



جمهوری اسلامی افغانستان

وزارت تحصیلات عالی

ریاست موسسات تحصیلات عالی خصوصی

موسسه تحصیلات عالی خواجه عبدالله انصاری (ره)

دانشکده کامپیوتر ساینس

دیپارتمنت دتاییس



پایان نامه مقطع لیسانس

طراحی و ساخت سیستم مدیریت دواخانه

استاد راهنما: محمد حسنی

محصل: همایون عظیمی

زمستان ۱۳۹۴ - هرات، افغانستان

این صفحه عمداً فالی گذاشته شده است!

جمهوری اسلامی افغانستان

وزارت تحصیلات عالی

ریاست موسسات تحصیلات عالی خصوصی

موسسه تحصیلات عالی خواجه عبدالله انصاری (ره)

دانشکده کامپیوتر ساینس

دیپارتمنت دتاییس

پایان نامه مقطع لیسانس

طراحی و ساخت سیستم مدیریت دواخانه

استاد راهنما: محمد حسنی

محصل: همایون عظیمی

زمستان ۱۳۹۴ - هرات، افغانستان

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

اهدا

با سپاس از سه وجود مقدس:

آنان که ناتوان شدند تا ما به توانایی برسیم...

موهایشان سپید شد تا ما رو سفید شویم...

و عاشقانه سوختند تا گرمابخش وجود ما و روشنگر راه مان باشند...

پدرم

مادرم

اساتید عزیزم

چکیده:

طرح حاضر بیانی از نحوه ساخت و نمونه عملکرد یک دتایس در مدیریت خرید و فروش دوا در دواخانه های افغانستان می باشد. با توجه به این که تعداد دواخانه ها در افغانستان همه روزه رو به افزایش می باشد بر آن شدیم تا سیستمی را به عنوان پایان نامه خویش مد نظر گرفته تا قسمت کوچکی از مسائل پیش روی ملت عزیزمان افغانستان را برداشته و یا به نحوی حل نماییم.

دواخانه هایی که حداقل یک پایه کامپیوتر و یک کارمند با حداقل دانش کامپیوتری را دارا باشند می توانند به راحتی سیستم حاضر را استفاده نموده و تمامی امورات حسابداری خویش را به راحت ترین شکل ممکن شفافیت بخشیده و به آن سرعت دهند. علاوه بر آن برای آن عده از عزیزانی که خواهان ساخت یک سیستم مدیریت دواخانه می باشند می توانند با استفاده از این رساله تکنیک ها و ابزار های مورد استفاده ساخت سیستم های مدیریتی را آموخته و برای امور مشابه دست به طراحی بزنند. قابل ذکر است محققین عزیزی که خواهان رشد و انکشاف سیستم حاضر می باشند نیز می توانند امور تکراری از قبل انجام شده را به راحتی پشت سر گذاشته و در امر توسعه سیستم کنونی گام های بیشتری بردارند.

فهرست مطالب

۱	فصل اول – کلیات تحقیق
۲	بیان مسئله
۲	سوال تحقیق
۲	سوال اصلی
۲	سوالات فرعی
۳	فرضیه تحقیق
۳	فرضیه اصل
۳	فرضیه جانشین
۳	اهداف تحقیق
۳	هدف اصلی
۳	هدف فرعی
۳	اهمیت و ضرورت تحقیق
۴	پیشینه تحقیق
۵	روش تحقیق
۵	موانع و محدودیت های تحقیق
۵	زمان بندی اجرای تحقیق
۶	فصل دوم – معرفی دواخانه و نحوه خرید و فروش دوا
۷	نحوه کارکرد دواخانه
۷	خرید دوا توسط بیمار
۸	معرفی اشیای موجود و رابطه ها
۸	بیمار و بیماری:
۸	دوا (دارو):
۸	شرکت های پخش دوا:
۹	فاکتور (بیل):
۹	چک های پرداخت پول
۹	پزشک:

نسخه:	۱۰
انتبار:	۱۰
فصل سوم – نحوه طراحی و ساخت دتاییس	۱۱
دتاییس دیاگرام دتاییس	۱۲
دتاییس و جداول	۱۲
مشخصه های موجودیت ها	۱۵
نرمال سازی ها	۱۹
گزارش های اس کیو ال	۲۰
گزارش های محاسباتی	۲۱
موجودی انتبار	۲۱
فصل چهارم – معرفی ابزارها و نحوه ساخت رابط کاربری	۲۵
معرفی و نحوه ساخت رابط کاربری	۲۶
ابزار های مورد استفاده	۲۶
نرم افزار Visual Studio	۲۶
زبان برنامه نویسی C#	۲۷
Silverlight	۲۸
معرفی Microsoft Light Switch	۲۸
ساخت پروژه جدید با LightSwitch	۲۹
ارتباط نرم افزار و پایگاه داده	۳۰
تنظیمات و مشخصات پروژه	۳۲
ساخت Screen	۳۴
افزودن ستون های محاسباتی	۳۵
Validation معتبر سازی	۳۷
گزارش های چاپی	۳۷
ساخت Setup نرم افزار	۳۷
فصل پنجم – مراحل نصب و نحوه استفاده از سیستم	۴۱
مراحل نصب و استفاده سیستم	۴۲
بستر سخت افزاری	۴۲

۴۲	بستر نرم افزاری
۴۳	منابع داتلود نرم افزار های مورد نیاز
۴۳	مراحل نصب سیستم
۴۶	فصل ششم – بررسی نقاط ضعف و قوت سیستم مدیریت دواخانه
۴۷	نتیجه گیری
۴۷	پیشنهادهات
۴۸	پیوست
۴۹	منابع

۷	تصویر ۱ - چارت تعقیبی ورود و خروج بیمار
۱۳	تصویر ۲ - رابطه یک به چند بیمار و نسخه
۱۳	تصویر ۳ - رابطه یک به چند بین دوا و نسخه
۱۳	تصویر ۴ - رابطه چند به چند بین نسخه و دوا
۱۴	تصویر ۵ - دیاگرام کامل دتابیس مدیریت دواخانه
۱۵	تصویر ۶ - بیمار و مشخصه های آن
۱۶	تصویر ۷ - انبار و مشخصه های آن
۱۶	تصویر ۸ - دکتر و مشخصه های آن
۱۷	تصویر ۹ - دوا و مشخصه های آن
۱۷	تصویر ۱۰ - شرکت های پخش دوا و مشخصه های آن
۱۸	تصویر ۱۱ - جزئیات نسخه و مشخصه های آن
۱۸	تصویر ۱۲ - بل و مشخصه های آن
۲۲	تصویر ۱۳ - جداول مرتبط برای دریافت موجودی انبار
۲۴	تصویر ۱۴ - نتیجه گزارش موجودی انبار در نرم افزار SSMS
۲۹	تصویر ۱۵ - نحوه ساخت پروژه جدید در LightSwitch
۳۰	تصویر ۱۶ - دو انتخاب برای ایجاد دتابیس در LightSwitch
۳۰	تصویر ۱۷ - نحوه ارتباط دتابیس در SSMS با LightSwitch
۳۱	تصویر ۱۸ - ایجاد ارتباط با SSMS و انتخاب دتابیس ساخته شده
۳۲	تصویر ۱۹ - ساختار درختی پروژه LightSwitch و زیر مجموعه ها
۳۳	تصویر ۲۰ - پنجره Properties سیستم دواخانه
۳۳	تصویر ۲۱ - پنجره Properties رابط کاربری سیستم مدیریت دواخانه
۳۴	تصویر ۲۲ - نحوه ساخت Screen جدید در LightSwitch
۳۵	تصویر ۲۳ - راست چین کردن و تغییر نام فرم ها
۳۶	تصویر ۲۴ - نحوه افزودن ستون های محاسبه ای
۳۸	تصویر ۲۵ - انتخاب نوع خروجی نرم افزار
۳۸	تصویر ۲۶ - انتخاب نوع سرویس ها
۳۹	تصویر ۲۷ - محل ذخیره Setup نرم افزار
۳۹	تصویر ۲۸ - مشخصات دتابیس شامل آدرس، نام دتابیس، نام کاربری و پسورد
۴۰	تصویر ۲۹ - انتخاب برنامه های مورد نیاز برای نصب برنامه
۴۰	تصویر ۳۰ - مشخصات کلی از خروجی نرم افزار
۴۴	تصویر ۳۱ - نصب Setup نرم افزار
۴۴	تصویر ۳۲ - صفحه اصلی نرم افزار مدیریت دواخانه
۴۵	تصویر ۳۳ - افزودن نسخه جدید در سیستم
۱۴	جدول ۱ - عبارت های دری نام جداول در دتابیس
۱۹	جدول ۲ - نمونه جدول، پیش از نرمال سازی

فصل اول – کلیات تحقیق

اهمیت دواخانه در اجتماع کنونی یک ضرورت غیر قابل انکار است که تمامی انسان ها در هر طبقه علمی، فرهنگی و یا اقتصادی با آن به نوعی سر و کار دارند؛ از این رو در تمامی شهرها، ولسوالی ها و حتی اکثر قریه ها وجود یک دواخانه حتمی بوده و هر کدام به نوعی خدمات صحتی را به بیماران خویش ارائه میدارند.

نظر به حجم کاری هر دواخانه امورات هر یک پیچیدگی های خاص خود را دارد. طور مثال یک دواخانه در مرکز شهر (شهر نو) عرضه و تقاضا بسیار بالا بوده و مدیریت انبار و قفسه ها کاری دشوار برای کارمندان آن می باشد و یا این که روزانه چه مقدار مفاد و یا ضرر از فروش روزانه بدست آمده است گاهی غیر ممکن میشود. اضافه از آن در صورتی که خواسته باشد اطلاعاتی راجع به مقدار دواهایی که در یک فاصله زمانی فروخته شده و یا این که تاریخ تولید و مصرف اجناس را ردیابی کند نیز دشواری های خاص خود را دارد. شرکت های پخش دارو که به طور عمده دوا را به یک دواخانه عرضه میکنند، تمامی مبادلات پولی آن ها نیز به صورت اقساط بوده و به طور هفته وار و یا ماه وار پرداخت می شود که ثبت تمامی اجناس آورده شده و پول های پرداخت شده نیز از چالش های هر دواخانه می باشد.

سیستمی که در انتهای این پایان نامه ساخته خواهد شد، یک راه حل جامع برای مشکلات بالا می باشد.

سوال تحقیق

سوال اصلی

نحوه طراحی و ساخت یک سیستم مدیریت دواخانه چگونه می باشد؟

سوالات فرعی

سیستم طراحی شده چه مقدار در امر شفافیت کار کارکنان یک دواخانه کمک کرده است؟

برای کار کردن با این سیستم آیا به افراد متخصص در زمینه کامپیوتر نیاز است؟

حیطه کاری این سیستم در کدام قسمت های یک دواخانه می باشد؟

فرضیه تحقیق

فرضیه اصل

پس از طراحی این سیستم قابلیت ثبت تمامی دواهای خریداری شده از شرکت های مختلف، ثبت فروش و اجرای فاکتور فروش برای تمامی بیماران و قابلیت مدیریت انبار را این سیستم در بر داشته باشد و بتواند چالش های ذکر شده در بیان مسئله را به حداقل برساند..

فرضیه جانشین

روش های سنتی کنونی با این که نواقص زیادی را دارد، اما هنوز هم پاسخگوی نیازهای اساسی یک دواخانه در مقیاس کوچک که تعداد مراجعین کم دارد پاسخگو است؛ اما با توجه به افزایش حجم کار در دواخانه های بزرگ، وجود یک سیستم جایگزین که در زمینه مدیریت دواخانه باشد، امری ضروریست.

اهداف تحقیق

هدف اصلی

ایجاد راه اندازی یک پایگاه داده برای مدیریت انبار، دارایی و مراجعین یک دواخانه و همچنین محاسبات قسمت های مالی دواخانه که بتواند میزان مفاد و ضرر که از فروش دوا به دست آمده است را حساب کند.

هدف فرعی

ایجاد یک برند جدید در زمینه مدیریت دواخانه که بتواند مسائل عمومی یک دواخانه را بدون این که از سرعت کار بکاهد ساخته شود. همچنین این سیستم بتواند نمایندگی از جمهوری اسلامی افغانستان در میان یکی از سیستم های جامع کاربران زبان فارسی باشد.

اهمیت و ضرورت تحقیق

نظر به آماری که از دواخانه ها در شهر کابل در سایت رایزن بازرگانی سفارت ایران ثبت شده است:

- تعداد داروخانه های بزرگ کابل (درجه ۱) ۵۰ عدد
- تعداد داروخانه های متوسط کابل (درجه ۲) ۱۴۰۰ عدد

- تعداد داروخانه های حومه کابل (درجه ۳) ۶۰۰ عدد
 - تعداد کلینیک های موجود در کابل ۱۵۰ عدد
 - تعداد بیمارستانهای کابل ۴۳ عدد بیمارستان شخصی که به طور متوسط دارای ۱۰الی ۲۰ تخت می باشد
- پس با توجه به آمار دواخانه های پایتخت، بالاتر از ۲۰۰۰ دواخانه موجود می باشد که همه روزه با امور تکراری چون محاسبات پیچیده، ثبت اسناد، انبار داری و بسیاری کارهای دیگر نیز سرکار دارند؛ پس با وجود سیستم جامع و معتبر علاوه بر دواخانه های پایتخت بسیاری از ولایت های بزرگ چون هرات، بلخ، بامیان، جلال آباد، قندهار و دیگر ولایات را نیز میتوان تحت پوشش قرار داد.

پیشینه تحقیق

تحقیق حاضر نظر به این که پیشینه فروش دوا بسیار قدیمی بوده، قطعاً کاری نو نیست؛ اما افغانستان با توجه به مسائل داخلی چند دهه اخیر از این روند مستثنی می باشد. با توجه به این که رشته کامپیوتر و دتایس هنوز نتوانسته جای خود را در میان جوامع به طور گسترده باز کند، شرکت های تولید نرم افزار تا کنون اقدام به ساخت چنین سیستمی نکرده اند. اما برخی از دواخانه ها که با دیدی متفاوت به این بخش می نگرند سیستم هایی که توسط کشور های همسایه ساخته شده است را در برخی از دواخانه های شهر راه اندازی کرده و از آن استفاده می کنند.

این سیستم ها به طور عموم ساخت کشور ایران و یا پاکستان می باشد. اکثر این سیستم ها نظر به معیار های همان کشور ها ساخته شده است و بسیاری موارد اضافی و مغایر با شیوه کار افغانستان می باشد. به طور مثال بیمه درمانی یکی از خدماتی است که در کشور ایران به وفور بین شهروندان این کشور استفاده می شود و در تمامی سیستم های مدیریت دواخانه جا سازی شده است. اما هیچ یک از بیمه های درمانی کشور ایران در افغانستان معنایی ندارد.

روش تحقیق

برای طراحی هر دتایس، طراحان آن باید با اداره متقاضی دتایس به طور مستقیم رابطه برقرار کرده و تمامی قسمت های مختلف سیستم را توسط کارمندان آن اداره دریابند. ازین رو روش تحقیق در این منوگراف میدانی می باشد. اما در قسمت طراحی سیستم که توسط زبان های برنامه نویسی مختلف انجام می شود، کماکان از روش کتابخانه ای نیز در قسمت حل مسائل برنامه نویسی به صورت استاندارد باید استفاده کنیم.

موانع و محدودیت های تحقیق

مانع بزرگی که در قسمت جمع آوری منابع می باشد، مسئله حریم خصوصی است که اکثر کسانی که به نوعی به تجارت مصروف هستند صلاح کار خود را در پنهان نگه داشتن سیستم کاری خویش می دانند. رقابت هایی که بین شرکت های همکار می باشد این موضوع را برای محققین که برای جمع آوری اطلاعات به این مراکز مراجعه میکنند دشوار میسازد.

محدودیت دیگری که در این قسمت می باشد کارکرد غیر استاندارد بسیاری از دواخانه ها است که می تواند در ساخت این سیستم موانعی را ایجاد کند.

زمان بندی اجرای تحقیق

جمع آوری اطلاعات (۷ روز)

تحلیل و طراحی دیاگرام پایگاه داده (۵ روز)

اجرای گزارش های مربوط به سیستم (۷ روز)

طراحی ظاهر (ماکت) نرم افزار (۵ روز)

اجرای توابع (فانکشن های) نرم افزار در قسمت رابط کاربری (۵۰ روز)

تست و اجرای نرم افزار (۱۰ روز)

فصل دوم – معرفی دواخانه و نحوه خرید و فروش دوا

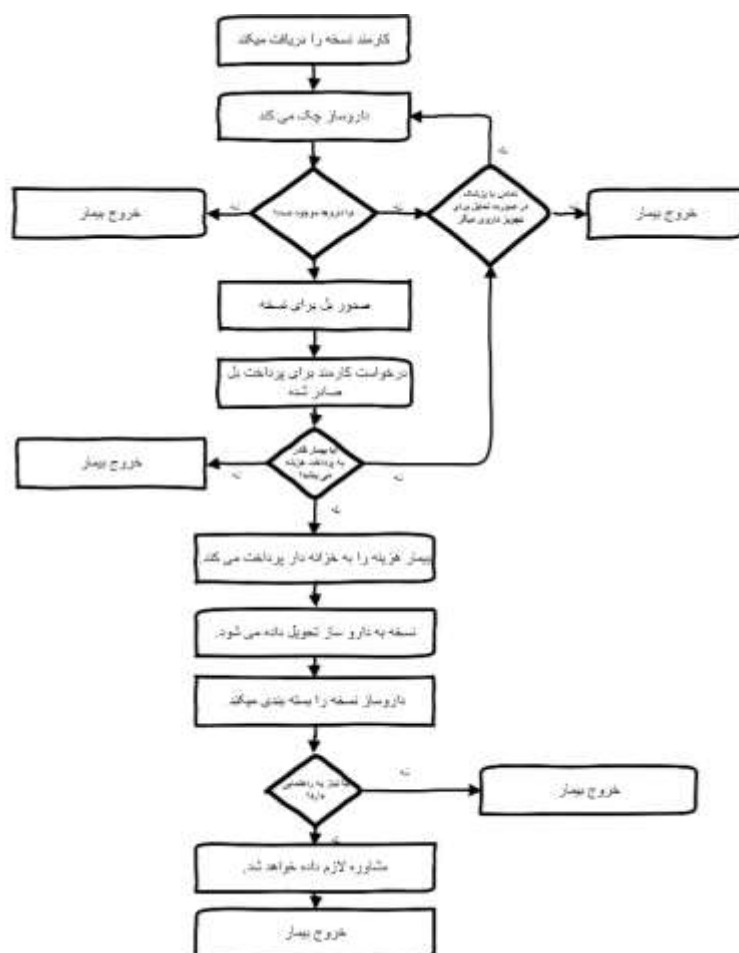
نحوه کارکرد دواخانه

خرید دوا توسط بیمار

طبق قانون کشوری افغانستان اکثر دواهایی که توسط بیماران خریداری می شود باید با دستور پزشک باشد و پس از ارائه نسخه توسط بیمار، دواساز که شخص مسلکی در رشته Pharmacy^۱ باشد می تواند دواي مورد نظر را در اختیار بیمار قرار دهد.

به طور مفصل می توان چنین گفت که پس از ورود بیمار، دواساز اطمینان خود را از موجودیت دواهای تجویز شده در انبار دواخانه حاصل نموده و اقدام به صدور قبض پرداخت هزینه دوا توسط بیمار را صادر کند؛ در صورت

قادر بودن بیمار به پرداخت هزینه ها، بیمار موظف به پرداخت هزینه به صندوقدار خواهد بود و میتواند دواي تجویز شده توسط پزشک را دریافت نماید. در این میان عوامل احتمالی مبنی بر موجود بودن و یا نبودن دواهای تجویز شده، قدرت پرداخت هزینه دواها توسط بیمار و برخی اتفاقات دیگر نیز رخ می دهد که در شکل ۱-۲ می توانید نموداری از لحظه ورود بیمار تا خروج آن را مشاهده نمایید. در این قسمت تمامی اتفاقاتی که ممکن در جریان اجرای یک نسخه روی دهد را نشان داده است.



تصویر ۱- چارت تعقیبی ورود و خروج بیمار

به هم ریختگی نظم درونی بدن را بیماری گویند و این به هم ریختگی میتواند بر اثر عوامل زیست محیطی و یا ناهنجاری های داخل بدن باشد. شخصی که یکی از علایم بیماری را در خود مشاهده کند بیمار نامیده میشود. در چنین شرایطی شخص بیمار برای درمان خویش به پزشک معالج خود مراجعه کرده و برای درمان خویش تلاش میکند.

هر آنچه منجر به درمان بیماری در انسان شود دوا گفته می شود. در گذشته داروهای گیاهی برای درمان بیماری های مختلف کاربرد فراوانی داشت که هنوز هم در بسیاری از کشور ها از این شیوه برای درمان استفاده می کنند؛ اما با گذشت زمان و پیشرفت دانش بشر، دارو های شیمیایی جایگزین دارو های گیاهی شده و امروزه سهم عظیمی را در درمان بیماری های مختلف بشر را دارا می باشد. هدف ما در اینجا دارو های شیمیایی بوده که مفصل تر در مورد آن صحبت خواهیم نمود.

دوا ها به طور عموم دو نام را دارا می باشند، که یک نام از ترکیب شیمیایی دوا و نام دیگر آن توسط شرکت سازنده انتخاب شده و به بازار عرضه می گردد. مشخصه های دیگر هر دوا عبارت است از کشور سازنده که در افغانستان کشورهای عربی، ایرانی، پاکستانی و تا حدودی اروپایی بالاترین حد مصرف را دارا می باشد. مقدار وزنی دوا ها و نوع استفاده آن ها نیز در تفکیک دوا ها تعیین کننده است. تاریخ تولید، تاریخ مصرف هر دوا از مشخصه های بارز آن می باشد. دوا های شیمیایی به طرق مختلف مورد استفاده قرار می گیرد که به شکل آمپول، قرص (تبلت)، سرم، پماد، شربت، شیاف، اسپری، قطره چکان و بسیاری از انواع دیگر می باشد.

در هر یک از شهر های افغانستان مارکت هایی برای توزیع و پخش دوا موجود می باشد که دوا فروشان می توانند ادویه مورد نیاز خویش را از این مارکت ها تهیه کنند؛ با این حال هم هر شرکت پخش با استخدام چند کارمند به

عنوان بازاریاب نمونه دواهای جدید خود را به دواخانه های مختلف شهر به طور روزمره فرستاده و با توافق بین دواخانه ها دواهای مورد نیاز هر دواخانه را تامین می کنند. بسیاری از شرکت ها به خاطر جلب بیشتر دواخانه ها به طور اقساط نیز معاملات خویش را انجام میدهند و به طور عموم در آخر هر هفته مقداری از هزینه دواهای فروخته شده را از دواخانه ها جمع آوری میکنند.

فاکتور (بیل):

هر گاه دواخانه برای تهیه دواي خود از شرکت های پخش درخواست دوا می کند، شرکت های پخش برای این که مشخصات دوا های فروخته شده را برای خروج آن ها از انبار خودشان، تحویل دهی دوا به دواخانه دار و همچنان به نوعی سندی بین طرفین باشد را صادر میکنند. این سند دارای تاریخ صدور، اقلام فروخته شده، تعداد، قیمت واحد و قیمت مجموعه می باشد. فاکتور هایی که به صورت نقد از طرف دواخانه دار پرداخت گردد، نماینده فروش شرکت پخش دارو با امضای خویش مصداق رسید پول دواي فروخته شده خواهد شد؛ در غیر در آرشیف هر دو طرف به صورت باقی داری اضافه خواهد شد و در اقساط چند ماهه به فروش خواهد رسید.

چک های پرداخت پول

شرکت های پخش برای این که سندی را از دریافت پول دواي فروخته شده از دواخانه ها دریافت کرده باشند، قبض هایی را که نام شرکت پخش و برخی مشخصات دیگر را دارا میباشد را با اضافاتی از قبیل تاریخ، مبلغ دریافت شده از سوی دواخانه، مناصفه (نصف) مبلغ دریافت شده و همچنان مشخصات شخص دریافت کننده (بازاریاب شرکت پخش) را در آن ذکر می کنند. این قبض ها دارای یک شماره سریال نیز می باشد.

پزشک:

شخصی که بتواند نوعی از بیماری های فیزیکی و یا روانی افراد را مداوا کند پزشک نامیده می شود. دکتر واژه دیگر لاتینی هست که برای معرفی شخص مورد نظر استفاده می شود و در افغانستان گاهی دکتور دوکتور و یا داکتر نیز نوشته می شود. پزشک ها نظر به تخصصی که در هر قسمت داشته باشند از یک دیگر متمایز می شوند.

برخی ها بیماری های عمومی، امراض داخله و برخی ها هم متخصص در قسمتی دیگر مانند چشم، ارتوپدی، فیزیوتراپی و غیره نیز می باشند.

نسخه:

نسخه که با نشان Rx نیز شناخته می شود، از چندین قسمت تشکیل شده است که نام، شماره تماس، رشته تحصیلی، آدرس معاینه خانه و برخی دیگر اطلاعات عمومی پزشک صادر کننده را دارا می باشد. به طور عموم تمامی مشخصات ذکر شده بر روی اوراق با اندازه های مختلف از قبل چاپ شده است. پزشک نظر به نوع بیماری شخص مراجعه کننده دواهای مورد نیاز را بر روی نسخه نوشته و بیمار می تواند با رجوع به دواخانه ها دواي مورد نیاز خویش را تهیه نماید. در قسمت اول نسخه، نوع دواي صادره توسط پزشک نوشته می شود. سپس نام شرکتی و یا نام کیمیایی دواي مورد نظر، در مقابل آن تعداد و سپس مقدار استفاده دوا در ۲۴ ساعت توسط پزشک نوشته می شود.

انبار:

با توجه به نام انبار، به راحتی میتوان گفت که انبار محلی برای ذخیره اجناس می باشد. حال انبار ها میتوانند در مکان های مختلف و با ظرفیت های متفاوت باشند.

دلیل انتخاب انبار برای موجودیت به این خاطر گرفته شده است که در صورتی که بیش از چند مکان برای ذخیره و نگهداری دوا ها موجود باشد، موقعیت دواي مورد نیاز خود را بتوانیم به راحتی پیدا کنیم.

فصل سوم – نحوه طراحی و ساخت دتاییس

دیاگرام دتایس

دتایس و جداول

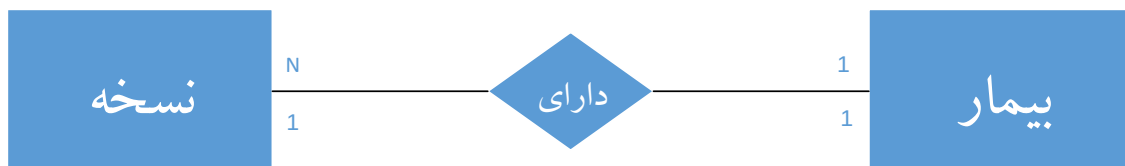
دتایس ترکیبی از جداول مختلف می باشد که در هر یک از آن ها مقادیر مختلفی ثبت می شود. نام این جداول متناسب با نام موجودیت هایی که قرار است اطلاعات آن را ذخیره کنیم انتخاب می شود. به طور مثال اگر قرار است در این جدول مشخصات بیمار ثبت شود جدول به نام بیمار و ستون های هر جدول مشخصه هایی که ارزش ثبت را داشته باشند انتخاب می کنیم. مثلاً (نام، تخلص، نام پدر، تاریخ تولد، محل سکونت، شماره تماس، اثر انگشت، عکس، شماره تذکره، میزان تحصیلات) و بسیاری موارد دیگر این چینی نیز میتواند به حیث مشخصه (Attribute) منظور شود.

با توجه به مطالبی که در فصل دوم به آن اشاره شد، موجودیت ها در یک دواخانه عبارت اند از (دوا، بیمار، نسخه، پزشک، شرکت های پخش دوا، انبار های نگهداری دوا، فاکتور ها، قبض های پرداختی) که هر کدام به نوعی در مورد آن توضیح داده شد، می باشد. در این میان برای ذخیره اطلاعات مربوط به هر یک از اشیای مذکور جدول خاصی را باید مد نظر گرفت. از آنجایی که اطلاعات جدول ها ممکن است با یکدیگر مرتبط باشند در طراحی یک پایگاه داده باید تمامی موارد منطقی آن پیش از هرگونه اقدام بر روی کاغذ ترسیم شود. وقتی بحث از رابطه جدول ها در یک پایگاه داده به میان آید تنها رابطه یک به یک، یک به چند و چند به چند شناخته شده است. برای اطلاعات بیشتر در مورد رابطه ها به [این لینک](#)² مراجعه کنید.

حال برای درک بیشتر از مفهوم جداول و رابطه ها در پایگاه داده، و چرایی انتخاب این رابطه ها را به طور کل شرح می دهیم.

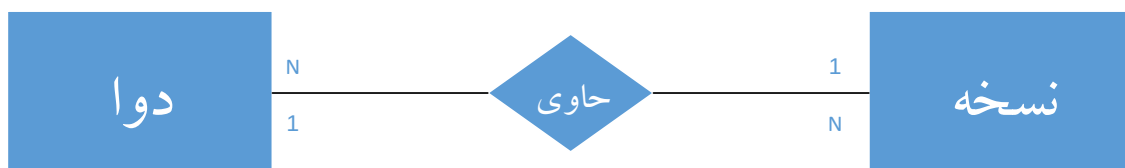
² <https://www.youtube.com/watch?v=NvrpuBAMddw>

رابطه بین بیمار و نسخه یک رابطه یک به چند می باشد؛ چراکه یک بیمار میتواند چندین نسخه را از دواخانه دریافت کند و از سمت دیگر یک نسخه، فقط مختص به یک بیمار است. رابطه این دو را میتوانید در رسم ۱-۳ مشاهده نمایید.



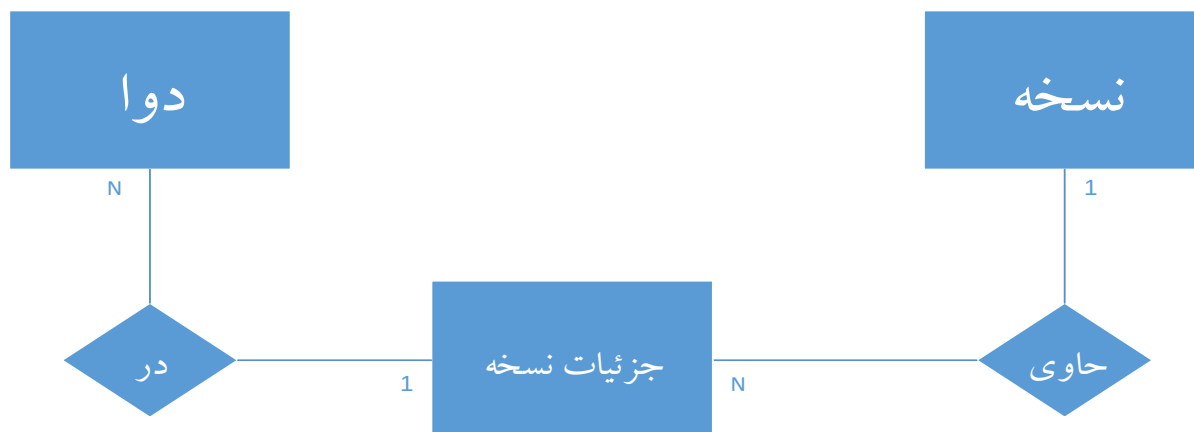
تصویر ۲- رابطه یک به چند بیمار و نسخه

رابطه دیگری که میتوان تعریف کرد، مشخصات دارویی نسخه می باشد. به این معنا که در یک نسخه میتواند چندین دوا را در بر داشته باشد و یک دوا میتواند در چندین نسخه تکرار شود. پس نتیجه این رابطه یک رابطه چند به چند می باشد.



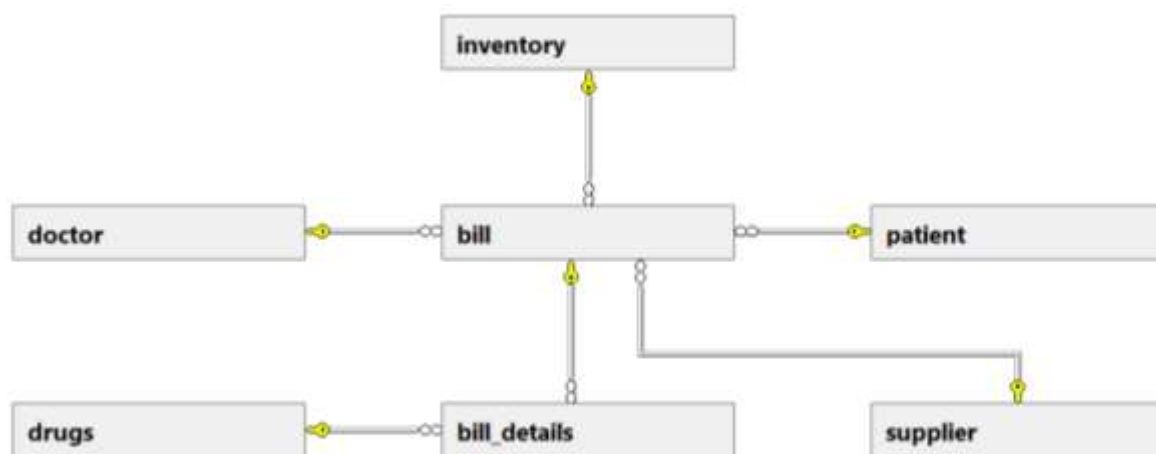
تصویر ۳- رابطه یک به چند بین دوا و نسخه

رابطه های چند به چند یک تعریف منطقی است و در هنگام پیاده سازی در نهایت به یک رابطه یک به چند تبدیل میشود، اما یک جدول دیگر میانجی دو جدول اصلی میگردد.



تصویر ۴ - رابطه چند به چند بین نسخه و دوا

نظر به موجودیت هایی که پس از جمع آوری اطلاعات به دست آمد، رابطه های هر یک از آن ها در تصویر زیر به نمایش گذاشته شده است. سمتی که با کلید زرد رنگ نشانه گذاری شده است سمت ۱ و سمت دیگری که با علامت لایتناهی علامت گذاری شده است، سمت چند می باشد.



تصویر ۵ - دیاگرام کامل دتابیس مدیریت دواخانه

انبار	Inventory
بیمار	Patient
نسخه، فاکتور	Bill
پزشک	Doctor
شرکت پخش دوا	Supplier
جزئیات نسخه	Bill Details
دوا	Drugs

جدول 1- عبارت های دری نام جداول در دتابیس

- فاکتور و نسخه، دو موجودیتی که در فصل دوم آن اشاره شد، در طرح حاضر به نام بیل (Bill) مسمی شده است، چرا که مشخصات (Attribute) هر دو موجودیت همسان می باشند.

مشخصه های موجودیت ها

مشخصه هایی که برای هر یک از موجودیت ها مد نظر گرفته شده است بر اساس اطلاعاتی می باشد که از دواخانه ها جمع آوری شده است. نا گفته نماند که صرف مهمترین اطلاعاتی که برای یک دواخانه دار مهم هست در این طراحی مد نظر گرفته شده است. در صورتی که مشخصه دیگری مد نظر یک دواخانه دار باشد میتوان به راحتی در قسمت طراحی دتابیس به آن اضافه نمود و در سیستم به طور عملی پیاده کرد.

- ستون id در تمامی موجودیت ها و منیث کلید اصلی (Primary Key) تعریف شده است. کلید اصلی به منظور شناسایی یک رکورد مشخص در جدول می باشد. به طوری که به هیچ عنوان تکراری و یا اطلاعات خالی (Null) در آن درج نمی شود. همچنان رابط مشخص و خوبی برای ایجاد رابطه های هر یک از جداول می باشد.

بیمار (Patient)

شماره پرونده بیمار، نام، تخلص، نام پدر، تاریخ تولد، تاریخ ثبت در سیستم، آدرس، ولایت و در نهایت شهر و شهرستان به حیث مشخصات آن مد نظر گرفته شده است.

patient	
	id
	dossierno
	patientname
	lastname
	fathername
	dateofbirth
	registerdate
	lineaddress
	province
	district

تصویر ۶ - بیمار و مشخصه های آن

انبار (Inventory)

تنها نام و ظرفیت انبار مشخصه های این موجودیت می باشد. ظرفیت انبار در طراحی پایگاه داده به ما کمک می کند تا نظر به نیاز خویش در مقدار خرید دوا تصمیم گیری کنیم.

inventory	
id	
name	
capacity	

تصویر ۷ - انبار و مشخصه های آن

پزشک (Doctor)

نام پزشک، تخلص، شماره تماس، درجه تحصیل، آدرس مطب و عکس مشخصه هایی می باشد که از پزشک در پایگاه داده ما ثبت می گردد. در بسیاری از دواخانه ها این که دارو توسط کدام پزشک تجویز شده است زیاد مد نظر گرفته نمی شود. اما برخی از دواخانه ها این امر برایشان مهم هست.

doctor	
id	
name	
lastname	
mobilen	
degree	
address	
photo	

تصویر ۸ - دکتر و مشخصه های آن

دوا (Drugs)

مشخصه های دوا عبارت اند از نام شرکتی، نام ترکیبی، وزن، واحد وزن و نحوه استفاده دوا می باشد.

drugs	
	id
	companyname
	realname
	weight
	unit
	typeofuse

تصویر ۹ - دوا و مشخصه های آن

شرکت پخش دوا (Supplier)

نام شرکت های پخش، شخص ارتباطی، آدرس، دو ستون برای شماره های تماس و ستون آخر تلفن ثابت شرکت پخش ثبت می گردد.

supplier	
	id
	name
	contactperson
	address
	mobilen1
	mobilen2
	phoneno

تصویر ۱۰ - شرکت های پخش دوا و مشخصه های آن

مشخصات نسخه/فاکتور (Bill Details)

در این جدول دو ستون، اطلاعات خود را از دیگر جداول دریافت میکنند که در حقیقت قیمت کلید اصلی یکی از اشیا موجود در جدول دیگر باید در این ستون ها ذخیره شوند. مشخصه اول آی دی دوا هایی می باشد که قرار است در یک نسخه قرار گیرد و ستون بعدی آی دی نسخه/بل در آن قرار میگیرد. دو ستون دیگر، تعداد دوا و قیمت واحد بوده که در حقیقت بیانگر تعداد دواى تجویز شده و قیمت خرید/فروش دوا در آن ثبت می گردد.

bill_details	
id	
drugid	
billid	
quantity	
unitprice	

تصویر ۱۱ - جزئیات نسخه و مشخصه های آن

نسخه/فاکتور (Bill)

ستون ها در بل متفاوت از تمامی دیگر جداول می باشد. ستون های اصلی که در این جدول مد نظر گرفته شده است نوعیت بل و ستون آی دی که در تمامی جداول تکراری آمده است می باشد. چهار مشخصه ی دیگری که برای بل مد نظر گرفته شده است، آی دی دیگر جداول می باشد که کلید خارجی نیز یاد می شود. آی دی بیمار، آی دی انبار، آی دی شرکت پخش و آی دی دکتر چهار ستونی می باشد که در این جدول به دلیل ارتباطی که با دیگر جداول دارد مد نظر گرفته شده است.

bill	
id	
patientid	
inventoryid	
supplierid	
doctorid	
billtype	

تصویر ۱۲ - بل و مشخصه های آن

نوعیت بل ها می تواند در هر رکورد متفاوت باشد. به طور مثال یک (نسخه، فاکتور خرید اجناس و یا کوتیشن) می تواند باشد.

در صورتی که بل ما از نوع فاکتور باشد، آی دی مریض و آی دی پزشک هیچ گونه قیمتی در یک رکورد نمیگیرد و در مقابل در صورتی که نسخه باشد، آی دی شرکت پخش به هیچ عنوان قیمتی نمیگیرد. برای کوتیشن نیز میتوان گفت که تنها آی دی شرکت پخش قیمت میگیرد و باقی آن کاملاً خالی می ماند.

نرمال سازی ها

سیستم های پایگاه داده به دو نوع OLTP^۳ و OLAP^۴ می باشد. در سیستم های OLTP اطلاعات همه روزه تغییراتی چون (افزودن، تغییر، حذف) اعمال می شود و در این سیستم ها اطلاعات نباید به شکل تکراری وارد شود، چراکه سرعت، دقت و صحت اطلاعات در دست داشته کاهش و یا حتی امکان اشتباه کامل آن نیز رخ می دهد.

مزایای نرمال سازی

حذف داده های تکراری، جلوگیری از نوشتن کد های اضافی که داده های تکراری را یکسان سازد، کم عرض بودن جداول و سهولت در افزودن اطلاعات، افزایش به کار گیری ایندکس های کلاستر که باعث افزایش سرعت Join ها در پایگاه داده می شود و بسیاری موارد دیگر از مزایای نرمال سازی می باشد.

به جدول زیر دقت کنید:

آی دی	نام بیمار	تخلص	موبایل	نام دوا	تعداد	قیمت فروش
۱	فرهاد	فراز	۰۷۹۹۸۸۷۷۶۶	آموکسیسیلین	۲	AFN ۱۵۰
۱	فرهاد	فراز	۰۷۸۸۷۷۶۶۵۵	پاراسیتامول	۲	AFN ۳۰
۲	کریم	کریمی	۰۷۸۶۷۶۶۶۵۶	آموکسیسیلین	۲	AFN ۱۵۰
۳	فرهاد	فراز	۰۷۹۹۸۸۷۷۶۶	دگزامتازون	۴	AFN ۸۰

جدول ۲ - نمونه جدول پیش از نرمال سازی

^۳ On-Line Transaction Processing

^۴ On-Line Analytical Processing

در صورتی که در جدول بالا نام فرهاد به اشتباه وارد شده باشد، برای تغییر آن چند ردیف را باید ویرایش نمود؟ در عین حال برای حذف آن چگونه؟ شماره تماس های ثبت شده کدام یک دارای صحت است؟ در صورتی که فرهاد بیش از یک شماره تماس داشته باشد چگونه؟ حال با توجه به قوانین نرمال سازی که اولین بار توسط Edgar Frank Codd معرفی شد نرمال سازی ها را میتوان شروع کرد.

قابل ذکر است که تمامی مراحل نرمال سازی ها در طراحی دیاگرام حاضر مد نظر گرفته شده است. اما برخی از مفاهیم نرمال سازی ها نظر به این که حجم اطلاعات در یک سیستم دواخانه به حد اعلا نمیرسد، مد نظر گرفته نشده است. به طور مثال شماره تماس های شرکت های پخش دوا در دو ستون مد نظر گرفته شده است که در صورت خالی ماندن آن هیچ تاثیری در قسمت فضای اشغال شده محسوس نمی شود. چرا که تعداد رکورد هایی که برای شرکت پخش ثبت می شود بیش از ۱۰ نخواهد شد. اما در صورتی که شرکت پخش بیش از دو شماره تماس داشته باشد قطعاً موضوع کمی پیچیده خواهد شد و طراح دتایس موظف به اعمال نرمال سازی می باشد.

گزارش های اس کیو ال

با توجه به این که در تکنولوژی استفاده شده برای ساخت رابط کاربری این نرم افزار نیازی به نوشتن گزارش های ساده نیست اما باز هم به شرح برخی از این گزارش ها می پردازیم.

در تمامی پایگاه های اطلاعات چهار عملیه Create, Read, Update, Delete که به نام CRUD نیز یاد میشود امری معمول است که باید طراحان نظر به نیاز خود و در مواقع ضرورت از آن استفاده کنند.

عملیه Create در حقیقت افزودن یک رکورد جدید از اطلاعات در یکی از جداول می باشد که در ساختار زبان اس کیو ال به شکل زیر باید نوشته شود:

`insert into <table name> (cul1, cul2, culN) values (value1, value2, valueN)`

حال برای مثال برای افزودن یک پزشک باید از دستور ذیل استفاده کرد:

INSERT INTO doctor (name, lastname, degree, mobileno, address) VALUES ('
فاروق', 'حسینی', 'متخصص چشم', '0700xxxxxx', 'سرک ۶۴ متره غوردرواز')

عملیه Read با دستور Select (col1, col2, colN) from <table name> قابل اجرا می باشد. به طور

مثال تمامی پزشک های ثبت شده در پایگاه داده را میتوان اینگونه خروجی گرفت:

SELECT id, name, lastname, mobileno, degree, address, photo FROM doctor

اما در صورتی که خواسته باشیم نام یک پزشک خاص و یا مجموعه ای با مشخصات خاصی را از جداول خروجی

بگیریم از گزینه where استفاده میکنیم.

SELECT * FROM doctor WHERE id = 1 OR name = 'سید فاروق'

در دستور فوق، پزشکی که با آی دی ۱ باشد و یا نام او برابر با سید فاروق باشد خروجی گرفته خواهد شد. علامت

ستاره به معنای انتخاب تمامی ستون هایی می باشد که در جدول پزشک تعریف شده است.

دستور Update نیز یکی از پر کاربرد ترین دستورات در طراحی تمامی دتایس ها می باشد. این دستور به منظور

اعمال تغییرات در رکورد هایی می باشد که از پیش به نوعی اشتباه ثبت شده باشند.

برای اعمال تغییرات باید مجموعه و یا رکورد مورد نظر انتخاب گردد و تغییرات جدید بر روی آن اعمال شود.

Update <table name> SET (col1 = 'value1', col2 = 'value2', colN = 'valueN') where <condition>

حال نام پزشکی که با آی دی شماره ۱ در پایگاه داده ما ثبت است را توسط دستور update در ستون شماره

تماس تغییرات خویش را اعمال میکنیم:

UPDATE doctor SET mobileno = '07xxxxxxx' WHERE id = 1

دستور delete نیز یکی دیگر از دستورات پر کاربرد می باشد که برای حذف یک رکورد می باشد.

delete <table name> where <condition>

حال برای حذف یک پزشک از دستور زیر میتوان استفاده نمود:

DELETE doctor WHERE id = 1

گزارش های محاسباتی

موجودی انبار

در صورتی که به دیاگرام پایگاه داده نظری بیندازید خواهید دید که بعد از ایجاد چندین فاکتور و نسخه و قرار

دادن دواهای مختلف در هر فاکتور و یا نسخه نمیتوان به راحتی تعداد دوا هایی که در سیستم موجود است را به

دست آورید. در اینجا با نحوه نوشتن گزارشی که بتوان میزان دواهای موجود در سیستم را محاسبه کند آشنا خواهیم شد. به تصویر زیر دقت کنید:



تصویر ۱۳ - جداول مرتبط برای دریافت موجودی انبار

تنها ستونی که در جدول بل (bill) کلید نیست، ستون billtype می باشد که مقادیر مختلف چون (نسخه، فاکتور، کوتیشن) را میتواند در خود ذخیره کند. حال در صورتی که بل صادر شده از نوع نسخه باشد به معنای خروج دوا از دواخانه و در صورتی که بل از نوع فاکتور باشد معنای ورود دوا به دواخانه را می دهد و در مقابل جدول bill_details می باشد که رابطه یک به چند را با جدول بل دارد؛ به گونه ای که هر بل می تواند انواع دواهای مختلف با مقدار مختلف را در بل های مختلف در خود جای دهد. حال برای این که تعداد هر نوع از دواهای فروخته شده در دتابیس را بخواهیم محاسبه کنیم، تمامی ستون های مورد نیاز از سه جدول فوق را باید با استفاده از دستور select انتخاب نموده و با اعمال شرط های مورد نیاز موجودی دوا هایی که در نسخه ها به فروش رسیده است را به دست آوریم. از آنجایی که مقدار دوا ها را می خواهیم پیدا کنیم، ستون هایی که نیاز است برایمان نمایش داده شود (companyname, realname) را انتخاب می نمایم و یک ستون محاسباتی با استفاده از فانکشن sum که ستون تعداد (quantity) را جمع کند را ایجاد میکنیم. بعد از آن نام هر سه جدول (bill_detail bd, drugs d, bill b) را بعد انتخاب می کنیم. در قسمت شرطیه انتخاب ها باید صرف بل هایی که مقادیر ورودی آن معادل به نسخه باشد، ستون b.id معادل با bd.billid باشد و drugid برابر با d.id

باشد. حال استفاده از دستور Group By تمامی دواهای فروخته شده را گروپ کرده و دستور خود را اجرا میکنیم.

```
SELECT  companyname, realname,
        SUM(quantity) TotalSold,
        b.billtype
FROM    bill_details bd ,
        drugs d ,
        bill b
WHERE   b.billtype = N'نسخه'
        AND bd.billid = b.id
        AND drugid = d.id
GROUP BY companyname , realname, b.billtype
```

حال برای این که موجودی دواي فروخته شده را به دست آورید صرف در قسمت شرط دستور بالا به جای نسخه از فاکتور استفاده کنید. آنگاه تمامی دوا هایی که از شرکت های پخش خریداری کرده اید نمایش داده خواهد شد.

کار ما تا به اینجا تمام نشده است. چرا که باید مقدار دواهای فروخته شده را از مقدار دواهای خریداری شده کم نموده تا موجودی دقیق انبار پیدا شود. نظر به نوعیت بل، در صورتی که فاکتور باشد قیمتی عددی ما مثبت و در صورتی که نوعیت بل نسخه باشد، قیمت عددی ما منفی تلقی می شود، در فرمول sum که در دستور بالا نوشته شده است از جمله شرطیه case استفاده نموده و در صورتی که نوعیت بل، فاکتور باشد ضرب عدد ۱ و در صورتی که نوعیت بل نسخه باشد، در عدد ۱- ضرب میکنیم؛ در قسمت شرط تمامی بل هایی که نسخه یا فاکتور باشند را انتخاب می کنیم.

```
SELECT TOP ( 100 ) PERCENT
        d.companyname ,
        d.realname ,
        SUM(( CASE WHEN b.billtype = N'
        THEN 1
        THEN -1
        END ) * bd.quantity) AS available
FROM    dbo.bill_details AS bd
INNER JOIN  dbo.bill AS b ON bd.billid = b.id
INNER JOIN  dbo.drugs AS d ON bd.drugid = d.id
WHERE   ( b.billtype = N'
        ' )
        OR ( b.billtype = N'
        ' )
GROUP BY d.companyname ,
        d.realname
ORDER BY d.companyname
```

حال خواهیم دید که تمامی دواهای موجود در سیستم محاسبه شده و در ستون available نمایش داده می شود.

	companyname	realname	available
1	Aderonax	Monsipewentor	-6
2	Adnipepentor	Adnipefazz	-52
3	Adpeban	Repebiman	55
4	Adtininex	Barobommicator	-59
5	Adtumower	Tippickaquax	-26
6	Ciperanor	Lomdimexicator	6
7	Cipvenimin	Upquestilistor	-78
8	Dopnipar	Inpebonax	-93
9	Dopweror	Repickower	115
10	Emquestegicator	Tiptumover	53
11	Emzapazz	Vardudewistor	155
12	Endowastentor	munor	25

تصویر 14 - نتیجه گزارش موجودی انبار در نرم افزار SSMS

دلیل این که قیمت منفی در این جدول ثبت شده است اطلاعاتی است پیش از پیاده سازی Validation ها بوده است.

فصل چهارم – معرفی ابزارها و نحوه ساخت رابط کاربری

معرفی و نحوه ساخت رابط کاربری

ابزار های مورد استفاده

تا به حال میتوان گفت که تمامی کارهایی را که انجام داده ایم در ساختار زیربنایی پروژه مان بوده است. اما حال تلاش برای ساخت رابط کاربری (User Interface) یا UI برای کنترل سیستم را انجام خواهیم داد. نظر به این که پایگاه داده ما از شرکت مایکروسافت می باشد، ابزار هایی را که نیز برای ساخت رابط کاربری نرم افزار مان استفاده نموده ایم را بر اساس آخرین تکنولوژی های روز شرکت مایکروسافت انتخاب کرده ایم. تکنولوژی Silver Light، زبان برنامه نویسی C# (سی شارپ) و ابزار قدرتمند Light Switch که محصولی دیگر از شرکت مایکروسافت برای ساخت نرم افزار با سرعت بالا (Rapid Development) می باشد را به کار گرفته ایم. توسط این ابزار ها به راحتی میتوان فرم های مختلف را با قابلیت CRUD تنها با چند کلیک ساده آماده ساخت. در قسمت های بعدی این منوگراف با نحوه کار با LightSwitch آشنایی کامل با چگونگی ساخت نرم افزار های تجاری مختلف آشنا خواهید شد.

نرم افزار Visual Studio

Visual Studio یک مجموعه از برنامه هایی است که ارتباط بسیار نزدیک با هم دارند که مایکروسافت آن را به توسعه دهندگان و برنامه نویسان برنامه های کاربردی اهدا نمود تا آنها را وادار نماید در محیطی توسعه یافته بر روی پلت فرم های ویندوز و دات نت به ساخت برنامه های خود بپردازند. نرم افزار Visual Studio می تواند برای نوشتن برنامه های کنسولی، ویندوزی، سرویس های ویندوز، برنامه های کاربردی موبایل، برنامه های کاربردی ASP.NET و سرویسهای وب ASP.NET بنا به انتخاب شما همراه با زبانهایی مانند C#, C++، VB.NET، J# مورد استفاده قرار گیرد. با Visual Studio واقعا چه کارهایی میتوان انجام داد؟ در زیر تعدادی از کاربردهایی را که برای تولید آنها میتوان از Visual Studio استفاده نمود معرفی گردیده اند:

تکنولوژی WPF: برای برنامه های کاربردی ویندوزی که با استفاده از NET framework نوشته میشوند. نرم افزار ویژوال استدیو بی نظیر است، همچنان Windows Services یا سرویس ها برنامه های کاربردی هستند که در پس زمینه ویندوز اجرا میشوند. ASP.NET یک تکنولوژی قدرتمند که برای طراحی و ساخت صفحات وب دینامیک استفاده میشود. علاوه بر آن نیز، ابزار های قدرتمند دیگری نظیر Light Switch نیز برای کاربرانی که می خواهند با سرعت بالا نرم افزار های تجاری را ایجاد کنند به آن اضافه شده است.

زبان برنامه نویسی C#

زبان C# یک زبان سطح بالا، شی گرا و همه منظوره است که به وسیله شرکت مایکروسافت همزمان با پیدایش لایه نرم افزاری جدید آن به نام NET. ابداع و توسعه پیدا کرده است. از نرم افزارهای متنوع و گوناگونی از جمله نرم افزارهای اداری و برنامه های کاربردی تحت وب گرفته تا نرم افزارهایی برای تلفن همراه و بازی های کامپیوتری، با زبان C# و با استفاده از لایه NET. تولید می شود.

زبان C# شباهت زیادی به زبان های C و Java دارد و ویژگی هایی را از آنها تقلید کرده، یا بعضی امکانات آنها را بهبود داده است. تلاش شده است که بهترین ویژگی ها گردآوری شود، اما برخلاف زبان جاوا که متن باز است، C# در انحصار و اختیار سازنده آن یعنی شرکت مایکروسافت است. زبان های C و Java هر دو به زبان C بر می گردند که در سال ۱۹۷۰ ابداع شد و معروفیت آن به دلیل نوشتن سیستم عامل UNIX به وسیله آن بود. زبان C، یک زبان حرفه ای است که دست برنامه نویس را برای نوشتن برنامه و دسترسی به سخت افزار، باز می گذارد و دارای انعطاف بسیار زیادی است، به همین دلیل کمتر اشتباهات منطقی برنامه نویس را کنترل می نماید. اما در زبان C#، در هنگام ترجمه و همچنین اجرای برنامه دقت زیادی بر روی تطبیق و به کارگیری داده ها صورت می گیرد تا از اشتباهات دستوری برنامه نویس یا کاربر جلوگیری نماید.

همان طور که برای تهیه و پخت یک غذا، به مواد اولیه، لوازم آشپزی و دستور پخت نیاز داریم، برای تولید یک برنامه نیز، به یک کامپیوتر یا لپ تاپ، لوازم برنامه نویسی (یک ویرایش گر متنی و یک برنامه مترجم) و همچنین به یک الگوریتم نیاز داریم. اگر یک کامپیوتر با سیستم عامل ویندوز ۷ یا بالاتر در دسترس باشد تقریباً تمام مواد

اولیه و لوازم مورد نیاز را در اختیار داریم. ناگفته نماند، در صورتی که نرم افزار Visual Studio را روی سیستم خود داشته باشید تمامی امکانات برنامه نویسی با زبان های مبتنی بر تکنولوژی NET. مایکروسافت را دارا می باشید.

Silverlight

Microsoft Silverlight ابزاری است که به راحتی می تواند بر روی انواع سیستم عامل ها و انواع نرم افزار های جستجوی اینترنت نصب و اجرا شود و مایکروسافت این ابزار را برای سازندگان برنامه های کاربردی اینترنت طراحی کرده است. Silver Light هم اکنون به صورت رسمی عرضه شده است و امکانات پیشرفته نرم افزاری نظیر تعامل با کاربران، نمایش انیمیشن و همچنین پخش فایل های صوتی و تصویری را پشتیبانی می کند. هم اکنون در محیط ویندوز، با مرورگر های IE و Firefox سازگار بوده و ضمناً با مرورگر Safari تنها در سیستم عامل Mac OS X سازگار است.

معرفی Microsoft Light Switch

اغلب برنامه های تجاری که امروزه ساخته می شوند به نحوی با داده ها سر و کار دارند. نشان دادن اطلاعات به کاربر، کیوری، گزارش گرفتن اطلاعات، گزارش گیری (Report) و کار کردن با دیتابیس های مختلف بخش هایی از تمامی برنامه های تجاری امروزی می باشد.

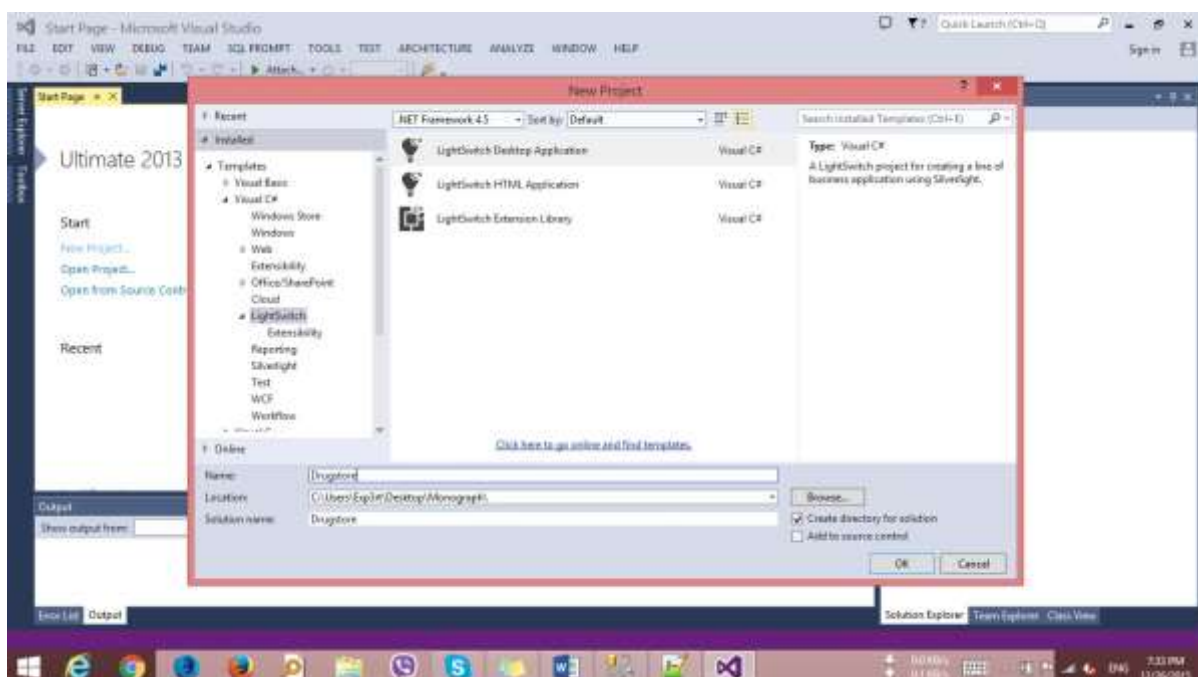
تکنولوژی **Light Switch** در ویژوال استودیو بستری برای طراحی و ساخت برنامه های تجاری به صورت سریع و ساده است. تکنولوژی Light Switch به ما این امکان را می دهد که به صورت سریع و آسان برنامه های داده محور ایجاد کنیم. برنامه هایی که از دیتابیس های مختلف استفاده می کنند، اغلب با فراخوانی سرویس های مختلف، داده های مورد نظر را پردازش می کنند. از دیگر قابلیت های Light Switch فراهم کردن برنامه ای است که بر روی اغلب پلت فرم و دستگاه ها قابل استفاده باشد. با استفاده از Light Switch می توان:

- برنامه های تحت وب را بر اساس HTML5 طوری طراحی کرد که بر روی اکثر دستگاه ها قابل استفاده باشند.
- از دیتاسورس های مختلف از قبیل SharePoint، SQL و OData استفاده کرد.
- از کد نویسی های روزمره در کار کردن با داده ها جلوگیری کرد.
- از سیستم های مبتنی بر Cloud از قبیل Azure و Office 365 استفاده کرد.
- برنامه ی مورد نظر را با استفاده از جدیدترین تکنولوژی ها و روش های ساخت برنامه ایجاد کرد.

ساخت پروژه جدید با LightSwitch

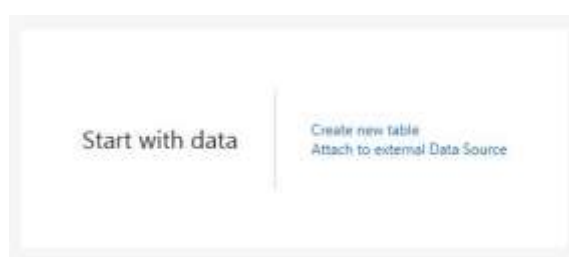
برای ساخت رابط کاربری نرم افزار با LightSwitch ابتدا Visual Studio را باز نموده و پس از آن اقدام به ساخت یک پروژه جدید خواهیم کرد.

New Project -> Visual C# -> LightSwitch -> LightSwitch Desktop Application



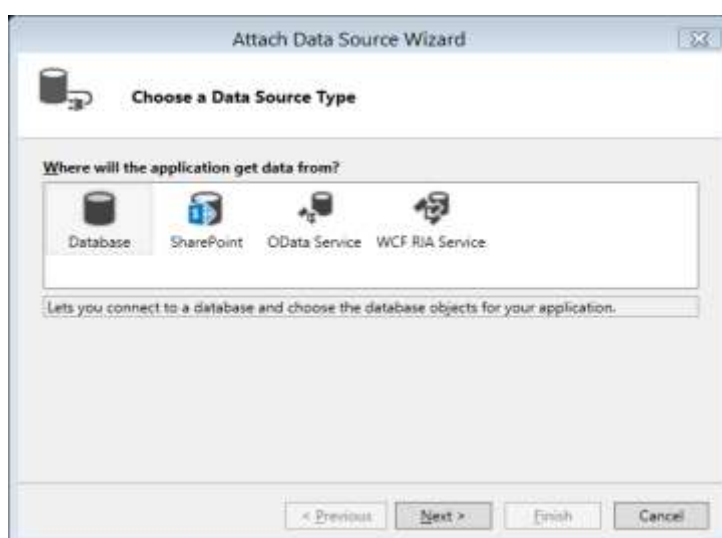
تصویر ۱۵ - نحوه ساخت پروژه جدید در LightSwitch

پس از این که پروژه به طور کامل ساخته و در سیستم ذخیره شود در اولین نگاه با صفحه Start در میان صفحه و Solution Explorer نیز در سمت راست صفحه ظاهر خواهد شد. در تب دو گزینه ای که میتوان از آن استفاده کرد Create New Table و Attach to External Data Source می باشد. گزینه اول قابلیت ساخت جداول بدون نیاز به دتاییس خارجی می باشد و بیشتر در نرم افزار هایی که تحت وب از آن استفاده می شود کاربرد دارد. با این گزینه میتوانید تمامی جداول و رابطه هایی که بین جداول مد نظر تان است را ترسیم نموده و به دیگر ابزار ها هیچ نیازی نداشته باشید. (برای Desktop Application هایی که تک کاربره هستند، پیشنهاد می شود ساخت جداول را در قالب خود نرم افزار ایجاد نمایید)



تصویر ۱۶ - دو انتخاب برای ایجاد دتاییس در LightSwitch

حال با استفاده از Attach to external Data Source دتایسی را که در فصل قبل ایجاد کرده بودیم را به پروژه خود می افزاییم.



تصویر ۱۷ - نحوه ارتباط دتاییس در SSMS با LightSwitch

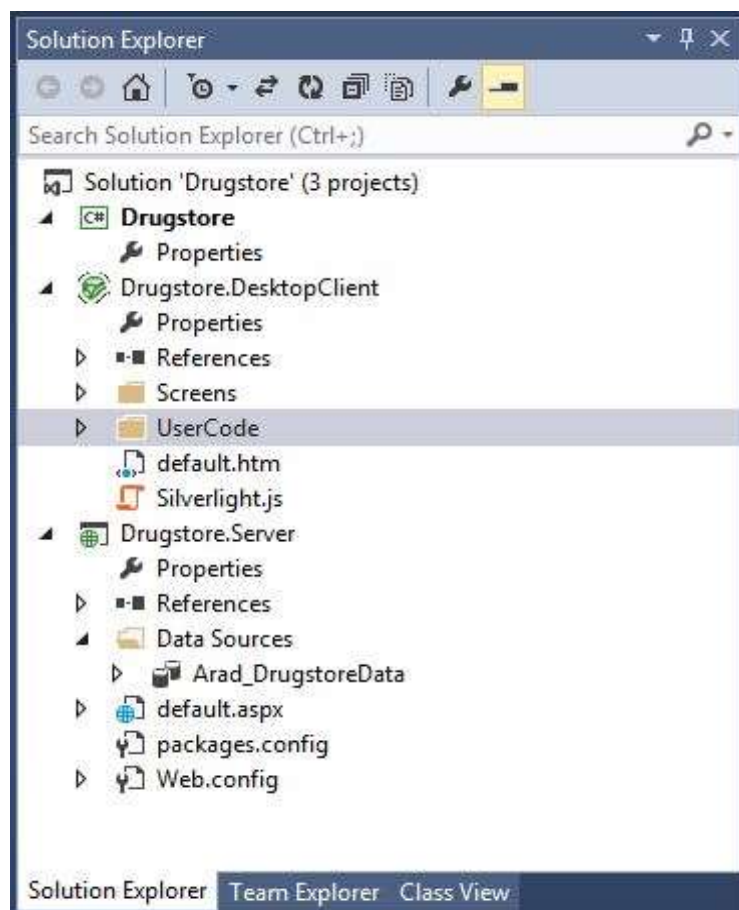
حال پس از انتخاب منبع اطلاعاتی خود کلید Next را فشار داده و در پنجره باز شده ابتدا نام سرور و آدرس آن را تنظیم نموده و اطلاعات ورود به سرور را وارد مینمایید و در قسمت آخر نام پایگاه داده ای را که پیش از آن ساخته اید را انتخاب نموده و کلید OK را فشار دهید.

تصویر ۱۸ - ایجاد ارتباط با SSMS و انتخاب دتابیس ساخته شده

حال با انتخاب تمامی جداول و نمایه هایی که در پایگاه داده ساخته ایم را انتخاب کرده و به پروژه خود اضافه نموده و دکمه Finish را می فشاریم. حال پروژه ما با دتابیس ارتباط برقرار کرده و تمامی ستون ها و رابطه ها را با تمامی جزئیات در خود گنجانیده است.

حال به قسمت Solution Explorer که یکی از قسمت های پر کاربرد و مهم کار بر روی پروژه مان هست می پردازیم.

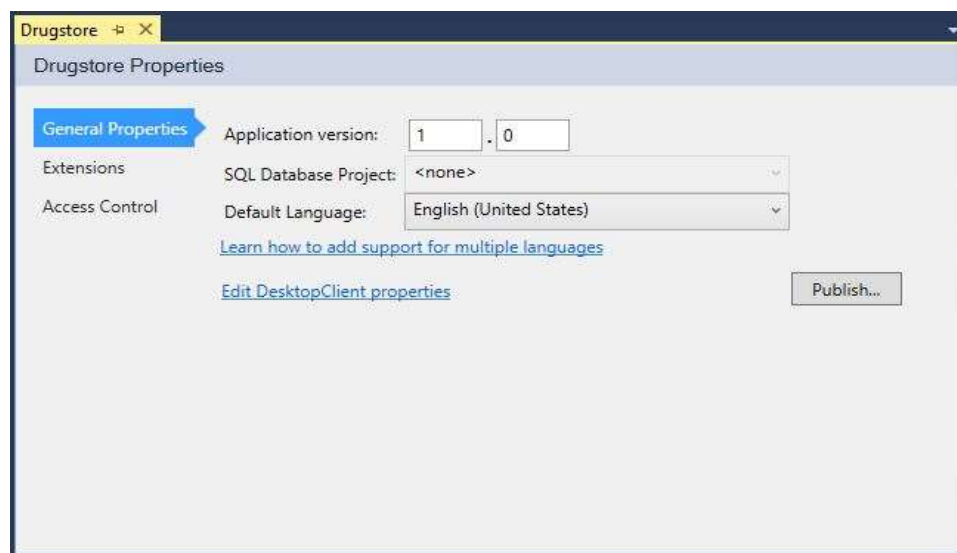
در Solution Explorer که سه پروژه را در خود گنجانیده است، قسمت اول آن Drugstore، قسمت دوم آن Drugstore.DesktopClient و قسمت سوم آن Drugstore.Server می باشد که هر کدام وظایف خاص خود را دارا می باشند.



تصویر ۱۹ - ساختار درختی پروژه LightSwitch و زیر مجموعه ها

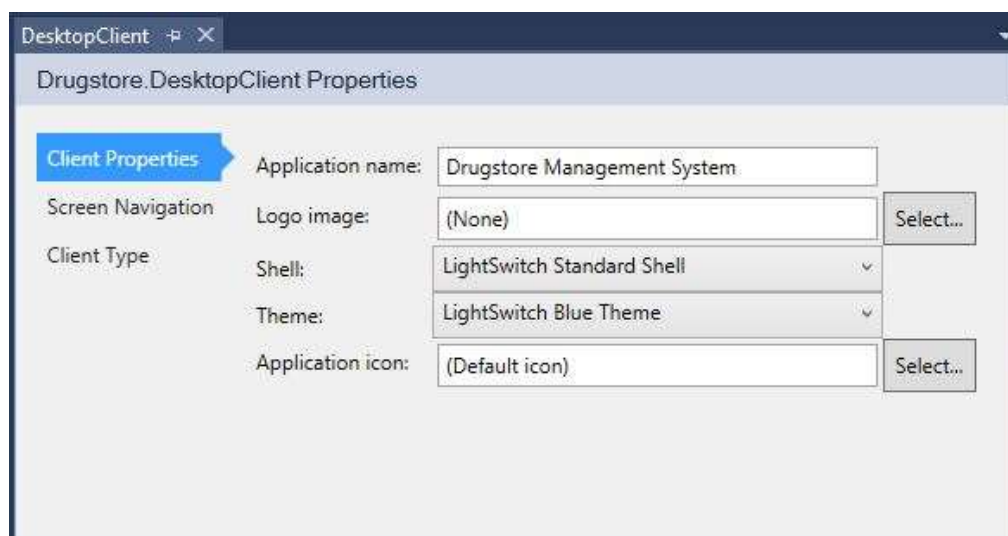
در قسمت Drugstore -> Properties مشخصات کلی سیستم (General Properties) میباشد که شامل ورژن نرم افزار استخراج پروژه به صورت فایل exe. و زبان رابط کاربری نرم افزار می باشد. قسمت بعدی آن Extensions قرار دارد که افزونه های مورد استفاده در سیستم را از این قسمت می توان فعال و یا غیر فعال

ساخت و در نهایت Access Control که سطح دسترسی کاربران به قسمت های مختلف سیستم تعریف شده و هر کاربر نظر به روادید مدیر سیستم می تواند برای کاربرانی که میخواهند از سیستم استفاده کنند سطوح مختلف را ایجاد کند.



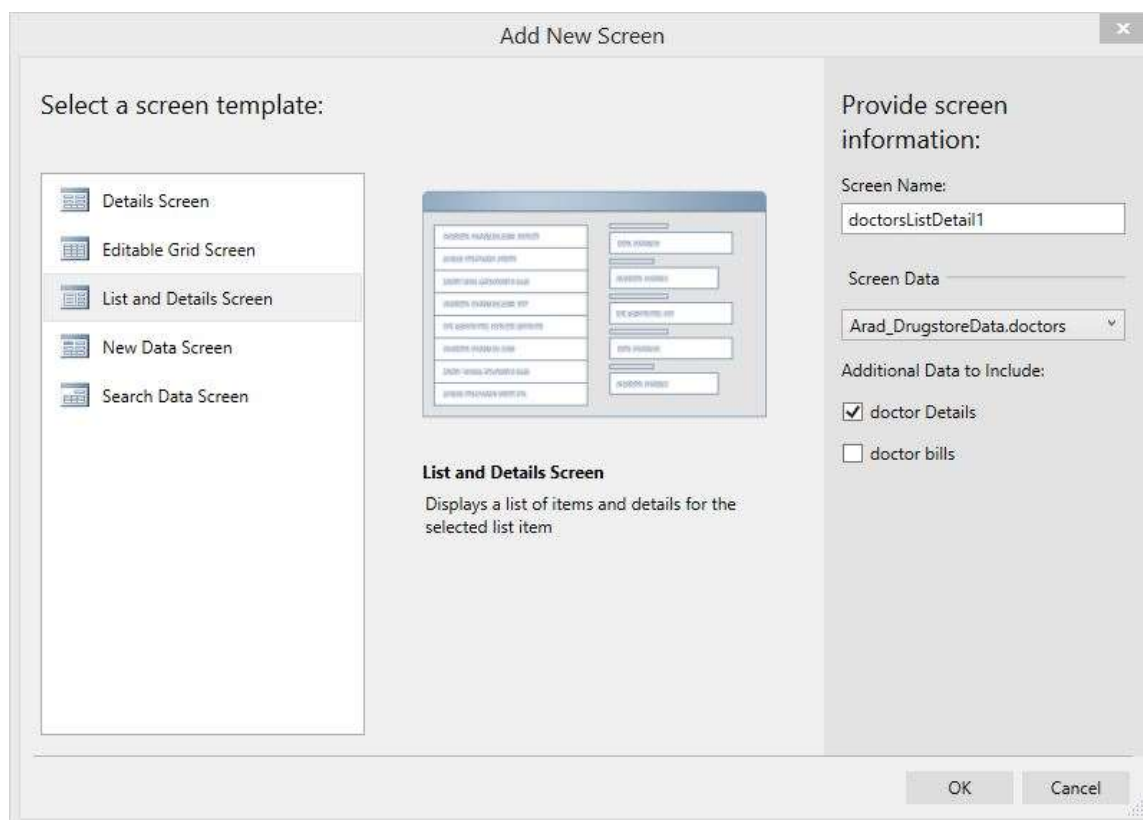
تصویر ۲۰ – پنجره Properties سیستم دواخانه

قسمت بعدی Drugstore.DesktopClient دارای چندین بخش مختلف که از جمله مهمترین بخش های رابط کاربری می باشد. چراکه نام نرم افزار، لوگو، قالب های طراحی نرم افزار، Navigation و نوعیت خروجی نرم افزار که تحت وب و یا به شکل نصبی باشد را میتوان انتخاب نمود.



تصویر ۲۱ – پنجره Properties رابط کاربری سیستم مدیریت دواخانه

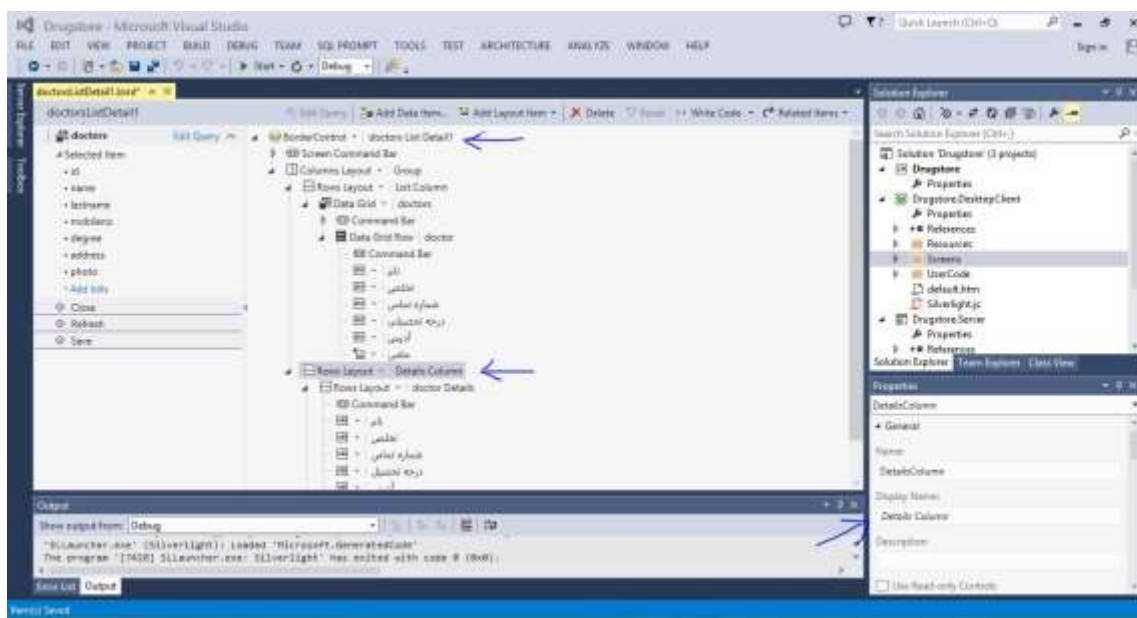
حال برای ساخت صفحات برای ورود اطلاعات از به قسمت نیاز بر آن است تا روی پوشه Screen که در قسمت Drugstore.DesktopClient قرارداد دارد راست کلیک نموده و با کلیک کردن بر روی Add Screen اقدام به ساخت اولین قسمت از رابط کاربری برای ورود اطلاعات مربوط به یک پزشک نمایم.



تصویر ۲۲ – نحوه ساخت Screen جدید در LightSwitch

از لیست نمونه های Screen های ساخته شده List and Details Screen را انتخاب نموده و پس از انتخاب نام برای Screen جدول دکتر را انتخاب و دکمه Ok را بفشارید. حال اولین فرم که قابلیت نمایش تمامی پزشک های ثبت شده در دتابیس به شمول افزودن، حذف، ویرایش و خروجی گرفتن در یک فایل اکسل را دارا می باشد ساخته اید.

برای این که فرم خود را به زبان فارسی ترجمه نمایید، باید بر روی تک تک ستون های جداول کلیک نموده و نامی را که باید برای کاربر نمایش دهد تغییر دهید. همچنان برای راست چین کردن فرم رود اطلاعات با استفاده از افزونه Border Control اقدام به راست چین کردن فرم خود می کنید.

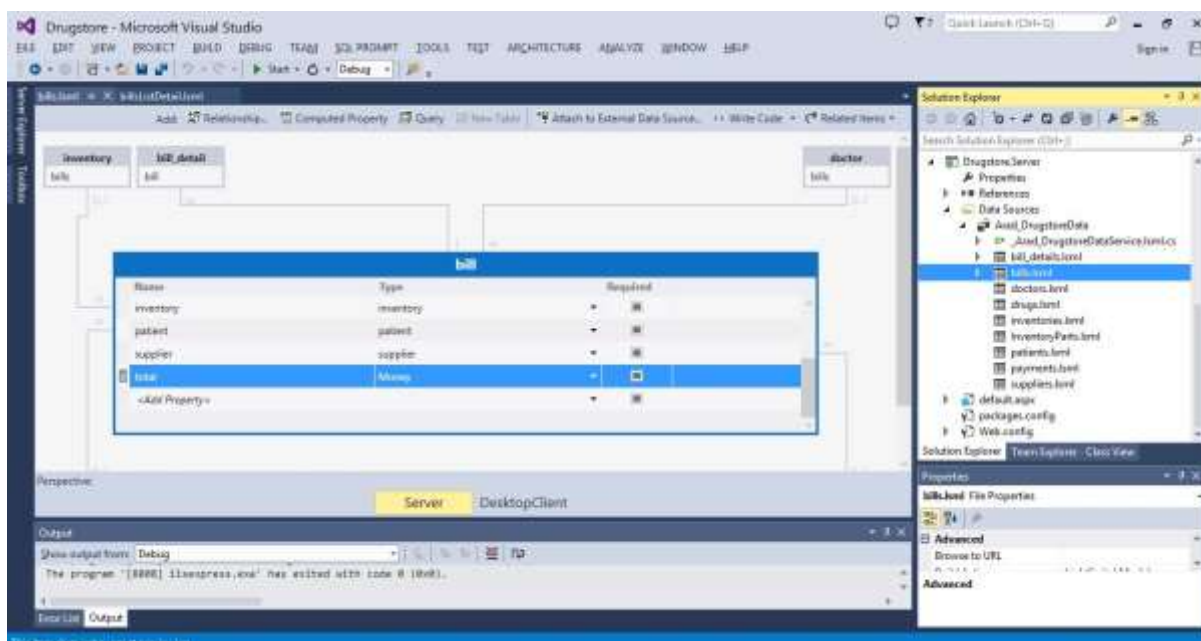


تصویر ۲۳ - راست چین کردن و تغییر نام فرم ها

حال به همین منوال تمامی Screen هایی که برای پروژه خود نیاز دارید را ساخته و به طور اتوماتیک در منوی سیستم اضافه خواهد شد.

افزودن ستون های محاسباتی

حال نسخه و مشخصات نسخه که به نام های (bill and bill_details) در سیستم می باشد را پیش از اقدام به ساخت یک Screen ستون محاسباتی آن را در قسمت Dragstore.Server اضافه میکنیم و با استفاده از کد های سی شارپ قیمت کل برای نسخه و قیمت جزء برای هر دوا را محاسبه می کنیم.



تصویر ۲۴ – نحوه افزودن ستون های محاسبه ای

بر روی ستون های محاسبه ای که در جدول bill و bill_details اضافه نموده ایم برای ستون محاسبه ای bill که در تصویر بالا به نام total ثبت شده است از قسمت Write Code و متود total_Compute کد های مربوط به محاسبه آن را می نویسیم.

```
partial void total_Compute(ref decimal result)
{
    result = bill_details.Sum(b => b.unitprice * b.quantity);
}
```

در کد های بالا تعداد دوا در جدول bill_details را ضرب قیمت فروش آن نموده و مجموع آن را توسط متود sum در متغیر result ذخیره می نماییم.

همچنان برای نمایش ریز قیمت های دوا که به نام sub_total در جدول bill_detail تعریف شده است صرف تعداد دوا را ضرب قیمت ورودی می نماییم.

```
partial void subtotal_Compute(ref decimal result)
{
    // ریز قیمت های هر دارو در نسخه ها
    result = quantity * unitprice;
}
```

برای اعتبار دهی به قیمت هایی که توسط کاربران در سیستم وارد می شود نیز راه کار هایی موجود است که می توان آن ها را در یک جدول و یا یک ستون افزود. این قیمت ها با جملات شرطیه و در متود Validate می توان ایجاد نمود. طور مثال تعداد دوا در هیچ یک از نسخه ها نباید که کمتر از ۱ باشد. در صورتی که کاربر مقدار عددی صفر و یا عدد منفی را وارد نمود یک پیغام خطا باید کاربر را از این عملش متوجه سازد.

برای این کار در جدول bill_detail ستون quantity را انتخاب نموده و از قسمت Write Code متود quantity_Validate را انتخاب میکنیم. در قسمت کد نویسی برنامه با جمله شرطیه if کاربر را از نوع اشتباهی که در قسمت وارد کردن اطلاعات انجام داده است با خبر می سازیم.

```
partial void quantity_Validate(EntityValidationResultsBuilder results)
{
    if (this.quantity < 1)
    {
        results.AddPropertyError(
            "حداقل قیمت ورودی برای تعداد دوا باید ۱ باشد.");
    }
}
```

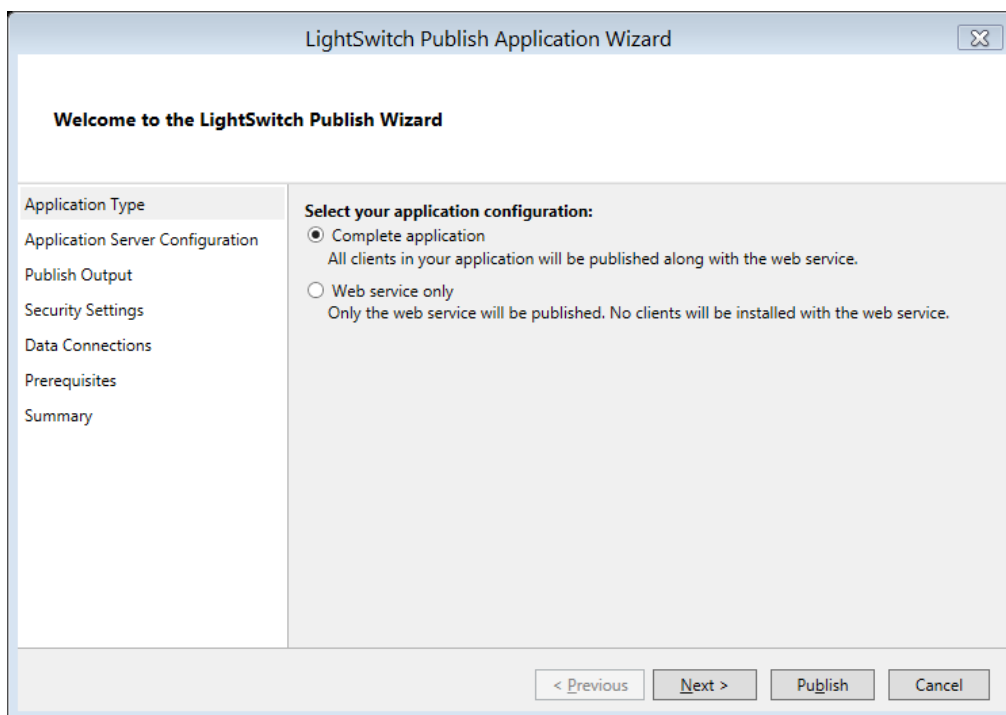
گزارش های چاپی

متاسفانه LightSwitch به طور پیش فرض

ساخت Setup نرم افزار

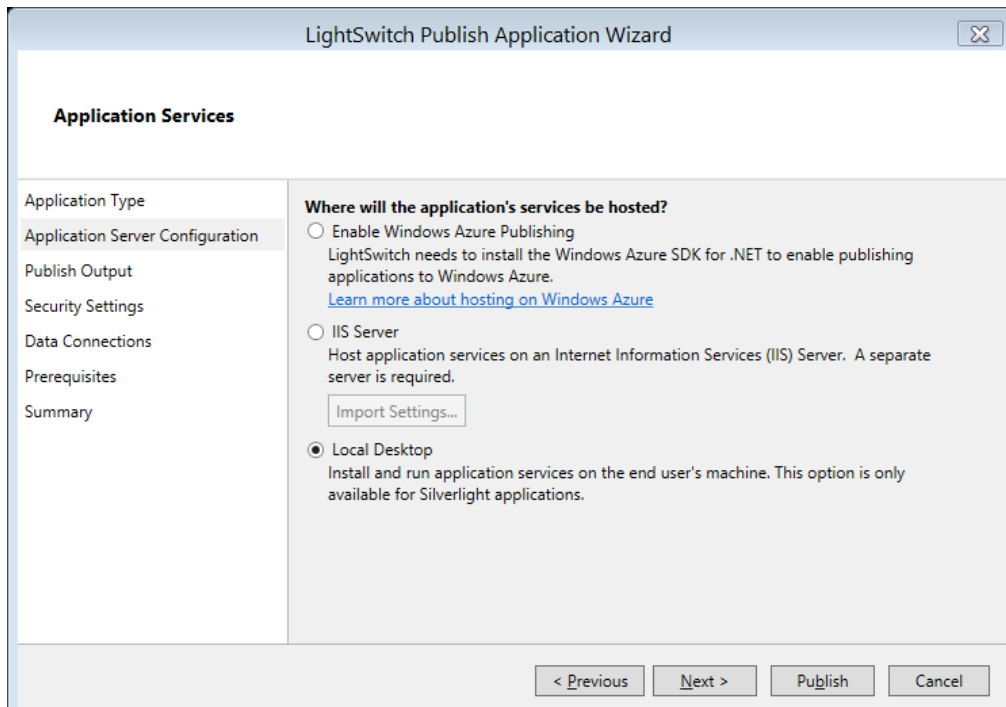
پس از ساخت کامل Screen ها و تمامی اعتبار سازی ها (Validation) حال نوبت به استخراج یا Publish کردن نرم افزار می پردازیم. برای شروع در Drugstore->Properties در Solution Explorer را باز نموده و با فشردن دکمه Publish بر طبق نیاز خود تنظیمات را وارد میکنیم.

در صفحه نوعیت برنامه گزینه Complete application را انتخاب نموده و کلید Next را بفشارید.



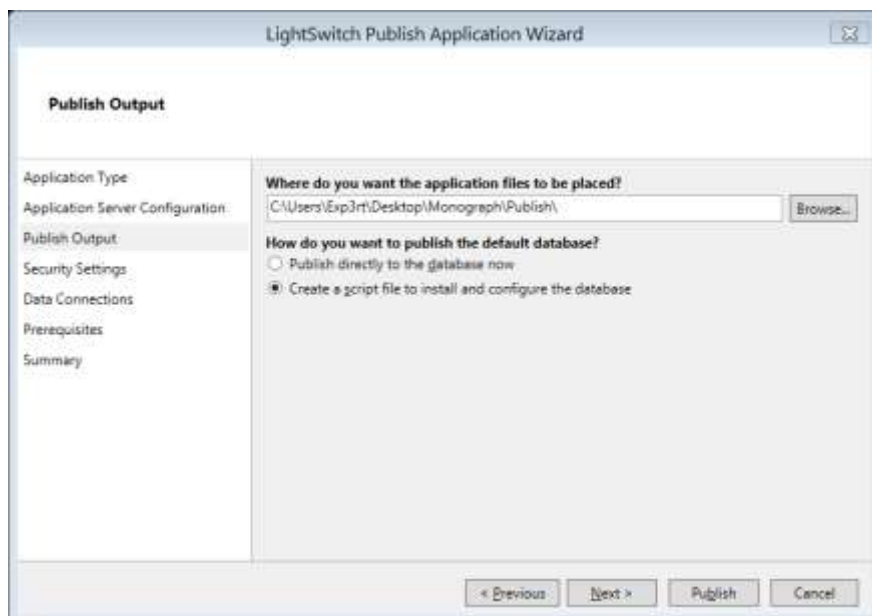
تصویر ۲۵ - انتخاب نوع خروجی نرم افزار

در صفحه بعدی گزینه پیش فرض Local Desktop را انتخاب نموده و Next را می فشاریم.



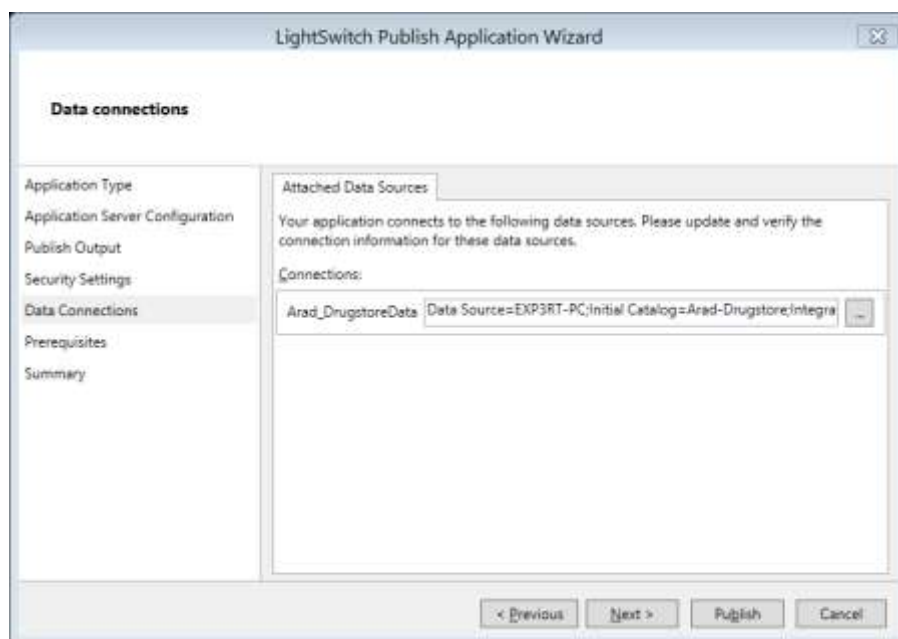
تصویر ۲۶ - انتخاب نوع سرویس ها

محل خروجی فایل Setup را در این قسمت باید تنظیم نماییم. به طور پیش فرض محلی که برای ذخیره سورس کد نرم افزار انتخاب کرده اید برای خروجی Setup انتخاب شده است که میتوانید تغییر دهید.



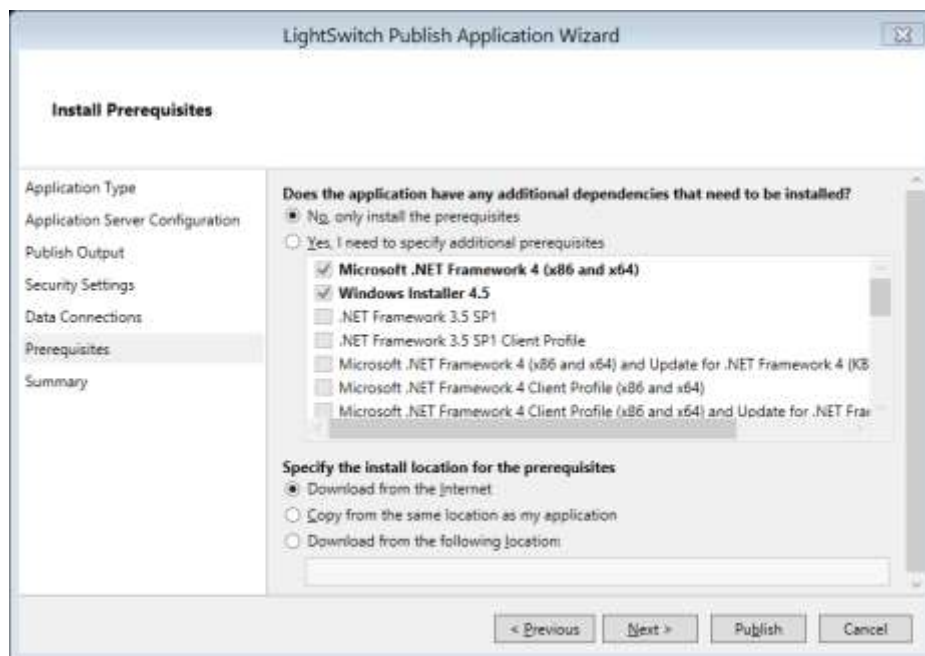
تصویر ۲۷ – محل ذخیره Setup نرم افزار

قسمت Security Setting را به حالت پیش فرض مد نظر گرفته و در قسمت Data Connections آدرس موقعیت دتاییس SQL و نام دتاییس که در آن ساخته اید را تنظیم نمایید.



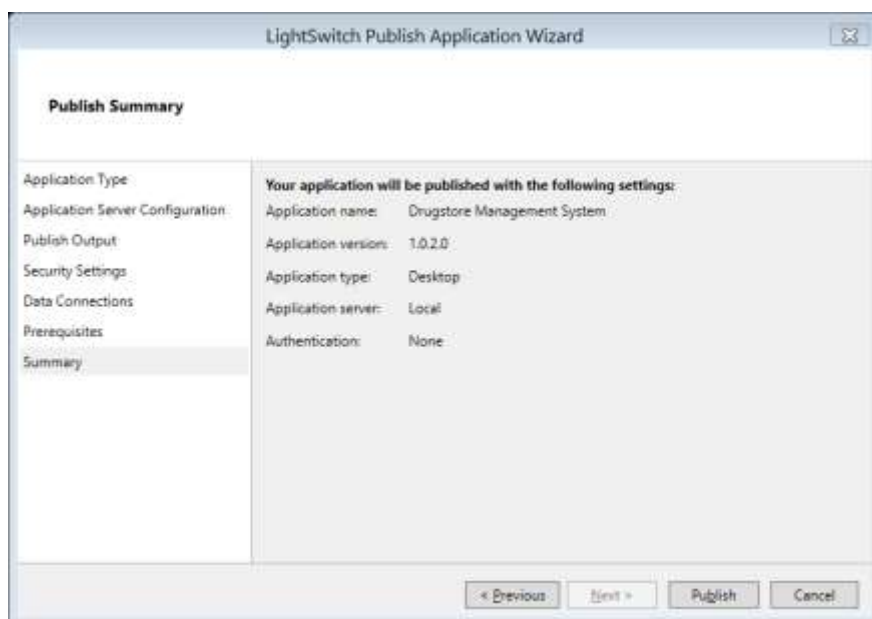
تصویر ۲۸ – مشخصات دتاییس شامل آدرس، نام دتاییس، نام کاربری و پسورد

در صفحه بعد نرم افزارهایی که وابسته به اجرای سیستم می باشد را می توانید انتخاب نمایید که هنگام نصب نرم افزار به کاربر آن ها بر روی سیستم کاربر برای اجرای پروژه هشدار دهد.



تصویر ۲۹ - انتخاب برنامه های مورد نیاز برای نصب برنامه

در نهایت صفحه Summary خلاصه ای از مشخصات سیستم را نمایش داده و با فشردن گزینه Publish نرم افزار در محل تعیین شده استخراج خواهد شد.



تصویر ۳۰ - مشخصات کلی از خروجی نرم افزار

فصل پنجم – مراحل نصب و نحوه استفاده از سیستم

برای کاربرانی که خواهان ساخت این پروژه به صورت عملی می باشند باید خاطر نشان شد که نرم افزار Visual Studio 2013 نیاز به 1.6 GHz و یا بالاتر از آن مورد ضرورت است. در قسمت حافظه رم در 1 GB و در صورتی که بر روی یک ماشین مجازی از آن استفاده می کنید 1.5 GB حافظه رم نیاز است. فضای خالی مورد ضرورت بر روی هارد دیسک 10 GB با فایل سیستم NTFS، سرعت دور هارد دیسک 5400 RPM و همچنان کارت گرافیک با قابلیت پشتیبانی از DirectX 9 و قدرت نمایش 1024 x 768 را دارا باشد.

اجرای دتایس برای هر دو کاربری که قصد طراحی و یا استفاده آن را دارند در این سیستم ضروری می باشد. حداقل سخت افزار مورد نیاز برای اجرای پایگاه داده SQL Express شرکت مایکروسافت برای رم 512 MB و برای دیگر ورژن های SQL Server بالاتر از 1 GB حافظه رم نیاز است. اما پیشنهاد شرکت مایکروسافت برای اجرای پایدار برای SQL Express حافظه 1 GB و برای دیگر ورژن ها 4 GB پیشنهاد شده است. پردازنده پیشنهادی مایکروسافت 1.4 GHz با معماری x64 یا همان ۶۴ بیتی می باشد و با پردازنده های AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T به خوبی سازگار می باشد.

سیستم عامل مورد نیاز برای نصب نرم افزار مدیریت دواخانه را میتوان بر روی انواع ورژن های ویندوز شرکت مایکروسافت از جمله ویندوز XP, 7, Vista, 7, 8, 8.1 و ویندوز ۱۰ نصب نمایید. نا گفته نماند که در قسمت Configuration Manager نوع رابط کاربری را نظر به نوع معماری پردازنده مورد نظر خود می توانید تنظیم کنید تا نرم افزار هنگام خروجی گرفتن صرف برای معماری مورد نظر شما که x86 و یا x64 ساخته شود.

برای نرم افزار هایی که با LightSwitch ساخته می شود چارچوب کاری NET. و همچنان پلاگین Microsoft Silverlight را چه برای ساخت سیستم و چه برای اجرا باید نصب نمایید. البته برای اجرای آن باید از ورژن SDK که در سایت مایکروسافت قرار دارد استفاده کنید. در مورد ورژن قابل نصب، پیشنهاد ما این است که آخرین نسخه از نرم افزارهایی که نام بردیم را نصب نمایید. در نسخه های ویندوز 8.1, 8 و نسخه ویندوز 10 چارچوب کاری NET. به طور پیش فرض نصب می باشد.

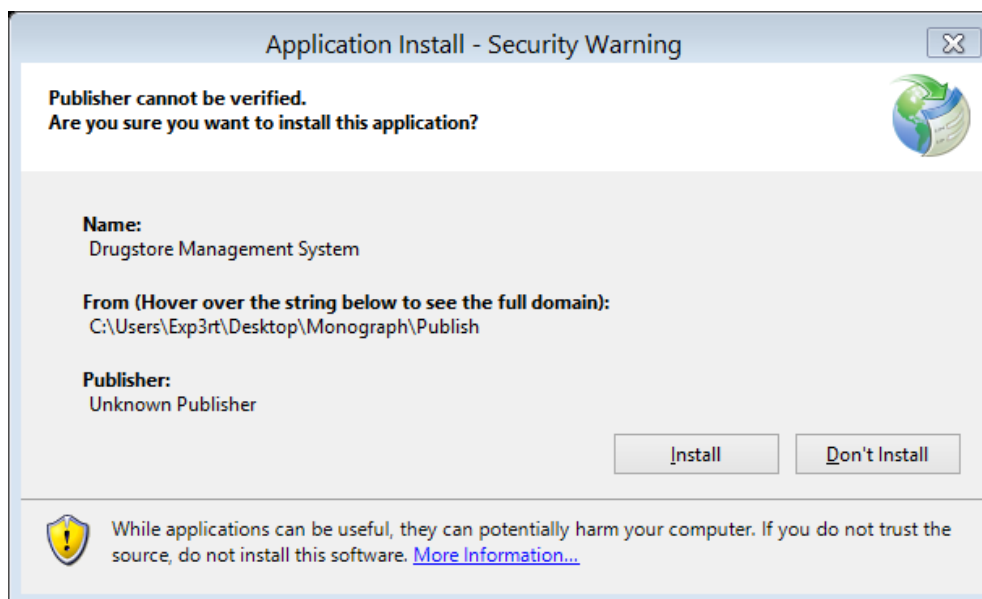
نظر به این که خروجی گزارش ها با فرمت فایل Excel و یا Word ایجاد میشود، نصب بسته نرم افزاری Office بر روی سیستم کاربر حتمی می باشد. حداقل نسخه مورد نیاز برای دریافت خروجی از گزارش های سیستم Office 2007 می باشد که کاربر باید حتما بر روی سیستم خود نصب داشته باشد.

منابع دانلود نرم افزار های مورد نیاز

برای دانلود نرم افزار های مورد نیاز سایت های معتبر زیادی موجود می باشد؛ اما معتبرترین مکانی که برای دریافت نرم افزار های مورد ضرورت می توانید استفاده کنید، وبسایت رسمی خود شرکت مایکروسافت می باشد که تمامی نسخه های نرم افزار مورد نیاز را برای دانلود فراهم ساخته است. نظر به این که برخی از کاربران در کشور افغانستان توانایی پرداخت قیمت این نرم افزار ها را ندارند می توانند از وبسایت های فارسی که نرم افزار های ذکر شده را با نحوه نصب و کرک در خود گنجانیده اند دریافت کنند.

مراحل نصب سیستم

پس از نصب ابزارهایی که در قسمت قبلی به معرفی گرفته شد حال نوبت به اجرای نرم افزار مدیریت دواخانه رسیده است که میتوان با چند کلیک ساده اقدام به نصب آن نمود. در صورتی که مراحل استخراج پروژه که در فصل قبل توضیح داده شد با موفقیت انجام شده باشد، هم اکنون می توانید فایل Setup آن را از محل ذخیره آن باز نمود. با اجرای Setup.exe صفحه نصب نرم افزار به شکل زیر باز خواهد.



تصویر ۳۱ - نصب Setup نرم افزار

فقط کافی است تا دکمه Install را بفشارید تا نرم افزار روی سیستم نصب گردد.

در صورتی که نرم افزار با موفقیت کامل نصب شود صفحه اصلی نرم افزار به طور خودکار باز شده و می توانید به صفحه مدیریت دواخانه وارد شده و میتوانید.

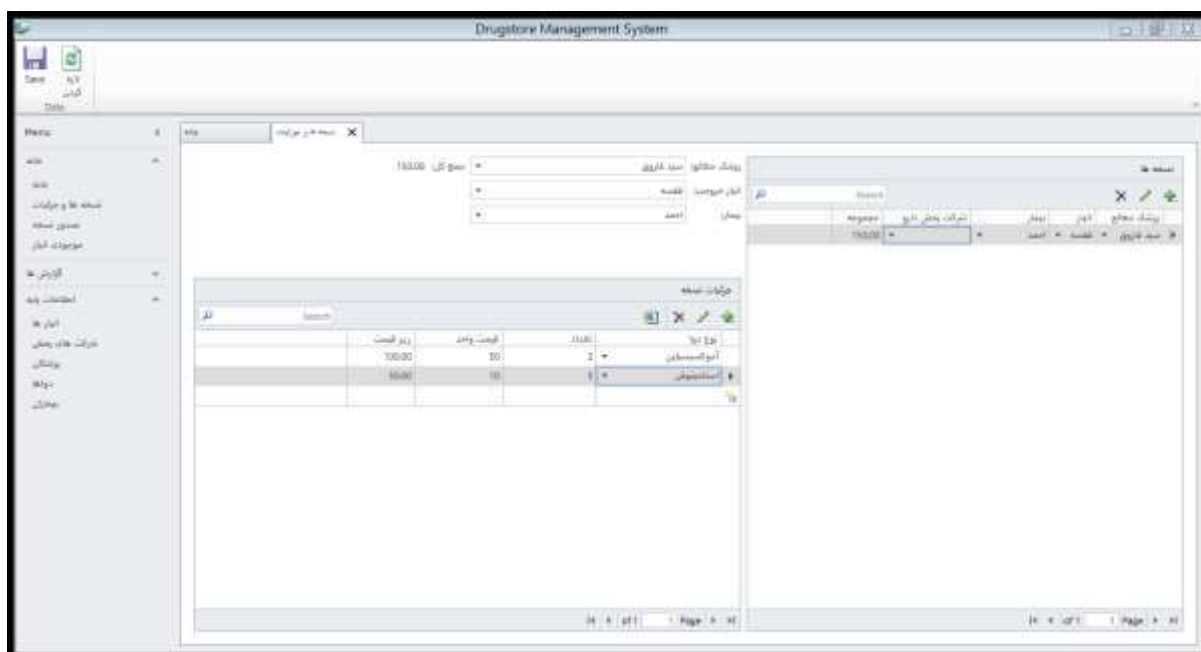
در صفحه اول چندین آیکون مورد ضروری می باشد که هر کدام به یکی از فرم های مربوط به ذخیره اطلاعات ارتباط میدهد. همچنان در Navigation سمت چپ برنامه نیز میتوان به تمامی فرم های مورد نیاز دسترسی داشت.



تصویر ۳۲ - صفحه اصلی نرم افزار مدیریت دواخانه

هم اکنون برای صدور نسخه بر روی آیکون آن و یا از پنل سمت چپ بر روی نسخه ها و جزئیات کلیک می نمایید تا صفحه مورد نظر باز شود.

در چوکات سمت راست می توانید نسخه جدید خویش را با مشخصاتی از قبیل پزشک معالج و نام بیمار را افزوده و در ستون میانی مشخصات نسخه که شامل چه دوا هایی می شوند را اضافه کنید. نکته قابل ذکر این است که قیمت مجموعی نسخه و همچنان هر خط از دوا هایی که به نسخه افزوده اید در آن واحد محاسبه شده و کاربر به راحتی می تواند آن را مشاهده کند.



تصویر ۳۳ - افزودن نسخه جدید در سیستم

فصل ششم - بررسی نقاط ضعف و قوت سیستم مدیریت دواخانه

نتیجه گیری

با توجه به سوال تحقیق دیدیم که با استفاده از تکنولوژی های طراحی نرم افزار شرکت مایکروسافت به راحتی میتوان نرم افزار های تجاری قدرتمندی را ساخت. هرچند تکنولوژی LightSwitch محدودیت هایی را در قسمت چاپ فرم های مورد نیاز از قبیل فاکتور، نسخه، میزان درآمد، موجودی انبار و غیره گزارش های این چینی را دارا نمی باشد، در مقابل شرکت های همکار مایکروسافت نظیر Syncfusion با ساختن نرم افزار های جانبی قدرتمند دیگری این خلاء را پر کرده و می توان فرم ها و گزارش های مختلفی را با استفاده از این نرم افزار ها به صورت چاپ لحظه ای با پرینتر های موجود و یا دریافت خروجی در قالب PFT, Excel, Word و یا HTML را ساخت.

سیستم مدیریت دواخانه ای که هم اکنون ساخته شده است، تمامی قابلیت هایی که در فرضیه تحقیق فصل اول گنجانیده شده است به طور کامل به وقوع پیوسته است. چراکه

نتیجه کلی از سیستم که آیا به سوالات اصلی و فرعی تحقیق پاسخ داده است یا خیر؟

آیا فرضیه های تحقیق به وقوع پیوسته است یا خیر؟

پیشنهادهات

افزودن قسمت کنترل تاریخ تولید و مصرف داروهای وارده در انبار یکی از قسمت های مهم در مدیریت دواهای موجود در دواخانه هست که در این سیستم لحاظ نشده است. این قسمت را می توان با استفاده از دستورات اس کیوال و همچنان قابلیت های Validation در تکنولوژی LightSwitch اعمال نمود که از جمله فعالیت های محققین دیگر می تواند به حساب آید.

در صورتی که به تصویر ۳۱ در فصل پنجم نظری بیندازید خواهید دید که یک هشدار Warning به نصب کننده نرم افزار مبنی بر این که Publisher و یا نشر کننده این نرم افزار ناشناخته است را نمایش میدهد.

نصب برنامه به شکل پیشرفته تر و بدون پیغام خطا در مورد سازنده

دواخانه دار برای افزودن دوا ها در هر نسخه می بایست نام تمامی دواهای موجود در یک نسخه را به صورت دستی تایپ کند، با در نظر داشت این که هم اکنون سیستم های پیشرفته بارکد خوان موجود می باشد قسمت افزودن دوا به نسخه ها به راحتی می تواند توسط بارکد خوان به سیستم اضافه شود که در پروژه حاضر در این زمینه هیچ فعالیتی نشده است.

استفاده از بارکد خوان برای ورود اطلاعات در پایگاه داده

پیوست

تمامی نرم افزار های مورد نیاز برای طراحی، ساخت و اجرای این نرم افزار مورد ضرورت می باشد در قالب ۴ دی وی دی ضمیمه این سند شده است.

دی وی دی شماره یک شامل نرم افزار Visual Studio 2013 به همراه افزونه های مورد نیاز از قبیل LightSwitch Components, Border Control, Pixata Custom Controls می باشد.

دی وی دی شماره دو شامل نرم افزار SQL Express و SQL Server Management Studio می باشد که در طراحی دتاییس و ساخت گزارش ها کاربرد به سزایی دارد.

دی وی دی شماره سه در بر گیرنده نسخه نهایی ویندوز 8.1 و Runtime Libraries شامل Silverlight, Microsoft C++ و برخی دیگر نرم افزار های مورد نیاز می باشد.

دی وی دی شماره چهار شامل سورس کد نرم افزار سیستم مدیریت دواخانه و هم چنان Setup کامل نرم افزار برای کسانی که میخواهند از سیستم استفاده نمایند قرار دارد.

1. Database System Concepts Sixth Edition by Abraham Silberschatz - Yale University
2. Beginning Visual Studio® LightSwitch Development by István Novák
3. Normalization Is a Nice Theory by David Adams & Dan Beckett
4. <http://iranattache-afghan.ir/news/?id=751>
5. <http://www.tebyan.net/newindex.aspx?pid=71071>
6. <http://goo.gl/D1SG8Z>
7. <https://msdn.microsoft.com/en-us/library/jj635143.aspx?f=255&MSPPError=-2147217396>
<http://www.msdn.com>
8. http://www.qbssoftware.com/products/SQL_Server/system-requirements/_prodmssqlsrv
9. http://www.qbssoftware.com/products/Visual_Studio_2013/system-requirements/_prodmsvs2013

۱۰. دواخانه روشنا

۱۱. دواخانه رضا ریاض

Abstract

طرح حاضر بیانی از نحوه ساخت و نمونه عملکرد یک دتایس در مدیریت خرید و فروش دوا در دواخانه های افغانستان می باشد. با توجه به این که تعداد دواخانه ها در افغانستان همه روزه رو به افزایش می باشد بر آن شدیم تا سیستمی را به عنوان پایان نامه خویش مد نظر گرفته تا قسمت کوچکی از مسائل پیش روی ملت عزیزمان افغانستان را برداشته و یا به نحوی حل نماییم.

دواخانه هایی که حداقل یک پایه کامپیوتر و یک کارمند با حداقل دانش کامپیوتری را دارا باشند می توانند به راحتی سیستم حاضر را استفاده نموده و تمامی امورات حسابداری خویش را به راحت ترین شکل ممکن شفافیت بخشیده و به آن سرعت دهند. علاوه بر آن برای آن عده از عزیزانی که خواهان ساخت یک سیستم مدیریت دواخانه می باشند می توانند با استفاده از این رساله تکنیک ها و ابزار های مورد استفاده ساخت سیستم های مدیریتی را آموخته و برای امور مشابه دست به طراحی بزنند. قابل ذکر است محققین عزیزی که خواهان رشد و انکشاف سیستم حاضر می باشند نیز می توانند امور تکراری از قبل انجام شده را به راحتی پشت سر گذاشته و در امر توسعه سیستم کنونی گام های بیشتری بردارند.



Islamic Republic of Afghanistan

Ministry of Higher Education

Khaja Abdullah-e-Ansari Institute of Higher Education

Faculty of Computer Science

Department of Database

Subject

Design and Development of Drugstore
Management System

Supervisor: Mohammad Hassany

Prepared: Homayoon Azimi

Winter 2015 – Herat, Afghanistan