده ها) خواهیم پرداخت.
* در فصل نهم به بررسی State Chart Diagram پرداخته خواهد شد.
* در فصل دهم مروری بر زبان برنامه نویسی مورد استفاده خواهیم داشت و به بررسی علل انتخاب این زبان خواهیم پرداخت.

**مقدمه**

با بهره گیری از پایه کامپیوتر و زبانهای برنامه نویسی بسیار قوی ، سیستمهای نرم افزاری زیادی در سراسر جهان پا به عرصه حیات نهاده اند. پروژه های نرم افزاری مختلفی طراحی و پیاده سازی گردیده اند که می توان اطلاعات ثبت شده در آنها را با سرعت بسیار زیادی بدست آورد.

این گونه نرم افزارها مزایای بسیار زیادی دارند. البته در حال حاضر با بهره گیری از ویژگیهای شئ گرایی به قابلیتهای این سیستمها اضافه شده اند و امنیت اطلاعات را بسیار بالا برده و بیشتر و بهتر می توان از ابزار و امکانات سخت افزاری و نرم افزاری موجود استفاده کرد.

**هدف سیستم جاری**

هدف از طراحی و پیاده سازی این سیستم ، نر م افزاری است که با آن بتوان در آن اطلاعات مراجعان به یک مطب را مورد ثبت و بازیابی قرار داد و بتوان گزارشهای متوعی از آنها بدست آورد.

**زبان تحلیل سیستم**

تحلیل سیستم با استفاده از زبان مدلسازی یکنواخت UML انجام خواهد شد. زبان مدلسازی یکنواخت یا Unified Modeling Language) UML)، یک زبان مدلسازی است که برای تحلیل وطراحی سیستمهای شی‌گرا بکار می‌رود. UML اولین بار توسط شرکت Rational ارائه شد و پس از آن از طرف بسیاری از شرکت‌های کامپیوتری و مجامع صنعتی و نرم‌افزاری دنیا مورد حمایت قرار گرفت. به طوریکه تنها پس از یک سال ، توسط گروه Object Management Group ، به عنوان زبان مدلسازی استاندارد پذیرفته شد. UML تواناییها و خصوصیات بارز فراوانی دارد که می‌تواند به طور گسترده‌ای در تولید نرم‌افزار استفاده گردد.

**تاریخچة UML**

دیدگاه شی‌گرایی (Object Oriented) از اواسط دهه 1970 تا اواخر دهه 1980 در حال مطرح شدن بود. در این دوران تلاشهای زیادی برای ایجاد روشهای تحلیل و طراحی شی‌گرا صورت پذیرفت. در نتیجة این تلاشها بود که در طول 5 سال یعنی 1989 تا 1994، تعداد متدولوژیهای شی‌گرا از کمتر از 10 متدولوژی به بیش از 50 متدولوژی رسید. تکثر متدولوژیها و زبانهای شی‌گرایی و رقابت بین اینها به حدی بود که این دوران به عنوان "جنگ متدولوژیها" لقب گرفت. از جمله متدولوژیهای پرکاربرد آن زمان می‌توان از Booch، OOSE، OMT، Fusion، Coad-Yourdan، Shlayer-Mellor وغیره نام برد. فراوانی و اشباع متدولوژیها و روشهای شی‌گرایی و نیز نبودن یک زبان مدلسازی استاندارد، باعث مشکلات فراوانی شده بود. از یک طرف کاربران از متدولوژیهای موجود خسته شده بودند، زیرا مجبور بودند از میان روشهای مختلف شبیه به هم که تفاوت کمی در قدرت و قابلیت داشتند یکی را انتخاب کنند. بسیاری از این روشها، مفاهیم مشترک شی‌گرایی را در قالبهای مختلف بیان می‌کردند که این واگرایی و نبودن توافق میان این زبانها، کاربران تازه‌کار را از دنیای شی‌گرایی زده می‌کرد و آنها را از این حیطه دور می‌ساخت. عدم وجود یک زبان استاندارد، برای فروشندگان محصولات نرم‌افزاری نیز مشکلات زیادی ایجاد کرده بود.

اولین تلاشهای استانداردسازی از اکتبر 1994 آغاز شد، زمانی که آقای Rumbaurgh صاحب متدولوژی OMT به آقای Booch در شرکت Rational پیوست و این دو با ترکیب متدولوژیهای خود، اولین محصول ترکیبی خود به نام "روش یکنواخت" را ارائه دادند. در سال 1995 بود که با اضافه شدن آقای Jacobson به این دو، روش یکنواخت ارائه شده با روش OOSE نیز ترکیب شد واین خود سبب ارائة UML نسخة 0.9 در سال 1996 گردید. سپس این محصول به شرکتهای مختلفی در سراسر جهان به صورت رایگان ارائه شد و استقبال شدید شرکت‌ها از این محصول و تبلیغات گسترده شرکت Rational، سبب آن شد که گروه OMG، نسخة 1.0 UML را به عنوان زبان مدلسازی استاندارد خود بپذیرد. تلاشهای تکمیلی UML استاندارد ادامه پیدا کرد و نسخة 1.1 آن در سال 1997 و نسخه 1.3 آن در سال 1999 ارائه گردید.

**UML چیست؟**

UML یا زبان مدلسازی یکنواخت، زبانی است برای مشخص کردن (Specify)، مصورسازی (Visualize)، ساخت (Construction) و مستندسازی (Documenting) سیستمهای نرم‌افزاری و غیر نرم‌افزاری و نیز برای مدلسازی سیستمهای تجاری. اما چرا مدل و مدلسازی؟

ایجاد یک مدل برای سیستمهای نرم‌افزاری قبل از ساخت یا بازساخت آن، به اندازه داشتن نقشه برای ساختن یک ساختمان ضروری و حیاتی است. بسیاری از شاخه‌های مهندسی، توصیف چگونگی محصولاتی که باید ساخته شوند را ترسیم می‌کنند و همچنین دقت زیادی می‌کنند که محصولاتشان طبق این مدلها و توصیفها ساخته شوند. مدلهای خوب و دقیق در برقراری یک ارتباط کامل بین افراد پروژه، نقش زیادی می‌توانند داشته باشند. شاید علت مدل کردن سیستمهای پیچیده این باشد که تمامی آن را نمی‌توان یکباره مجسم کرد، بنابراین برای فهم کامل سیستم و یافتن و نمایش ارتباط بین قسمتهای مختلف آن، به مدلسازی می‌پردازیم. UML زبانی است برای مدلسازی یا ایجاد نقشه تولید نرم‌افزار.

به عبارت دیگر، یک زبان، با ارائه یک فرهنگ لغات ویک مجموعه قواعد، امکان می‌دهد که با ترکیب کلمات این فرهنگ لغات و ساختن جملات، با یکدیگر ارتباط برقرار کنیم. یک زبان مدلسازی، زبانی است که فرهنگ لغات و قواعد آن بر نمایش فیزیکی و مفهومی آن سیستم متمرکزند. برای سیستمهای نرم‌افزاری نیاز به یک زبان مدلسازی داریم که بتواند دیدهای مختلف معماری سیستم را در طول چرخة تولید آن، مدل کند.

فرهنگ واژگان و قواعد زبانی مثل UML به شما می‌گویند که چگونه یک مدل را بسازید و یا چگونه یک مدل را بخوانید. اما به شما نمی‌گویند که در چه زمانی، چه مدلی را ایجاد کنید. یعنی UML فقط یک زبان نمادگذاری (Notation) است نه یک متدولوژی. یک زبان نمادگذاری شامل نحوة ایجاد و نحوة خواندن یک مدل می‌باشد، اما یک متدولوژی بیان می‌کند که چه محصولاتی باید در چه زمانی تولید شوند و چه کارهایی با چه ترتیبی توسط چه کسانی، با چه هزینه‌ای، در چه مدتی و با چه ریسکی انجام شوند.

**ویژگیهای UML**

UML دارای ویژگیهای بارز فراوانی است که در این قسمت به آنها می‌پردازیم. UML یک زبان مدلسازی است اما چیزی فراتر از چند نماد گرافیکی است. بطوریکه در ورای این نمادها، یک سمانتیک (معناشناسی) قوی وجود دارد، بطوریکه یک تولیدکننده می‌تواند مدلهایی تولید کند که تولید‌کننده‌های دیگر و یا حتی یک ماشین آن را بخواند و بفهمد. بنابراین یکی دیگر از نقش‌های مهم UML "تسهیل ارتباط" بین اعضای پروژه و یا بین تولیدکنندگان مختلف می‌باشد. این ارتباط بسیار مهم است. شاید دلیل اصلی اینکه تولید نرم‌افزار به صورت فریبنده‌ای دشوار است، همین عدم ارتباط مناسب بین اعضای پروژه باشد و اگر در تولید نرم‌افزار، بین اعضای پروژه گزارشهای هفتگی و مداوم وجود داشته باشد، بسیاری از این دشواریها برطرف خواهد شد.

البته این را هم باید در نظر گرفت که UML کمی پیچیده است و این به خاطر آن است که سعی شده است نمودارهایی فراهم شود که در هر موقعیتی و با هر ترتیبی قابل استفاده باشند. دلیل دیگر پیچیدگی از آنجا ناشی می‌شود که UML ترکیبی است از زبانهای مختلف، که برای حفظ سازگاری و جمع کردن خصوصیات مثبت آنها، ناگزیر از پذیرش این پیچیدگی می‌باشد.

UML موفقیت طرح را تضمین نمی‌کند، اما در عین حال خیلی چیزها را بهبود می‌بخشد. به عنوان مثال استفاده از UML، تا حد زیادی، هزینه‌های ثابتی نظیر آموزش و استفاده مجدد از ابزارها را در هنگام ایجاد تغییر در سازمان و طرحها کاهش می‌دهد.

مساله دیگر اینکه، UML یک زبان برنامه‌نویسی بصری (visual) نیست، اما مدلهای آن را می‌توان مستقیماً به انواع زبانهای مختلف ارتباط داد. یعنی امکان نگاشت از مدلهای UML به کد زبانهای برنامه‌نویسی مثل Java و VC++ وجود دارد که به این عمل "مهندسی روبه‌جلو" می‌گویند. عکس این عمل نیز ممکن است؛ یعنی این امکان وجود دارد که شما بتوانید از کد یک برنامه زبانی شی‌گرا، مدلهای UML معادل آن را بدست آورید. به این عمل "مهندسی معکوس" می‌گویند. مهندسی روبه‌جلو و معکوس از مهمترین قابلیتهای UML به شمار می‌روند، البته نیاز به ابزار Case مناسبی دارید که از این مفاهیم پشتیبانی‌کنند.

اگر با زبانهای مدلسازی دیگر کار کرده باشید، برای کار با UML مشکل چندانی نخواهید داشت. اما برای شروع کار با UML به عنوان اولین زبان مدلسازی، بهتر است فقط با نمودارهای خاصی کار کنید. برای این کار بهتر است ابتدا با نمودارهای مورد کاربرد و تعامل کار کنید و پس از مدتی کار و آشنا شدن با ویژگیهای اولیة آن، به یادگیری و استفاده از نمودارها واجزای دیگر بپردازید. در مقایسه با زبانهای مدلسازی دیگر مثلER و زبان فلوچارتی DR، زبان UML نمودارهای قویتر و قابل‌فهمتری را ارائه می‌دهدکه شامل تمامی مراحل چرخة حیات تولید نرم‌افزار (تحلیل، طراحی، پیاده‌سازی و تست) می‌شود.

یکی دیگر از ویژگیهای مهم UML این است که مستقل از متدولوژی یا فرایند تولید نرم‌افزار می‌باشد و این بدان معنی است که شما برای استفاده از UML، نیاز به استفاده از یک متدولوژی خاص ندارید و می‌توانید طبق متدولوژی‌های قبلی خود عمل کنید با این تفاوت که مدلهایتان را با UML نمایش می‌دهید. البته مستقل‌بودن از متدولوژی و فرایند تولید، یک مزیت برای UML می‌باشد؛ زیرا بسیاری از انواع پروژه‌ها و سیستمها نیاز به متدولوژی خاص خود دارند. اگر UML در پی پیاده کردن همة اینها بر می‌آمد، یا بسیار پیچیده می‌شد و یا استفاده خود را محدود می‌کرد. البته متدولوژیهایی براساس UML در حال شکل‌گیری می‌باشند.

از دیگر ویژگیهای UML می‌توان به پشتیبانی از مفاهیم سطح بالای شی‌گرایی مثل Collaboration، Framework، Pattern و Component اشاره کرد. همچنین UML با استفاده از یک سری مکانیزمهای گسترش‌پذیر امکان می‌دهد که بتوان زبانهای مدلسازی جدیدتری (با گسترش مفاهیم پایه‌ای موجود) ایجادکرد.

**در ادامه به بررسی و تحلیل سیستم مطب خواهیم پرداخت.**

**فصل اول: Use Case Diagrams**

در این فصل به بررسی و شناخت Use Case ها ، Actor ها ، رسم Use Case Diagram ها و در نهایت نوشتن سناریویی برای هر یک از Use Case Diagram ها خواهیم پرداخت. در ابتدا بایستی بدانی که مفاهیم هر یک از عبارات فوق چیست؟

**1-1- Use case چیست؟**

Use Case ها که در فاز آنالیز پروژه برای شناساسیی و تقسیم بندی فعالیت های سیستم استفاده می شوند و می تونند به عنوان سرویس ها یا کارکردهایی که سیستم برای کاربران خودش فراهم می کنه نیز توصیف بشوند.

**دو دیدگاه وجود دارد:** یکی داخلی , دید ساختاری و دیگری خارجی و دید وظیفه گرایی(task Oriented)

در دیدگاه اول ما باید کلاسها و متدها را تعریف کنیم و سپس واسطهای کاربری (user interface)را تعریف کنیم. مشکل اینجاست که برای کاربر مهمترین چیز رفتار سیستم است ولی واسطهای کاربری تنها قسمت آخر فرآیند را تعریف می کنند. و این مارو به سمت مشکلاتی می بره نظیر اینکه سیستم تمام کارکردهایی که ما می خواهیم در اختیارمون قرار نمی ده و یا کارکردهایی رو داره که مورد نیاز ما نبوده.

در دید دوم ،سیستم از Actor ها و فعالیتها و کلاسهایی که به فعالیتها وصل شده اند پشتیبانی می کند.در این دیدگاه هیچ کار ناخواسته ای وجود ندارد و سیستم تمام فعالیتهای کاربر را پشتیبانی می کند که همه آنها در Use case Diagram نمایش داده می شود.

**1-2- Actor چیست؟**

Actor هـا نقشـهایـی را ارائـه مـی دهـنـد کـه تـوســط کـاربــران سیــــسـتمـــهای اطـلاعـاتی (Information System=IS) انجام می شود. این Actor ها می توانند انسانها ,کامپیوترها , سخت افزارها و حتی نرم افزار ها باشند. تنها چیزی که آنها را Actor می کند این است که آنها باید بیرون از قسمتی باشند که توسط سیستم به use case ها تقسیم شده است ویکسری ورودی برای سیستمهای اطلاعاتی فراهم می کنند و یکسری از آنها خروجی می گیرند.

**1-3- دیاگرام use case چیست؟**

دیاگرامهای use case با استفاده از Use case و Actor عملکرد (Functionality) سیستم رامدلسازی می کنند.

**1-4- ارتباطات بین use case ها چگونه است؟**

ارتباط بین use case ها یا به صورت Extends است و یا به صورت Uses .Uses دلالت بر این دارد که یک use case برای انجام وظیفه و فعالیتش نیازمند use case دیگری است. Extends دلالت بر این دارد که use case ی یک امکان و گزینه اختیاری برای use case دیگر است که در بعضی شرایط از آن استفاده می کند.

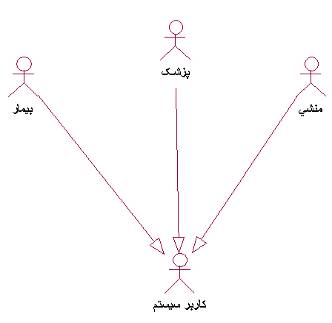
**1-5- شناسائی Actor های سیستم**

Actor ها در حقیقت مکانیسمی برای طبقه بندی External User ها هستند. Actor ها می توانند در چهار گروه User ، Applications ، Devices و External Events باشند. در سیستم جاری با دو نوع اکتور سرو کار داریم:

* Actor کاربر
* Actor External Events

**1-5-1- Actor کاربر**

کاربران سیستم در حالت کلی شامل Actor مدیر وب سایت و Actor بازدیدکنندگان وب سایت می باشند.



Actor منشی و Actor بیمار و Actor پزشک با رابطه Generalization با Actor کاربر سیستم ارتباط دارند و تمامی خواص Actor کاربر سیستم را دارا هستند. در حالت کلی ، کاربر به استفاده کننده سیستم نرم افزاری مطب گفته می شود که سایر تعاریف Actor های سیستم از آن مشتق می شود. (ارث بری دارند)

مفهوم ارث بری در نمودار فوق بدین معنی است که در حالت کلی Actor کاربر سیستم دارای یک سری اطلاعات اعم از اطلاعات تماس ، اطلاعات شناسنامه ای و . . . می باشد که می توان در تعریف Actor ها آنها را در کلاس کلی تعریف نمود و در تعریف سایر خواص مربوط به Actor ها موارد تکراری را از کلاس اصلی مشتق گرفت.

**1-5-2- Actor پزشک**



**شرح:** پزشک کسی است که کار اصلی در مطب را انجام می دهد. بیماران را ویزیت می نماید ، برای آنها نسخه صادر می نماید و بر روند کار منشی مطب نظارت دارد.

**1-5-3- Actor بیمار**

****

**شرح:** بیمار کسی است که برای معالجه خود به مطب مراجعه نموده و پس از ویزیت توسط دکتر ، نسخه های خود را دریافت می نماید.

**1-5-4- Actor منشی مطب**

منشی مطب کسی است که تمامی عملیات مربوط به پذیرش و زمانبندی مراجعه بیماران را به عهده دارد و البته کارهای جانبی و خدمات جانبی مطب را نیز انجام می دهد.



**1-5-5- External Events Actor**

این گونه Actor ها به صورت Periodic با سیستم محاوره دارند. در سیستم جاری یک Timer برای ثبت ونگهداری زمان شروع و پایان استفاده کاربران سیستم (تمامی اکتورها) تعبیه گردیده است.

icon_lesson_timer

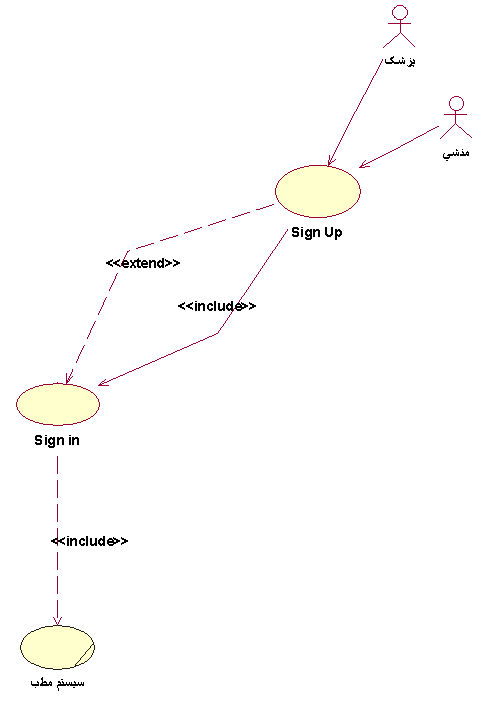
سایر **External Events Actor** در هنگام نیاز بررسی خواهند شد.

**1-6- شناسائی Use Case ها**

Functionality های زیر به درخواست مشتری پس از چند مرحله مصاحبه شناسایی و استخراج گردیده است:

1. ثبت نام در سیستم (Sign Up)
2. ورد به سیستم ( Log In / Sign In)
3. مشاهده فهرست زمانی ویزیت بیماران (Main Menu)
4. تنظیم برنامه زمانبندی برای مراجعه بیماران (Time Slice Definition)
5. صندوق پیام مدیر (Admin Mailbox)
6. صندوق پیام منشی (Secretary Mailbox)
7. لیست ویزیتها و تجویزات انجام شده ( Visitd List)
8. ویراش لیست بیماران و تجویزات توسط مدیر (Edit Users Profiles )

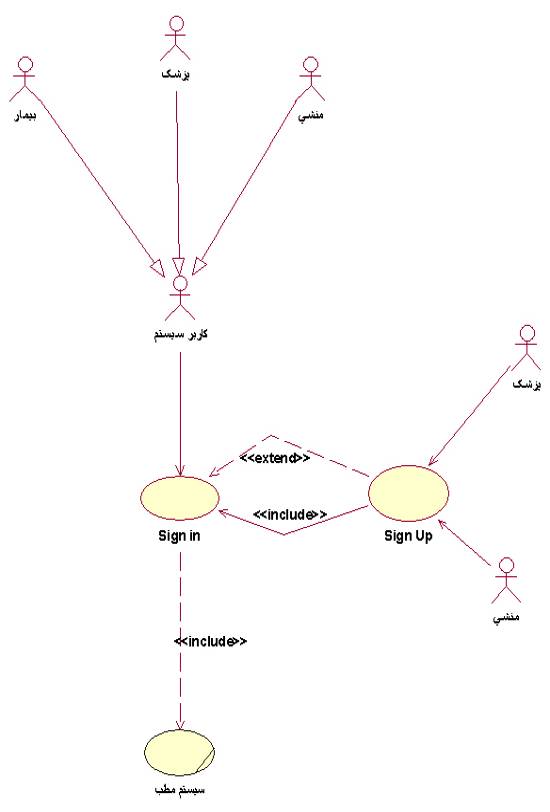
**1-6-1- Sing Up Use Case**



**1-6-1-1- سناریوی Sing Up Use Case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سناریوی شماره 1** | | **سیستم مطب** | | | **نام: ثبت نام در سیستم (Sign Up)**  **Actor: منشی و پزشک** |
| **شرح سناریو:** در این سناریو ، Actor پس از فشردن کلید ثبت نام به بخش ورود اطلاعات وارد شده و پس از وارد نمودن اطلاعات کاربری ، سیستم به کاربر نمایش داده خواهد شد و کاربر پس از تائید به عملیات ثبت نام خود پایان می دهد.  پس از این مرحله نام کاربری و کلمه عبور به کاربر نمایش داده خواهد شد. | | | | | |
| **Assumption**: منظور از Actor در این Use Case پزشک و منشی یا منشیان مطب می باشند که در هنگام ورود برای اولین بار به سیستم اطلاعات کاربری خود را وارد می نمایند تا در هنگام انجام عملیات ، شرح عملیات انجام شده با نام کاربری آنها ثبت شود. | | | | | |
| **Include:** Sign In | | | **Extended:** Sign In | | |
| **(Pre-Condition):** کاربر برای ثبت نام به بخش ثبت نام مراجعه می نماید. | | | | | |
| **مراحل:** | **Actor** | | | **System** | |
| 1- کاربر درخواست ثبت نام از سیستم را می نماید. | | | 2- سیستم فرم مربوط به ورود اطلاعات ثبت نام کاربر را نمایش می دهد. | |
| 3- کاربر اطلاعات درخواستی را وارد می نماید. (اعم از اطلاعات شناسنامه ای ، اطلاعات تماس ، عکس و . . .) و کلید ثبت نام را فشار می دهد. | | | 4- سیستم اطلاعات ثبت شده را برای تائید به کاربر نمایش میدهد. | |
| 5-کاربر با کلیک بر روی کلید تائید ، ثبت نام خدو را تائید می نماید. | | | 6- عملیات کلیک بر روی کلید تائید ثبت نام در سیستم ثبت می گردد و مجوز فعالسازی حساب کاربری صادر می گردد. | |
| 7- نام کاربری و کلمه عبور به کاربر نمایش داده می شود. | |
| **Constraints:** پس از ورود نام کاربری برای ثبت نام در صورتی که نام وارد شده قبلا ثبت شده باشد به کاربر پیغام مناسبی مبنی بر انتخاب نام کاربری دیگر نمایش داده خواهد شد. | | | | | |
| **Post-Condition:** کاربران ثبت نامی پس از تائید ثبت نام خود (پس کلیک بر روی کلید تائید حساب کاریری) می توانند وارد سیستم شده و امکانات سیستم استفاده نمایند. | | | | | |
| **Exception:** کاربر نمی تواند بدون ثبت نام وارد سیستم شود. عملیات مجوز ثبت نام نیز باید توسط مدیر سیستم با سطوح دسترسی خاص انجام شود. | | | | | |

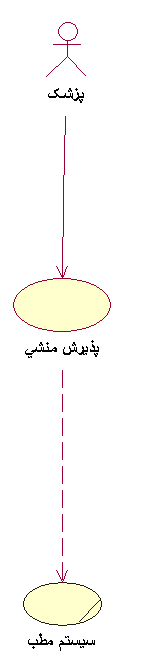
**1-6-2- Sing In Use Case**



**1-6-2-1- سناریوی Sing In Use Case**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سناریوی شماره 2** | | **سیستم مطب** | | | **نام: ثبت نام در سیستم (Sign In)**  **Actor: منشی و پزشک** |
| **شرح سناریو:** در این سناریو ، کاربر پس از وارد نمودن نام کاربری و کلمه عبور خود و تائید صحت مشخصات از طرف سیستم ، وارد سیستم می شود و می تواند از امکانات کامل نرم افزار مطب استفاده نماید. | | | | | |
| **Assumption**: منظور از کاربر در این Use Case همان پزشک و منشی است که قبلا ثبت نام نموده اند و ثبت نام آنها تکمیل و تائید شده است. | | | | | |
| **Include:** Sign Up | | | **Extended:** | | |
| **(Pre-Condition):** ثبت نام کاربر قبلا تکمیل شده و کاربر دارای کلمه عبور و نام کاربری می باشد. | | | | | |
| **مراحل:** | **Actor** | | | **System** | |
| 1- کاربر بر روی کلید درخواست ورود کلیک می نماید. | | | 2- سیستم فرم مربوط به ورود نام کاربری و کلمه عبور را نمایش می دهد. | |
| 3- کاربر نام کاربری و کلمه عبور را وارد می نماید و کلید ورود را فشار می دهد. | | | 4- سیستم پس از بررسی نام کاربری و کلمه عبور ، وارد صفحه اصلی سیستم نرم افزاری مطب می گردد. | |
|  | | | 5- لیست زمانبندی بیماران در صورت وجود به کاربر نمایش داده خواهد شد. | |
| **Constraints:** نام کاربری ترکیبی از حروف و ارقام است و نباید کمتر از 6 حرف باشد. | | | | | |
| **Post-Condition:** کاربر وارد صفحه اصلی نرم افزار مطب می گردد و می تواند از تمامی امکانات نرم افزار استفاده نماید. | | | | | |
| **Alternative Flow:** در صورتی که نام کاربری و کلمه عبور وارد شده معتبر نباشد ، پیغامی مبنی بر ورود مجدد اطلاعات از طرف سیستم صادر می گردد. | | | | | |
| **Exception:** در صورتی که کاربر رمز عبور خود را فراموش کرده باشد بایستی درخواست کلمه عبور جدید و یا قبلی را از سیستم بنماید و سیستم باید بتواند کلمه عبور را به کاربر با پرسیدن برخی سوالات امنیتی ارائه دهد. | | | | | |

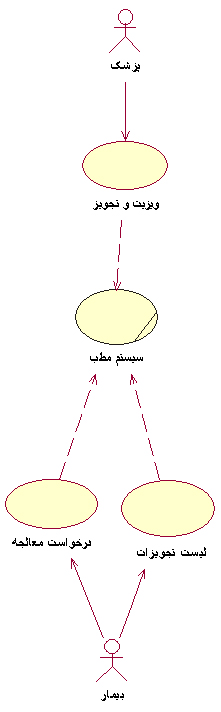
**1-6-3- Use Case پذیرش منشی**



**1-6-3-1- سناریوی Use Case پذیرش منشی**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سناریوی شماره 3** | | **سیستم مطب** | | | **نام: پذیرش منشی**  **(Staff Accepting)**  **Actor: منشی و پزشک** |
| **شرح سناریو:** در این سناریو ، پزشک پس از ورود به سیستم ، درخواستهای مربوط به استخدام را بررسی می نماید و یک منشی با توجه به سوابق کاری و تخصص حرفه ای انتخاب می نماید. | | | | | |
| **Assumption**: منظور از کاربر در این Use Case همان پزشک و منشی است که قبلا ثبت نام نموده اند و ثبت نام آنها تکمیل و تائید شده است. | | | | | |
| **Include:** | | | **Extended:** | | |
| **(Pre-Condition):** درخواست برای استخدام قبلا انجام شده و در خواست گنندگان دارای کد مخصوص برای مشاهده نتیجه درخواست ثبت نام می باشند. | | | | | |
| **مراحل:** | **Actor** | | | **System** | |
| 1- کاربر بر روی کلید درخواست استخدام جدید کلیک می نماید. | | | 2- سیستم فرم مربوط به ورود اطلاعات مورد نیاز برای استخدامی را به کاربر سیستم نمایش می دهد. | |
| 3- کاربر اطلاعات مورد نیاز برای شرکت در آزمون استخدامی متقاضی را وارد می نماید و کلید ثبت را فشار می دهد. | | | 4- سیستم پس از بررسی صحت اطلاعات از نظر موراد در خواستی ، کدی منحصر به فرد در اختیار کاربر قرار می دهد. | |
|  | | | 5- لیست زمانبندی مصاحبات به درخواست کنندگان نمایش داده می شود. | |
| **Constraints:** اطلاعات وارد شده باید دقیقا بر اساس استاندارد مورد نیاز باشند به عنوان مثال کد ملی یک عدد 10 رقمی می باشد. | | | | | |
| **Post-Condition:** کاربر وارد صفحه درخواست ثبت نام استخدام می گردد و می تواند از اطلاعات درخواست کنندگان استخدام را ثبت نماید. | | | | | |
| **Alternative Flow:** در صورتی که اطلاعات اعتبار نداشته باشند باید پیغامی داده شود. | | | | | |
| **Exception:** در صورتی که در خواست کنندگان استخدامی ، اطلاعات مورد نیاز خود را همراه نداشته باشند بایستی این امکان وجود داشته باشد که اطلاعات اولیه ثبت شود و در زمان مراجعه بعدی سایر اطلاعات وارد سیستم شود. | | | | | |

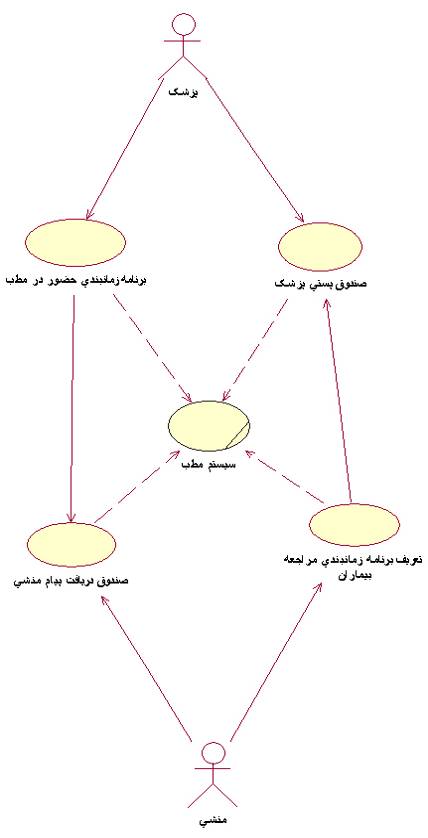
**1-6-4- Use Case تجویزات و معاینات**



**1-6-4-1- سناریوی Use Case معاینات و تجویزات**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سناریوی شماره 4** | | **سیستم مطب** | | | **نام: تجویزات معاینات**  **(Declare & Medical)**  **Actor: بیمار و پزشک** |
| **شرح سناریو:** در این سناریو ، پزشک پس از ورود به سیستم ، درخواستهای مربوط به معاینات را بررسی می نماید و اجازه ورود یک بیمار را صادر می نماید. (اولویت پس از موارد Emergency با زمان ثبت نام می باشد) | | | | | |
| **Assumption**: منظور از کاربر در این Use Case همان پزشک و بیمار است که قبلا ثبت نام نموده اند و ثبت نام آنها تکمیل و تائید شده است. | | | | | |
| **Include:** پزشک ، بیمار | | | **Extended:** صفحه اصلی سیستم مطب | | |
| **(Pre-Condition):** درخواست برای معاینات توسط بیمار قبلا انجام شده و در خواست گننده دارای کد مخصوص و زمان مراجعه جهت معاینه و تجویز می باشند. | | | | | |
| **مراحل:** | **Actor** | | | **System** | |
| 1- پزشک بر روی مشاهده لیست بیماران کلیک می نماید. | | | 2- سیستم فرم مربوط به بیماران موجود را بر اساس موارد اورژانسی و زمان قبت نام به پزشک نمایش می هد. | |
|  | | | 3- پزشک اجازه ورود بیمار را صادر می نماید. | |
| 4- بیمار جهت معاینات و تجویزات به پزشک مراجعه می نماید. | | | 5- پزشک معاینات و تجویزات را انجام می دهد و در سیستم ثبت می نماید. | |
|  |  | | | 6- سیستم نتیجه تجویزات را جهت دریافت نسخه به بیمار ارائه می دهد. | |
| **Constraints:** کاربر باید اطلاعات درخواستی تجویزات انجام شده سیستم را وارد نماید. | | | | | |
| **Post-Condition:** کاربر وارد صفحه تجویزات و معاینات شده و نتیجه معاینات و تجویزات انجام شده را در سیستم ثبت می نماید. | | | | | |
| **Alternative Flow:** در صورتی که بیمار سوابقی داشته باشد بایستی از طرف سیستم پیغامی صادر شود. | | | | | |
| **Exception:** در صورتی که برای درخواست کننده (بیمار) زمان حالی وجود نداشته باشد سیستم باید به گونه ای آدرس محل دیگری را برای مراجعه بیمار ، اعلام نماید. | | | | | |

**1-6-5- Use Case برنامه زمانبندی**

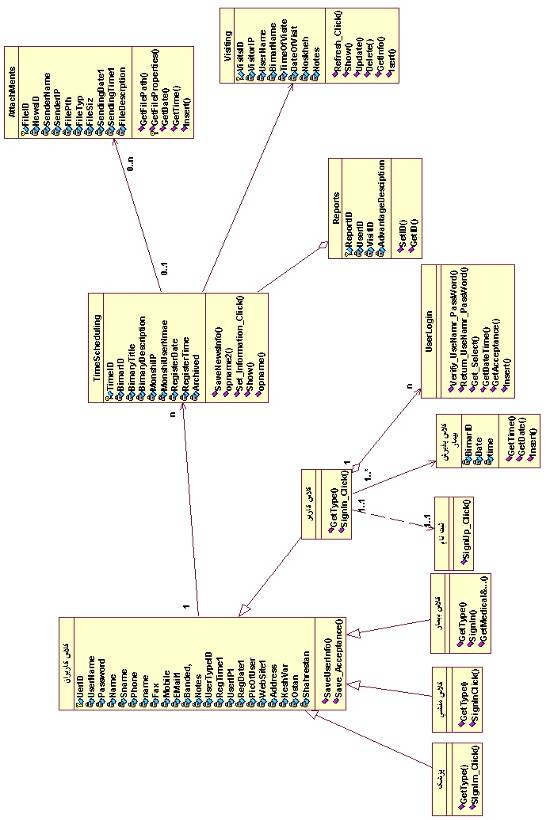


**1-6-5-1- سناریوی Use Case برنامه زمانبندی**

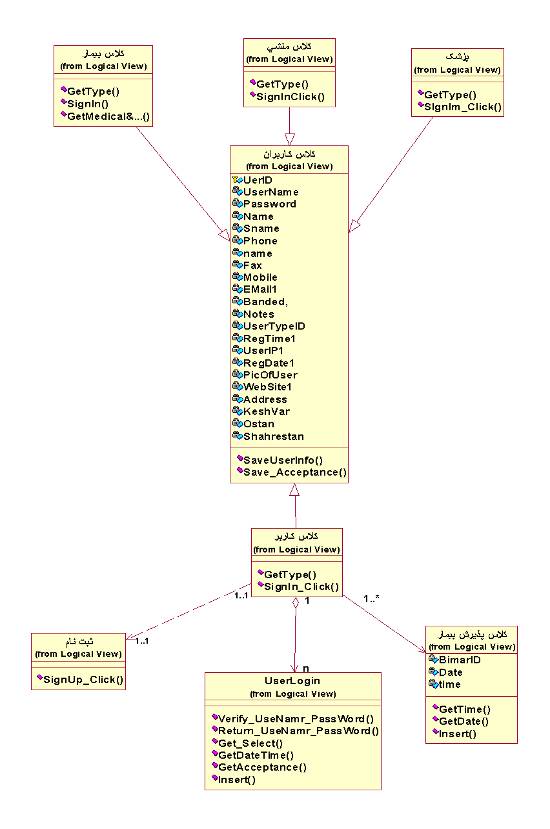
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **سناریوی شماره 5** | | **سیستم مطب** | | | **نام: برنامه زمانبندی**  **(Time Scheduling)**  **Actor: منشی و پزشک** |
| **شرح سناریو:** در این سناریو ، ابتدا منشی برنامه زمانبندی مراجعه بیماران را به پزشک ارائه می دهد. سپس پزشک برنامه را مورد بررسی قرار داده سپس برنامه زمانبندی حضور را اعلام می نماید. | | | | | |
| **Assumption**: منظور از کاربر در این Use Case همان پزشک و منشی است که قبلا ثبت نام نموده اند و ثبت نام آنها تکمیل و تائید شده است. | | | | | |
| **Include:** پزشک ، منشی | | | **Extended:** صفحه اصلی سیستم مطب | | |
| **(Pre-Condition):** زمان خالی برای تعریف برنامه زمانبندی وجود دارد. | | | | | |
| **مراحل:** | **Actor** | | | **System** | |
| 1- منشی فهرست درخواست بیماران را برای پزشک ارسال می نماید. | | | 2- پزشک پس از ورود به سیستم پیغامی مبنی بر وجود پسغام جدید در صندوق پستی مشاهده می نماید. | |
|  | | | 3- پزشک وارد صندوق پستی می شود و لیست بیماران را مشاهده می نماید. | |
|  | | | 4- پزشک برنامه زمانبندی حضور در مطب را به صندوق پستی منشی ارسال می نماید. | |
|  | 5- منشی اطلاعات را مشاهده می نماید. | | |  | |
| **Constraints:** کاربر باید زمان دقیق مربوط به معاینات هر بیمار را مشخث نماید. | | | | | |
| **Post-Condition:** پس از اعلام برنامه زمان بندی توسط پزشک ، منشی می تواند شمان معاینات را به پزشک اعلام نماید. | | | | | |
| **Alternative Flow:** در صورتی که بیمار نتواند به موقع در محل حاضر شود باید بیمار بعدی بلافاصله جایگزین گردد. | | | | | |
| **Exception:** پس از تائید یا رد درخواست معاینات ، اطلاعات به عنوان آرشیو ذخیره شوند. | | | | | |

**فصل دوم: Class Diagram**

**2-1- نمودار کلاس سیستم مطب**



**2-1- 1-نمودار کلاس سیستم مطب (بخش کاربران)**

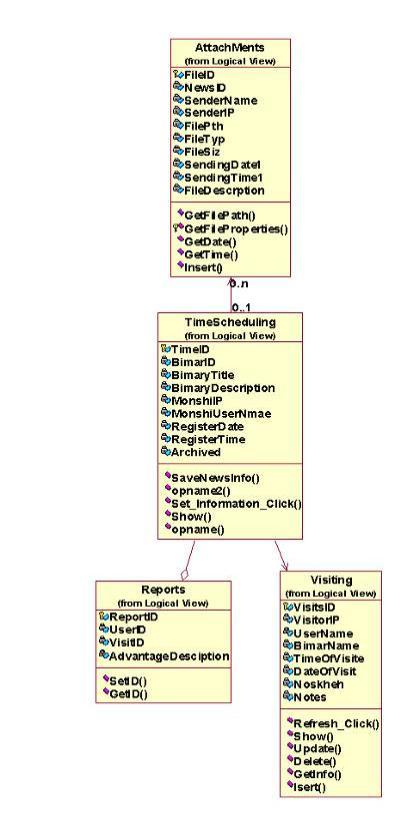


**2-1- 1-1- توضیحات نمودار کلاس سیستم مطب (بخش کاربران)**

در این بخش یک کلاس اصلی به نام کلاس کاربران داریم که کلاسهای پزشک ، بیمار و منشی مطب از آنها مشتق می شود. نوع ارتباط از نوع Generalization می باشد.

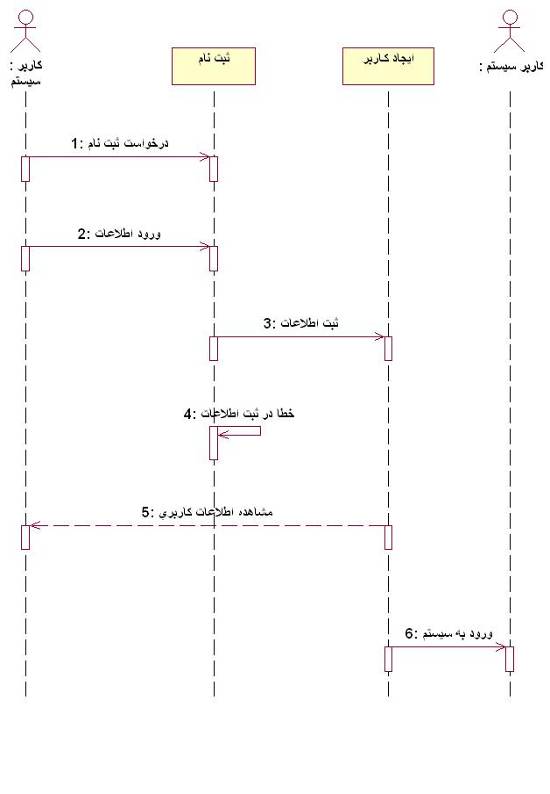
اعمال اصلی کلاس کاربر شامل موارد ورود به سیستم ، پذیرش بیمار ، ثبت نام می باشد. هر کدام از کلاسها ، دارای یک سری متد می باشند که با استفاده از این متدها عملیات خاصی را انجام می دهند. به عنوان مثال متد GetTime() زمان جاری سیستم را برمی گرداند و پایگاه داده در هنگام استفاده از کلاس ذخیره می نماید.

**2-1- 2-نمودار کلاس سیستم مطب (بخش معاینات و تجویزات)**

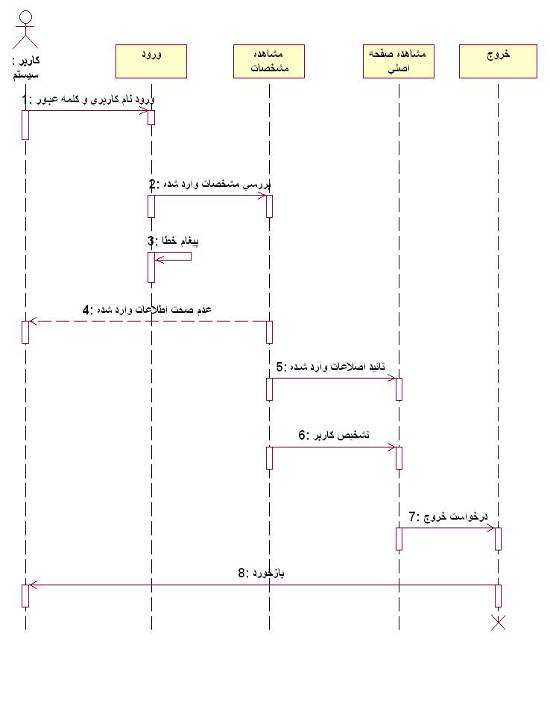


**فصل سوم: Sequence Diagram**

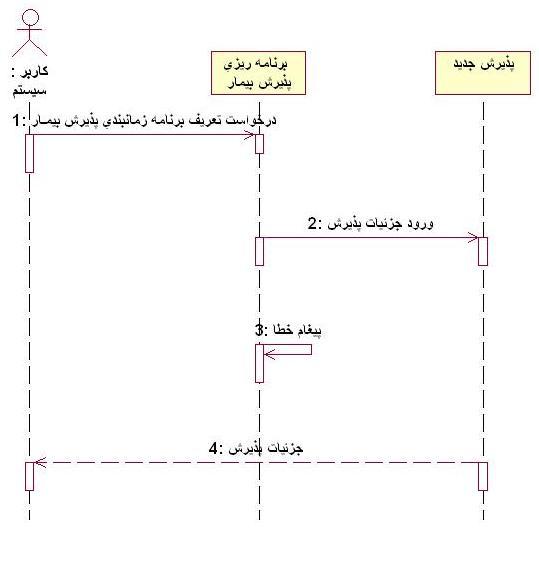
**3-1- نمودار توالی 1 – ثبت نام کاربر جدید**



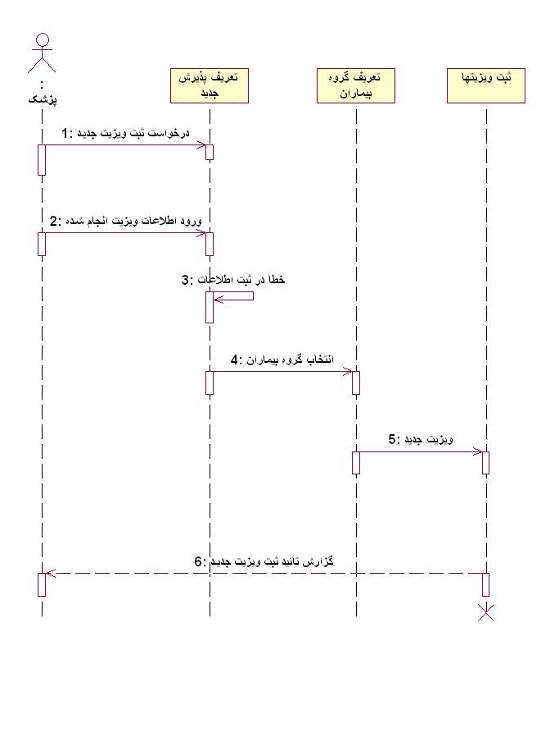
**3-2- نمودار توالی 2 - ورود به سیستم**



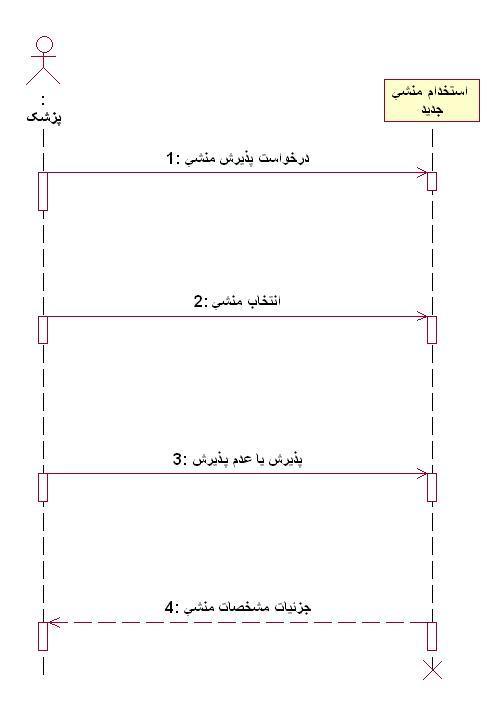
**3-3- نمودار توالی 3 - عملیات برنامه ریزی پذیرش بیمار جدید**



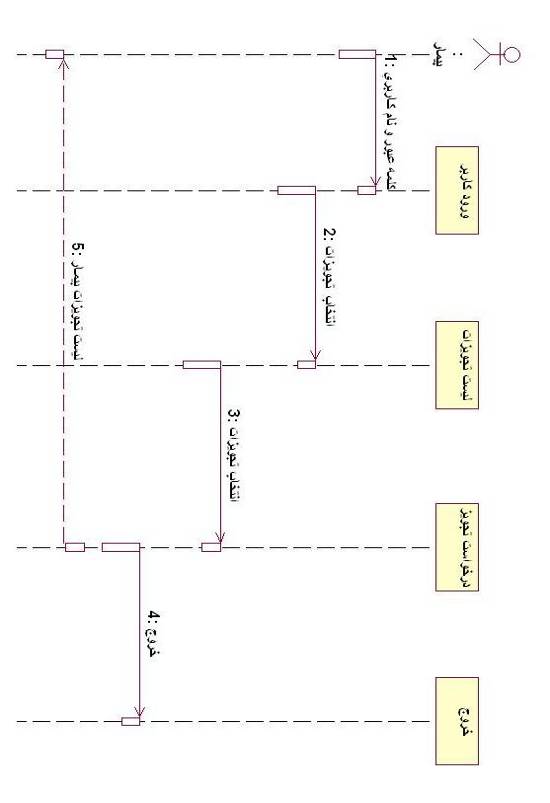
**3-4- نمودار توالی 4 - عملیات ثبت پذیرش بیمار ویزیت شده**



**3-5- نمودار توالی 5 - عملیات استخدام منشی**

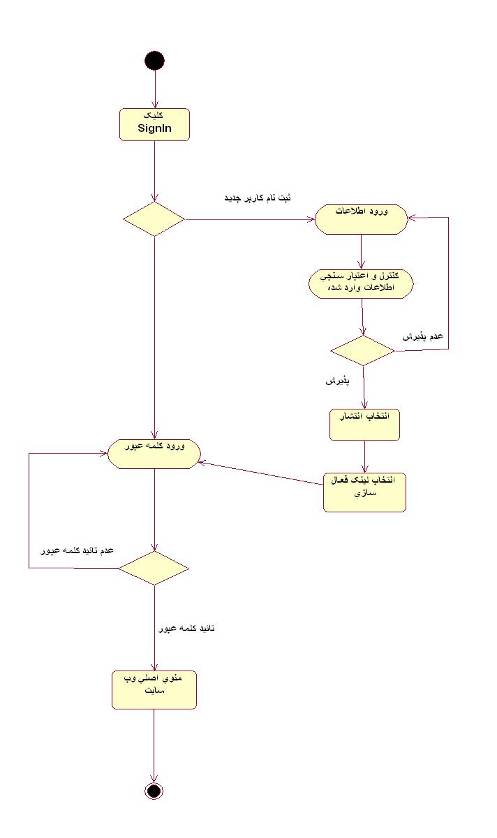


**3-6- نمودار توالی 6 - عملیات درخواست تجویزات انجام شده**

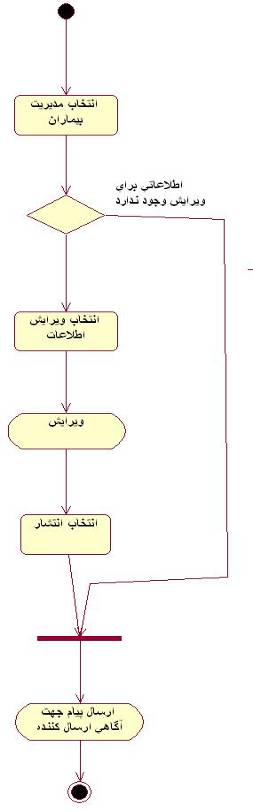


**فصل چهارم: Activity Diagram**

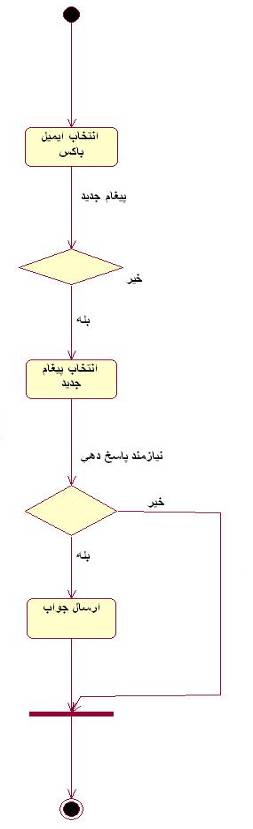
* 1. **نمودار فعالیت 1 – ثبت نام و ورد به سیستم**

****

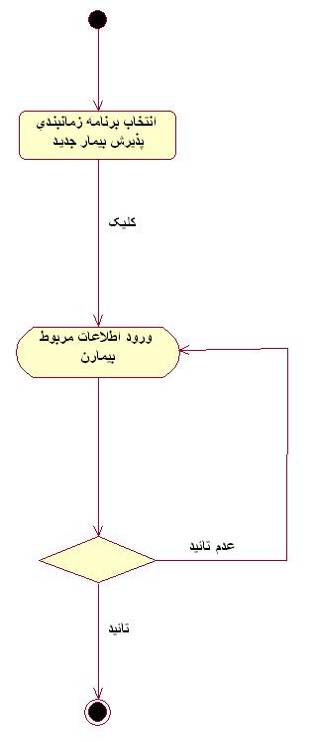
* 1. **نمودار فعالیت 2 – ویرایش اطلاعات بیماران**

****

* 1. **نمودار فعالیت 3 - صندوق پستی**

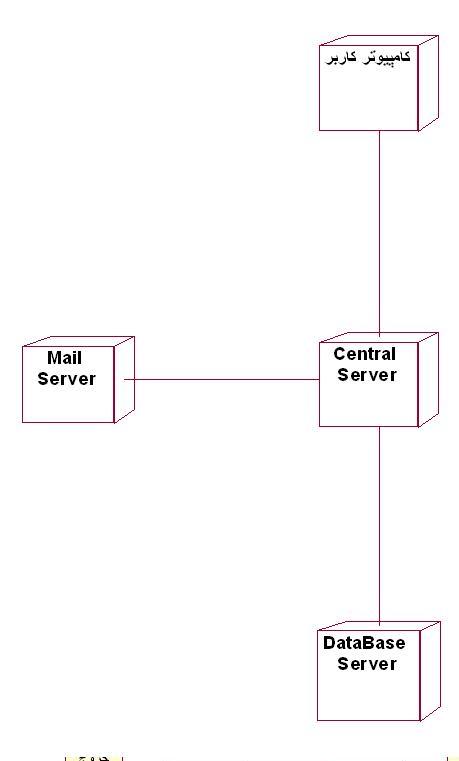
****

* 1. **نمودار فعالیت 4- برنامه زمانبندی پذیرش بیماران**

****

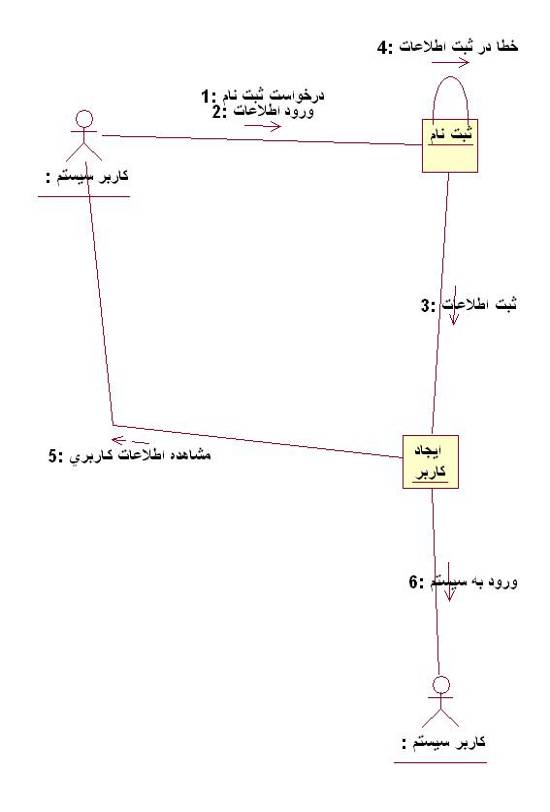
**فصل پنجم: Domain Diagram**

**5-1- نمودار Domain سیستم مطب**

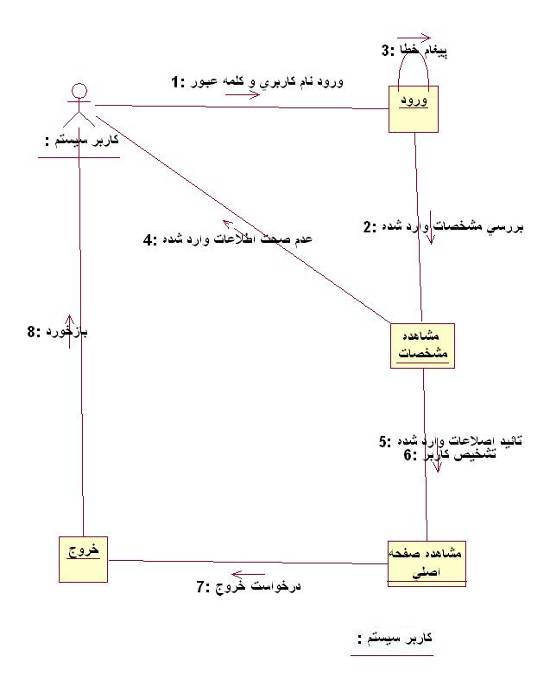
****

**فصل ششم: Collaboration Diagrams**

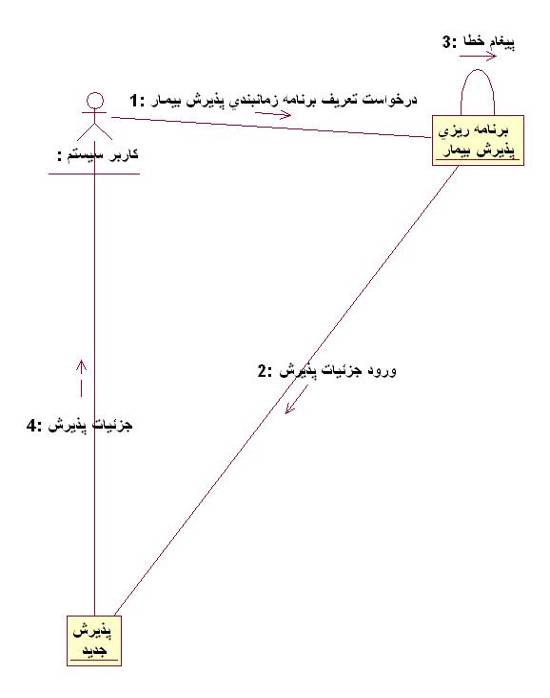
**6-1- نمودار همکاری – ثبت نام کاربر جدید**



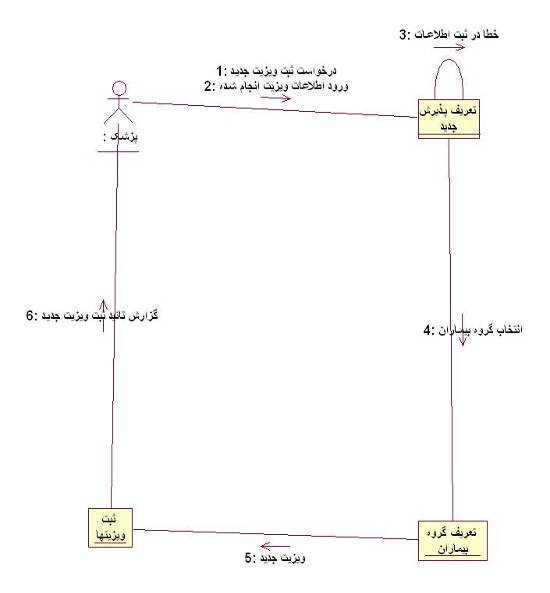
**6-2- نمودار همکاری 2 - ورود به سیستم**



**6-3- نمودار همکاری - عملیات برنامه ریزی پذیرش بیمار جدید**



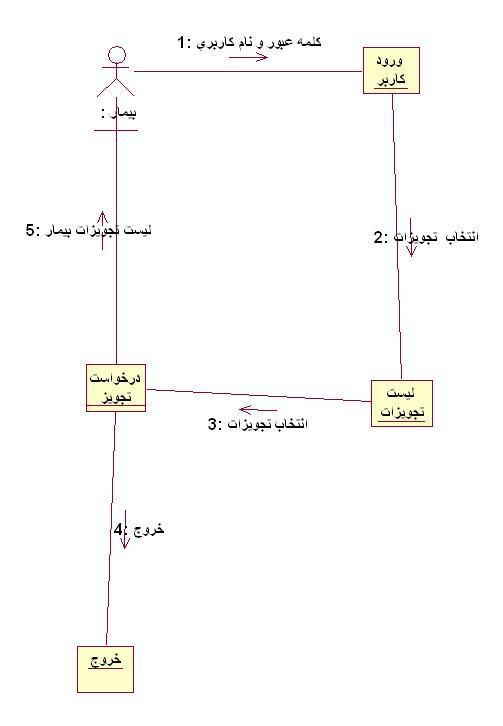
**6-4- نمودار همکاری - عملیات ثبت پذیرش بیمار ویزیت شده**



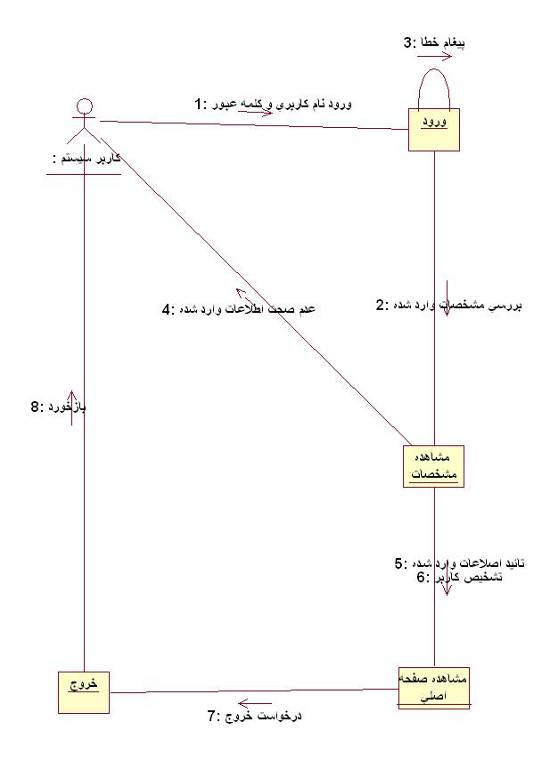
**6-5- نمودار همکاری - عملیات استخدام منشی**



**6-6- نمودار همکاری - عملیات درخواست تجویزات انجام شده**

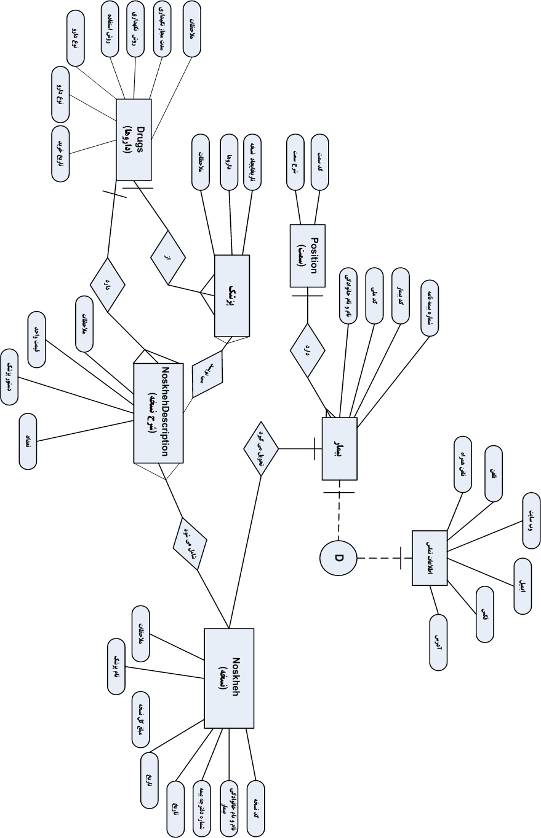


**6-7- نمودار همکاری - عملیات ورود به سیستم**

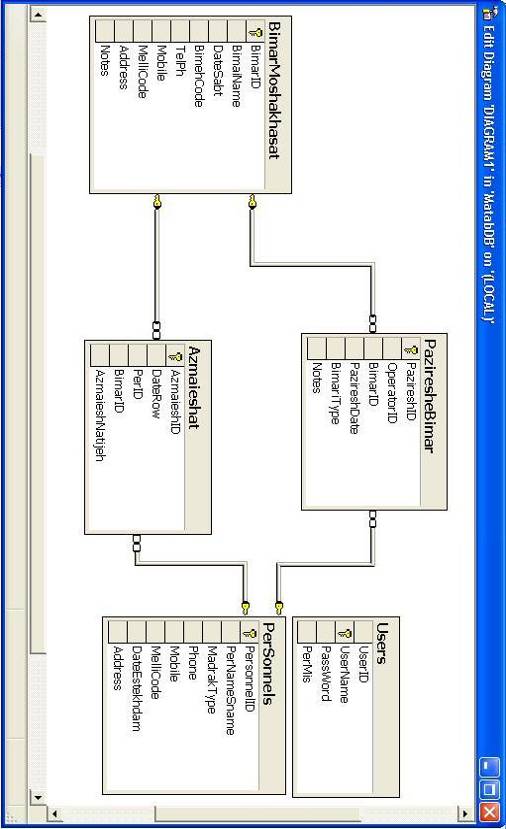
****

**فصل هفتم: Entity Relation Diagram (ER)**

**7-1- نمودار ER منطقی سیستم مطب**

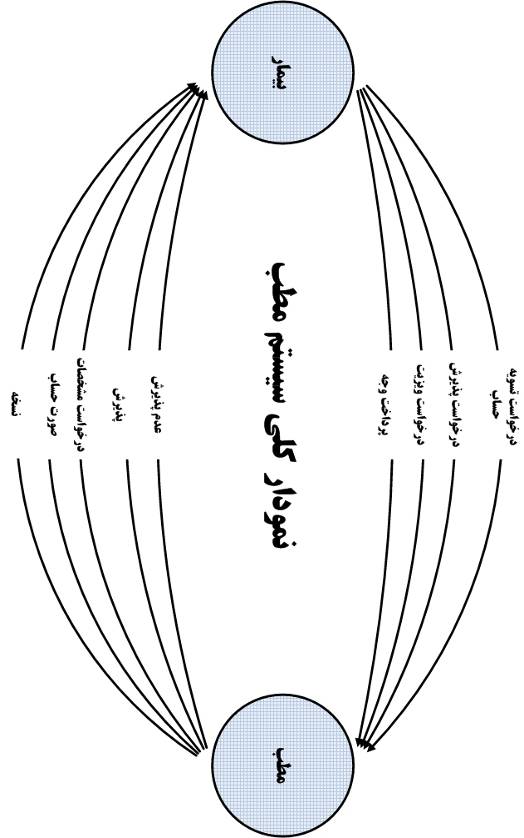
****

**7-2- نمودار ER فیزیکی سیستم مطب**

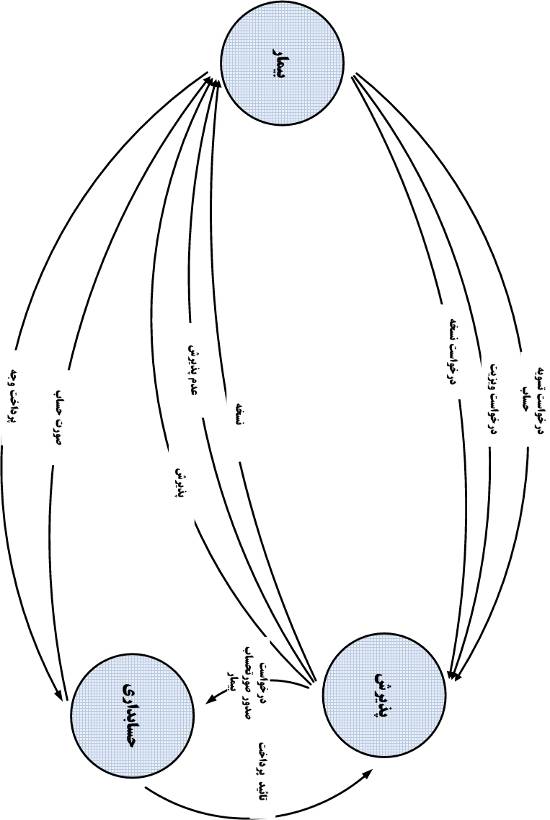
****

**فصل هشتم: Data Flow Diagrams**

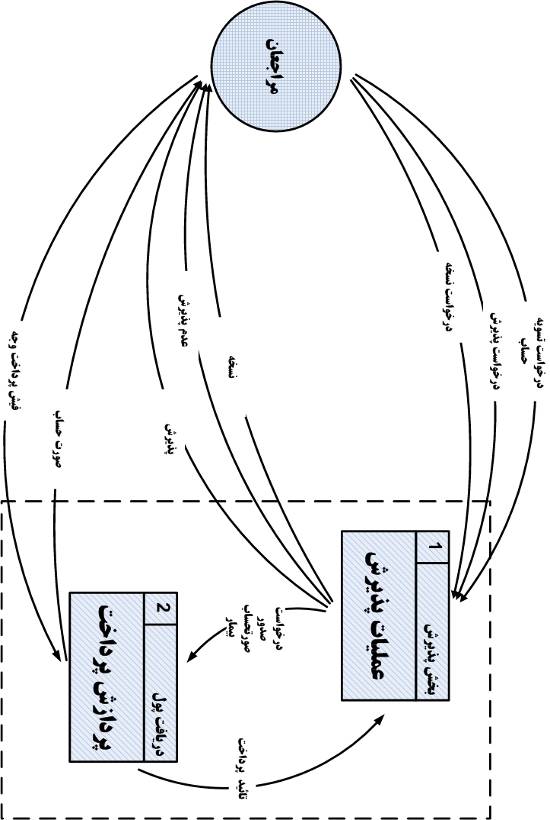
**8-1- نمودار DFD سطح صفر**

****

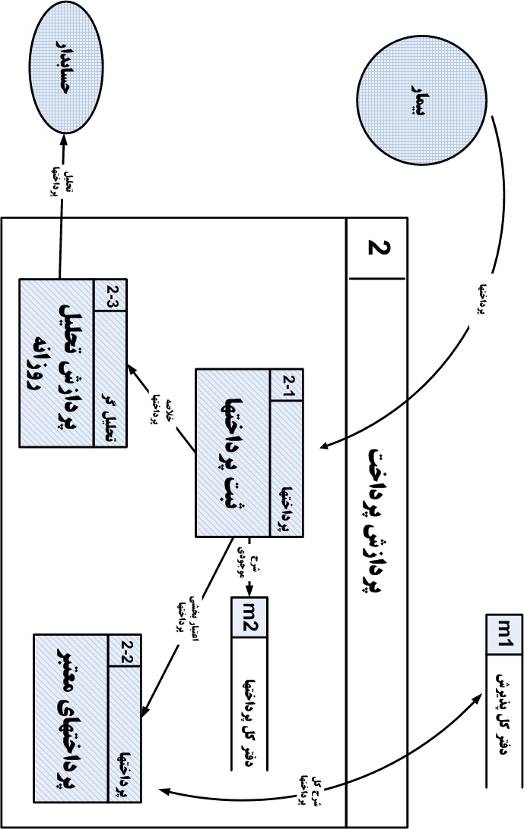
**8-2- نمودار DFD سطح یک**

****

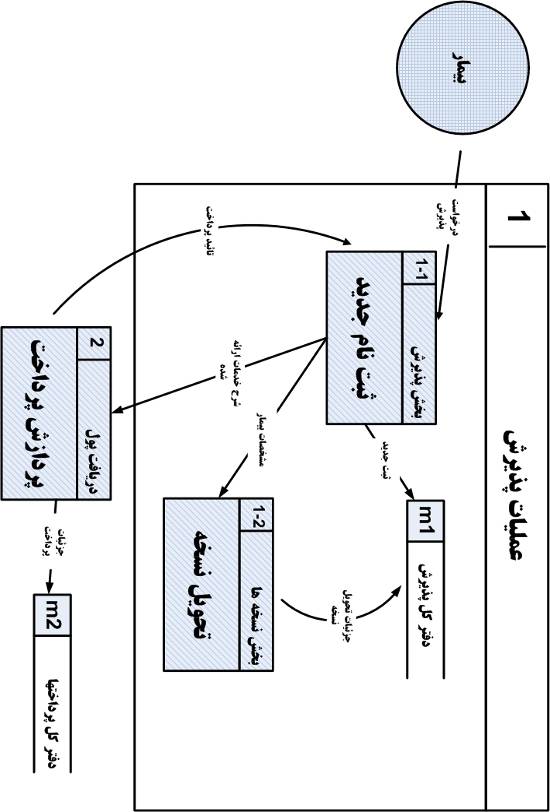
**8-3- نمودار DFD سطح یک (جداسازی مجموعه مورد نظر)**

****

**8-4- نمودار DFD سطح دو (پردازش پرداخت)**

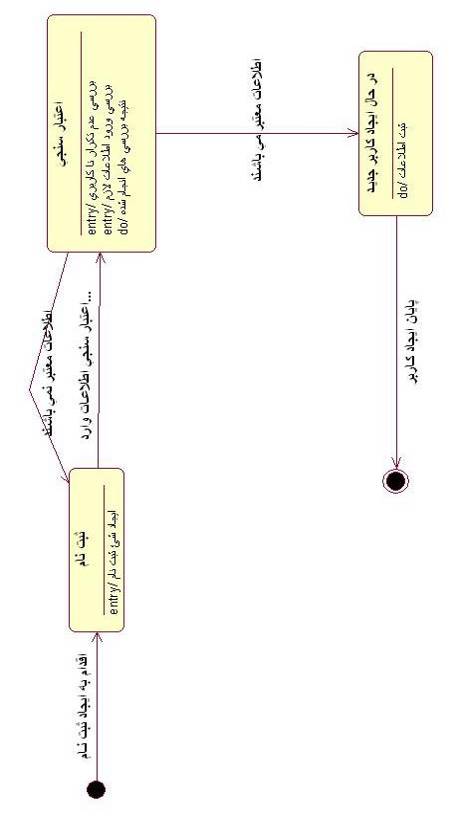
****

**8-5- نمودار DFD سطح دو (پذیرش بیمار)**

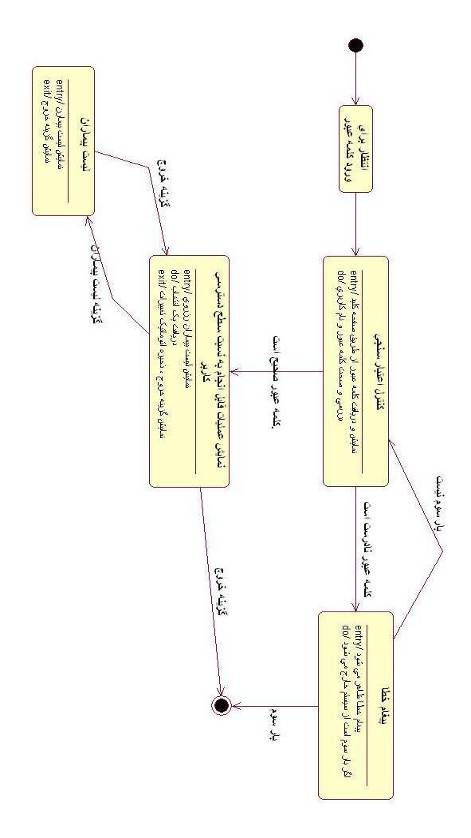
****

**فصل نهم: State Chart Diagrams**

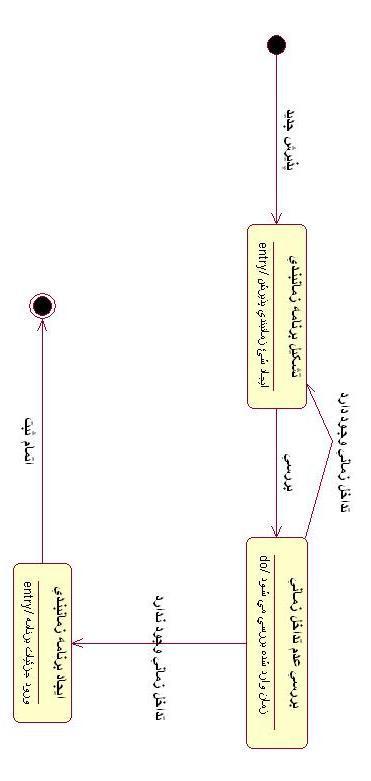
**9-1- نمودار وضعیت – ثبت نام کاربر جدید**



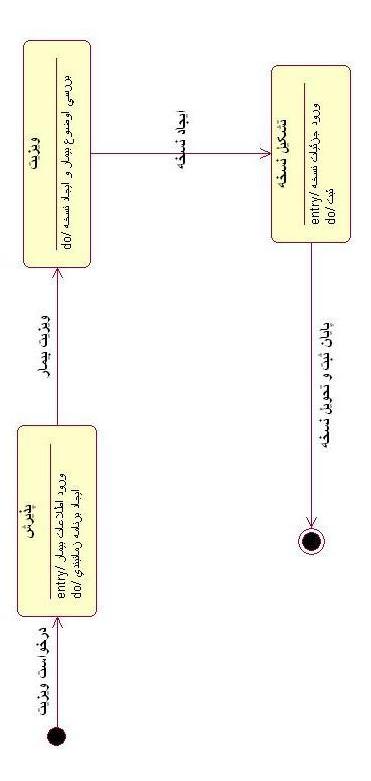
**9-2- نمودار وضعیت - ورود به سیستم**



**9-3- نمودار وضعیت - عملیات برنامه ریزی پذیرش بیمار جدید**



**9-4- نمودار وضعیت - عملیات ثبت پذیرش بیمار ویزیت شده**



**فصل دهم: زبان برنامه نویسی**

C# یکی از زبانهایی است که می توان از آن برای ایجاد برنامه هایی که در .Net CLR اجرا می شوند، استفاده کرد. C# شکل تکامل یافته ای از زبانهای c و c++ است که توسط مایکروسافت، برای کار در پلت فرم .Net، ایجاد شده است.همان طور که دستور زبان c# ساده تر از C++ است، توسعه ی برنامه های کاربردی با استفاده از C# نیز آسانتر از استفاده از C++ است.



در بعضی مواقع، کد C# کمی طولانی تر از c++ است، که این نتیجه ی این است که C# یک زبان type-safe است (بر خلاف c++). به این معنی که زمانی که یک نوع داده ای (Type) به یک داده اختصاص داده می شود، این داده ، نمی تواند خودش را به نوع غیر مرتبط دیگر تبدیل کند.به همین دلیل قواعد محکمی وجود دارد که باید هنگام تبدیل نوع های داده ای به هم، به آن توجه داشت. پس از این نظر، اغلب برای انجام کاری در C# باید کد بیشتری را، نسبت به زمانی که بخواهیم همان کار را در C++ انجام دهیم، بنویسیم. اما فایده ی این کار این است که کد ما توانایی و قابلیت بیشتری دارد و همچنین اشکال زدایی در آن ساده تر و آسانتر صورت می گیرد.

سي‌شارپ همچون زبان برنامه‌نويسي جاوا زباني‌ست شئ‌گرا و سطح بالا (high level). محصول شرکت Microsoft و بر پايه .NET از آنجايي که شئ‌گرائي و سطح بالا بودن از ابزارهاي مديريّت مؤثّر و کارآمد پيچيدگي در فضاي پيچيد? اينترنت مدرن مي باشند، در واقع مي‌شود جاوا و سي‌شارپ را از جمله زبان هاي اصلي براي ايجاد و انجام برنامه‌هاي کاربردي تحت وب (web applications) و خدمات وب دانست.

بر اساس ادّعاي شرکت مايکروسافت، اين زبان در سال 2000 توسط تيمي به سرکردگي آندرس هلزبرگ و نيز سکات ويلتاموث ساخته شد. سي‌شارپ که فقط براي دات نت است در مجموعه NET Platform SDK. ارائه گرديد که در محيط هاي برنامه نويسي استوديوي بصري دات نت (Visual Studio .NET)، در نسخه‌هاي 2003 و 2005 آن موجود است. دستورات زبان سي‌شارپ مانند جاوا سطح بالا تر از C و ++C است و از VB ساده تر. اين زبان همانند پيدايش زبان C انقلابي را در امر برنامه نويسي موجب شد، چراکه به طور همزمان ميتواند امکانات سطح پاين و سطح بالا را به بهترين شکل پشتيباني کند. در طراحي اين زبان براي جلوگيري از پيدايش Bug در زمان اجرا بسيار تلاش شده و اجازه هر کاري (مانند ساخت اشاره گر) را نميدهد(مگر اينکه اصرار بر انجام اين کارها داشته باشيد).

سکّوي دات نت

در ماه ژوئن سال 2000 ميلادي بود که شرکت مايکروسافت ابداع و ايجاد سکّوي (platform) جديد برنامه نويسي خود را موسوم به دات نت (Microsoft .NET) اعلام نمود. در جهت بهبود سکّوهاي پيشين مايکروسافت، دات نت مدل تازه‌اي را براي ايجاد نرم افزار هاي کاربردي ارائه مي‌دهد که در آنجا زبان هاي گوناگون مي توانند در کنار هم قرار گيرند و با يکديگر به همکاري بپردازند. اين درست مدل کوچک‌تري ست از تمامي فضاي اينترنت بدانگونه که عوامل گوناگون ميتوانند در سطوحي گسترده به همکاري اقدام کنند.

انعطاف پذيري سي شارپ

سي شارپ زباني است کاملا شي گرا و بر پايه .NET Framework اين زبان مانند بسياري از زبانهاي شي گراي ديگر از فايل هاي کتابخانه اي Net. استفاده ميکند و همچنين فقط بر روي سيستم عامل Microsoft Windows XP SP2(يا بعد از آن) که .NET Framework بر روي آن نصب باشد اجرا ميشود سي شارپ به کلي براي دات نت طراحي شده و علاوه بر توليد پروژه هاي Console Programs در توليد Windows Application هم بسيار قوي است سي شارپ بر پايه نظريه اوليه سي طراحي شده و همچنين بسياري از صفات خود را از C++ و Java و Delphi به ارث برده است ولي به گفته مايکروسافت بر پايه C براي رقابت با Java و به سادگي VB طراحي شده است. در کل سي شارپ يک زبان مدرنيزه شي گرا ميباشد که در محيط Visual Studio بسيار قدرت يافته است .

منبع اطلاعات در سي شارپ

سي شارپ ميتواند به خوبي با Database ها ارتباط برقرار کرده و آنها را ويرايش و يا از آنها اطلاعات بگيرد اين ويژگي که در Visual Studio .NET به خوبي پشتيباني شده کار را براي کاربر بسيار راحت کرده تا جايي که فقط با چند دستور ساده ميتوان با منبع اطلاعات ارتباط برقرار کرد. در اين زبان از بانک اطلاعات به صورت Connectionless استفاده ميشود. به اين شکل که کل بانک به داخل يک Dataset بر روي حافظه اصلي کپي ميشود و سپس استفاه ميشود.اين روش به منظور کاهش ترافيک شبکه طراحي شده است.

فايلهاي توليدي در سي شارپ

فايلهايي که بيشتر با کاربر سر و کار دارند عبارتند از : فايلهاي با پسوند .Sln : اين فايل سطح بالاي Solution است که براي هر برنامه يک فايل از اين نوع موجود است . هر فايل Solution يک يا چند فايل پروژه را در خود دارد. فايلهاي با پسوند .Csproj: اين يک فايل پروژه C# است .هر فايل پروژه يک يا چند فايل سورس دارد . فايل هاي سورس در يک پروژه بايد به يک زبان برنامه نويسي نوشته شود . فايلهاي با پسوند Cs: اين يک فايل سورس است و کد برنامه در اين فايل نوشته مي شود و اين فايل شامل کدي است که ويژوال استوديو به صورت خود کار ايجاد مي کند . فايل Assemblyinfo.cs: اين يک فايل سورس ديگر است با اين تفاوت که مي توان از اين فايل براي اضافه کردن مواردي مثل اسم نويسنده و تاريخ نوشتن برنامه و امثال آن به برنامه استفاده مي شود. فايل .Ico: اين آيکون برنامه است .آيکوني که در زمان اجراي برنامه در نوار وظيفه قرار مي گيرد.

ابزارهاي ديگر در سي شارپ

سي شارپ نه فقط مخصوص ساخت يک برنامه DOS يا Windows Form است بلکه ميتوان با آن نرم افزارهاي کاربردي تري مانند Movie Collection ها را به صورت سفارشي کمپايل کرد حتي قدرت بسيار بالاي اين زبان در ساخت Screen Saver ها هم بسيار زبان زد است که اين خود نوعي مزيت نسبت به C++ يا ديگر نرم افزارهاي برنامه نويسي به شمار ميرود ولي اين زبان براي اين کارهاي ساده طراحي نشده است. از جمله کارهايي که اين زبان ميتواند انجام دهد طراحي نرم افزار براي SmartPhone,Pocket PC ,Windows CE, Linux است. در ضمن اين زبان براي طراحي وب سايت به زبان ASP.Net نيز به کار برده ميشود. همچينين براي کار با Registry و فايلها مناسبترين زبان است. در زمينه کار با بانکهاي اطلاعاتي و APIها نيز بسيار قوي است.

حرف آخر

اگذر ميخواهيد زبان برنامه نويسي را به طور حرفه اي ادامه دهيد و قبلاً زبان هاي مانند وي بي را كار كرده ايد بهترين گزينه سي شارپ ميباشد.

**ویژگی‌ها**

برخی از تفاوت‌های زبان سی شارپ با C و ++C عبارتند از:

هیچ تابع یا متغیر سراسری(Global) وجود ندارد، تمام متدها و اعضا بایستی در داخل کلاس‌ها تعریف شوند.این امر ممکن است، هر چند برای استفاده از متغیر ها و توابع عمومی باید از متدها و متغیرها در کلاس های عمومی استفاده کرد.

متغیر های عمومی، بر خلاف C و ++C ، نمی توانند بلاک های پیوستی را در بر بگیرند.

سی شارپ دارای یک نوع داده بولی است (bool).برخی از عبارت‌ها مانند while و if که شرطی هستند، نیازمند یک عبارت نوع بولی هستند.همان طور که ++C نیز دارای نوع داده بولی است،این نوع داده به راحتی میتواند به یا از Integerها تبدیل شود، و عبارتی مانند (if(a نیازمند این امر است که a از یک نوع قابل تبدیل به bool یا اشاره گر باشد.کامپایلر سی شارپ برنامه نویس را در این شرایط مجبور به استفاده از عباراتی می‌کند که به درستی یک مقدار bool را برمیگردانند. بنابراین دستوری مانند (if(a = b باعث بروز خطا می‌شوند.(به جای = بایستی از == استفاده شود)

در سی شارپ ، اشاره گرهای به حافظه بایستی فقط در داخل بلوکهای unsafe استفاده شوند و برنامه در این حالت برای اجرا نیاز به اجازه از کاربر دارد.بیشتر دسترسی شی از طریق شی امن است که یا همیشه در حال اشاره به شی صحیح موجود است یا یک مقدار Null دارد. اشاره گری به شی به درد نخور یا بلاک حافظه رندم غیر ممکن است.اشاره گر نا امن می تواند به نمونه ای از value-type ، آرایه، رشته یا بلاکی که حافظه به آن داده شده است اشاره نماید.کدی که به عنوان نا امن علامت نخورده باشد، هنوز می تواند اشاره گر ها را از سیستم بازیابی یا در آن ذخیره کند ولی نمی تواند مرجع جدیدی به آنها اختصاص دهد.

حافظه ساماندهی شده نمی تواند صریحا آزاد شود، ولی به طور خودکار به عنوان به درد نخور تلقی می شود.انتخاب آدرس های به درد نخور حافظه نفوذ ناپذیر است. هم چنین #C با استفاده از عبارات، پشتیبانی مستقیمی از پایان اجباری می کند(پشتیبانی از اصطلاح Resource Acquisition Is Initialization).

وراثت چندگانه از کلاس‌ها در این زبان پشتیبانی نمی‌شود.البته یک کلاس امکان ارث بری از تعداد نامحدود واسط‌ها را دارد.پشتیبانی نکردن از وراثت چندگانه به دلیل اهداف معماری این زبان در CLI و برای جلوگیری از پیچیدگی است.

سی شارپ بسیار typesafe تر از C++ است. تنها تبدیلات ضمنی مثل تبدیل نوع داده کوچکتر به بزرگتر یا تبدیل نوع مشتق شده به نوع پایه به طور پیش فرض و بدون خطا صورت می‌پذیرد.هیچ تبدیل ضمنی ای میانBooleanها و Integerها وجود ندارد و هر تبدیل user-defined بایستی به صراحت با یکی از کلمات explicit یا implicit نشانه گذاری شود. تبدیل b به a در حالتی که a یک Integer و b یک double باشد در زبان C++ مجاز است اما در سی شارپ به یک خطای زمان کامپایل منجر می‌شود(بایستی به صورت explicit تعریف شود)

اعضای Enumeration در داخل محدوده شخصی خود قرار دارند. #C قابلیت syntactic sugar را برای توابع متداول، اکسسور ها و ماجول های کسول شده در یک کلاس به صورت ویژگی ها قرار داده است.

اکسسورها که خاصیت نیز گفته می‌شوند در زبان سی شارپ قادر به کنترل دسترسی اعضا و معتبرسازی داده‌ها هستند.

تمام انواع بازتابی(Reflection) و بازیابی(Recovery) قابل استفاده‌است. در حال حاظر (3 جوان 2008) دارای 77 کلمه رزرو شده است.

**سیستم یکپارچه شده**

سی شارپ دارای یک سیستم نوع یکپارچه‌است که به آن CTS می گویند.این بدان معناست که تمام انواع،شامل موارد اصلی مانند Integerها، مشتق شده از System.Object هستند.به عنوان مثال، هر نوع یک متد به نام ToString() را به ارث می‌برد.بخاطر کارائی،انواع اولیه (و انواع مقداری) به طور داخلی فضایی برای آنها بر روی پشته در نظر گرفته می‌شود.

**انواع داده**

CTS داده ها را به دو نوع تقسیم می کند: نوع مقداری ، نوع مرجعی

انواع داده ای توده ساده ای از داده می باشند.نمونه های انواع داده ای نه حویت مرجعی دارند و نه مفاهیم مقایسه مراجع را. برای مقایسه برابری یا عدم برابری انواع داده ای، خود مقدار داده ها را با یکدیگر مقایسه می کنیم مگر اینکه عملگر های مشابه دوباره تعریف شده باشند.مقادیر داده های مرجعی همیشه یک مقدار پیش فرض دارند و همیشه می توانند ایجاد و یا کپی شوند.یکی دیگر از محدودیت های انواع داده ای این ات که آنها نمی توانند از یکدیگر مشتق شوند(ولی میتوانند اشتراکاتی داشته باشند) و هم چنین نمی توانند در سازنده مقدار دهی اولیه شوند. مثالی از انواع داده ای، بعضی از انواع اولیه مانند int و float و char و System.DateTime می باشند. در مقابل، انواع مرجعی مفهوم تعریف مرجعی را دارند( که در آن هر نمونه از نوع مرجع، به طور ذاتی از دیگر نمونه ها جدا می شود، حتی اگر داده هر دو نمونه یکی باشد). این دقیقا نمونه مشابه مقایسه تساوی یا عدم تساوی داده های مرجعی است، که در آن آزمایش برای مرجع ها از داده ای ها سریع تر است. در کل نه همیشه امکان تعریف نمونه مرجعی وجود دارد و نه امکان کپی یا نمایش مقادیر مقایسه دو نمونه.ولی به هر حال انواع مرجعی خاص می توانند این اعمال را از طریق سازنده های عمومی یا اجرای واسط های مشابه(مثل ICloneable یا IComparable) انجام دهند.نمونه هایی از انواع مرجعی، اشیا، System.Stringو Sysmet.Array می باشند. هر دو نوع داده قابلیت **انعطاف** توسط تعریف به وسیله کاربر را دارند.

## Boxing و EnBoxing

Boxing عمل تبدیل مقدار نوع داده ای به نوع مرجع مشابه آن می باشد.

مثال:

int foo = 42; *// Value type...*

object bar = foo; *// foo is boxed to bar.*

UnBoxing عمل تبدیل نوع مرجع به نوع داده ای می باشد. مثال:

int foo = 42; *// Value type.*

object bar = foo; *// foo is boxed to bar.*

int foo2 = (int)bar; *// Unboxed back to value type.*

سی شارپ به [برنامه نویس](http://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87_%D9%86%D9%88%DB%8C%D8%B3) با استفاده از کلمه کلیدی Struct اجازه می‌دهد تا انواع مقداری User-defined را ایجاد کند. از دیدگاه برنامه نویسی، آنها کلاس‌های سبک وزن به نظر می‌رسند. برخلاف کلاس‌ها (که بر روی heap قرار میگیرند) و شبیه به انواع اولیه استاندارد مانند انواع مقداری Structها نیز بر روی پشته قرار می‌گیرند.آنها همچنین می‌توانند قسمتی از یک شئ باشند، یا در یک آرایه مرتب شوند، بدون حافظه غیر مستقیمی که به طور معمول برای انواع کلاس تخصیص می‌یابد.

**فصل یازدهم: Source Code های سیستم مطب**

توضیحات کلاس FirstClass

using System;

using System.Collections.Generic;

using System.Text;

using System.Data;

using System.Data.SqlClient;

فایلهای سرآیند

namespace MatabSystemCS

{

class FirstClass

{

SqlConnection con;

public SqlCommand cmd;

DataTable dt;

SqlDataAdapter da;

public FirstClass()

{

con = new SqlConnection("server = (local); Database = MatabDB; trusted\_connection = yes;");

cmd = new SqlCommand();

dt = new DataTable();

cmd.Connection = con;

da = new SqlDataAdapter(cmd);

}

public DataTable dbOut(string sql)

{

cmd.CommandText = sql;

con.Open();

da.Fill(dt);

con.Close();

return dt;

}

public DataTable SP\_dbOut(string SP\_Name)

{

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = SP\_Name;

con.Open();

da.Fill(dt);

con.Close();

return dt;

}

public void exeCommand(string strName)

{

con.Open();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = strName;

cmd.ExecuteNonQuery();

con.Close();

}

public void exeCommand1(string strName)

{

try

{

cmd.CommandText = strName;

con.Open();

cmd.ExecuteNonQuery();

}

catch { }

finally

{

con.Close();

}

}

}

}



namespace MatabSystemCS.TagiirateSystemiForms

{

public partial class frmTarifekarbaran : Form

{

public frmTarifekarbaran()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox2.Text == "" || comboBox1.Text == "")

{

MessageBox.Show("تمامی مقادیر باید انتخاب شوند");

}

else

{

FirstClass dbCheck = new FirstClass();

DataTable dtChek = new DataTable();

dtChek = dbCheck.dbOut("SELECT TOP 100 PERCENT UserName FROM Users WHERE (UserName = '" + textBox1.Text.Trim() + "') ORDER BY UserName");

if (dtChek.Rows.Count <= 0)

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@UserName", SqlDbType.NVarChar).Value = textBox1.Text.Trim();

db.cmd.Parameters.Add("@PassWord", SqlDbType.NVarChar).Value = textBox2.Text.Trim();

db.cmd.Parameters.Add("@PerMis", SqlDbType.SmallInt).Value = Convert.ToUInt16(comboBox1.Text.Trim());

db.exeCommand("sp\_Users\_Insert");

MessageBox.Show("نام کاربری جدید ثبت شد");

grdFill();

}

else

{

MessageBox.Show("این نام کاربری قبلا ثبت شده است");

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" )

{

MessageBox.Show("لطفا نام کاربری را وارد نمائید");

}

else

{

FirstClass dbCheck = new FirstClass();

DataTable dtChek = new DataTable();

dtChek = dbCheck.dbOut("SELECT TOP 100 PERCENT UserName FROM Users WHERE (UserName = '" + textBox1.Text.Trim() + "') ORDER BY UserName");

if (dtChek.Rows.Count <= 0)

{

MessageBox.Show("نام کاربری قبلا ثبت نشده");

}

else

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@UserName", SqlDbType.NVarChar).Value = textBox1.Text.Trim();

db.exeCommand("sp\_Users\_DeleteRow");

MessageBox.Show("رکورد مورد نظر حذف گردید");

grdFill();

}

}

}

private void frmTarifekarbaran\_Load(object sender, EventArgs e)

{

grdFill();

}

private void grdFill()

{

FirstClass db = new FirstClass();

DataTable dt = new DataTable();

dt = db.dbOut(@"SELECT TOP 100 PERCENT UserName AS [نام کاربری], PassWord AS [کلمه عبور], PerMis AS [سطح دسترسی] FROM Users ORDER BY UserName ");

dataGridView1.DataSource = dt;

}

}



namespace MatabSystemCS.ParvandeForms

{

public partial class frmPaziresheBimar : Form

{

public frmPaziresheBimar()

{

InitializeComponent();

}

private void frmPaziresheBimar\_Load(object sender, EventArgs e)

{

grdFill();

cmb1Fill();

cmb2Fill();

}

private void cmb1Fill()

{

FirstClass db = new FirstClass();

comboBox1.DataSource = db.dbOut("SELECT TOP 100 PERCENT PersonnelID, PerNameSname FROM PerSonnels ORDER BY PersonnelID");

comboBox1.ValueMember = "PersonnelID";

comboBox1.DisplayMember = "PerNameSname";

}

private void cmb2Fill()

{

FirstClass db = new FirstClass();

comboBox2.DataSource = db.dbOut("SELECT TOP 100 PERCENT BimarID, BimaiName FROM BimarMoshakhasat ORDER BY BimarID DESC");

comboBox2.ValueMember = "BimarID";

comboBox2.DisplayMember = "BimaiName";

}

private void grdFill()

{

FirstClass db = new FirstClass();

DataTable dt = new DataTable();

dt = db.dbOut(@"SELECT PaziresheBimar.PazireshID AS [کد پذیرش], PerSonnels.PerNameSname AS [نام اپراتور], BimarMoshakhasat.BimaiName AS [نام بیمار],

PaziresheBimar.PazireshDate AS [تاریخ پذیرش], PaziresheBimar.BimariType AS [نوع بیماری], PaziresheBimar.Notes AS ملاحظات

FROM BimarMoshakhasat INNER JOIN PaziresheBimar ON BimarMoshakhasat.BimarID = PaziresheBimar.BimarID INNER JOIN PerSonnels ON PaziresheBimar.OperatorID = PerSonnels.PersonnelID");

dataGridView1.DataSource = dt;

}

private void dbIns()

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@OperatorID", SqlDbType.Int).Value = comboBox1.SelectedValue;

db.cmd.Parameters.Add("@BimarID", SqlDbType.Int).Value = comboBox2.SelectedValue;

db.cmd.Parameters.Add("@PazireshDate", SqlDbType.NVarChar).Value = textBox1.Text.Trim();

db.cmd.Parameters.Add("@BimariType", SqlDbType.NVarChar).Value = textBox2.Text.Trim();

db.cmd.Parameters.Add("@Notes", SqlDbType.NVarChar).Value = textBox3.Text.Trim();

db.exeCommand("sp\_PaziresheBimar\_Insert");

}

private void btnReg\_Click(object sender, EventArgs e)

{

txtNulCheck();

dbIns();

grdFill();

txtClear();

}

private void txtNulCheck()

{

if (textBox1.Text == "")

{

textBox1.Text = "Null";

}

if (textBox2.Text == "")

{

textBox2.Text = "Null";

}

if (textBox3.Text == "")

{

textBox3.Text = "Null";

}

}

private void txtClear()

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox9.Text == "")

{

MessageBox.Show("کد پذیرش وارد نشده");

}

else

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@PazireshID", SqlDbType.Int).Value = textBox9.Text.Trim();

db.exeCommand("sp\_PaziresheBimar\_DeleteRow");

textBox9.Text = "";

MessageBox.Show("مشخصات مورد نظر حذف شد");

grdFill();

txtClear();

}

}

}

}



namespace MatabSystemCS.ParvandeForms

{

public partial class frmSabtepersonnel : Form

{

public frmSabtepersonnel()

{

InitializeComponent();

}

private void frmSabtepersonnel\_Load(object sender, EventArgs e)

{

grdFill();

}

private void grdFill()

{

FirstClass db = new FirstClass();

DataTable dt = new DataTable();

dt = db.dbOut(@"SELECT TOP 100 PERCENT PersonnelID AS [کد پرسنلی], PerNameSname AS [نام و نام خانوادگی], MadrakType AS [مدرک تحصیلی], Phone AS تلفن,

Mobile AS موبایل, MelliCode AS [کد ملی], DateEstekhdam AS [تاریخ استخدام], Address AS آدرس FROM dbo.PerSonnels ORDER BY PersonnelID");

dataGridView1.DataSource = dt;

}

private void dbIns()

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@PerNameSname", SqlDbType.NVarChar).Value = textBox1.Text.Trim();

db.cmd.Parameters.Add("@MadrakType", SqlDbType.NVarChar).Value = textBox2.Text.Trim();

db.cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = textBox3.Text.Trim();

db.cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = textBox4.Text.Trim();

db.cmd.Parameters.Add("@MelliCode", SqlDbType.NVarChar).Value = textBox5.Text.Trim();

db.cmd.Parameters.Add("@DateEstekhdam", SqlDbType.NVarChar).Value = textBox6.Text.Trim();

db.cmd.Parameters.Add("@Address", SqlDbType.NVarChar).Value = textBox7.Text.Trim();

db.exeCommand("sp\_PerSonnels\_Insert");

}

private void txtNulCheck()

{

if (textBox3.Text == "")

{

textBox3.Text = "Null";

}

if (textBox4.Text == "")

{

textBox4.Text = "Null";

}

if (textBox5.Text == "")

{

textBox5.Text = "Null";

}

if (textBox6.Text == "")

{

textBox6.Text = "Null";

}

if (textBox7.Text == "")

{

textBox7.Text = "Null";

}

}

private void txtClear()

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

textBox7.Text = "";

}

private void btnReg\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox2.Text == "")

{

MessageBox.Show("فیلد نام یا مدرک تحصیلی باید وارد شوند");

}

else

{

txtNulCheck();

dbIns();

grdFill();

txtClear();

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox9.Text == "")

{

MessageBox.Show("کد پرسنلی وارد نشده");

}

else

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@PersonnelID", SqlDbType.Int).Value = textBox9.Text.Trim();

db.exeCommand("sp\_PerSonnels\_DeleteRow");

textBox9.Text = "";

MessageBox.Show("مشخصات مورد نظر حذف شد");

grdFill();

txtClear();

}

}

}

}

****

namespace MatabSystemCS.ParvandeForms

{

public partial class frmSabteazmaieshat : Form

{

public frmSabteazmaieshat()

{

InitializeComponent();

}

private void frmSabteazmaieshat\_Load(object sender, EventArgs e)

{

grdFill();

cmb1Fill();

cmb2Fill();

}

private void cmb1Fill()

{

FirstClass db = new FirstClass();

comboBox1.DataSource = db.dbOut("SELECT TOP 100 PERCENT PersonnelID, PerNameSname FROM PerSonnels ORDER BY PersonnelID");

comboBox1.ValueMember = "PersonnelID";

comboBox1.DisplayMember = "PerNameSname";

}

private void cmb2Fill()

{

FirstClass db = new FirstClass();

comboBox2.DataSource = db.dbOut("SELECT TOP 100 PERCENT BimarID, BimaiName FROM BimarMoshakhasat ORDER BY BimarID DESC");

comboBox2.ValueMember = "BimarID";

comboBox2.DisplayMember = "BimaiName";

}

private void grdFill()

{

FirstClass db = new FirstClass();

DataTable dt = new DataTable();

dt = db.dbOut(@"SELECT TOP 100 PERCENT dbo.Azmaieshat.AzmaieshID AS [کد معاینات], dbo.BimarMoshakhasat.BimaiName AS [نام بیمار],

dbo.PerSonnels.PerNameSname AS [نام اپراتور], dbo.Azmaieshat.AzmaieshNatijeh AS [تجویزات و آزمایشات]

FROM dbo.Azmaieshat INNER JOIN dbo.BimarMoshakhasat ON dbo.Azmaieshat.BimarID = dbo.BimarMoshakhasat.BimarID INNER JOIN

dbo.PerSonnels ON dbo.Azmaieshat.PerID = dbo.PerSonnels.PersonnelID ORDER BY dbo.Azmaieshat.AzmaieshID");

dataGridView1.DataSource = dt;

}

private void dbIns()

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@PerID", SqlDbType.Int).Value = comboBox1.SelectedValue;

db.cmd.Parameters.Add("@BimarID", SqlDbType.Int).Value = comboBox2.SelectedValue;

db.cmd.Parameters.Add("@DateRow", SqlDbType.NVarChar).Value = textBox1.Text.Trim();

db.cmd.Parameters.Add("@AzmaieshNatijeh", SqlDbType.NVarChar).Value = textBox2.Text.Trim();

db.exeCommand("sp\_Azmaieshat\_Insert");

}

private void btnReg\_Click(object sender, EventArgs e)

{

txtNulCheck();

dbIns();

grdFill();

txtClear();

}

private void txtNulCheck()

{

if (textBox1.Text == "")

{

textBox1.Text = "Null";

}

if (textBox2.Text == "")

{

textBox2.Text = "Null";

}

}

private void txtClear()

{

textBox1.Text = "";

textBox2.Text = "";

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox9.Text == "")

{

MessageBox.Show("کد پذیرش وارد نشده");

}

else

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@AzmaieshID", SqlDbType.Int).Value = textBox9.Text.Trim();

db.exeCommand("sp\_Azmaieshat\_DeleteRow");

textBox9.Text = "";

MessageBox.Show("مشخصات مورد نظر حذف شد");

grdFill();

txtClear();

}

}

}

}

****

namespace MatabSystemCS.ParvandeForms

{

public partial class frmSabtenameBimar : Form

{

public frmSabtenameBimar()

{

InitializeComponent();

}

private void frmSabtenameBimar\_Load(object sender, EventArgs e)

{

grdFill();

}

private void grdFill()

{

FirstClass db = new FirstClass();

DataTable dt = new DataTable();

dt = db.dbOut(@"SELECT TOP 100 PERCENT BimarID AS [کد بیمار], BimaiName AS [نام و نام خانوادگی], DateSabt AS [تاریخ ثبت نام], BimehCode AS [شماره دفترچه بیمه],

TelPh AS تلفن, Mobile AS موبایل, MelliCode AS [کد ملی], Address AS آدرس, Notes AS ملاحظات FROM dbo.BimarMoshakhasat ORDER BY BimarID");

dataGridView1.DataSource = dt;

//dataGridView1.DataSource = db.dbOut(@"SELECT TOP 100 PERCENT BimarID AS [کد بیمار], BimaiName AS [نام و نام خانوادگی], DateSabt AS [تاریخ ثبت نام], BimehCode AS [شماره دفترچه بیمه],

// TelPh AS تلفن, Mobile AS موبایل, MelliCode AS [کد ملی], Address AS آدرس, Notes AS ملاحظات FROM dbo.BimarMoshakhasat ORDER BY BimarID");

}

private void dbIns()

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@BimaiName", SqlDbType.NVarChar).Value = textBox1.Text.Trim();

db.cmd.Parameters.Add("@DateSabt", SqlDbType.NVarChar).Value = textBox2.Text.Trim();

db.cmd.Parameters.Add("@BimehCode", SqlDbType.Int).Value = textBox3.Text.Trim();

db.cmd.Parameters.Add("@TelPh", SqlDbType.NVarChar).Value = textBox4.Text.Trim();

db.cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = textBox5.Text.Trim();

db.cmd.Parameters.Add("@MelliCode", SqlDbType.NVarChar).Value = textBox6.Text.Trim();

db.cmd.Parameters.Add("@Address", SqlDbType.NVarChar).Value = textBox7.Text.Trim();

db.cmd.Parameters.Add("@Notes", SqlDbType.NVarChar).Value = textBox8.Text.Trim();

db.exeCommand("sp\_BimarMoshakhasat\_Insert");

}

private void txtNulCheck()

{

if (textBox3.Text == "")

{

textBox3.Text = "0";

}

if (textBox4.Text == "")

{

textBox4.Text = "Null";

}

if (textBox5.Text == "")

{

textBox5.Text = "Null";

}

if (textBox6.Text == "")

{

textBox6.Text = "Null";

}

if (textBox7.Text == "")

{

textBox7.Text = "Null";

}

if (textBox8.Text == "")

{

textBox8.Text = "Null";

}

}

private void txtClear()

{

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

textBox5.Text = "";

textBox6.Text = "";

textBox7.Text = "";

textBox8.Text = "";

}

private void btnReg\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox2.Text == "")

{

MessageBox.Show("فیلد نام یا تاریخ ثبت نام باید وارد شوند");

}

else

{

txtNulCheck();

dbIns();

grdFill();

txtClear();

}

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox9.Text == "")

{

MessageBox.Show("کد بیمار وارد نشده");

}

else

{

FirstClass db = new FirstClass();

db.cmd.Parameters.Add("@BimarID", SqlDbType.Int).Value = textBox9.Text.Trim();

db.exeCommand("sp\_BimarMoshakhasat\_DeleteRow");

textBox9.Text = "";

MessageBox.Show("مشخصات مورد نظر حذف شد");

grdFill();

txtClear();

}

}

}

}

****

namespace MatabSystemCS

{

public partial class MatabMDI : Form

{

private int childFormNumber = 0;

public MatabMDI()

{

InitializeComponent();

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

this.Close();

}

private void newToolStripMenuItem\_Click(object sender, EventArgs e)

{

ParvandeForms.frmSabtenameBimar frm = new MatabSystemCS.ParvandeForms.frmSabtenameBimar();

frm.MdiParent = this;

frm.Show();

}

private void openToolStripMenuItem\_Click(object sender, EventArgs e)

{

ParvandeForms.frmSabtepersonnel frm = new MatabSystemCS.ParvandeForms.frmSabtepersonnel();

frm.MdiParent = this;

frm.Show();

}

private void selectAllToolStripMenuItem\_Click(object sender, EventArgs e)

{

TagiirateSystemiForms.frmTarifekarbaran frm = new MatabSystemCS.TagiirateSystemiForms.frmTarifekarbaran();

frm.MdiParent = this;

frm.Show();

}

private void optionsToolStripMenuItem\_Click(object sender, EventArgs e)

{

MessageBox.Show("سیستم مطب - نسخه 1.1 - زبان برنامه نویسی : سی شارپ - پایگاه داده اس کیو ال");

}

private void saveAsToolStripMenuItem\_Click(object sender, EventArgs e)

{

ParvandeForms.frmPaziresheBimar frm = new MatabSystemCS.ParvandeForms.frmPaziresheBimar();

frm.MdiParent = this;

frm.Show();

}

private void printSetupToolStripMenuItem\_Click(object sender, EventArgs e)

{

ParvandeForms.frmSabteazmaieshat frm = new MatabSystemCS.ParvandeForms.frmSabteazmaieshat();

frm.MdiParent = this;

frm.Show();

}

private void toolBarToolStripMenuItem\_Click(object sender, EventArgs e)

{

GozareshatForms.frmRpt1 frm = new MatabSystemCS.GozareshatForms.frmRpt1();

frm.MdiParent = this;

frm.Show();

}

private void statusBarToolStripMenuItem\_Click(object sender, EventArgs e)

{

GozareshatForms.frmRpt2 frm = new MatabSystemCS.GozareshatForms.frmRpt2();

frm.MdiParent = this;

frm.Show();

}

private void MatabMDI\_Load(object sender, EventArgs e)

{

}

private void MatabMDI\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

}

}

**فصل دوازدهم: SQL Scripts**

CREATE TABLE [dbo].[Azmaieshat] (

[AzmaieshID] [int] IDENTITY (1, 1) NOT NULL ,

[DateRow] [nvarchar] (10) COLLATE Arabic\_CI\_AS NULL ,

[PerID] [int] NULL ,

[BimarID] [int] NULL ,

[AzmaieshNatijeh] [nvarchar] (300) COLLATE Arabic\_CI\_AS NULL

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[BimarMoshakhasat] (

[BimarID] [int] IDENTITY (1, 1) NOT NULL ,

[BimaiName] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[DateSabt] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[BimehCode] [int] NULL ,

[TelPh] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[Mobile] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[MelliCode] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[Address] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[Notes] [ntext] COLLATE Arabic\_CI\_AS NULL

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

CREATE TABLE [dbo].[PaziresheBimar] (

[PazireshID] [int] IDENTITY (1, 1) NOT NULL ,

[OperatorID] [int] NULL ,

[BimarID] [int] NULL ,

[PazireshDate] [nvarchar] (10) COLLATE Arabic\_CI\_AS NULL ,

[BimariType] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[Notes] [nvarchar] (350) COLLATE Arabic\_CI\_AS NULL

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PerSonnels] (

[PersonnelID] [int] IDENTITY (1, 1) NOT NULL ,

[PerNameSname] [nvarchar] (80) COLLATE Arabic\_CI\_AS NULL ,

[MadrakType] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[Phone] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[Mobile] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[MelliCode] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[DateEstekhdam] [nvarchar] (10) COLLATE Arabic\_CI\_AS NULL ,

[Address] [nvarchar] (350) COLLATE Arabic\_CI\_AS NULL

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[Users] (

[UserID] [int] IDENTITY (1, 1) NOT NULL ,

[UserName] [nvarchar] (50) COLLATE Arabic\_CI\_AS NOT NULL ,

[PassWord] [nvarchar] (50) COLLATE Arabic\_CI\_AS NULL ,

[PerMis] [smallint] NULL

) ON [PRIMARY]

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Azmaieshat\_DeleteRow

-- Create date: 2009/06/09 02:46:57 ق.ظ

-- Description: This stored procedure is intended for deleting a specific row from Azmaieshat table

-- ==========================================================================================

Create Procedure sp\_Azmaieshat\_DeleteRow

@AzmaieshID int

As

Begin

Delete Azmaieshat

Where

AzmaieshID = @AzmaieshID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Azmaieshat\_Insert

-- Create date: 2009/06/09 02:46:57 ق.ظ

-- Description: This stored procedure is intended for inserting values to Azmaieshat table

-- ==========================================================================================

Create Procedure sp\_Azmaieshat\_Insert

@DateRow nvarchar(10),

@PerID int,

@BimarID int,

@AzmaieshNatijeh nvarchar(300)

As

Begin

Insert Into Azmaieshat

(DateRow,PerID,BimarID,AzmaieshNatijeh)

Values

(@DateRow,@PerID,@BimarID,@AzmaieshNatijeh)

Declare @ReferenceID int

Select @ReferenceID = @@IDENTITY

Return @ReferenceID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Azmaieshat\_SelectAll

-- Create date: 2009/06/09 02:46:57 ق.ظ

-- Description: This stored procedure is intended for selecting all rows from Azmaieshat table

-- ==========================================================================================

Create Procedure sp\_Azmaieshat\_SelectAll

As

Begin

Select

AzmaieshID,

DateRow,

PerID,

BimarID,

AzmaieshNatijeh

From Azmaieshat

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Azmaieshat\_SelectRow

-- Create date: 2009/06/09 02:46:57 ق.ظ

-- Description: This stored procedure is intended for selecting a specific row from Azmaieshat table

-- ==========================================================================================

Create Procedure sp\_Azmaieshat\_SelectRow

@AzmaieshID int

As

Begin

Select

AzmaieshID,

DateRow,

PerID,

BimarID,

AzmaieshNatijeh

From Azmaieshat

Where

AzmaieshID = @AzmaieshID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Azmaieshat\_Update

-- Create date: 2009/06/09 02:46:57 ق.ظ

-- Description: This stored procedure is intended for updating Azmaieshat table

-- ==========================================================================================

Create Procedure sp\_Azmaieshat\_Update

@AzmaieshID int,

@DateRow nvarchar(10),

@PerID int,

@BimarID int,

@AzmaieshNatijeh nvarchar(300)

As

Begin

Update Azmaieshat

Set

DateRow = @DateRow,

PerID = @PerID,

BimarID = @BimarID,

AzmaieshNatijeh = @AzmaieshNatijeh

Where

AzmaieshID = @AzmaieshID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_BimarMoshakhasat\_DeleteRow

-- Create date: 2009/06/08 03:48:57 ق.ظ

-- Description: This stored procedure is intended for deleting a specific row from BimarMoshakhasat table

-- ==========================================================================================

Create Procedure sp\_BimarMoshakhasat\_DeleteRow

@BimarID int

As

Begin

Delete BimarMoshakhasat

Where

BimarID = @BimarID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_BimarMoshakhasat\_Insert

-- Create date: 2009/06/08 03:48:57 ق.ظ

-- Description: This stored procedure is intended for inserting values to BimarMoshakhasat table

-- ==========================================================================================

CREATE Procedure sp\_BimarMoshakhasat\_Insert

@BimaiName nvarchar(50),

@DateSabt nvarchar(10),

@BimehCode int,

@TelPh nvarchar(50),

@Mobile nvarchar(50),

@MelliCode nvarchar(50),

@Address nvarchar(50),

@Notes ntext

As

Begin

Insert Into BimarMoshakhasat

(BimaiName,DateSabt,BimehCode,TelPh,Mobile,MelliCode,Address,Notes)

Values

(@BimaiName,@DateSabt,@BimehCode,@TelPh,@Mobile,@MelliCode,@Address,@Notes)

Declare @ReferenceID int

Select @ReferenceID = @@IDENTITY

Return @ReferenceID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_BimarMoshakhasat\_SelectAll

-- Create date: 2009/06/08 03:48:57 ق.ظ

-- Description: This stored procedure is intended for selecting all rows from BimarMoshakhasat table

-- ==========================================================================================

Create Procedure sp\_BimarMoshakhasat\_SelectAll

As

Begin

Select

BimarID,

BimaiName,

DateSabt,

BimehCode,

TelPh,

Mobile,

MelliCode,

Address,

Notes

From BimarMoshakhasat

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_BimarMoshakhasat\_SelectRow

-- Create date: 2009/06/08 03:48:57 ق.ظ

-- Description: This stored procedure is intended for selecting a specific row from BimarMoshakhasat table

-- ==========================================================================================

Create Procedure sp\_BimarMoshakhasat\_SelectRow

@BimarID int

As

Begin

Select

BimarID,

BimaiName,

DateSabt,

BimehCode,

TelPh,

Mobile,

MelliCode,

Address,

Notes

From BimarMoshakhasat

Where

BimarID = @BimarID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_BimarMoshakhasat\_Update

-- Create date: 2009/06/08 03:48:57 ق.ظ

-- Description: This stored procedure is intended for updating BimarMoshakhasat table

-- ==========================================================================================

Create Procedure sp\_BimarMoshakhasat\_Update

@BimarID int,

@BimaiName nvarchar(50),

@DateSabt float,

@BimehCode int,

@TelPh nvarchar(50),

@Mobile nvarchar(50),

@MelliCode nvarchar(50),

@Address nvarchar(50),

@Notes ntext

As

Begin

Update BimarMoshakhasat

Set

BimaiName = @BimaiName,

DateSabt = @DateSabt,

BimehCode = @BimehCode,

TelPh = @TelPh,

Mobile = @Mobile,

MelliCode = @MelliCode,

Address = @Address,

Notes = @Notes

Where

BimarID = @BimarID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Moaienat\_Tajvizat\_DeleteRow

-- Create date: 2009/06/08 03:51:31 ق.ظ

-- Description: This stored procedure is intended for deleting a specific row from Moaienat\_Tajvizat table

-- ==========================================================================================

Create Procedure sp\_Moaienat\_Tajvizat\_DeleteRow

@MoaienehID int

As

Begin

Delete Moaienat\_Tajvizat

Where

MoaienehID = @MoaienehID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Moaienat\_Tajvizat\_Insert

-- Create date: 2009/06/08 03:51:31 ق.ظ

-- Description: This stored procedure is intended for inserting values to Moaienat\_Tajvizat table

-- ==========================================================================================

Create Procedure sp\_Moaienat\_Tajvizat\_Insert

@MoaieneDate float,

@PazireshID int,

@TajvizatDaroo nvarchar(350),

@Azmaieshat nvarchar(350)

As

Begin

Insert Into Moaienat\_Tajvizat

(MoaieneDate,PazireshID,TajvizatDaroo,Azmaieshat)

Values

(@MoaieneDate,@PazireshID,@TajvizatDaroo,@Azmaieshat)

Declare @ReferenceID int

Select @ReferenceID = @@IDENTITY

Return @ReferenceID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Moaienat\_Tajvizat\_SelectAll

-- Create date: 2009/06/08 03:51:31 ق.ظ

-- Description: This stored procedure is intended for selecting all rows from Moaienat\_Tajvizat table

-- ==========================================================================================

Create Procedure sp\_Moaienat\_Tajvizat\_SelectAll

As

Begin

Select

MoaienehID,

MoaieneDate,

PazireshID,

TajvizatDaroo,

Azmaieshat

From Moaienat\_Tajvizat

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Moaienat\_Tajvizat\_SelectRow

-- Create date: 2009/06/08 03:51:31 ق.ظ

-- Description: This stored procedure is intended for selecting a specific row from Moaienat\_Tajvizat table

-- ==========================================================================================

Create Procedure sp\_Moaienat\_Tajvizat\_SelectRow

@MoaienehID int

As

Begin

Select

MoaienehID,

MoaieneDate,

PazireshID,

TajvizatDaroo,

Azmaieshat

From Moaienat\_Tajvizat

Where

MoaienehID = @MoaienehID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Moaienat\_Tajvizat\_Update

-- Create date: 2009/06/08 03:51:31 ق.ظ

-- Description: This stored procedure is intended for updating Moaienat\_Tajvizat table

-- ==========================================================================================

Create Procedure sp\_Moaienat\_Tajvizat\_Update

@MoaienehID int,

@MoaieneDate float,

@PazireshID int,

@TajvizatDaroo nvarchar(350),

@Azmaieshat nvarchar(350)

As

Begin

Update Moaienat\_Tajvizat

Set

MoaieneDate = @MoaieneDate,

PazireshID = @PazireshID,

TajvizatDaroo = @TajvizatDaroo,

Azmaieshat = @Azmaieshat

Where

MoaienehID = @MoaienehID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PaziresheBimar\_DeleteRow

-- Create date: 2009/06/08 03:52:07 ق.ظ

-- Description: This stored procedure is intended for deleting a specific row from PaziresheBimar table

-- ==========================================================================================

Create Procedure sp\_PaziresheBimar\_DeleteRow

@PazireshID int

As

Begin

Delete PaziresheBimar

Where

PazireshID = @PazireshID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PaziresheBimar\_Insert

-- Create date: 2009/06/08 03:52:07 ق.ظ

-- Description: This stored procedure is intended for inserting values to PaziresheBimar table

-- ==========================================================================================

CREATE Procedure sp\_PaziresheBimar\_Insert

@OperatorID int,

@BimarID int,

@PazireshDate nvarchar(10),

@BimariType nvarchar(50),

@Notes nvarchar(350)

As

Begin

Insert Into PaziresheBimar

(OperatorID,BimarID,PazireshDate,BimariType,Notes)

Values

(@OperatorID,@BimarID,@PazireshDate,@BimariType,@Notes)

Declare @ReferenceID int

Select @ReferenceID = @@IDENTITY

Return @ReferenceID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PaziresheBimar\_SelectAll

-- Create date: 2009/06/08 03:52:07 ق.ظ

-- Description: This stored procedure is intended for selecting all rows from PaziresheBimar table

-- ==========================================================================================

Create Procedure sp\_PaziresheBimar\_SelectAll

As

Begin

Select

PazireshID,

OperatorID,

BimarID,

PazireshDate,

BimariType,

Notes

From PaziresheBimar

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PaziresheBimar\_SelectRow

-- Create date: 2009/06/08 03:52:07 ق.ظ

-- Description: This stored procedure is intended for selecting a specific row from PaziresheBimar table

-- ==========================================================================================

Create Procedure sp\_PaziresheBimar\_SelectRow

@PazireshID int

As

Begin

Select

PazireshID,

OperatorID,

BimarID,

PazireshDate,

BimariType,

Notes

From PaziresheBimar

Where

PazireshID = @PazireshID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PaziresheBimar\_Update

-- Create date: 2009/06/08 03:52:07 ق.ظ

-- Description: This stored procedure is intended for updating PaziresheBimar table

-- ==========================================================================================

Create Procedure sp\_PaziresheBimar\_Update

@PazireshID int,

@OperatorID int,

@BimarID int,

@PazireshDate float,

@BimariType nvarchar(50),

@Notes nvarchar(350)

As

Begin

Update PaziresheBimar

Set

OperatorID = @OperatorID,

BimarID = @BimarID,

PazireshDate = @PazireshDate,

BimariType = @BimariType,

Notes = @Notes

Where

PazireshID = @PazireshID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PerSonnels\_DeleteRow

-- Create date: 2009/06/08 03:52:34 ق.ظ

-- Description: This stored procedure is intended for deleting a specific row from PerSonnels table

-- ==========================================================================================

Create Procedure sp\_PerSonnels\_DeleteRow

@PersonnelID int

As

Begin

Delete PerSonnels

Where

PersonnelID = @PersonnelID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PerSonnels\_Insert

-- Create date: 2009/06/08 03:52:34 ق.ظ

-- Description: This stored procedure is intended for inserting values to PerSonnels table

-- ==========================================================================================

CREATE Procedure sp\_PerSonnels\_Insert

@PerNameSname nvarchar(80),

@MadrakType nvarchar(50),

@Phone nvarchar(50),

@Mobile nvarchar(50),

@MelliCode nvarchar(50),

@DateEstekhdam nvarchar(10),

@Address nvarchar(350)

As

Begin

Insert Into PerSonnels

(PerNameSname,MadrakType,Phone,Mobile,MelliCode,DateEstekhdam,Address)

Values

(@PerNameSname,@MadrakType,@Phone,@Mobile,@MelliCode,@DateEstekhdam,@Address)

Declare @ReferenceID int

Select @ReferenceID = @@IDENTITY

Return @ReferenceID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PerSonnels\_SelectAll

-- Create date: 2009/06/08 03:52:34 ق.ظ

-- Description: This stored procedure is intended for selecting all rows from PerSonnels table

-- ==========================================================================================

Create Procedure sp\_PerSonnels\_SelectAll

As

Begin

Select

PersonnelID,

PerNameSname,

MadrakType,

Phone,

Mobile,

MelliCode,

DateEstekhdam,

Address

From PerSonnels

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PerSonnels\_SelectRow

-- Create date: 2009/06/08 03:52:34 ق.ظ

-- Description: This stored procedure is intended for selecting a specific row from PerSonnels table

-- ==========================================================================================

Create Procedure sp\_PerSonnels\_SelectRow

@PersonnelID int

As

Begin

Select

PersonnelID,

PerNameSname,

MadrakType,

Phone,

Mobile,

MelliCode,

DateEstekhdam,

Address

From PerSonnels

Where

PersonnelID = @PersonnelID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_PerSonnels\_Update

-- Create date: 2009/06/08 03:52:34 ق.ظ

-- Description: This stored procedure is intended for updating PerSonnels table

-- ==========================================================================================

Create Procedure sp\_PerSonnels\_Update

@PersonnelID int,

@PerNameSname nvarchar(80),

@MadrakType nvarchar(50),

@Phone nvarchar(50),

@Mobile nvarchar(50),

@MelliCode nvarchar(50),

@DateEstekhdam float,

@Address nvarchar(350)

As

Begin

Update PerSonnels

Set

PerNameSname = @PerNameSname,

MadrakType = @MadrakType,

Phone = @Phone,

Mobile = @Mobile,

MelliCode = @MelliCode,

DateEstekhdam = @DateEstekhdam,

Address = @Address

Where

PersonnelID = @PersonnelID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Users\_DeleteRow

-- Create date: 2009/06/08 03:52:57 ق.ظ

-- Description: This stored procedure is intended for deleting a specific row from Users table

-- ==========================================================================================

CREATE Procedure sp\_Users\_DeleteRow

@UserName nvarchar(50)

As

Begin

Delete Users

Where

UserName = @UserName

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Users\_Insert

-- Create date: 2009/06/08 03:52:57 ق.ظ

-- Description: This stored procedure is intended for inserting values to Users table

-- ==========================================================================================

Create Procedure sp\_Users\_Insert

@UserName nvarchar(50),

@PassWord nvarchar(50),

@PerMis smallint

As

Begin

Insert Into Users

(UserName,PassWord,PerMis)

Values

(@UserName,@PassWord,@PerMis)

Declare @ReferenceID int

Select @ReferenceID = @@IDENTITY

Return @ReferenceID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Users\_SelectAll

-- Create date: 2009/06/08 03:52:57 ق.ظ

-- Description: This stored procedure is intended for selecting all rows from Users table

-- ==========================================================================================

Create Procedure sp\_Users\_SelectAll

As

Begin

Select

UserID,

UserName,

PassWord,

PerMis

From Users

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Users\_SelectRow

-- Create date: 2009/06/08 03:52:57 ق.ظ

-- Description: This stored procedure is intended for selecting a specific row from Users table

-- ==========================================================================================

Create Procedure sp\_Users\_SelectRow

@UserID int

As

Begin

Select

UserID,

UserName,

PassWord,

PerMis

From Users

Where

UserID = @UserID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS OFF

GO

-- ==========================================================================================

-- Entity Name: sp\_Users\_Update

-- Create date: 2009/06/08 03:52:57 ق.ظ

-- Description: This stored procedure is intended for updating Users table

-- ==========================================================================================

Create Procedure sp\_Users\_Update

@UserID int,

@UserName nvarchar(50),

@PassWord nvarchar(50),

@PerMis smallint

As

Begin

Update Users

Set

UserName = @UserName,

PassWord = @PassWord,

PerMis = @PerMis

Where

UserID = @UserID

End

GO

SET QUOTED\_IDENTIFIER OFF

GO

SET ANSI\_NULLS ON

GO

**منابع و مواخذ**

**الف) کتب**

1. کتاب برنامه نویسی به زبان VB.NET تالیف آقای مهندس زاهد
2. کتاب تجزیه و تحلیل سیستمهای شی گرایی تالیف آقای مهندس جبیب مظاهری
3. کتاب پایگاه های داده تالیف آقای دکتر ندیمی شهرکی
4. کتاب بررسی زبانهای برنامه نویسی نوشته مهندس فرشاد حاجی محمدی
5. کتاب سیستمهای نرم افزاری هوشمند تالیف دکتر شیرین مرادخانی

**ب) آدرسهای اینترنتی**

* + www. geekpoint.net
  + www. vbdotnetheaven.com
  + wwww. visualbasic.about.com
  + www.shareme.com
  + forum.hammihan.com
  + www.windowsecurity.com
  + ukoln.ac.uk
  + www.webservicessummit.com
  + www.agilemodeling.com