



دانشگاه شمال

دانشکده فنی و مهندسی

گروه مهندسی کامپیوتر

# طراحی و پیاده سازی سیستم نرم افزاری ارتباط و اتوماسیون اداری

پروژه پایان نامه برای دریافت درجه کارشناسی

در رشته کامپیوتر گرایش نرم افزار

توسط :

سعید بزاز آبکنار

استاد راهنما :

مهندس حامد رجبی قمی

مهر ۱۳۸۷



## تقدیم به

مادرم که دعای خیرش همواره گره گشای مشکلاتم بوده و هست و  
پدر بزرگوarm که راهنمایی هایش همیشه روشنگر راه آینده ام  
بوده و خواهد بود.

## فهرست مطالب

۱	پیشگفتار.....
۲	چکیده .....
۳	۱ - فصل اول : مقدمه و امکان سنجی .....
۴	۱-۱: سیستم اتوماسیون اداری و دبیرخانه بدون کاغذ .....
۴	۲-۱: اجزاء پروژه .....
۴	۱-۲-۱: نقطه شروع: تحلیل .....
۴	۲-۲-۱: زیربنا: سخت افزار .....
۵	۳-۲-۱: اسکلت کار: موتور بانک اطلاعاتی .....
۵	۴-۲-۱: روکش کار: برنامه های کاربردی .....
۵	۵-۲-۱: شروع ارتباطات: شبکه و سرور .....
۵	۶-۲-۱: نصب حفاظ های امنیتی: محرمانگی .....
۵	۷-۲-۱: کسانی که باید با برنامه ها کار کنند: آموزش .....
۶	۸-۲-۱: آیا پس از تحویل کار تمام شده است: پشتیبانی .....
۷	۳-۱: بخشهای اصلی سیستم اتوماسیون اداری .....
۷	۱-۳-۱: کارتابل مکانیزه (کاربر سیستم) .....
۸	۲-۳-۱: امکانات .....
۸	۳-۱: کارتابل WEB .....
۹	۲-۳-۱: امور دبیرخانه .....
۹	۱-۲-۳-۱: قسمت نامه های وارده .....
۹	۲-۲-۳-۱: قسمت تایپ و نامه های صادره .....
۱۰	۳-۲-۳-۱: مسئول ارتباطات .....
۱۰	۴-۲-۳-۱: امکانات .....
۱۱	۳-۳-۱: تقویم شخصی و تقویم سازمانی .....
۱۱	۴-۳-۱: سیستم ارتباط و تماسها .....
۱۱	۵-۳-۱: مدیریت گزارش ها .....

۱۱	..... امکانات : ۱-۵-۳-۱
۱۲	..... مدیریت سیستم : ۶-۳-۱
۱۳	..... فصل دوم : تجزیه و تحلیل و طراحی
۱۴	..... ۱-۲: مدل سازی معنایی داده ها (نمودار های ER)
۱۷	..... ۲-۲: طراحی پایگاه رابطه ای (روش بالا به پایین )
۲۰	..... ۳-۲: تجزیه و تحلیل
۲۰	..... ۱-۳-۲: مشخصات سیستم ارتباط و اتوماسیون اداری سبا (نگارش ۱/۰۰)
۲۱	..... ۲-۳-۲: مفاهیم اولیه سیستم اتوماسیون اداری
۲۱	..... ۱-۲-۳-۲: دبیرخانه
۲۱	..... ۲-۲-۳-۲: کارتابل
۲۱	..... ۳-۲-۳-۲: نامه وارده
۲۱	..... ۴-۲-۳-۲: نامه صادره
۲۱	..... ۵-۲-۳-۲: پیام داخلی
۲۲	..... ۶-۲-۳-۲: کاربر
۲۲	..... ۷-۲-۳-۲: سمت
۲۲	..... ۸-۲-۳-۲: نقش
۲۳	..... ۹-۲-۳-۲: بایگانی
۲۳	..... ۱۰-۲-۳-۲: نامه های وارده
۲۳	..... ۱۱-۲-۳-۲: نامه های صادره
۲۴	..... ۱۲-۲-۳-۲: پیام داخلی
۲۴	..... ۱۳-۲-۳-۲: ارجاع نامه
۲۵	..... ۱۴-۲-۳-۲: چارت سازمانی
۲۵	..... ۱۵-۲-۳-۲: قالب شماره نامه
۲۶	..... ۳-۳-۲: طرح شماتیک گردش اسناد اتوماسیون اداری سبا
۲۷	..... ۴-۳-۲: نمودار های UML
۲۷	..... ۱-۴-۳-۲: نمودار های مورد کاربرد ( UseCase )
۲۷	..... ۱-۱-۴-۳-۲: کارتابل مکانیزه
۲۸	..... ۳-۱-۴-۳-۲: اطلاعات پایه
۳۵	..... ۲-۴-۳-۲: نمودار های کلاس ( Class )
۳۷	..... ۳-۴-۳-۲: نمودار ترتیبی (توالی) ( Sequence )
۳۷	..... ۱-۳-۴-۳-۲: ارسال نامه
۳۸	..... ۲-۳-۴-۳-۲: خواندن نامه

۳۹	..... جستجو نامه	۳-۳-۴-۳
۴۰	..... ( مثل کاربر و سمت و دسترسی و ... )	۴-۳-۴-۳-۲
۴۱	..... فصل سوم : ابزار شناسی	۳-۴-۱
۴۲	..... چيست و چگونه شکل گرفت؟	۳-۴-۱-۱
۴۳	..... یک پلات فرم مناسب برای آینده	۳-۴-۱-۱-۱
۴۳	..... ایده های اولیه	۳-۴-۱-۲
۴۴	..... تولد دات نت	۳-۴-۱-۳
۴۴	..... مروری بر فریمورک دات نت	۳-۴-۱-۴
۴۶	..... شروع کار با ویژوال استودیو دات نت ۲۰۰۵	۳-۴-۲
۴۶	..... مقدمه	۳-۴-۲-۱
۴۶	..... اجرای MS VS.NET	۳-۴-۲-۲
۴۹	..... نحوه کار با SQL Server 2005	۳-۴-۳
۴۹	..... مواردی در خصوص نصب SQL Server 2005	۳-۴-۳-۱
۵۰	..... Reporting Service	۳-۴-۳-۲
۵۰	..... Analysis Service	۳-۴-۳-۳
۵۱	..... Notification Service	۳-۴-۳-۴
۵۱	..... Integration Services	۳-۴-۳-۵
۵۱	..... SQL Server Agent	۳-۴-۳-۶
۵۲	..... Authentication Mode	۳-۴-۳-۷
۵۲	..... Collation	۳-۴-۳-۸
۵۲	..... وصل شدن به SQL Server و ساخت Database جدید	۳-۴-۳-۹
۵۸	..... Recovery Model	۳-۴-۳-۱۰
۵۹	..... Compatibility Level	۳-۴-۳-۱۱
۵۹	..... Table ها و DataType ها	۳-۴-۳-۱۲
۶۰	..... Table سایر مشخصات	۳-۴-۳-۱۳
۶۴	..... تعریف سایر مشخصات جدول	۳-۴-۳-۱۳-۱
۶۷	..... محدود کننده های فیلد (Constraint)	۳-۴-۳-۱۳-۲
۷۳	..... فصل چهارم : پیاده سازی	۴-۴-۱
۷۴	..... اصول برنامه نویسی چند لایه	۴-۴-۱-۱
۷۹	..... مزیت های برنامه های چند لایه	۴-۴-۱-۱-۱
۷۹	..... Presentation انتخاب های لایه ی	۴-۴-۱-۲
۸۰	..... Business Logic انتخاب های لایه ی	۴-۴-۱-۳
۸۱	..... Data Access انتخاب های لایه ی	۴-۴-۱-۴
۸۲	..... ارسال داده از یک لایه به لایه ی دیگر	۴-۴-۱-۵

۸۳	۶-۱-۴: ساختار لایه ای نرم افزار ارتباط و اتوماسیون اداری سبا .....
۸۴	۲-۴: نمودار های UML پیاده سازی .....
۸۴	۱-۲-۴: نمودار جزء (Component Diagram) .....
۸۵	۲-۲-۴: نمودار استقرار (Deployment Diagram) .....
۸۶	۳-۴: واسطه گرافیکی کاربر و کامپوننت های آن و راهنمای استفاده .....
۸۹	۱-۳-۴: کارتل مکانیزه .....
۸۹	۱-۱-۳-۴: فرم اطلاعات کاربری .....
۹۰	۲-۱-۳-۴: فرم ارسال نامه .....
۹۵	۳-۱-۳-۴: فرم ویرایش پیش نویس ها .....
۹۶	۴-۱-۳-۴: فرم نامه های دریافتی .....
۹۸	۵-۱-۳-۴: فرم ارجاع نامه .....
۹۹	۶-۱-۳-۴: فرم بایگانی نامه .....
۱۰۰	۷-۱-۳-۴: فرم پیشرفت نامه .....
۱۰۱	۸-۱-۳-۴: فرم جستجو نامه .....
۱۰۲	۹-۱-۳-۴: فرم ثبت پرونده .....
۱۰۳	۱۰-۱-۳-۴: فرم دریافت و ارسال پیام شخصی .....
۱۰۴	۲-۳-۴: دبیرخانه .....
۱۰۴	۱-۲-۳-۴: فرم مشاهده نامه های صادره .....
۱۰۵	۲-۲-۳-۴: فرم مشاهده نامه های داخلی .....
۱۰۶	۳-۳-۴: اطلاعات پایه .....
۱۰۶	۱-۳-۳-۴: فرم تعریف واحد سازمانی .....
۱۰۷	۱-۳-۳-۴: فرم تعریف سمت سازمانی .....
۱۰۸	۲-۳-۳-۴: فرم تعریف سطح دسترسی .....
۱۰۹	۳-۳-۳-۴: فرم مدیریت کارمندان .....
۱۱۰	۴-۳-۳-۴: فرم مدیریت اشخاص حقیقی و حقوقی .....
۱۱۱	۵-۳-۳-۴: فرم های مدیریت اطلاعات نامه .....
۱۱۲	۵- فصل پنجم : نتیجه گیری .....
۱۱۵	۶- فهرست منابع .....

## فهرست اشکال

### ۲ - فصل دوم : تجزیه و تحلیل و طراحی

شکل ۱-۲: نمودار ER شخص	۱۴
شکل ۲-۲: نمودار رابطه شخص با نامه	۱۵
شکل ۳-۲: روابط نامه	۱۶
شکل ۴-۲: گردش نامه در سازمان	۲۶
شکل ۵-۲: نمودار مرد کاربرد کارتابل - نامه	۲۷
شکل ۶-۲: نمودار مورد کاربرد پیام	۲۸
شکل ۷-۲: نمودار مورد کاربرد پرونده	۲۸
شکل ۸-۲: نمودار مورد کاربرد دبیرخانه	۲۹
شکل ۹-۲: نمودار مورد کاربرد چارت سازمانی	۳۰
شکل ۱۰-۲: نمودار مورد کاربرد دسترسی ها	۳۱
شکل ۱۱-۲: نمودار مورد کاربرد سازمان خارجی	۳۱
شکل ۱۲-۲: نمودار مورد کاربرد اطلاعات پایه نامه	۳۲
شکل ۱۳-۲: نمودار مورد کاربرد تنظیمات	۳۳
شکل ۱۴-۲: نمودار مورد کاربرد تقویم سازمانی	۳۴
شکل ۱۵-۲: نمودار مورد کاربرد سیستم ارتباط	۳۴
شکل ۱۶-۲: نمودار کلاس	۳۵
شکل ۱۷-۲: نمودار ترتیبی ارسال نامه	۳۷
شکل ۱۸-۲: نمودار ترتیبی خواندن نامه	۳۸
شکل ۱۹-۲: نمودار ترتیبی جستجو نامه	۳۹
شکل ۲۰-۲: نمودار ترتیبی مدیریت اطلاعات پایه	۴۰



۴۵	۳ - ابزار شناسی .....
۴۷	شکل ۱-۳ : شروع .....
۴۸	شکل ۲-۳: خصوصیات ویژوال استدیو .....
۵۳	شکل ۳-۳ : وصل شدن به Sql Server .....
۵۴	شکل ۴-۳ : پایگاه داده .....
۵۵	شکل ۵-۳: پایگاه داده جدید .....
۵۶	شکل ۶-۳: پنجره ایجاد پایگاه داده جدید .....
۵۸	شکل ۷-۳ : پنجره خصوصیات ایجاد پایگاه جدید .....
۵۹	شکل ۸-۳ : ایجاد پایگاه جدید .....
۶۵	شکل ۹-۳ : ستون های جدول .....
۶۶	شکل ۱۰-۳ : Is Persisted .....
۶۸	شکل ۱۱-۳ : کلید خارجی .....
۶۹	شکل ۱۲-۳ : فرم ساخت کلید خارجی جدید .....
۷۰	شکل ۱۳-۳: انتخاب رابطه ها .....
۷۱	شکل ۱۴-۳ : نمای کلید ساخته شده .....
۷۱	شکل ۱۵-۳ : ساخت محدودیت .....
۷۳	۴ - پیاده سازی .....
۷۶	شکل ۱-۴ : نمای سه لایه .....
۷۸	شکل ۲-۴ : نمای ۷ لایه .....
۷۸	شکل ۳-۴ : نمای ۳ لایه برنامه اتوماسیون سبا .....
۸۱	شکل ۴-۴ : معماری لایه ای برنامه سه لایه .....
۸۲	شکل ۵-۴ : ساختار درختی کامپوننت های تشکیل دهنده نرم افزار .....
۸۳	شکل ۶-۴ : اجزای سازنده برنامه اتوماسیون سبا .....
۸۴	شکل ۷-۴ : نمودار اجزای نرم افزار اتوماسیون اداری سبا .....
۸۵	شکل ۸-۴ : نمودار استقرار سیستم اتوماسیون سبا .....
۸۶	شکل ۹-۴ : صفحه اصلی نرم افزار .....
۸۸ و ۸۷	شکل ۱۰-۴ : بخشهای مختلف منوی ریبون .....
۸۸	شکل ۱۱-۴ : پنجره در باره نرم افزار .....
۸۹	شکل ۱۲-۴ : صفحه اطلاعات کاربر .....
۹۰	شکل ۱۳-۴ : فرم ارسال نامه .....
۹۱	شکل ۱۴-۴ : فرم ارسال نامه - بخش رونوشت .....

شکل ۴-۱۵ : فرم ارسال نامه - بخش متن نامه .....	۹۲
شکل ۴-۱۶ : فرم ارسال نامه - بخش متن امضای نامه .....	۹۳
شکل ۴-۱۷ : فرم ارسال نامه - بخش پیوست .....	۹۴
شکل ۴-۱۸ : فرم پیش نویس ها .....	۹۵
شکل ۴-۱۹ : فرم نامه دریافتی .....	۹۶
شکل ۴-۲۰ : فرم نمایش نامه .....	۹۷
شکل ۴-۲۱ : طریقه ذخیره پیوست .....	۹۷
شکل ۴-۲۲ : فرم ارجاع نامه .....	۹۸
شکل ۴-۲۳ : فرم بایگانی نامه .....	۹۹
شکل ۴-۲۴ : فرم پیشرفت نامه .....	۱۰۰
شکل ۴-۲۵ : فرم جستجو نامه .....	۱۰۱
شکل ۴-۲۶ : فرم ثبت پرونده .....	۱۰۲
شکل ۴-۲۷ : فرم پیام های دریافتی .....	۱۰۳
شکل ۴-۲۸ : فرم ارسال پیام .....	۱۰۳
شکل ۴-۲۹ : مشاهده نامه های صادره .....	۱۰۴
شکل ۴-۳۰ : فرم مشاهده نامه داخلی .....	۱۰۵
شکل ۴-۳۱ : فرم تعریف واحد سازمانی .....	۱۰۶
شکل ۴-۳۲ : فرم تعریف سمت .....	۱۰۷
شکل ۴-۳۳ : فرم تعریف سطح دسترسی .....	۱۰۸
شکل ۴-۳۴ : فرم مدیریت کارمندان .....	۱۰۹
شکل ۴-۳۵ : فرم مدیریت اشخاص حقیقی و حقوقی .....	۱۱۰
شکل ۴-۳۶ : فرم امنیت نامه .....	۱۱۱
شکل ۴-۳۷ : فرم وضعیت نامه .....	۱۱۱

## پیشگفتار

سیستم بایگانی اداره پلیس، سیستم تولید و قیمت تمام شده محصولات یک کارخانه، قسمت حسابداری یک مرکز تجاری، امور مربوط به پیگیری چکها در یک موسسه و..... همه و همه نمونه هایی از سیستم اداری هستند که در صورت کمک گرفتن از کامپیوتر برای انجام هر چه بهتر امور نمونه هایی از اتوماسیون اداری خواهند بود. نظام اداری را در یک بنگاه و موسسه اقتصادی، می توان به عنوان شاکله و پیونددهنده ارکان آن موسسه دانست. نظام اداری از یک سو بیانگر نوع و نحوه ارتباط بین ارکان داخل سازمان و از سویی دیگر، پل ارتباطی بین سازمان و خارج آن می باشد. بر همین اساس به نظر می رسد ابتدائی ترین امر در جهت ارتقای بهره وری و همچنین دستیابی هر چه بیشتر به اهداف سازمانی، ایجاد یک نظام اداری منسجم و کارآمد است.

بنابر این دلایل است که موسسات و بنگاههای اقتصادی از تمامی امکانات موجود در جهت بهبود و ارتقای نظام اداری خود بهره می برد.

از سویی دیگر با فراگیر شدن فناوری اطلاعات و نمایان شدن قالبها و روشهای گوناگون استفاده از آن، استفاده از این فناوری در به روز کردن نظام اداری، مورد توجه بسیاری واقع گردیده است.

سیستم اتوماسیون مشکلات ناخواسته و موانع موجود در نظام اداری سازمانها و موسسات اقتصادی را که برخی از آنها به شرح زیر می باشد، برطرف می نمایند.

۱. اتلاف هزینه و زمان در ارتباطات درون سازمانها
۲. اتلاف هزینه و زمان در ارتباطات بین سازمانها
۳. اتلاف هزینه و زمان در پیگیری ارجاعات و پیگیری مکاتبات
۴. عدم امکان نظارت پویا و دقیق بر سیستم اداری
۵. اتلاف هزینه و زمان در خصوص دسترسی به مکاتبات و نگهداری آنها
۶. عدم امکان دسترسی به نرم افزار از طریق اینترنت (بدون محدودیت زمان و مکان)

## چکیده

سیستم اتوماسیون اداری یک سیستم جامع و یکپارچه نرم افزاری برای مدیریت گردش اطلاعات و مکاتبات اداری و سازمانی است. یک ابزار نرم افزاری قوی، سریع و دقیق برای ورود، بازیابی، جستجو، مشاهده، ویرایش، گزارش گیری و گردش مجموعه اطلاعات سازمانی - اداری در بین تمام کارکنان و مدیران سازمان متناسب با ساختار آن است. گردش اطلاعات سازمانی - اداری می تواند شامل انواع مکاتبات، نامه های وارده، نامه های صادره و داخلی و یا انواع دستورات، گزارش ها، یادداشت ها، پیغام ها، مستندات، گزارشات فنی و یا نمابرها و به طور کلی تمام اطلاعات الکترونیکی و غیر الکترونیکی باشد.

بیشتر سازمانها و موسسات بزرگ اداری، از مکاتبات گسترده ای با بقیه سازمانها و موسسات برخوردارند. این مکاتبات گسترده و همچنین پاسخگو به مراجعه کنندگان همواره مشکل ساز بوده و هزینه زیادی را به لحاظ زمانی و مالی به سازمان متحمل می نماید.

به منظور پاسخگویی به نیازهای فوق، یک سیستم نرم افزاری اتوماسیون اداری را طراحی می نماییم.

## فصل اول : مقدمه وامکان سنجی

## ۱-۱: سیستم اتوماسیون اداری و دبیرخانه بدون کاغذ

این سیستم کار مدیریت و گردش اسناد، ارتباط با سایر سیستمهای اداری و مالی و پشتیبانی و تولید، ارجاع کارها، ثبت تمامی مکاتبات سازمان و اطلاعات مرتبط با آنها و ارائه این اطلاعات به افراد ذینفع (و ذیصلاح) در موقع لزوم، با دسته بندی مناسب و طبقه بندی صحیح را انجام می دهد.

همواره سازمانها و موسسات و شرکتها تمایل به بهبود در بهره وری سازمان در بخشهای مختلف داشته و دارند. سیستم های اتوماسیون اداری ابزاری قدرتمند جهت افزایش بهره وری و مدیریت بهتر را در اختیار مدیران سازمانها قرار می دهد.

سیستم اتوماسیون اداری سبا با بهره گیری از یک تحلیل دبیرخانه ای و استفاده از روشهای نوین عرصه سیستمهای اتوماسیون اداری و داشتن یک سیستم منسجم و یکپارچه طراحی و پیاده سازی شده است.

## ۱-۲: اجزاء پروژه

### ۱-۲-۱: نقطه شروع: تحلیل

در اولین دیدار مشتری شرایط و مشخصات کاری خود را برای ما تشریح میکند و توقعاتی را که از سیستم مورد نظر خود دارد بیان می نماید. ما نیز بعنوان تولید کننده برنامه پس از دریافت این اطلاعات از مشتری می کوشد تا ضمن امکان سنجی اجرای پروژه پیشنهاداتی در جهت بهبود سیستم به مشتری ارائه نماید. در پایان این مرحله نتیجه حاصله بصورت گزارشی به مشتری ارائه میگردد که اولاً آیا کامپیوتری کردن یا تغییر سیستم فعلی به صلاح هست یا خیر؟ ثانیاً در صورت مثبت بودن سوال اول کدامیک از نیازهای اولیه مشتری قابل جوابگویی توسط سیستم خواهد بود و چه امکاناتی علاوه بر آنها میتواند در اختیار مشتری قرار گیرد؟ ثالثاً جهت اجرای پروژه چه منابعی - از قبیل سخت افزار، هزینه اجرا و مدت زمان لازم جهت تکمیل پروژه- مورد نیاز خواهند بود.

### ۱-۲-۲: زیربنا: سخت افزار

بدیهی است جهت راه اندازی سیستم نیازمند داشتن تعدادی دستگاه از قبیل کامپیوتر، چاپگر، امکانات شبکه و .... دارید. پس از مرحله تحلیل لیست لوازم مورد نیاز در اختیار مشتری قرار میگردد و با توجه به گستردگی بازار

سخت افزار میتوانید براحتی اجناس مورد نظرتان را با مناسبترین قیمت تهیه کنید و سپس مرحله نصب و راه اندازی سیستم را به ما بسپارید.

### ۱-۲-۳: اسکلت کار: موتور بانک اطلاعاتی

موتور SQL Server میتواند با توجه به حجم کار و امکانات مورد تقاضا برای این منظور انتخاب گردد. شایان ذکر است هر چه موتور مورد نظر قویتر باشد امنیت، سرعت و ظرفیت اطلاعات بالاتر خواهد رفت و در مقابل بخاطر پیچیدگی بیشتر هزینه و زمان پیاده سازی بیشتر خواهد بود.

### ۱-۲-۴: روش کار: برنامه های کاربردی

موتورهای بانک اطلاعاتی امکان ارتباط مستقیم با کاربر را فراهم نمی سازند و فقط امکانات ایجاد ساختار بانک و همچنین ورود اطلاعات بصورت ابتدایی را بوجود می آورند. جهت ساختن یک رابط کاربر کارا و زیبا از زبان برنامه نویسی تحت ویندوز C#.net استفاده می شود و آنچه استفاده کننده می بیند فرمها و گزارشهایی است که بدین طریق تهیه و از طریق رابط بانک اطلاعاتی ویندوز با موتور بانک اطلاعاتی ارتباط دارند. جهت ایجاد برنامه از آخرین ابزار برنامه سازی میکروسافت بنام Visual Studio .NET استفاده می کند.

مزیت اصلی این بسته برنامه سازی امکان تولید برنامه های معمولی تحت ویندوز جهت کار در محل و همینطور تولید برنامه های تحت وب با ASP .NET جهت استفاده از راه دور و پیاده سازی سیستم های بسیار بزرگ می باشد.

### ۱-۲-۵: شروع ارتباطات: شبکه و سرور

اگر لازم است هر بخش شرکت بخشی از اطلاعات را وارد کامپیوتر کند یا اینکه حجم زیاد اطلاعات شما را بر آن می دارد که از تعدادی کامپیوتر جهت ورود همزمان اطلاعات استفاده کنید و از همه مهمتر اینکه مدیران شرکت مایلند از طریق کامپیوتر روی میز خود وضعیت شرکت را در هر لحظه کنترل کنند، شما به یک شبکه نیاز دارید. شبکه از بعد سخت افزاری یعنی وصل کردن تمام کامپیوترهای موجود از طریق سیم (که طول آن می تواند از ۳۰ سانتی متر تا ۲۰۰ متر باشد) به یک سویچ مرکزی و از بعد نرم افزاری شبکه یعنی داشتن یک کامپیوتر مرکزی یا سرور که وظیفه نگهداری تمام-یا اکثر- اطلاعات و شناسایی تمام کاربران و دادن مجوز به آنان جهت دستیابی به اطلاعات را بر عهده دارد.

### ۱-۲-۶: نصب حفاظ های امنیتی: محرمانگی

مفهوم امنیت در سیستمهای کامپیوتری اینست که هیچ کس نتواند اطلاعاتی را بخواند و یا چیزی را تغییر دهد مگر اینکه از طرف مدیریت سازمان جهت انجام این کارها مجاز شناخته شده باشد. بانکهای اطلاعاتی جدید امنیتی بسیار قوی و قابل اتکا فراهم میکنند و می توانید مطمئن باشید که کاربری که صرفا جهت گزارش گیری به سیستم معرفی کرده اید امکان انجام هیچ تغییری را در اطلاعات ندارد و حتی اگر کسی کامپیوتر شما را بدزدد نخواهد توانست اطلاعات شما را بخواند.

### ۱-۲-۷: کسانی که باید با برنامه ها کار کنند: آموزش

افراد با تجربه ، سرمایه اصلی سازمان شما هستند و کامپیوتر فقط ابزاری است که سرعت عمل ، دقت و بهره وری آنان را افزایش میدهد. آموزش خوب نه فقط یاد دادن چگونه کار کردن با کامپیوتر و برنامه ها به افراد که ایجاد رابطه دوستانه در جهت بوجود آوردن شوق ، انگیزه و علاقه برای اتوماسیون و ایجاد تغییرات در سیستم اداری نیز هست. در صورت نیل به این مهم است که نرم افزار تهیه شده زود در سیستم اداری جا می افتد ، عیب یابی می شود و سیستم اداری بوسیله آن ارتقا می یابد.

### ۸-۲-۱ : آیا پس از تحویل کار تمام شده است: پشتیبانی

پشتیبانی دو مرحله کاملاً متفاوت دارد، در مرحله اول نرم افزار هر قدر خوب طراحی شده باشد و قبل از تحویل مورد آزمایش قرار گرفته باشد ناگزیر پس از تحویل نیاز به اصلاحاتی خواهد داشت و ایراداتی وجود دارد که فقط پس از بکارگیری عملی نرم افزار خود را ظاهر خواهد ساخت. دوره نخست پشتیبانی حداکثر بمدت چند ماه از تحویل نرم افزار و با رفع اشکالات خاتمه خواهد یافت.

نرم افزاری که بصورت اصولی طراحی شده باشد پس از طی مرحله اول پشتیبانی کامل میشود و هرگز مستهلک نخواهد شد. شروع مرحله دوم پشتیبانی جهت تطبیق نرم افزار با شرایط جدیدی است که در سازمان شما بوجود خواهد آمد و این مرحله پس از پایان مرحله اول آغاز شده تا زمانی که نرم افزار کار میکند ادامه خواهد داشت. گفتنی است که هر نرم افزار تا حدی خاصی تحمل پذیرش تغییرات را دارد و ایجاد تغییرات اساسی پیش بینی نشده و یا تغییرات جزئی مکرر طی سالیان متمادی باعث از دور خارج شدن تدریجی برنامه خواهد شد.

مشخصات فنی :
<p>Platform</p> <ul style="list-style-type: none"> <li>▪ Microsoft Windows XP/2000/2003/Vista Methodology</li> <li>▪ Object Oriented Design, UML Based</li> <li>▪ Standard Relational Database Design (SQL)</li> <li>▪ Client/Server Architecture</li> </ul> <p>Technicality</p> <ul style="list-style-type: none"> <li>▪ .NET 2.0</li> <li>▪ Microsoft C#</li> <li>▪ Full Managed Code</li> <li>▪ Microsoft SQL Server 2005</li> </ul>



### ۱-۳: بخشهای اصلی سیستم اتوماسیون اداری

#### ▪ کارتابل مکانیزه :

وظیفه گردش انواع اطلاعات اداری و سازمانی مانند گردش نامه ها، دستورات ، گزارش ها، پیغام ها، یادداشت ها و انواع فرم های اطلاعاتی الکترونیکی و غیر الکترونیکی را بر عهده دارد .

#### ▪ امور دبیرخانه :

یک دبیرخانه و بایگانی الکترونیکی است که ثبت و حفظ اطلاعات انواع نامه ها، پیوست ها و محل فیزیکی نگهداری این نامه ها را در قالب یک ساختار درختی دارد .

#### ▪ تقویم شخصی و تقویم سازمانی:

در تقویم شخصی مواردی که شخص نیاز به یادآوری دارد ذکر می گردند و در تقویم سازمانی مواردی که کل سازمان به یادآوری آن نیاز دارند ثبت می گردند.

#### ▪ سیستم ارتباط و تماسها :

امکان برقراری کنفرانس همزمان بین چند کاربر(متن، صدا، تصویر) و ثبت هر ارتباط و تماس داخل و خارج سازمان

مشخصه های اطلاعاتی فراهم می نماید.

#### ▪ مدیریت سیستم:

قابلیت تعریف نقش ها، سمت ها و چارت سازمانی منطبق با وضعیت سازمان

### ۱-۳-۱: کارتابل مکانیزه (کاربر سیستم)

پس از دریافت و ثبت نامه های وارده در سیستم ، نامه ها به کاربر یا کاربران مورد نظر ارجاع می شوند ، در این مرحله می توان یک نامه را به یک یا چند کاربر بطور همزمان ارجاع داد ( مثل بخشنامه ها ، اطلاعیه ها و ... ) و

نیز این امکان وجود دارد که یک نامه بطور همزمان و با دستوراتی متفاوت به چند کاربر ارجاع شود. در ارجاع نامه ها نیز محدودیتی برای تعداد و طول متن ارجاعات وجود ندارد. ضمن آنکه می توان تمام ضمايم، شرح ارجاعات و سوابق یک نامه در دست اقدام را در رایانه مشخص نمود. در زمان ارجاع نامه تاریخ و ساعت ارجاع بصورت اتوماتیک ثبت می گردد و می توان تاریخ پیگیری نامه را نیز مشخص نمود. هر کاربر می تواند نسخه ای از نامه های مهم ارجاعی را در بایگانی راکدی که برای وی تعریف شده نگهداری نماید. ضمن آنکه در شرح ارجاعات می تواند گزارش کارهای انجام شده در ارتباط با موضوع نامه را به مدیر یا مدیران ارائه نماید. امکان تشخیص نامه های جدید اضافه شده به کارتابل کاربر، بازیافت نامه های بایگانی شده، نگهداری نامه های غیر ضروری در بایگانی شخصی افراد، امکان مشاهده کلیه عملیات انجام شده بر روی یک نامه بصورت بلوک دیاگرام از لحظه ورود تا بایگانی با ذکر نام کاربر، زمان و مدت انجام کار و امکان بایگانی یک نامه به همراه توضیح مربوطه از دیگر ویژگیهای این نرم افزار در بخش کارتابل می باشد. امکان ارسال پیغام از طریق این نرم افزار، یکی دیگر از قابلیت های کارتابل محسوب می شود. بر این اساس می توان هر گونه پیغام فوری و یا عادی را برای یک یا چند کاربر و یا برای تمامی کارکنان یک مجموعه اداری ارسال نمود. همچنین به همراه پیغام ارسالی، امکان فرستادن پیوستهائی بصورت مختلف ( نامه وارده، نامه صادره، متن تایپی، تصویر، فایل صوتی، فیلم و انواع فایلها بدون محدودیت فرمت آن ) یا پیگیری مجدد آنها وجود دارد.

### ۳-۱-۳-۱: امکانات

- امکان بازیافت و راه اندازی مجدد کارهایی که از گردش کار خارج شده اند.
- امکان تعلیق برای مدت دلخواه و یا لغو موارد کاری
- امکان دسترسی به انواع امکانات سیستم در فهرستهای جستجو
- امکان دسترسی به تاریخچه مکاتبات و مراحل کاری انجام شده به تفکیک مورد نظر
- مدیریت گزارش مکاتبات
- ثبت انواع مکاتبات و ارجاع
- رده بندی و طبقه بندی محتوای مستندات سازمان
- امکان ارسال گروهی نامه های صادره با یک شماره و یا شماره نامه های جداگانه.
- امکان حفظ سوابق تغییرات مستندات Word، ثبت کلیه تغییرات و امکان پیگیری اینکه چه کسی چه تغییری اعمال نموده است ( سوابق ایجاد و کلیه تغییرات هر نامه حفظ می شود ).
- امکان جستجو در مجموعه اطلاعات مربوط به مکاتبات

### ۳-۱-۳-۱: کارتابل WEB

در صورتی که سازمانی از امکانات Web این نرم افزار بهره مند شود کاربران می توانند کارتابل خود را از طریق اینترنت کنترل نمایند و ارجاعات مورد نظر را نیز انجام دهند.

### ۱-۳-۲: امور دبیرخانه

#### ۱-۳-۲-۱: قسمت نامه های وارده

نامه های وارده به نامه هایی اطلاق می شود که از سوی افراد ، موسسات و ... تهیه و به یک سازمان ارسال و پس از ثبت در بخش دبیرخانه در چرخه مکاتبات آن سازمان به جریان در می آیند . برنامه به صورتهای مختلف قابلیت دریافت نامه ها را دارا می باشد . نامه ها بصورت دستی، فکس ، شبکه (ارتباط بین سازمانی) و یا از طریق پست الکترونیکی (E-Mail) وارد سیستم می شوند . البته در مورد نامه های دستی ابتدا باید این نامه ها اسکن و ثبت شوند تا در چرخه سیستم قرار گیرند . این برنامه قابلیت دریافت نامه های وارده بصورت دستی ، فکس ، ایمیل و یا از طریق شبکه (ارتباط بین سازمانی) را دارا می باشد . در مرحله ثبت تاریخ ثبت ، شماره و ساعت ثبت و نام ثبات ( ثبت کننده ) بطور اتوماتیک ثبت می شود . در مرحله ثبت نامه ، این نرم افزار دارای قابلیتهایی همچون ثبت نامه های عادی و محرمانه ، بصورت کاملاً مستقل و با شماره های مستقل می باشد . تعیین فوریت اقدام (نامه های غیر فوری ، فوری و آنی ) نیز امکان پذیر است .

همچنین می توان نامه های مرتبط با نامه در حال ثبت را نیز مشخص نموده و با وارد کردن شماره مورد نظر ، در مرحله ارجاع نامه آنها را به پیوست نامه های در دست اقدام ، به کاربر یا کاربران مورد نظر ارجاع نمود. در مرحله ثبت نامه ، این امکان نیز پیش بینی شده است که نامه های هر حوزه از سازمان توسط همان حوزه و با شماره خاص هر یک ثبت گردد . همچنین علاوه بر امکان مشخص نمودن تعداد و نوع پیوستهای یک نامه ، می توان مشخصات سازمان یا شخص ارسال کننده را نیز به همراه تاریخ و شماره ارسال ، درج نمود . ضمن آنکه این برنامه قابلیت تشخیص تائیدیه های فکس و جلوگیری از ثبت مجدد آنها را دارا می باشد و در مواقعی که نامه ای با اطلاعات مطابق با یک نامه ثبت شده در گذشته مجدداً به چرخه ثبت وارد گردد ، سیستم بطور خودکار از ثبت دوباره آن جلوگیری می کند . همچنین در صورت بروز اشکال در اطلاعات ثبتی یک نامه ، امکان اصلاح مشخصات ثبتی نامه نیز پیش بینی شده و قابل انجام است .

#### ۱-۳-۲-۲: قسمت تایپ و نامه های صادره

جهت تایپ نامه ها در این برنامه ، کاربر می تواند هر نوع سربرگ مورد استفاده در سازمان و نیز برخی نامه های کلیشه ای و فرمهای اداری مانند احکام و امثالهم را برای سیستم تعریف نماید . همچنین می توان از نرم افزار MS Word نسخه های ۲۰۰۳ و ۲۰۰۷ برای تایپ نامه ها بهره گرفت . در این مرحله نام تایپیست ، تاریخ و ساعت تایپ بطور اتوماتیک ثبت می شود . می توان ترتیب اسامی مخاطبین و گیرندگان رونوشت را نیز تغییر داد و یا پس از تایپ ، متن تایپی را مورد بازبینی قرار داد . امکان مشخص نمودن گیرندگان داخلی رونوشت نیز وجود دارد . همچنین می توان نامه های مرتبط با نامه در دست اقدام را نیز ( مانند عطف ، پیرو و بازگشت ) و پیوستها و نوع آنها

و یا ارتباط نامه با طبقه یا کلاس همان خاص را تعیین نمود. نامه های صادره به نامه هایی اطلاق می شود که در درون یک سازمان تولید و بعد از تأیید و امضاء، توسط دبیرخانه ثبت و از سازمان خارج می شود.

تهیه یک نامه صادره ابتدا با تهیه یک پیش نویس توسط کارشناس مربوطه آغاز می شود. پیش نویس تهیه شده به تایپیست ارسال می گردد. تایپیست پس از تایپ متن نامه، آن را به کارشناس مربوطه بازمیگرداند تا پس از بازبینی، آنرا پاراف نماید. پس از پاراف نمودن کارشناس مربوطه، نامه به مدیر ارسال می شود تا وی تأیید و امضاء نماید. در مرحله بعدی برای نامه صادره، شماره و تاریخ ثبت می شود و سپس نامه از سازمان خارج می گردد. برای ارسال نامه نیز از طرق مختلف مانند روش دستی، از طریق فکس یا ایمیل و یا از طریق شبکه (ارتباط بین سازمانی) می توان اقدام نمود. رونوشت های نامه هایی که صاحبان آنها در داخل سازمان می باشند، بصورت اتوماتیک ثبت و به فرد مورد نظر ارجاع می شود. پس از امضای نهایی، برای نامه ها شماره ثبت می شود که در این مرحله سیستم بطور اتوماتیک تاریخ، ساعت و نام ثبت (ثبت کننده) را در کنار سایر مشخصات نامه درج می نماید. نامه های محرمانه را می توان با شماره های مجزا ثبت و صادر نمود.

همچنین سیستم قادر به ثبت و ارجاع اتوماتیک نامه های داخلی می باشد. امکان مشاهده پیش نمایش قبل چاپ و چاپ اتوماتیک نامه به تعداد مخاطبین و گیرندگان رونوشت به تفکیک وجود دارد. چاپ کلیه نامه های تایپ شده توسط ماشین نویس به وسیله چاپگرهای لیزری یا ماتریسی روی انواع سربرگ های اداری نیز برای این مرحله پیش بینی شده است.

### ۱-۳-۲-۳: مسئول ارتباطات

مسئول ورود دقیق اطلاعات فرستنده ها و گیرنده های مرتبط با سازمان است.

### ۱-۳-۲-۴: امکانات

- امکان ثبت انواع نامه و مستندات و گردش الکترونیکی آنها با روندهای تعریف شده
- امکان شماره دهی خودکار یا غیر خودکار
- ثبت نامه وارده و ارسال الکترونیکی آن به مقصد داخلی
- ثبت اقدامات انجام شده در گردش نامه
- قابلیت ردیابی سوابق و نامه های مرتبط در مراحل مختلف گردش مستندات
- امکان جستجوی نامه به روش های مختلف با استخراج گردش کار و تاریخچه فعالیت های انجام شده
- توانایی جستجوی قوی بر اساس فیلد دلخواه
- امکان بایگانی غیر متمرکز و درختی منطبق با ساختار بایگانی سازمان
- امکان طبقه بندی موضوعی مستند
- توانایی ایجاد پیوست نامه با گونه های مختلف اطلاعاتی (تصویر، نقشه، عکس و .....
- پی نوشت یا یادداشت دهی و گردش نامه
- قابلیت ایجاد پاسخ و پیرو یک نامه بصورت خودکار

- امکان ارسال گروهی نامه های صادره با یک شماره و یا شماره نامه های جداگانه .
- امکان دریافت و ارسال E-mail , Fax
- امکان ثبت انواع نامه و مستندات و گردش الکترونیکی آنها با روندهای تعریف شده محلی و فرامنطقه ای .
- امکان ثبت نتایج گزارشات و ارسال آن به سایر کاربران به عنوان یک مستند و یا پیوست .
- امکان چاپ پی نوشتهای یک نامه
- انتخاب چند امضاء کننده برای یک نامه بدون محدودیت

### ۱-۳-۳: تقویم شخصی و تقویم سازمانی

- امکان ثبت یادآوری توسط هر کاربر ب صورت شخصی
- امکان ثبت یادآوری های گروهی توسط مدیر هر بخش

### ۱-۳-۴: سیستم ارتباط و تماسها

- امکان ارسال پیام های متنی صوتی یا تصویری برای همکار یا گروهی از همکاران هنگام نیاز
- امکان برقراری کنفرانس بین چند کاربر(متن، صدا، تصویر)

### ۱-۳-۵: مدیریت گزارشها

#### ۱-۳-۵-۱: امکانات

- گزارش کارتابل
- مشاهده کارها و نامه های مرتبط با یک کار
- پیگیری و گزارش از وضعیت کارهای ارجاع شده
- گزارش از نامه ها بر اساس سال و شماره نامه
- گزارش از نامه ها بر اساس تاریخ دبیرخانه و نوع نامه
- گزارش از نامه بر اساس تاریخ نامه
- گزارش چرخه عمر یک کار یا نامه
- گزارش گیری بر اساس جستجوی آزاد

- گزارش‌های آماری متنوع بر حسب نوع نامه‌ها، تعداد نامه‌های ارسالی و دریافتی هر واحد و پیگیری‌ها و ارجاعات هر شخص یا هر واحد سازمانی

- استفاده از Crystal Report برای تولید گزارشها

### ۱-۳-۶: مدیریت سیستم

- ثبت عملکرد کاربران و کنترل آنها

- تعویض رمز عبور در داخل برنامه

- برخورداری از مستندات مناسب راهنمای کاربری و راهبری

- سهولت در مانیتور نمودن واحدهای کاری و روند اجرای امور

- امکان تعریف گروه‌های مختلف کاری جهت تسریع در امر مکاتبات

- امکان ارائه اطلاعات به ارباب رجوع نسبت به وضعیت هر نامه در هر بخش

- کنترل و تعیین مجوز اقدامات مختلف به کاربران با توجه به جایگاه سازمانی

- قابلیت تعریف نقش‌ها، سمت‌ها و چارت سازمانی منطبق با وضعیت سازمان

- پیگیری سوابق مکاتبات با استفاده از اعمال فیلترهای مورد نظر کاربر

- نگهداری سوابق کامل عملیات کاربران و مسئولیت‌های سازمانی

- جلوگیری از ورود کاربران به سیستم در مواقعی که لازم است اصلاحاتی انجام شود

- امکان گرفتن نسخ پشتیبان از اطلاعات به صورت روزانه، هفتگی و ماهیانه و خودکار

- قابلیت تعیین سطوح مختلف امنیتی و دسترسی به مستندات، ضمائم و پی‌نوشت‌ها برای کاربران

- امکان افزودن و یا حذف نمودن امکانات جانبی بر روی کلیه صفحات سیستم با توجه به سیاست کارفرما و

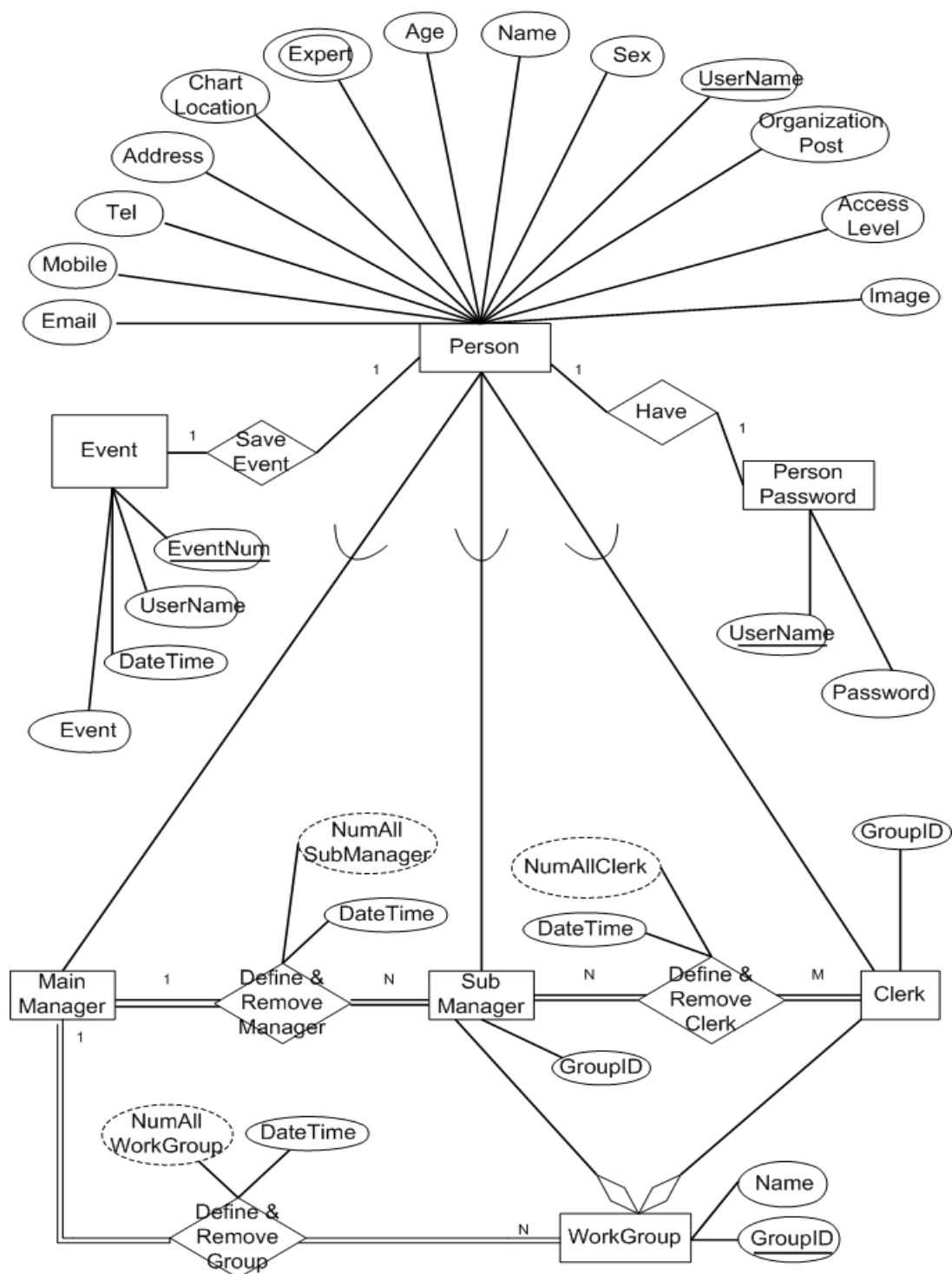
به صورت On line

- پیگیری و کنترل کارکرد و روند اجرای امور تک تک واحدها و کل سازمان به سهولت و سرعت به شکل

گزارش گیری و یا مشاهده لحظه ای .

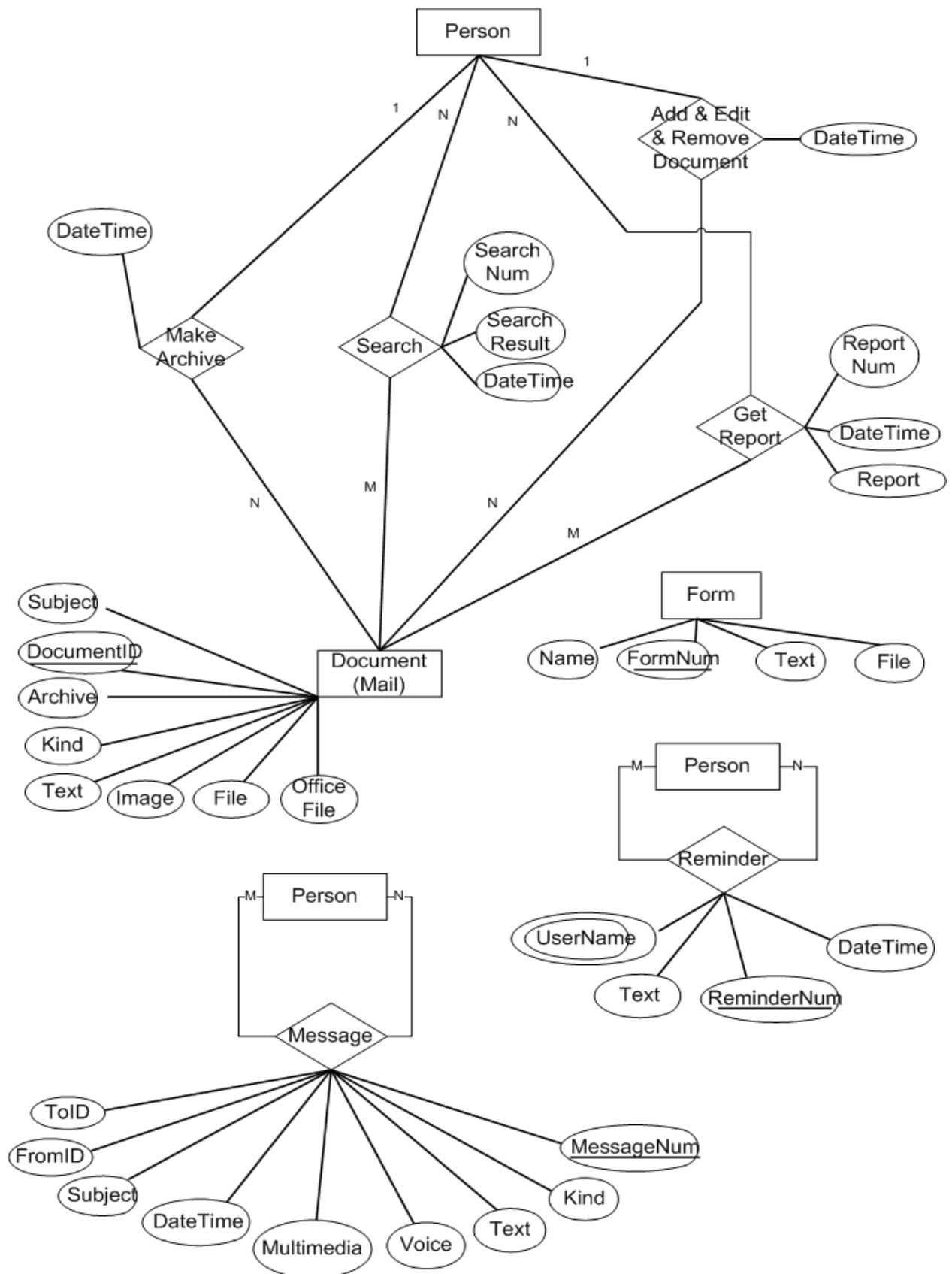
## فصل دوم : تجزیه و تحلیل و طراحی

## ۱-۲: مدل سازی معنایی داده ها (نمودار های ER)



شکل ۱-۲: نمودار ER شخص





شکل ۲-۲: نمودار رابطه شخص با نامه



## ۲-۲: طراحی پایگاه رابطه ای (روش بالا به پایین) :

Person (UserName , Name , Age , Sex , Tel , Mobile , Email , Address ,  
<sup>CK</sup>  
 AccessLevel , Image , OrganizationPost , Expert , ChartLocation , GroupID ,  
 UserType )

PersonPassword( UserName , Password )

CK . FK

Justification(UserName, DefineRemoveManager , DefineRemoveClerk ,  
<sup>CK . FK</sup>  
 DefineRemoveGroup , MakeArchive , Add & Edit & Remove Document ,  
 GetReport , Add & Remove Attach , Refer , ChangeState , SendDocument ,  
 AttachDocument)      \\Mojavez dastresi

Event(EventNum , UserName , DateTime , Event)

CK

FK

WorkGroup(GroupID , Name , Description)

CK

DefineRmoveSubManager(AdminID , SubID , NumAllSubManager , DateTime)

FK

FK

CK

DefineRmoveClerk(SubD , ClerkID , NumAllClerk , DateTime)

FK

FK

CK

DefineRmoveGroup(GroupID , NumAllWorkGroup , DateTime)

CK . FK

Form (FormNum , Name , Text , File)

CK

Reminder(ReminderNum , UserName , Text , DateTime)

CK

FK

Message(MessageNum , Kind , Text , Voice , Multimedia , DateTime , Subject ,  
FromID , ToID)

CK

FK

FK

Document(DocumentID , Subject , Archive , Kind , Text , Image , File , OfficeFile)

CK

MakeArchive(UserName , DocumentID , DateTime)

FK

FK

CK

AddEditRemoveDocument(UserName , DocumentID , DateTime)

FK

FK

CK

Note(NoteNum , Text , Subject)

CK

AddRemoveEditNote(UserName , NoteNum , DateTime)

FK

FK

CK

Reference(ReferenceNum , DocumentID , Text , ReferTo , ReferFrom)

FK

CK

DocumentState(StateNum , Name)

CK

ChangeState(StateNum , DocumentID , DateTime)

FK

FK

CK

AttachFile(AttachNum , DocumentID , File)

FK  
CK

AttachDocument(MainID , AttachID , DateTime)

FK FK

Reply(MainID , ReplyID , DateTime)

FK FK

SendDocument(DocumentID , CC , ToID , FromID , DateTime , Kind)

FK FK FK

## ۲-۳: تجزیه و تحلیل

### ۲-۳-۱: مشخصات سیستم ارتباط و اتوماسیون اداری سبا (نگارش ۱/۰۰)

- ثبت و گردش مکانیزه مکاتبات (وارد، داخلی، صادره) و پیام های داخلی
- تعریف دبیرخانه های مختلف سازمان و اجرای عملیات مستقل در هر دبیرخانه
- تعریف چارت سازمانی و تعریف سمت های ساختار سازمانی و محدوده مجاز اجرای عملیات هر

#### سمت در سیستم

- امکان تعریف جانشین برای سمت های سازمانی
- تعریف کاربران و محدوده مجاز اجرای عملیات کاربری
- انتساب سمت های مختلف به یک کاربر
- تعریف گروه های کاری از سمت ها و کاربران جهت تسهیل در امر تبادل مکاتبات و پیام
- تعریف طبقه بندی مکاتبات اداری (عادی، محرمانه، سری و...)
- تعریف سطوح دسترسی سمت ها به انواع مکاتبات طبقه بندی شده
- تعریف اولویت مکاتبات اداری (عادی، فوری، خیلی فوری و...)
- تعریف قالب شماره نامه (وارد، داخلی، صادره) در هر دبیرخانه
- امکان الحاق تصاویر و یا هر مورد دلخواه و اسکن اسناد و دریافت فاکس
- استفاده و کنترل مرکزی تاریخ در کلیه مراحل گردش مکاتبات
- تعریف مشخصات اشخاص حقیقی و حقوقی (طرف مکاتبات اداری)
- امکان افزودن و مشاهده انواع پیوست و درج نامه های عطف و پیرو
- ایجاد و مشاهده کارتابل مکاتبات، مکاتبات ارسالی و حذف شده به تفکیک سمت های سازمانی
- ایجاد و مشاهده کارتابل پیامها، پیامهای ارسالی و حذف شده به تفکیک کاربران
- اعلام دریافت نامه جدید
- امکان به گردش انداختن نامه های بایگانی شده
- نمایش راهنمای مصور فارسی در هر بخش از سیستم با قابلیت پیمایش در تمامی متن راهنما
- طراحی براساس متدلوژی شی گرا
- تولید براساس معماری .Net
- سیستم پایگاه داده MS-SQL Server 2005

- سیستم عامل ایستگاه کاری Windows 98/2000/XP
- سیستم عامل سرویس دهنده (Server, Advanced Server) Windows Server 2003

## ۲-۳-۲: مفاهیم اولیه سیستم اتوماسیون اداری

### ۲-۳-۲-۱: دبیرخانه:

مرکز مدیریت اسناد یک سازمان و یا زیر مجموعه ای از آن سازمان را دبیرخانه گویند. هر دبیرخانه می تواند به اشخاص مختلفی مدارک و نامه هایی را ارسال کند.

### ۲-۳-۲-۲: کارتابل:

کارتابل محل قراردادن نامه های ارسالی به یک سمت است. بدین معنی که اگر سمت " الف " نامه ای به سمت " ب " ارسال کند آن نامه در کارتابل سمت " ب " قرار می گیرد و او به محض ورود به سیستم و با اشاره ماوس به کارتابل می تواند نامه های دریافتی خود را مشاهده کند. سیستم به طور اتوماتیک در صورت ورود نامه به کارتابل در قالب پیام اطلاع می دهد.

### ۲-۳-۲-۳: نامه وارده:

به مدرکی اطلاق می شود که از خارج سازمان به مقصد داخل سازمان ارسال شده باشد. ثبت نامه وارده از طریق فرم ثبت نامه وارده صورت می پذیرد.

### ۲-۳-۲-۴: نامه صادره:

مدرکی است که از داخل سازمان به مقصد داخل یا خارج سازمان ارسال می گردد. ثبت نامه صادره از طریق فرم ثبت نامه صادره انجام می پذیرد.

### ۲-۳-۲-۵: پیام داخلی:

پیام داخلی به مدارکی اطلاق می شود که بین کاربران یک سازمان در گردش است. برای ثبت و یا مشاهده یک پیام داخلی از این فرم استفاده می شود. فرستنده و گیرنده پیام های داخلی همواره کاربران می باشند.

**۲-۳-۲-۶: کاربر:**

کاربر به شخصی اطلاق می شود که اجازه ورود به سیستم را دارد. با داشتن دو مشخصه نام کاربر و رمز عبور می توان وارد سیستم شد. مسئله مهم این است که یک کاربر مادامی که دارای سمت نباشد نمی تواند از بخش های مختلف سیستم اتوماسیون اداری که با ارسال نامه مرتبط می شود استفاده نماید. مثلاً "یک کاربر بدون سمت امکان صدور نامه، دریافت نامه، مشاهده نامه، و کلاً" انجام هر گونه عمل مرتبط با نامه را ندارد. یک کاربر در بخش تعریف کاربران قابل معرفی به سیستم است و نیز هر کاربر یک سری از امکانات مجاز کاربر را می تواند داشته باشد. جهت تعریف کاربر می توان از فرم تعریف کاربر استفاده کرد.

**۲-۳-۲-۷: سمت :**

سمت عنوانی است که در ساختار سازمانی تعریف شده و به کاربران منتسب می گردد. جابجایی مدارک و نامه ها در سیستم، مابین سمت های مختلف انجام می گیرد. مثلاً "مدیر بخش یک، به معاون بخش یک نامه ای را ارسال می کند. بنابراین سمت ها اساس گردش نامه در سیستم می باشند.

با مراجعه به فرم تعریف ساختار دبیرخانه می توان عمل تعریف سمت را انجام داد. تعریف سمت بخشی از تعریف ساختار سازمان می باشد. در فرم ساختار سازمان ابتدا باید دبیرخانه های مختلف سازمان را تعریف و چارت سازمانی موجود در هر محدوده کاری دبیرخانه را در قالب سمت ها به دبیرخانه ها افزود.

**۲-۳-۲-۸: نقش:**

در یک سیستم اتوماسیون اداری سمت های زیادی وجود دارند که از طریق این سیستم با یکدیگر ارتباط برقرار می کنند. این ارتباطات می تواند در شکل پیام و یا نامه باشد. جهت فرستادن پیام هیچگونه محدودیتی در مورد فرستنده و گیرنده وجود ندارد و هر کاربر تعریف شده در سیستم این امکان را دارد که به هر کاربر دیگری پیام ارسال کند ولی در مورد مکاتبات رسمی و نامه های اداری محدودیت های وجود دارد بدین ترتیب که افراد مجاز قابل دسترس برای هر سمت باید تعریف شوند و او نمی تواند به هر سمت دلخواه دیگری مدارک رسمی را ارسال کند. چنین مکانیزمی با فرم تعریف نقش ها قابل پیاده سازی می باشد.



**۲-۳-۹: بایگانی:**

بایگانی عبارت است از خارج کردن یک نامه از مسیر اداری و قراردادن آن در یک محل پوشه بایگانی . برای بایگانی یک نامه باید آن را به بایگان ارجاع داد. عمل بایگانی نامه توسط فرم بایگانی نامه انجام می گیرد.

**۲-۳-۱۰: نامه های وارده:**

نامه وارده به مدرکی اطلاق میشود که از خارج سازمان به مقصد داخل سازمان ارسال شده باشد . برای ثبت یک نامه وارده باید از یک فرم استفاده کرد . پس از ثبت شماره و تاریخ نامه وارده و مبداء نامه که میتواند شخص حقیقی یا حقوقی باشد باید گیرنده نامه را در بخش مقصد معرفی گردد . منظور از گیرنده نامه سمتی است درون سازمان که نامه وارده خطاب به وی نوشته شده است . این فرم از بخش های مختلفی تشکیل شده است که عبارتند از :

- مشخصات پایه
- متن و موضوع
- پرونده های موضوعی
- پیوست های نامه
- مشخصات تحویل

**۲-۳-۱۱: نامه های صادره :**

نامه صادره مدرکی است که از داخل سازمان به مقصد داخل یا خارج سازمان ارسال میگردد . برای ثبت یک نامه صادره باید از این فرم استفاده کرد . ابتدا باید گیرندگان داخلی و خارجی نامه را مشخص کنیم . سپس شرایط نامه را وارد و موضوع نامه را نیز در محل مربوطه ثبت کنیم . این فرم از بخش های مختلفی تشکیل شده است که عبارتند از :

- مشخصات پایه
- متن و موضوع

- پیوست های نامه

- مشخصات تحویل

## ۲-۳-۱۲: پیام داخلی :

پیام داخلی به مدارکی اطلاق می شود که بین کاربران یک سازمان در گردش است. برای ثبت و یا مشاهده یک پیام داخلی از این فرم استفاده می شود. فرستنده و گیرنده پیام های داخلی همواره کاربران می باشند.

تفاوت های زیادی بین یک پیام و گیرنده نامه وجود دارد.

که عبارتند از :

- پیام یک مدرک غیر رسمی می باشد به این شکل که همواره یک پیام قابل حذف می باشد  
حال آنکه نامه ها علاوه بر داشتن شماره و تاریخ ثبت غیر قابل حذف می باشند.

- پیام ها بین کاربران و نامه ها بین سمت ها در گردش می باشند.

دریافت کنندگان پیام می توانند پیام ها را به ۳ صورت اصل - رونوشت و یا رونوشت مخفی دریافت کنند.  
پس از مشاهده پیام می توان آنرا حذف کرد و به آن پاسخ گفت (Reply) و یا به کاربران دیگر ارجاع (forward) داد. همچنین قابلیت تبدیل پیام به نامه صادره در سیستم وجود دارد.

## ۲-۳-۱۳: ارجاع نامه :

برای ارجاع یک نامه از فرم ارجاع نامه استفاده می شود .

در این فرم به ازاء هر سمت دریافت کننده نامه میتوان به تفکیک موارد امنیت نامه ، اولویت نامه ، مهلت پاسخ و دستور را داشت .

یکی دیگر از امکانات این فرم افزودن کلید واژه پرونده موضوعی ، پیوست های نامه و پیروهای مختلفی است که میتوان آنها را به ازاء هر ارجاع به اصل نامه اضافه کرد .

**۲-۳-۲: چارت سازمانی:**

این فرم برای معرفی دبیرخانه ها و سمت های سازمانی به شکل درختی مورد استفاده قرار می گیرند . قبل از افزودن نام دبیرخانه ها بهتر است با یک سیستم منظم ، کد بندی خاصی برای دبیرخانه های سازمان و همچنین سمت ها در نظر گرفت و هنگام تعریف در قسمت کد آنها را وارد کرد .

دبیرخانه ها خود ساختار درختی دارند یعنی حالت سلسله مراتب سازمانی را در تعریف آنها باید در نظر گرفت.

بهتر است ابتدا ساختار درختی دبیرخانه ها بطور کامل وارد شوند و سپس به ورود اطلاعات مربوط به سمت ها پرداخت . نکته ای که وجود دارد این است که یک ند از نوع دبیرخانه می تواند فرزندی از نوع دبیرخانه و یا سمت داشته باشد . ولی یک ند از نوع سمت فقط فرزندی از نوع سمت می تواند داشته باشد .

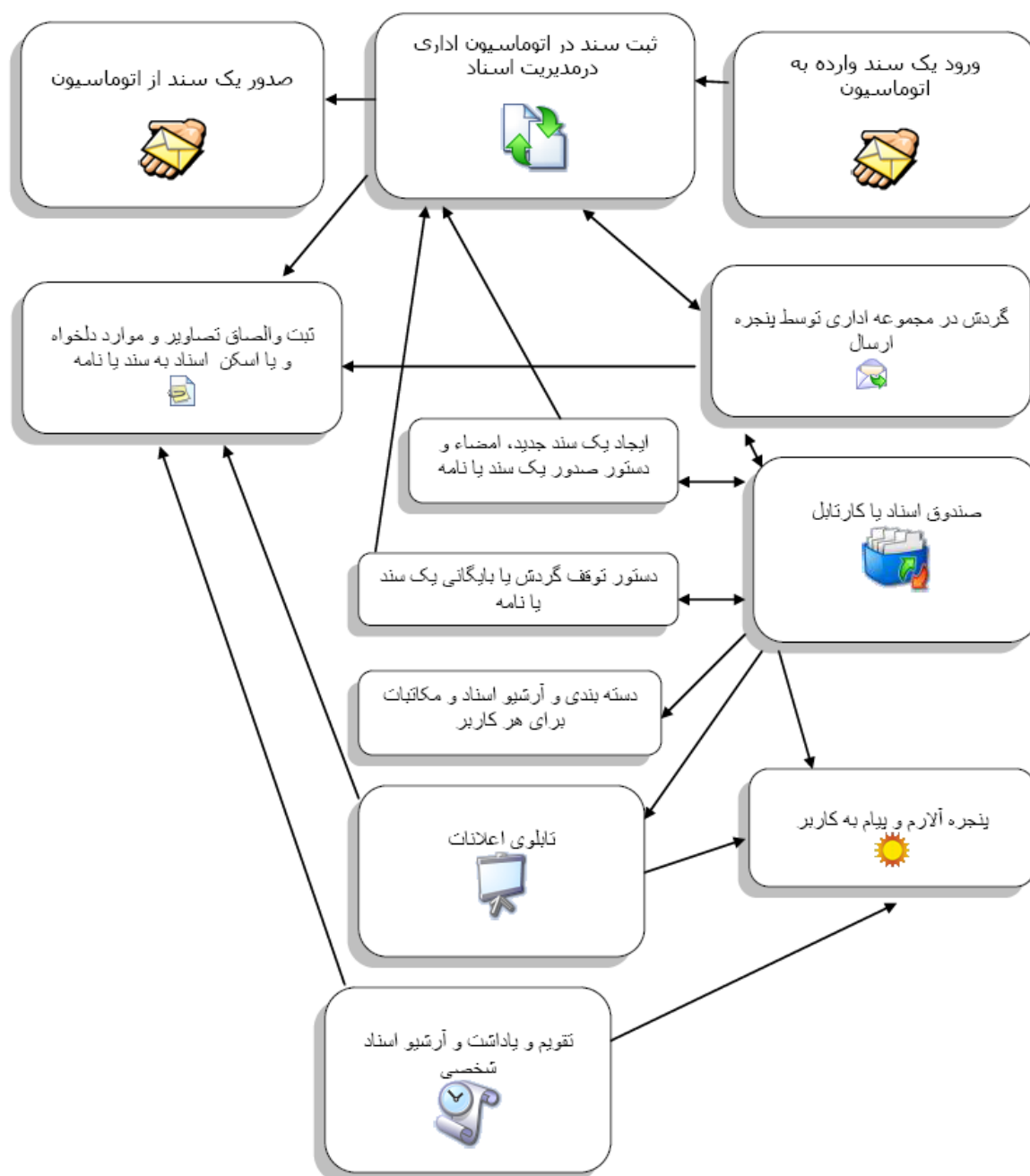
**۱- تعریف سمت****۲- نسبت دادن کاربر به سمت**

فرض بر این است که در این مرحله کاربران سیستم تعریف شده اند که این تعریف توسط فرم تعریف کاربر صورت می پذیرد.

**۲-۳-۲: قالب شماره نامه :**

با استفاده از این فرم میتوان قالب شماره نامه های کلیه دبیرخانه های موجود در سیستم را تعریف کرد . یکی از امکانات این فرم آن است که کلیه نامه ها میتوانند قالب شماره نامه یکسان داشته باشند و یا نامه های وارده و یا صادره به تفکیک هرکدام شماره نامه خاص خود را داشته باشند . در ضمن قالب های تعیین شده حتماً باید فیلدی به صورت شماره سریال داشته باشند این شماره سریال می تواند زوج و یا فرد و یا به صورت متوالی باشد .

## ۳-۳-۲: طرح شماتیک گردش اسناد اتوماسیون اداری سبا



شکل ۴-۲: گردش نامه در سازمان

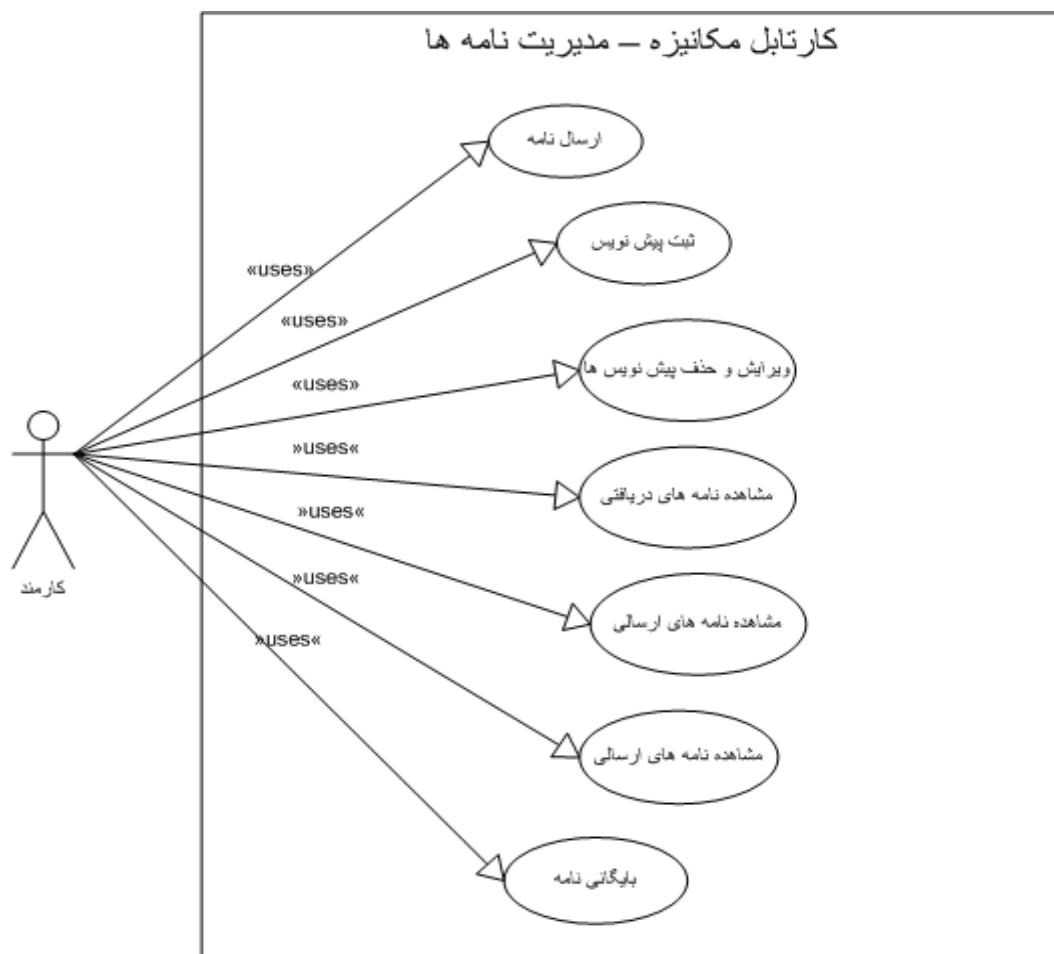
## ۲-۳-۴: نمودار های UML

در این قسمت به بررسی نمودار های UML برای سیستم اتوماسیون اداری می پردازیم.

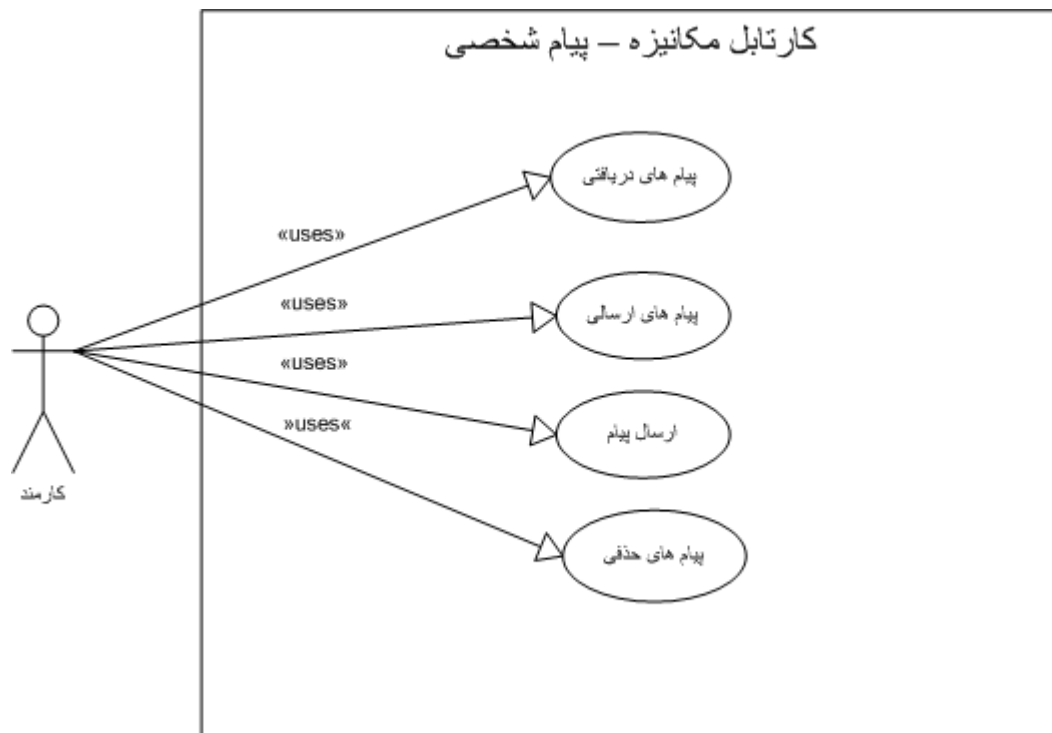
### ۲-۳-۴-۱: نمودار های مورد کاربرد (UseCase) :

در این نمودار کارها و تعامل هایی که کاربر با هر یک از بخش های نرم افزار داشته باشد، را ثبت می کنند.

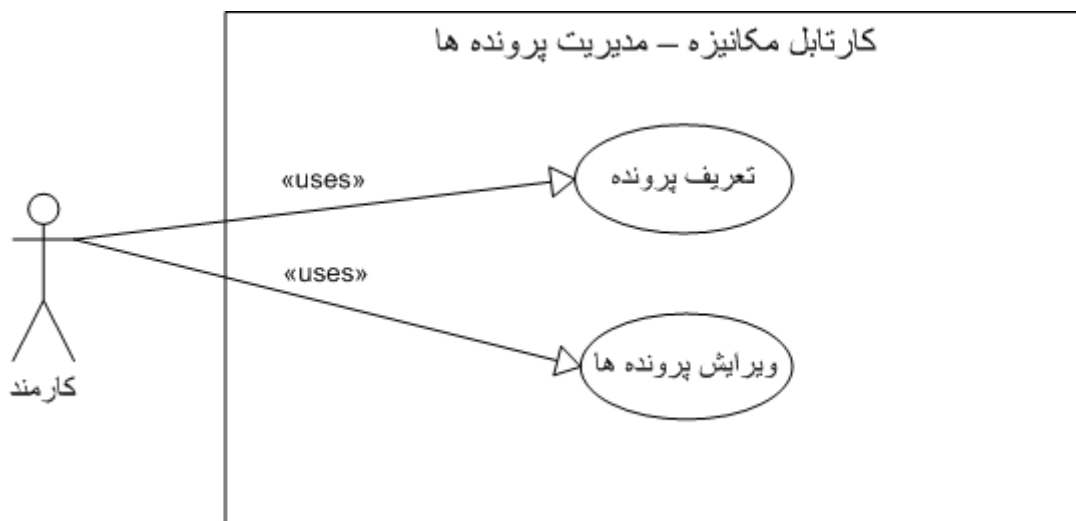
### ۲-۳-۴-۱-۱: کار تابل مکانیزه :



شکل ۲-۵: نمودار مورد کاربرد کار تابل - نامه

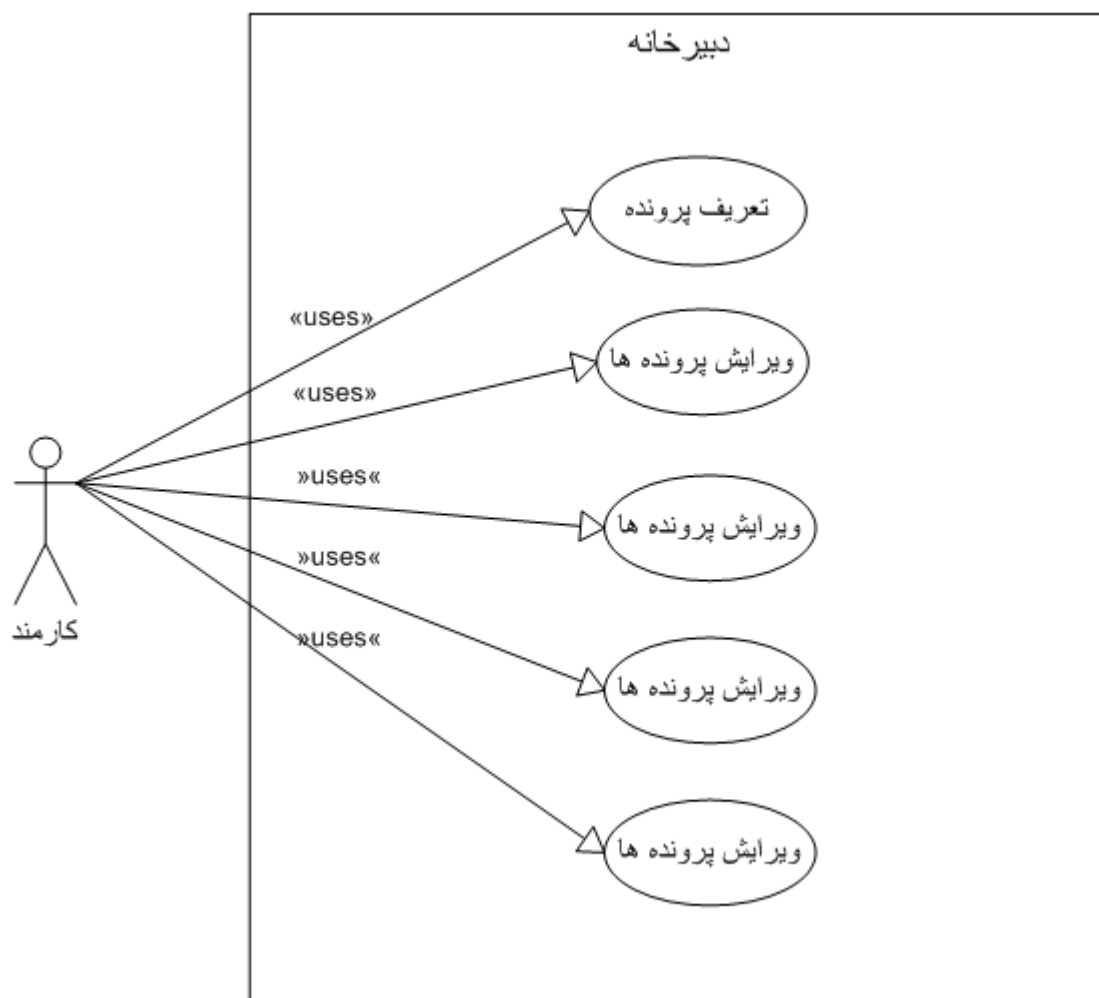


شکل ۶-۲: نمودار مورد کاربرد پیام



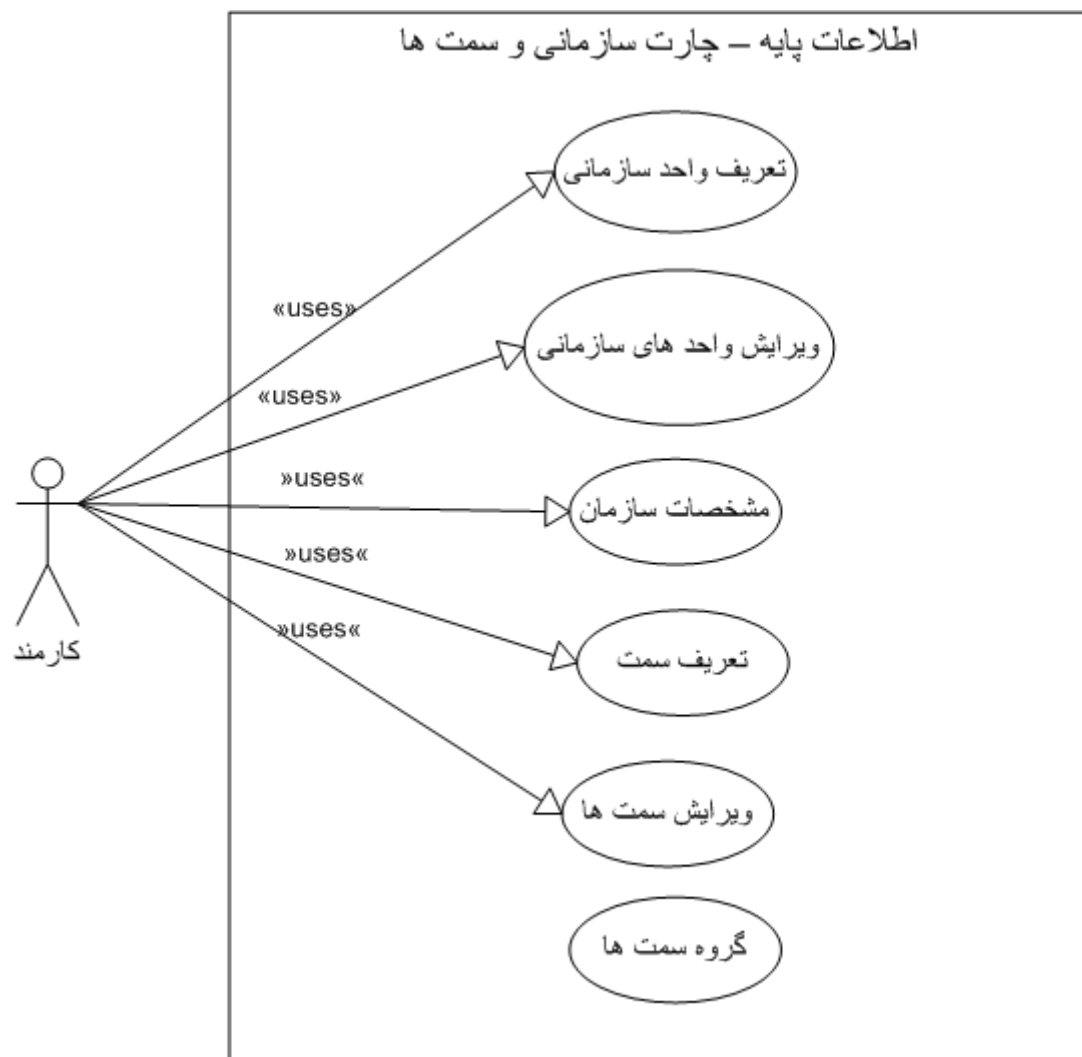
شکل ۷-۲: نمودار مورد کاربرد پرونده

۲-۳-۴-۱-۲: دبیرخانه :



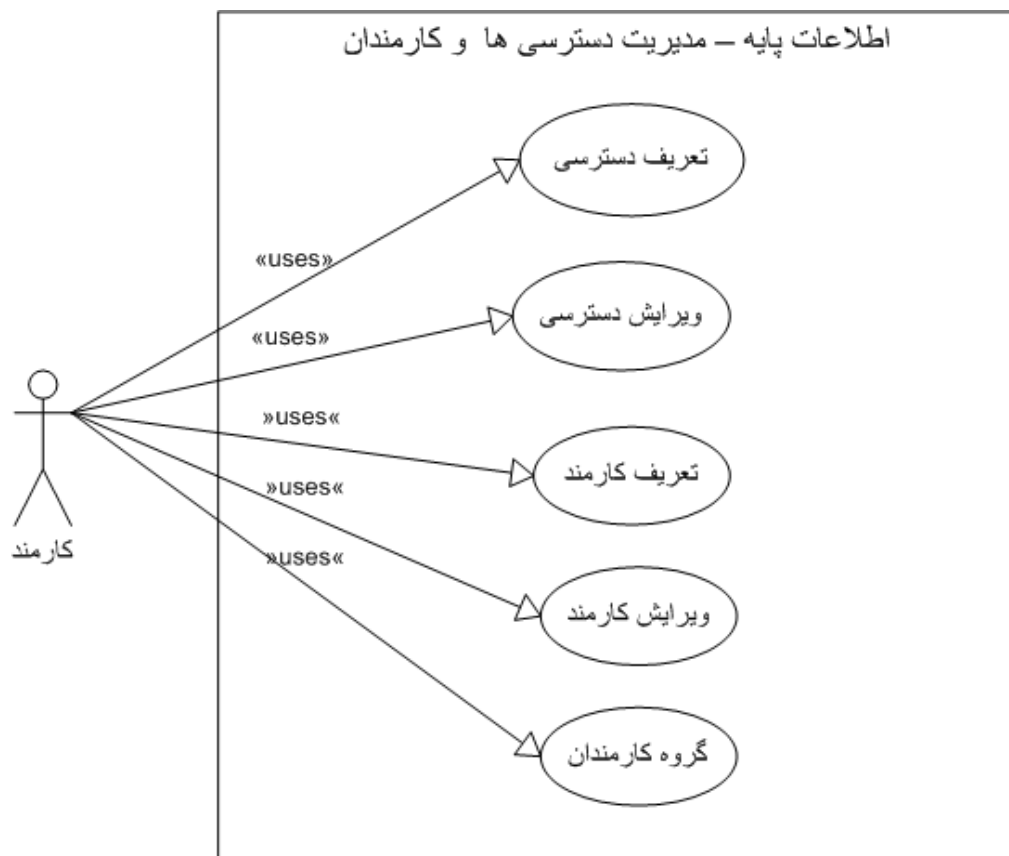
شکل ۲-۸: نمودار مورد کاربرد دبیرخانه

۲-۳-۴-۱-۳: اطلاعات پایه :

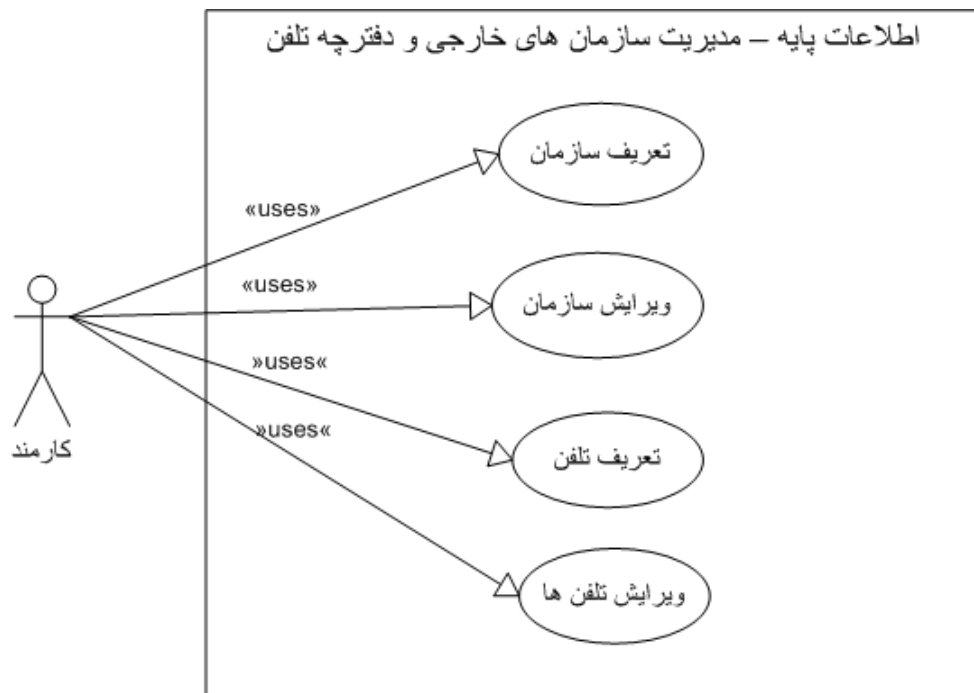


شکل ۲-۹: نمودار مورد کاربرد چارت سازمانی

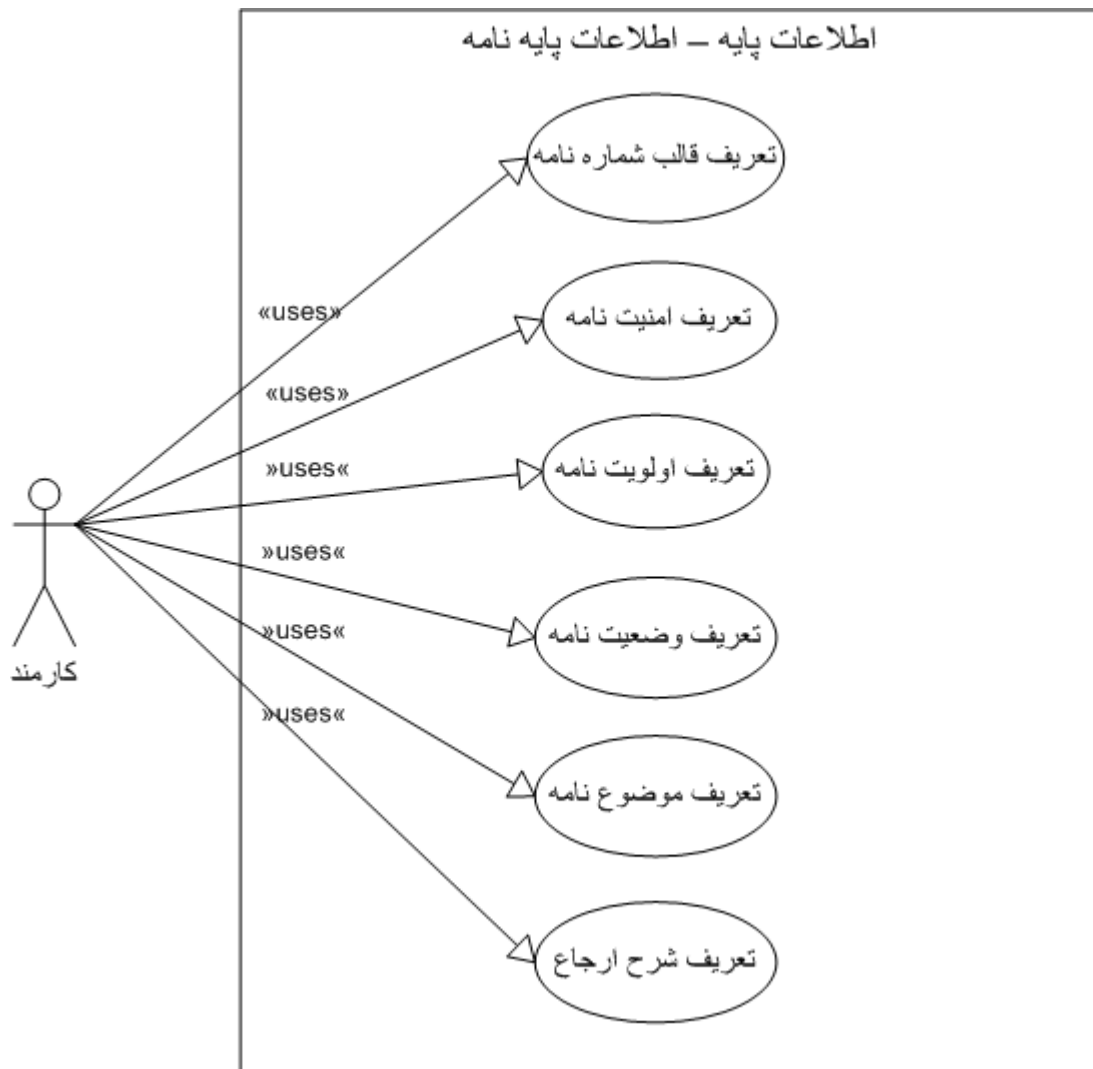




شکل ۲-۱۰: نمودار مورد کاربرد دسترسی ها

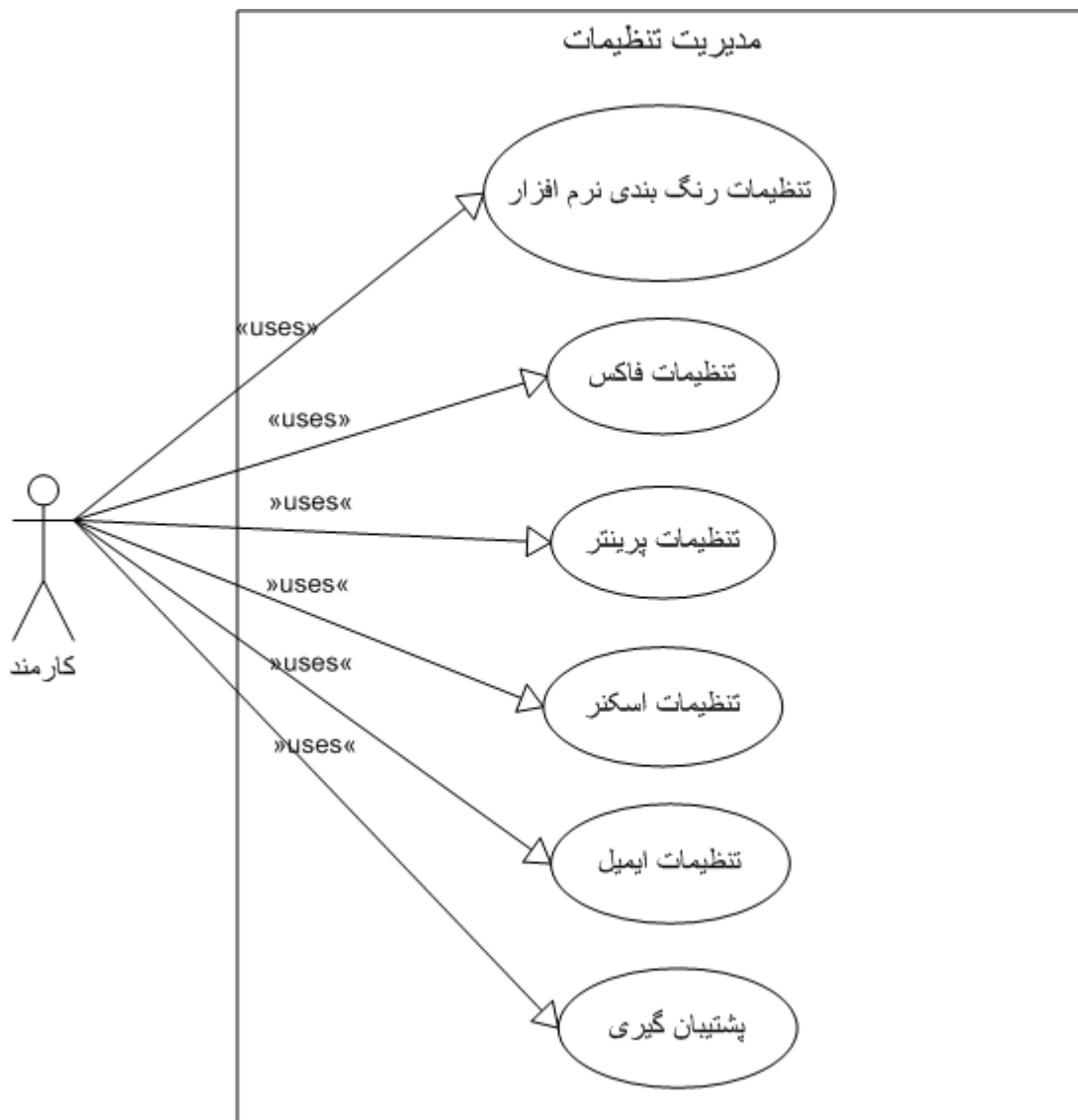


شکل ۲-۱۱: نمودار مورد کاربرد سازمان خارجی



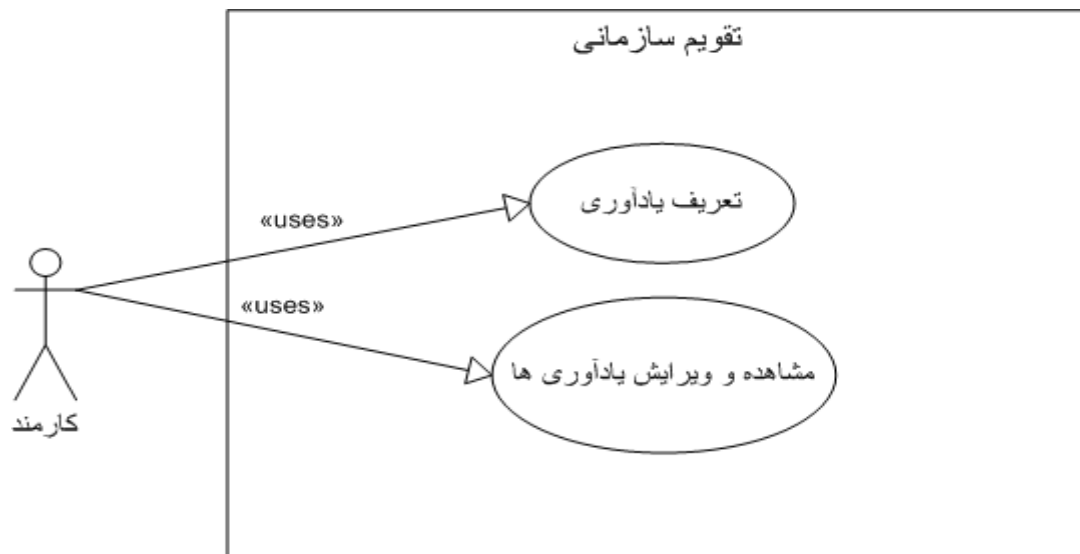
شکل ۲-۱۲: نمودار مورد کاربرد اطلاعات پایه نامه

۴-۱-۴-۳-۲: مدیریت تنظیمات :



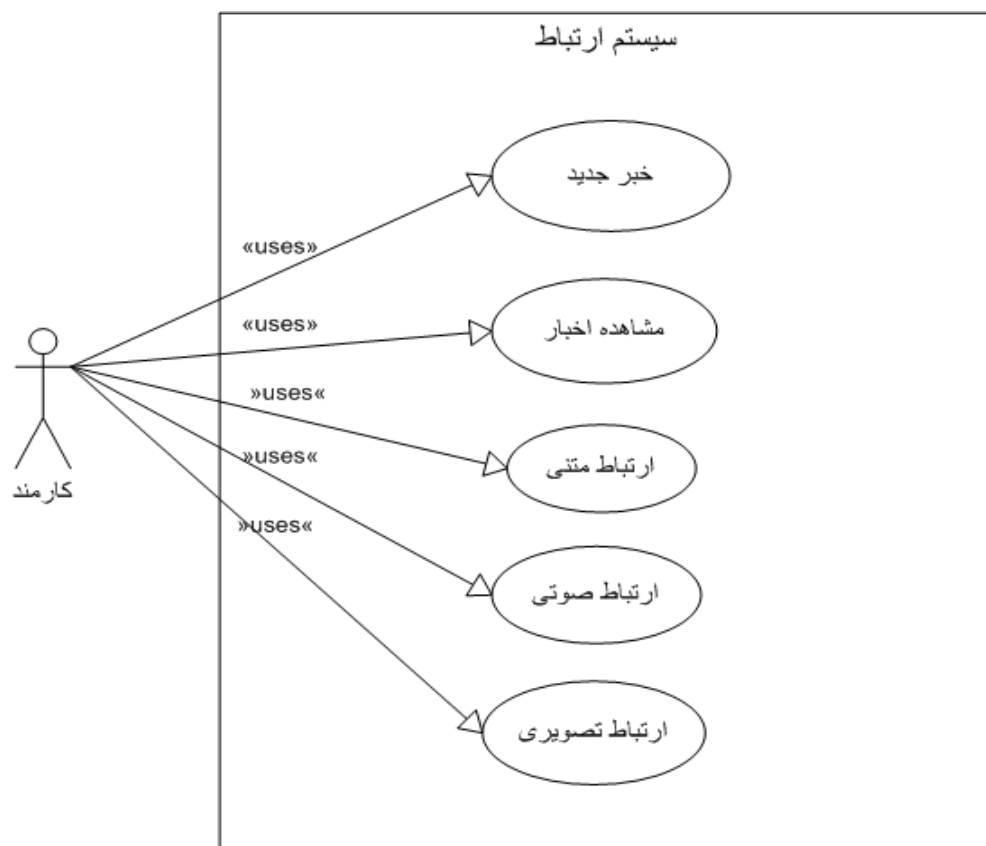
شکل ۲-۱۳: نمودار مورد کاربرد تنظیمات

۵-۱-۴-۳-۲: تقویم سازمانی :



شکل ۲-۱۴ : نمودار مورد کاربرد تقویم سازمانی

۶-۱-۴-۳-۲: سیستم ارتباط :



شکل ۲-۱۵ : نمودار مورد کاربرد سیستم ارتباط

## ۲-۳-۴-۲: نمودارهای کلاس ( Class ) :

این نمودارها برای نمایش ساختار موجودیت‌ها و اشیای موجد در نرم افزار به کار می‌روند و مشابه نمودارهای موجودیت‌های پایگاه داده هستند.

سازمائی واحد
decimal : واحد شماره-
string : واحد نام-
string : بالایی واحد-
() ها متد +

اداری سمت
int : سمت شماره-
string : سمت نام-
string : بالایی سمت-
string : سمت گروه-
() ها متد +

اداری سمت
int : سمت شماره-
string : سمت نام-
string : بالایی سمت-
string : سمت گروه-
() ها متد +

نویس پیش
string : کاربر-
string : نام-
string : متن-
() ها متد +

پرونده
string : پرونده نام-
() ها متد +

نامه
decimal : نامه شماره-
string : نامه تاریخ-
string : نامه اسم-
string : نامه صاحب-
string : فرستنده-
string : گیرنده-
string : موضوع-
string : اولویت-
string : امنیت-
string : وضعیت-
string : ارجاع شرح-
string : نامه متن-
int : آرشیو-
string : پرونده نام-
string : کننده امضا نام-
byte : امضا عکس-
() ها متد +

خارجی سازمان
string : شخص نوع-
string : شخص اسم-
string : تلفن-
string : فکس-
string : ایمیل-
string : سایت وب-
string : آدرس-
() ها متد +

پیام
decimal : پیام شماره-
string : پیام نام-
string : پیام صاحب-
string : فرستنده-
string : گیرنده-
string : پیام متن-
() ها متد +

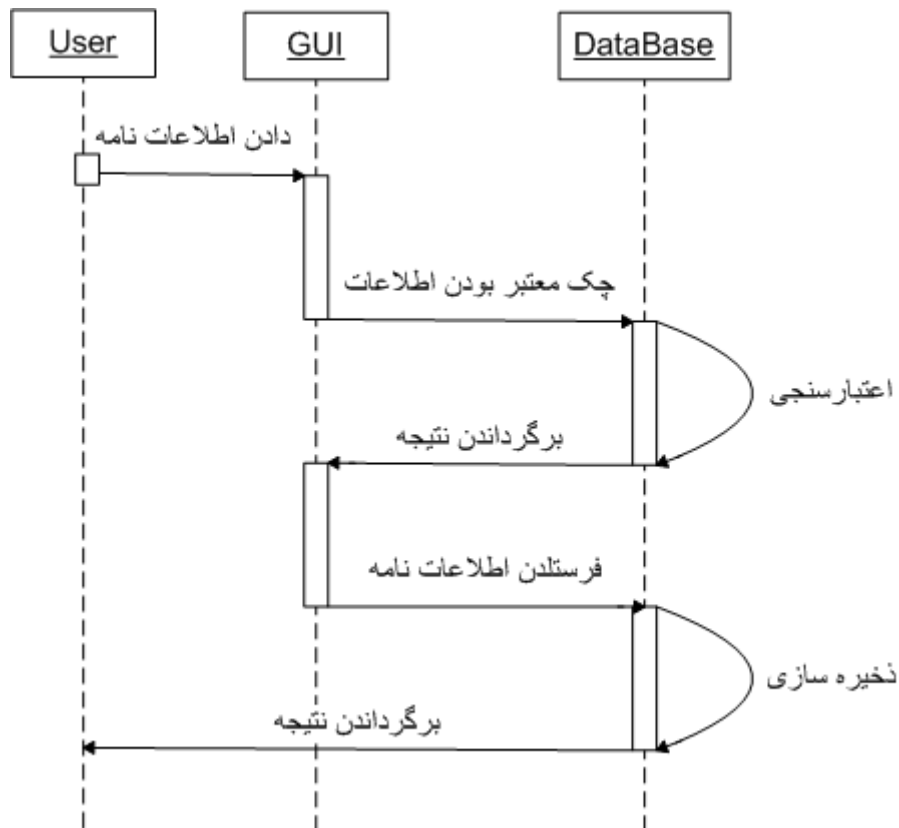
شکل ۲-۱۶: نمودار کلاس

دسترسی سطح
string : دسترسی نام-
int : نام ایجاد-
int : نویس پیش ایجاد-
int : ها نویس پیش ویرایش-
int : دریافتی های نام-
int : ارسال نام-
int : نام ارجاع-
int : نام آرشیو-
int : نام پیشرفت-
int : نام جستجوی-
int : پرونده تعریف-
int : پرونده ویرایش-
int : دریافتی های پیام-
int : ارسال های پیام-
int : پیام ارسال-
int : شده پاک های پیام-
int : وارده نام ثبت-
int : وارده نام مشاهده-
int : صادره های نام ویرایش-
int : صادره های نام مشاهده-
int : داخلی های نام مشاهده-
int : سازمانی واحد تعریف-
int : سازمانی واحد مشاهده-
int : سازمان مشخصات-
int : سمت تعریف-
int : ها سمت ویرایش-
int : ها سمت گروه-
int : دسترسی تعریف-
int : دسترسی ویرایش-
int : کارمند تعریف-
int : کارمند ویرایش-
int : کارمندان گروه-
int : سازمان تعریف-
int : سازمان ویرایش-
int : تلفن تعریف-
int : تلفن ویرایش-
int : نام امنیتی-
int : نام شماره قالب-
int : نام وضعیت-
int : نام اولویت-
int : نام موضوع-
int : ارجاع شرح-
int : افزار نرم بندی رنگ-
int : فاکس تنظیمات-
int : پرینتر تنظیمات-
int : ایمیل تنظیمات-
int : اسکنر تنظیمات-
int : گیری پشتیبان-
int : جدید یادآوری-
int : ها یادآوری ویرایش-
int : جدید خبر-
int : ها خبر ویرایش-
int : متنی ارتباط-
int : صوتی ارتباط-
int : ویدیویی ارتباط-
() ها متد+

## ۳-۴-۳-۲: نمودار ترتیبی (توالی) (Sequence):

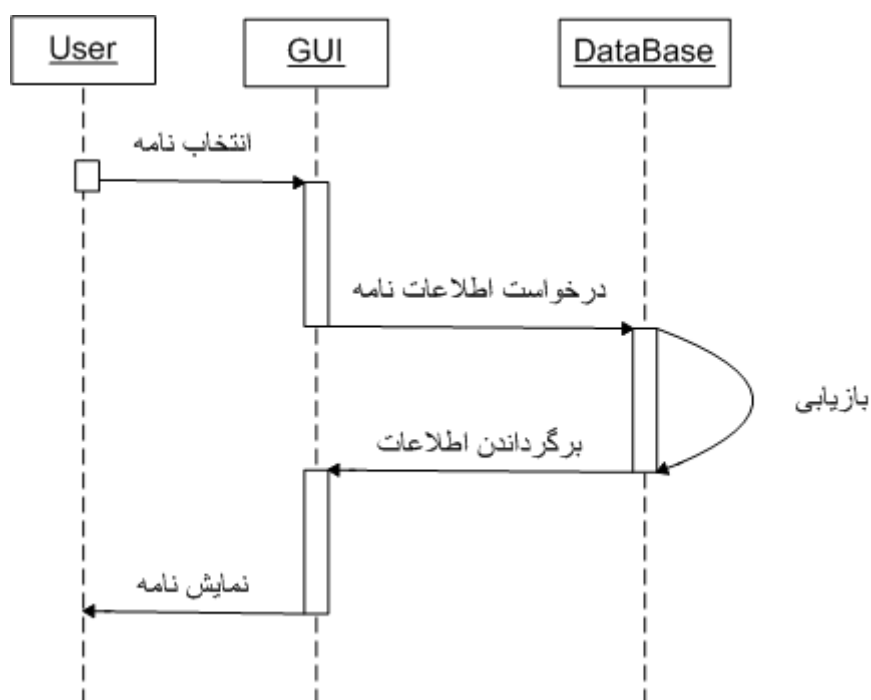
در این نمودار ما ترتیب انجام تراکنش های قسمتهای مختلف نرم افزار را ثبت می کنیم.

### ۳-۴-۳-۲: ارسال نامه :



شکل ۲-۱۷ : نمودار ترتیبی ارسال نامه

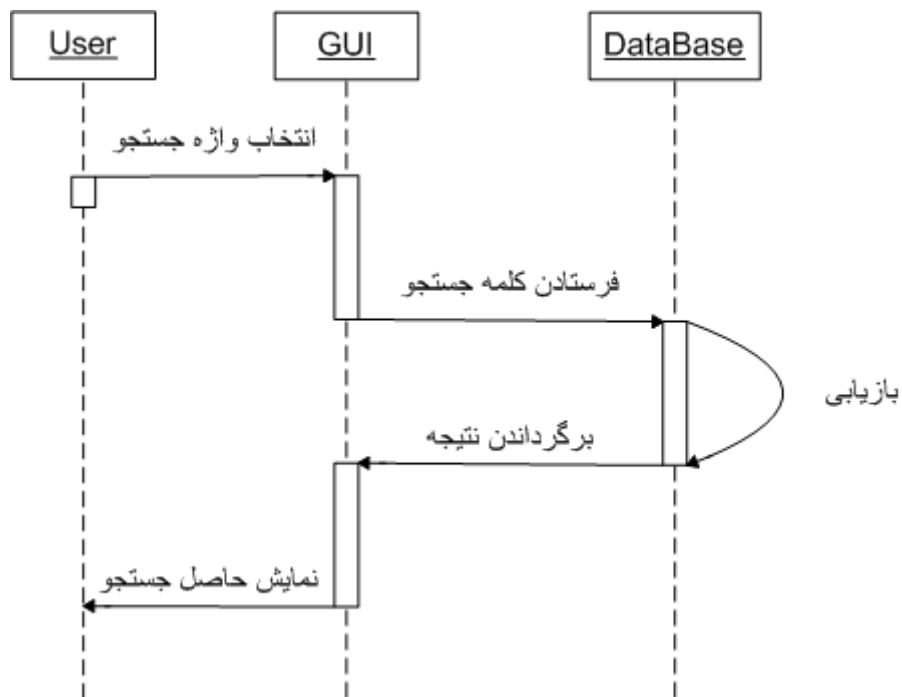
۲-۳-۴-۳-۲: خواندن نامه :



شکل ۲-۱۸ : نمودار ترتیبی خواندن نامه



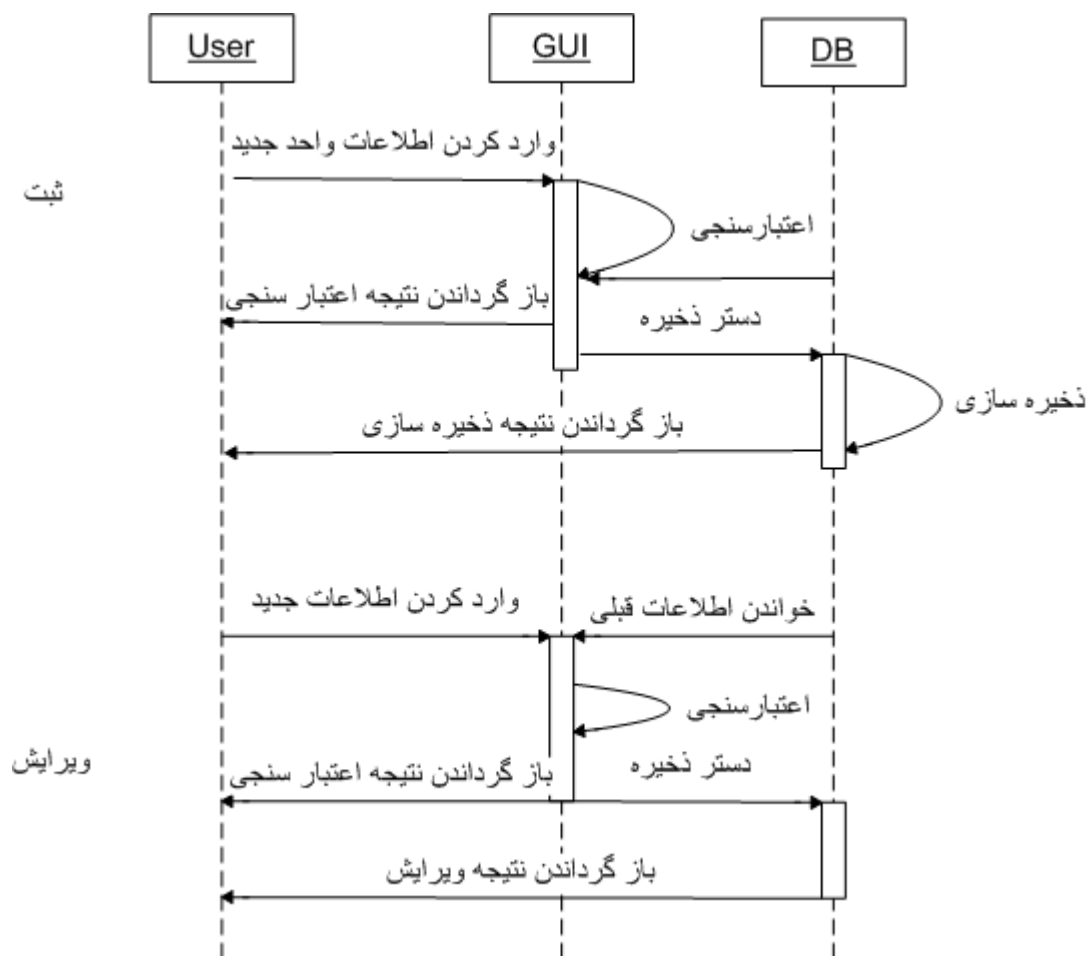
۳-۲-۴-۳-۳: جستجو نامه :



شکل ۲-۱۹: نمودار ترتیبی جستجو نامه

نمودارهای بالا عیناً برای پیام‌ها هم تکرار می شوند.

۴-۳-۴-۳-۲: ثبت و مدیریت اطلاعات پایه ( مثل کاربر و سمت و دسترسی و ... ) :



شکل ۲-۲۰: نمودار ترتیبی مدیریت اطلاعات پایه

## فصل سوم : ابزار سنجی

### ۳-۱: NET. چیست و چگونه شکل گرفت؟

**دات نت**، پلات فرم جدید مایکروسافت بمنظور تحقق نظریه: "نرم افزار بعنوان سرویس"، است. دات نت یک محیط پیاده سازی است که به کمک آن می توان اقدام به ایجاد و بکارگیری نرم افزار و نسل جدیدی از عناصر موسوم به "سرویس های وب"، نمود. تمامی محصولات اصلی مایکروسافت از ویژوال استودیو دات نت تا ویندوز و نهایتاً مجموعه آفیس، متأثر از پلات فرم فوق شده و خواهند شد. دات نت به پیاده کنندگان این امکان را خواهد داد که با زبان برنامه نویسی مورد علاقه خود، اقدام به پیاده سازی برنامه ها نمایند. ویژگی (پتانسیل) فوق از طریق معرفی (CLR(Common Language Runtime)، میسر شده است. در این مقاله قصد در بررسی دات نت و پتانسیل ها و قابلیت های گسترده آن است.

در جولای سال ۲۰۰۰، شرکت مایکروسافت در کنفرانس پیاده کنندگان حرفه ای (PDC)، در شهر Orlando ایالت کالیفرنیا، جزئیات بیشتری از نسل جدید پلات فرم خود (دات نت) بمنظور پیاده سازی برنامه های ویندوز و اینترنت را در اختیار علاقه مندان خصوصاً پیاده کنندگان نرم افزار قرار داد. محوریت ابتکار فوق، بر فریمورک دات نت استوار بوده و نشاندهنده یک پلات فرم مناسب به همراه کتابخانه های کلاس گسترده ای است که پتانسیل های متعددی را در اختیار قرار می دهد. یکی از نکات قابل توجه در پلات فرم فوق، استفاده از XML و SOAP بمنظور ارتباط بین نرم افزارها می موجود در اینترنت (نرم افزارهای مبتنی بر وب)، است. در این راستا مجموعه ای از محصولات مبتنی بر سرویس دهنده با نام سرویس دهندگان Enterprise دات نت، مطرح که به منزله نسل جدیدی از محصولات Backoffice مایکروسافت می باشند.

فریمورک دات نت، مدلی کاملاً جدید به منظور برنامه نویسی و بکارگیری نرم افزار را ارائه نموده است. "بیل گیتس"، در سخنرانی خود در PDC، بدین نکته اشاره داشت که در هر پنج تا شش سال ما شاهد یک تحول عمده در رابطه با پیاده سازی نرم افزار بوده ایم. آخرین موارد در این زمینه به سوئیچ از DOS به ویندوز در سال ۱۹۹۰ و گذر از پیاده سازی شانزده بیتی به سی و دو بیتی (از ویندوز widows 3.x به ویندوز NT/95) در اواسط دهه ۹۰ میلادی است.

با معرفی دات نت در PDC، پیاده کنندگان آن را معماری مناسبی برای پیاده سازی نرم افزار (برنامه های Desktop و برنامه های وب) مشاهده نمودند. ویژوال استودیو دات نت، اولین محصول مبتنی بر دات نت

مایکروسافت بود که در سال ۲۰۰۱ در اختیار علاقه مندان قرار گرفت. اهمیت دات نت برای مایکروسافت تا بدین حد است که در سال ۲۰۰۱، بیش از هشتاد درصد منابع بخش تحقیق و توسعه این شرکت در رابطه با آن صرف شده است. زبان سی شارپ، که زبانی جدید برای برنامه نویسی در دات نت است بعنوان زبان استاندارد برای پیاده سازی داخلی در شرکت مایکروسافت پذیرفته شده است.

### ۳-۱-۱: یک پلات فرم مناسب برای آینده

دات نت، اولین پلات فرم طراحی شده از صدر تا ذیل با در نظر گرفتن واقعیتی با نام اینترنت است. دات نت از یک ماشین مجازی خاص در این زمینه استفاده می نماید. ماهیت ماشین مجازی فوق، بگونه ای است که از API ویندوز فاصله و در این رابطه از یک کتابخانه کلاس استفاده می نماید که می توان به جرات این ادعا را داشت که تاکنون نظیر آن، ایجاد نشده است. امکان استفاده از زبانهای متعدد برنامه نویسی، وجود خواهد داشت. معماری دات نت، امکان ارتباط بین زبانها را به سادگی فراهم خواهد کرد. دات نت، یک رویکرد جدید در رابطه با پیاده سازی نرم افزار را مطرح نموده است. نگاه به دات نت، عمدتاً بصورت سیستم های توزیع شده است. با استفاده از XML، امکان اجرای توابع بر روی کامپیوترهای متفاوت یک سازمان و یا جهان فراهم و جلوه ای زیبا در همیاری بمنظور اجرای یک برنامه، به نمایش در خواهد آمد. از این منظر، سیستم ها از سرویس دهندگان تا سیستم های بدون کابل، قادر به اشتراک پلات فرم عمومی یکسانی خواهند بود. با استفاده از نسخه های دات نت که برای تمام آنها در دسترس خواهد بود، امکان ارتباط مناسب آنها با یکدیگر فراهم خواهد شد. دات نت، بمنظور طراحی و پیاده سازی برنامه های سنتی نیز راهکارها و امکانات مناسبی را ارائه تا از این طریق امکان پیاده سازی و بکارگیری این نوع از نرم افزارها، به سادگی انجام گیرد. برخی از تکنولوژی های ارائه شده در دات نت نظیر فرم های ویندوز، تلاشی در این راستا است.

### ۳-۱-۲: ایده های اولیه

از اواخر سال ۱۹۹۵، شرکت مایکروسافت توجهی خاص و قابل توجه نسبت به اینترنت نمود. هدف مایکروسافت در این زمینه پیوند بین پلات فرم ویندوز و اینترنت بود. ماحصل تلاش مایکروسافت در این زمینه ارائه مدل برنامه نویسی Windiws DNA، بود. در این راستا مجموعه ای از ابزارها و تکنولوژی های مبتنی بر اینترنت، طراحی و ارائه گردید ASP.، از اولین تلاش های مایکروسافت در این زمینه است. عملاً در این زمینه (مطرح شدن اسکریپت های مفسری) یک برگشت به عقب نسبت به پیاده سازی ساخت یافته و شی گراء را شاهد بوده ایم. طراحی، اشکال زدائی و نگهداری چنین کدهای غیر ساخت یافته ای مسائل خاص خود را خواهد داشت. سایر زبانها نظیر ویژوال بیسیک بصورت موفقیت آمیز در رابطه با برنامه نویسی بر روی اینترنت و پلات فرم مایکروسافت استفاده می گردید ولی اغلب از آن به منظور ایجاد عناصری که از طریق ASP، به خدمت گرفته می شدند، استفاده می گردید. در این رابطه تلاش های اندکی نیز در جهت ایجاد یک Interface مبتنی بر وب بر روی زبان های سنتی

نظیر webclasses در VB ، نیز انجام شد ولی هیچکدام از تلاش های فوق ، در سطح گسترده ای مورد استقبال و پذیرش قرار نگرفت . میکروسافت در صدد حل آشفتگی های همراه برنامه های ویندوز DNA بود . ویندوز DNA ، تصویری مناسب از یک معماری Three-Tire و مبتنی بر COM بود که تکنولوژی ASP در لایه Presentation، اشیاء Bussiness در لایه میانی و یک Engine بانک اطلاعاتی رابطه ای در لایه Data ، قرار می گرفت . مفاهیم همراه DNA ، کامل و بی عیب بود اما در زمان استفاده عملیاتی چالش های خاص خود را بدنبال داشت . پیاده سازی عناصر COM ، مستلزم یک سطح مناسب از دانش و مهارت است و می بایست زمان زیادی در این رابطه صرف گردد . بکارگیری نرم افزارهای DNA ، نیز مسائل خاص خود را داشت ( مسائل مربوط به ورژن ، نصب عناصر و عناصری که با آن مرتبط می باشند).

بموازات تلاش سایر شرکت ها در رابطه با ارائه راهکارهایی خاص بمنظور پیاده سازی برنامه های وب ، شرکت میکروسافت در صدد برطرف نمودن محدودیت های مدل برنامه نویسی DNA گردید .

### ۳-۱-۳: تولد دات نت

در اوایل سال ۱۹۹۸ ، گروهی از پیاده کنندگان نرم افزار در میکروسافت ، کار خود را بر روی نسخه ای جدید از ( IIS نسخه چهار ) ، به اتمام رساندند که دارای چندین ویژگی جدید در رابطه با ASP بود . در این راستا ، قابلیت های جدیدی بمنظور پیاده سازی برنامه های وب در ویندوز NT ، فراهم گردید. گروه پیاده کننده دارای ایده های متعددی برای اعمال اصلاحات جدید بودند . گروه فوق ، کار خود را بر روی یک معماری جدید برای پیاده سازی ایده های مطرح شده ، آغاز نمود . این پروژه ، ( Next Generation Window Services ) (NGWS). نامیده گردید. پس از ارائه ویژوال استودیو شش ، در اواخر سال ۱۹۹۸ ، تلاش برای ایجاد نسخه ای جدید از ویژوال استودیو در دستور NGWS ، قرار گرفت . گروه COM+/MTS در مدل پیشنهادی خود از یک Runtime ، عمومی برای تمامی زبانهای استفاده شده در ویژوال استودیو ، استفاده نمودند . تلاش افراد درگیر در پروژه NGWS ادامه یافت تا در نهایت، شرکت میکروسافت در کنفرانس پیاده کنندگان حرفه ای (PDC) ، دات نت را معرفی نمود .

### ۳-۱-۴: مروری بر فریمورک دات نت :

فریمورک دات نت ، تمامی لایه های پیاده سازی نرم افزار را از سطح سیستم عامل به بالا ، تحت پوشش قرار می دهد. فریمورک فوق، سطحی مناسب وقدرتمند از ارتباط و همبستگی بین تکنولوژی Presentation ، تکنولوژی های Component و تکنولوژی های Data را ارائه می نماید ( نظیر این ارتباط و همبستگی تاکنون در پلات فرم ویندوز مشاهده نشده است) . معماری فوق ، امکان طراحی و پیاده سازی برنامه های مبتنی بر اینترنت و محیط های Desktop ، را به سادگی فراهم و نیازهای هر گروه از نرم افزارهای فوق را بخوبی جواب می دهد. فریمورک دات نت از لایه پائین با عملیاتی نظیر مدیریت حافظه آغاز و به سمت بالا به منظور ارائه اینترفیس های

برنامه ها و کاربران، دنبال می شود. در بین لایه ها، لایه های سیستمی دیگر که هر یک دارای پتانسیل های خاصی برای پیاده کنندگان می باشند، وجود دارد.

(CLR) Common Language Runtime، بمنزله قلب فریمورک دات نت محسوب و Engine لازم بمنظور ارائه قابلیت های کلیدی را ارائه می نماید CLR. شامل عناصر اساسی دیگری نظیر (Common Type System (CTS، است. علاوه بر مدیریت حافظه، CLR، مراجعات به اشیاء و عملیات Garbage Collection را نیز انجام می دهد.

در لایه میانی، ما شاهد نسل جدیدی از سرویس های استاندارد نظیر ADO.NET و XML می باشیم. سرویس های فوق، تحت کنترل فریمورک بوده و امکان به کارگیری آنها بصورت جامع و استاندارد در بین تمامی زبانها، فراهم می گردد. بالاترین لایه، شامل اینترفیس های برنامه و کاربر است. فرم های ویندوز، روشی جدید به منظور ایجاد برنامه های Desktop مبتنی بر win32 می باشند. فرم های وب، یک رابط کاربر مناسب برای برنامه های مبتنی بر وب را ارائه می نمایند. سرویس های وب، مکانیزمی بمنظور ارتباط برنامه ها از طریق اینترنت و با استفاده از SOAP، می باشد. سرویس های وب، قابل مقایسه با عناصر COM و DCOM بوده با این تفاوت مهم که در این راستا از تکنولوژی های متعدد اینترنت، استفاده می گردد. فرم های وب و سرویس های وب، اینترفیس اینترنت دات نت را تشکیل و پیاده سازی آنان از طریق بخش دیگری در فریمورک دات نت که ASP.NET، نامیده می شود، محقق می گردد.

پتانسیل های موجود در هر لایه فریمورک دات نت، توسط هر یک از زبان های سازگار با دات نت، قابل استفاده خواهد بود. در پایان لازم است به این نکته اشاره گردد که در این رابطه (فریمورک دات نت) می توان از اینترفیس های مبتنی بر متن (کاراکتری) نیز استفاده کرد. این نوع برنامه ها اصطلاحاً Console Application، نامیده می شوند.

## ۲-۳: شروع کار با ویژوال استودیو دات نت ۲۰۰۵

### ۲-۳-۱: مقدمه:

محیط مجتمع دات نت فریم ورک ۲.۰ که تحت نرم افزار Microsoft Visual Studio .Net 2005 ارائه می گردد، محیطی است کامل و جامع از بهترین و قویترین زبانهای برنامه نویسی ویژوال مانند C#.NET، C++ .Net، VB.NET، و نیز ابزاری قدرتمند برای طراحی و انتشار صفحات وب توسط فناوری ASP.NET می باشد.

شاید برای شما نیز جالب باشد که بدانید از تمامی خواص زبانهای دات نت می توان در دیگر زبانهای دات نت استفاده کرد! به عنوان مثال فرض کنید یک کمپوننت به زبان VB.NET نوشته اید. حال می توانید از این بسته بدون هیچ تغییری در زبانهای دیگر مثل C#.NET و C++ .Net و ... استفاده کنید. این مساله به آن دلیل است که تمامی کدهایی که در قالب محیط دات نت نوشته می شوند قبل از کامپایل به کدهایی که به زبان میانی (Intermediate Language) یا IL معروفند تبدیل می شوند. در مورد IL مفصلاً بعداً توضیح خواهیم داد.

### ۲-۳-۲: اجرای MS VS.NET

هنگامی که برای اولین بار برنامه Microsoft Visual Studio .Net 2005 را اجرا می کنید تصویری مانند تصویر زیر مشاهده خواهید کرد:





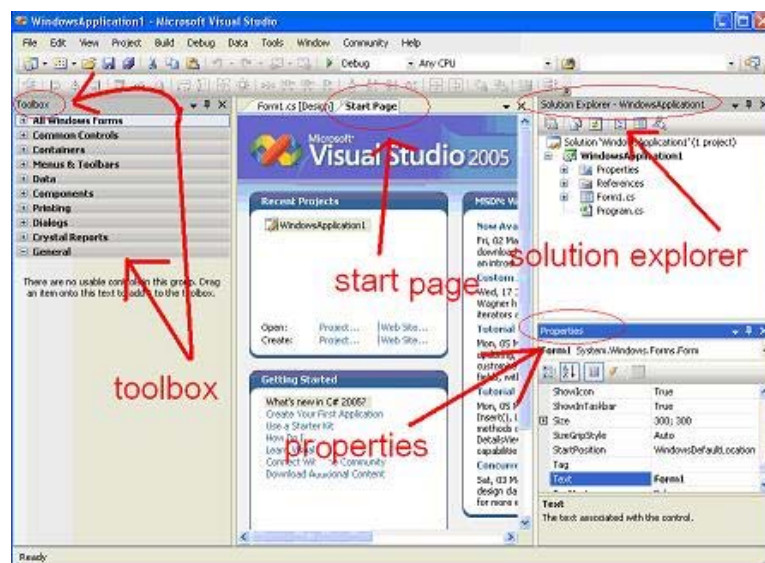
شکل ۳-۱: شروع

دقت کنید که این پنجره فقط بار اول که برنامه را اجرا می کنید ظاهر می شود. این پنجره جهت پیکربندی محیط دات نت بکار می رود. پیکر بندی محیط دات نت برای زبان های مختلف ممکن است کمی با هم تفاوت داشته باشند. این تفاوت جهت راحتتر بودن کاربر برای استفاده از آن زبانی است که در این پنجره انتخاب نموده است. البته در مورد انتخاب خود نگران نباشید. شما هر زمانی که اراده کنید می توانید تنظیمات محیط دات نت فریم ورک را مطابق میل خود پیکر بندی کنید.

چون قصد دارم در مورد زبان #C صحبت کنم، با انتخاب Visual C# Development Settings کار را ادامه می دهیم.

پس از انتخاب زبان #C بر روی Start Visual Studio کلیک کنید. پنجره فوق بسته شده و پنجره زیر نمایش داده خواهد شد:

متنی که بر روی این پنجره نوشته شده گواه صحبت‌هایی است که در بالا عنوان شد. پس از اینکه تنظیمات به پایان رسید پنجره اصلی نرم افزار باز خواهد شد. این پنجره در شکل زیر نشان داده شده است:



شکل ۳-۲: خصوصیات ویژوال استدیو

همانطور که در پنجره فوق مشخص کرده ام، سه پنجره به نام های **Solution Explorer** ، **Properties** و **Toolbox** ، و یک **Tab page** به نام **Start Page** وجود دارند.

توضیحات مختصری در مورد این پنجره ها می دهم و توضیحات مفصلتر را در درسهای آتی خواهم گفت.

**Start Page :**

این **Tab page** هر بار به هنگام راه اندازی **Visual Studio** نمایان می گردد. همانطور که در این **Tab page** می بینید قسمت هایی نظیر **Recent Projects** لیست پروژه هایی که اخیرا انجام گرفته (، **Create: project ... Website** ایجاد پروژه یا وب سایت جدید) **Open: Project ... Website** (باز کردن پروژه یا وب سایت موجود) می باشد. در این موارد بعدا توضیحات بیشتری ارائه خواهد شد.

### ۳-۳: نحوه کار با SQL Server 2005 :

خب! امیدوارم بدونید SQL Server چیه! کلاً دو جور دیتابیس داریم. دیتابیس‌های Desktop و دیتابیس‌های Client/Server.

دیتابیس‌های معروف Desktop عبارتند از:

Access – dBase

دیتابیس‌های Client/Server معروف هم عبارتند از:

DB2 – Oracle - MS SQLServer – MySQL

مهمترین ویژگی دیتابیس‌های Client/Server این هستش که کاربر بطور مستقیم به دیتابیس دسترسی نداره. یعنی باید از طریق یک سرویس واسط دستور خودش رو برای اجرا به سیستم بده. در حالی که در دیتابیس‌های Desktop اینگونه نیست و همین خصوصیت باعث میشه که دیتابیس‌های Client/Server نسبت به دیتابیس‌های Desktop امنیت بالاتری داشته باشند.

برای اینکه بفهمیم SQL Server 2005 روی یک کامپیوتر نصب شده یا نه، اکثراً دنبال برنامه Management Studio می‌گردید. در حالی که وجود برنامه Management Studio نه شرط لازم و نه شرط کافی برای وجود سرویس SQL Server 2005 روی یک سیستم می‌باشد. روش درست برای اطمینان از وجود SQL Server 2005 در یک سیستم، چک کردن مسیر زیر است:

Control Panel>Administrative Tools>Services>SQL Server (MSSQLServer)

یعنی سرویسی به نام SQL Server (MSSQLServer) باید در لیست سرویس‌های ویندوز موجود باشد.

### ۳-۳-۱: مواردی در خصوص نصب SQL Server 2005

اگر بخواهید SQL Server 2005 را روی کامپیوتر شخصی خودتون نصب کنید، توجه داشته باشید که اگر ویندوزتون XP هست نمی‌تونید نسخه Enterprise رو روی سیستمتون نصب کنید و باید از نسخه‌های Standard و یا Developer استفاده کنید. ضمناً ویندوز XP شما حتماً باید SP2 به بالا باشد.

ولی اگر بخواهید SQL Server 2005 رو برای شرکت و یا سازمانی نصب کنید، باید نسخه Enterprise رو روی Windows Server 2003 با SP1 به بالا نصب شود.

فراموش نکنید که قبل از نصب SQL Server 2005 (حتماً) DotNet Framework 2.0 و بالاتر) روی سیستمون نصب شده باشه. برای دیدن Minimum Requirement ها و Windows Compatibility ورژن‌های مختلف SQL Server 2005 میتونید این صفحه رو ببیند.

کلاً در یک کامپیوتر می‌توان تعداد ۶۴ سرویس SQL Server 2005 نصب کرد. ولی حداکثر ۴ عدد instance را میتوان نامگذاری و تعریف نمود.

در هنگام نصب SQL Server 2005 به یکسری اصطلاحات برمیخورید که شاید بعضی از اونها رو ندونید. من به اختصار اونها رو توضیح میدم:

### ۲-۳-۳: Reporting Service

اگر در زمان نصب SQL Server 2005 این گزینه فعال شود، بخش گزارشگیری خودکار روی سیستم شما نصب خواهد شد. این سرویس امکاناتی را برای ایجاد گزارش از بانک های اطلاعاتی مختلف در اختیاران قرار می دهد. گزارشهای ایجاد شده توسط این سرویس Web-enabled هستند و قابلیت پخش روی انواع دستگاهها را دارند. شما می توانید این گزارشات را با فرمتهای مختلف Excel و Word و PDF و Html و ... ایجاد کنید.

### ۳-۳-۳: Analysis Service

قابلیت پردازش های تجزیه و تحلیلی آنالاین (OLAP) با سرعت بالا، تجزیه و تحلیل پیشرفته برای مجموعه دیتابیس های پیچیده و بزرگ با استفاده از راههای متعدد ذخیره سازی اطلاعات. در حقیقت اگر بخواهیم روی دیتابیس های خیلی بزرگ گزارشگیری کنیم، پروسه ساخت گزارش خیلی کند است. برای این منظور یک مخزن اطلاعات جنبی در نظر گرفته می شود و گزارشات از روی آن تهیه می شود. نتیجه گزارش بر روی یک فضای جداگانه قرار می گیرد و چنانچه اطلاعات تغییر کند، تغییرات ایجاد شده بصورت اتوماتیک بر روی گزارشات اعمال می شود. فرض کنیم که پلیس ۱۱۰ یک نرم افزار در اختیار دارد که توسط آن آمار جرائم را نگهداری می کند. در این آمار،

پلیس اطلاعات مربوط به مکان و زمان وقوع جرم و نوع جرم مثلاً دزدی را نگهداری می کند. پس از مدتی پلیس می تواند اطلاعات مربوط به دزدی در یک مکان خاص را بررسی کند. مثلاً پلیس در بررسی و تحلیل اطلاعات خود پی می برد که در فلان محله خاص بین ساعت ۸ الی ۱۰ صبح هیچ مورد دزدی گزارش نشده است بنابراین می تواند نیروهای خود را در آن زمان خاص در آن محله خاص کاهش داده و در جای دیگری که امکان وقوع دزدی بیشتر است استفاده کند.

### Notification Service : ۴-۳-۳

این سرویس محیطی برای ایجاد برنامه هایی است که پیامهایی را ایجاد و ارسال میکنند. از این سرویس می توانید برای ایجاد و ارسال پیامهای شخصی و زمانبندی شده به هزاران یا میلیونها شخص یا دستگاههای گوناگون استفاده نمایید.

مثلاً تصور کنید که تعدادی کاربر در حال اتصال به یک بانک حاوی اطلاعات مربوط به ارزش سهام در بورس هستند. از آنجایی که ممکن است قیمت سهام هر شرکت یا مؤسسه برای تعدادی از کاربران از اهمیت زیادی برخوردار باشد، می توان این سیستم را طوری تنظیم کرد تا هرگاه ارزش سهام خاصی که مورد نظر هر کاربر است تغییر کرد، به صورت اتوماتیک به وی اعلام شود. کاربر هم می تواند این تغییرات را بروی برنامه کاربردی خود، تلفن همراه (در قالب SMS ، Windows Messenger) و یا ایمیل به صورت مرتب دریافت و مشاهده کند.

### Integration Services : ۵-۳-۳

این سرویسها یک پلت فرم هستند که راه حل هایی برای ایجاد یکپارچگی اطلاعات با سرعت بالا ارائه می کنند و شامل بسته های نرم افزاری پردازش extract, transform, and load برای data warehousing است.

### SQL Server Agent : ۶-۳-۳

به بیان ساده، این سرویس مسئول دفتر SQL SERVER است. مثلاً در صورت تعریف توسط ادمین، هر شب ساعت ۱۰ از اطلاعات کپی پشتیبان می گیرد.

### ۳-۳-۷: Authentication Mode

چنانچه گزینه‌ی Windows Authentication Mode انتخاب گردد، دسترسی کاربران و کلمه عبور آنان به SQL Server بر اساس وجود و درستی آن‌ها در ویندوز انجام می‌گیرد. ولی اگر Mixed Mode انتخاب شود باید برای دسترسی به SQL Server مجدداً نام کاربری و سطح دسترسی تعریف کرد.

### ۳-۳-۸: Collation

شمایی در SQL Server که نحوه Sort حروف الفبای یک زبان خاص و همچنین نحوه Compare کردن حروف الفبای آن زبان را در بر می‌گیرد.

تنظیمات collation به صورت پیش فرض از تنظیمات سیستم عامل گرفته می‌شود. و در صورتی که سیستم عامل شما در بخش Regional settings دارای تنظیمات لازم فارسی باشد، نیاز به تغییر در این بخش ندارید.

### ۳-۳-۹: وصل شدن به SQL Server و ساخت Database جدید

قبل از ادامه مبحث، به یه نکته کوچیک که یادم رفت در پست قبلیم بگم، اشاره کنم. تلفظ صحیح SQL، بصورت سیکوئل است. این تلفظ چیزی است که من در فیلم‌های آموزشی به لهجه آمریکایی شنیدم. حالا مطلب قبلی رو ادامه میدم.

پس از نصب SQL Server 2005 حالا می‌خواهیم به سرور وصل بشیم و تغییراتی را اعمال کنیم. برای این کار از نرم‌افزاری به نام SQL Server Management Studio که در صورت انتخاب، همراه با سرویس SQL Server 2005 نصب می‌شود استفاده می‌کنیم. همونطور که قبلاً هم گفتم، وجود این برنامه در کامپیوتر دلیلی بر وجود سرویس SQL Server روی آن کامپیوتر نیست. چون می‌توان این برنامه را بدون نصب سرویس SQL Server 2005 نصب نمود و همچنین می‌توان سرویس SQL Server 2005 را بدون نصب این برنامه، نصب نمود.

دسترسی به این برنامه، از مسیر زیر امکان‌پذیر است:

Start>All Programs>Microsoft SQL Server 2005>SQL Server Management Studio

و یا اینکه در Command Prompt تایپ کنید sqlwb

که پس از اجرا، پنجره‌ای شبیه عکس زیر را خواهید دید.



شکل ۳-۳: وصل شدن به Sql Server

برای وصل شدن به یک سرور باید نام آن سرور را بدانید. نام سرور می‌تواند به اشکال مختلف به شرح زیر

باشد:

OrionComputer: که در حقیقت نام کامپیوتر می‌باشد.

OrionComputer\InstanceName: همانطور که قبلاً گفتم، می‌توان در یک کامپیوتر،

چند Instance از SQL Server داشت

InstanceName: نام آن Instance در کامپیوتر OrionComputer می‌باشد.

172.150.12.54: که همان IP سرور می‌باشد.

172.150.12.54\InstanceName: دقیقاً همانند مورد دوم است. فقط در اینجا به جای نام

کامپیوتر، IP آن را ذکر می‌کنیم.

(Local): برای دسترسی به کامپیوتر Local به کار می‌رود که البته می‌توان به جای (Local) از نقطه نیز استفاده کرد (مانند شکل فوق Local)\InstanceName): ( دقیقاً مانند مواردی که در بالا ذکر شد.

AAA: همیشه برای هر سرویس SQL Server به اصطلاح یک Alias تعریف کرد. و من در این مثال فرض کردم Alias سرور من، AAA هست!

پس از وارد کردن نام سرور، باید تکلیف قسمت Authentication را روشن کرد. اگر در هنگام نصب SQL Server نحوه Authentication Mode روی Windows Authentication گذاشته باشید و به خودتون هم دسترسی داده باشید، در اینجا هم Windows Authentication رو وارد کنید و چون شناسایی شما توسط ویندوز انجام میشه، در اینجا لازم به وارد کردن یوزرنیم و پسورد نیست. اما اگر حالت Mixed Mode رو انتخاب کرده باشید، باید نام کاربری و کلمه عبور را وارد کنید. توجه داشته باشید که Authentication Mode رو بعد از نصب سرویس SQL Server هم میشه بصورت دستی تغییر داد.

حالا اگر همه چیز درست باشد، شما با کلیک بر روی دکمه Connect به سرور وصل میشید و بسته به نوع دسترسی، می‌توانید اطلاعات را مشاهده کرده و یا تغییراتی اعمال کنید.

پس از ورود به سرور از قسمت Object Browser می‌توانید به دیتابیس‌های موجود دسترسی پیدا کنید. (شکل زیر). البته موارد دیگری مانند Security, Server Objects, Replication و ... وجود دارند که ما در اینجا به آن‌ها اشاره نمی‌کنیم.



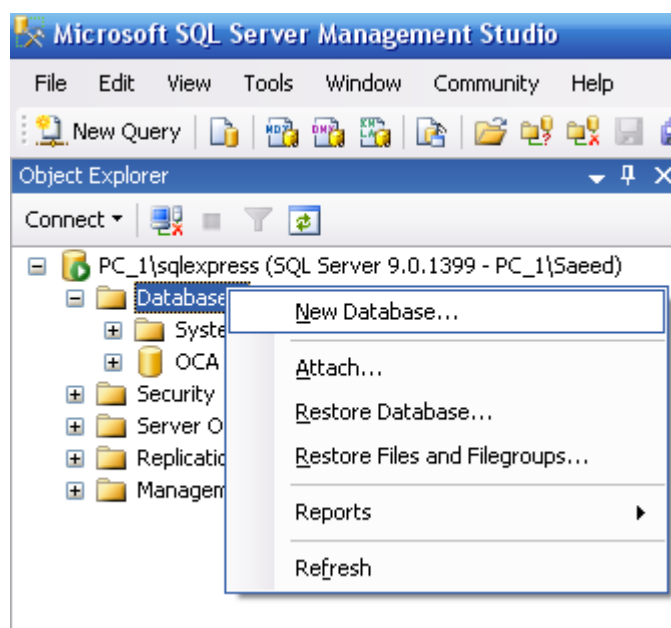
شکل ۳-۴: پایگاه داده



در شکل بالا، چهار Database وجود

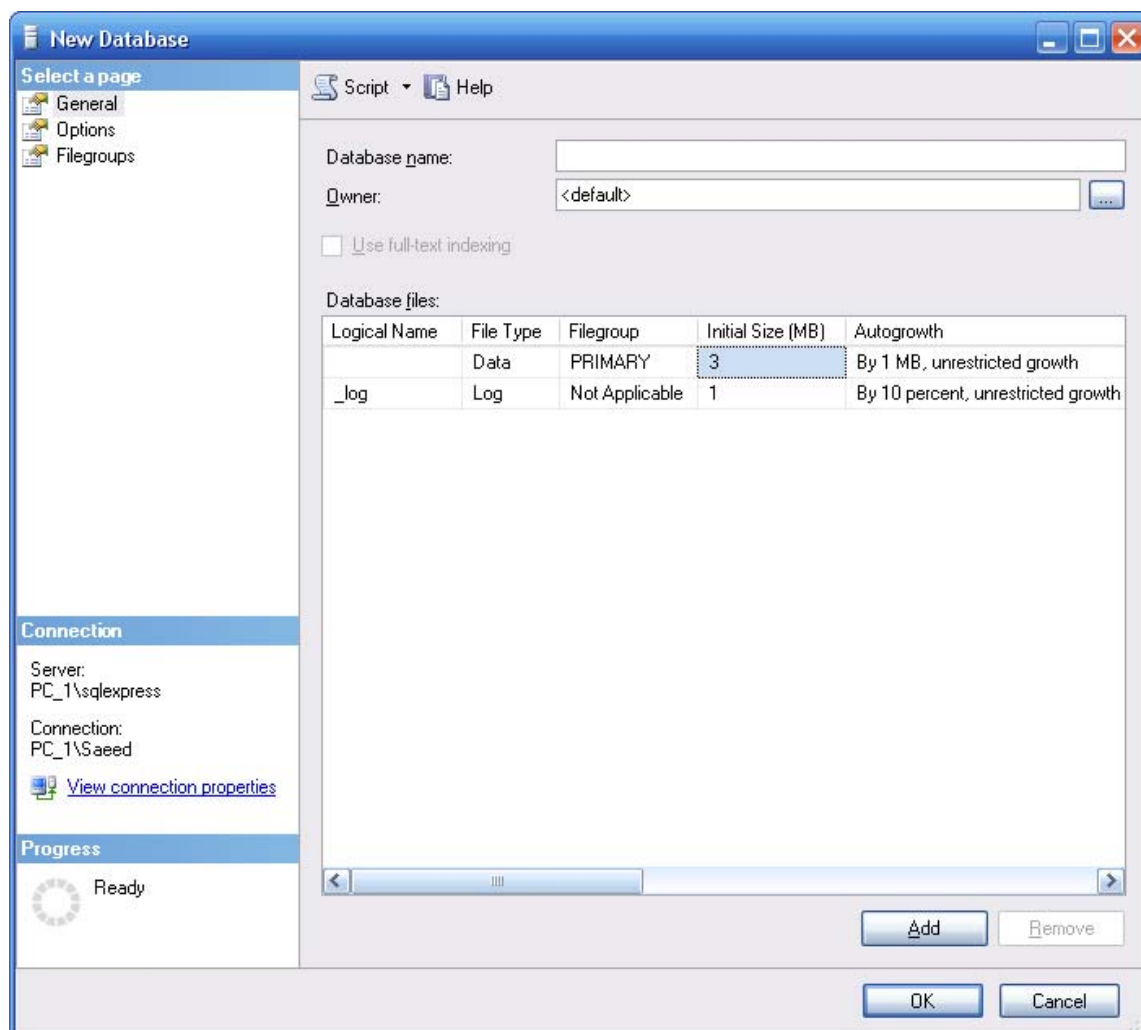
دارد AdventureWorks و AdventureWorksDW و Northwind و Pubs. که دو مورد اول از نمونه دیتابیس‌های ورژن ۲۰۰۵ هستند و دو مورد دوم یعنی Northwind و Pubs از نمونه دیتابیس‌های ورژن ۲۰۰۰ هستند. ما در آینده و برای یادگیری، احتیاج زیادی به دیتابیس Pubs داریم.

برای ساخت یک Database جدید، روی عبارت Databases کلیک راست کنید و بر روی New Database... کلیک کنید. (شکل زیر)



شکل ۳-۵: پایگاه داده جدید

سپس صفحه‌ای مانند شکل زیر باز می‌شود که باید خصوصیات دیتابیس را در آن ذکر کنید. برای ساخت یک دیتابیس ساده تنها نیاز به ذکر نام آن دیتابیس است.



شکل ۳-۶: پنجره ایجاد پایگاه داده جدید

سپس باید نوع فهرست‌گذاری برای کلمات متن تعیین گردد. اگر عنوان Use Full Text Indexing فعال گردد، برای تمام کلمات ایندکس ساخته خواهد شد و در زمان جستجوی کلمات، به سرعت متنی که دارای آن کلمه باشد، پیدا شده و ارائه می‌گردد. بدیهی است باید در این حالت برخی دستورات، متناسب با این حالت تغییر یابد. مثلاً اگر در حالت عادی دستور جستجوی کلمه "Test" بدین شرح باشد:

```
SELECT * FROM authors WHERE address LIKE '%Test%'
```

در حالت فعال بودن Full Text Indexing، باید دستور فوق بصورت زیر نوشته شود:

```
SELECT * FROM authors WHERE CONTAINS(address, '*Test*')
```

هر دیتابیس حداقل دارای دو نوع فایل به نام‌های زیر می‌باشد:

Master Data File - یا همان فایل mdf که در حقیقت فایل اطلاعات اصلی دیتابیس است.

Log Data File - یا همان فایل ldf که در حقیقت فایل History دستورات است.

در زمان تعریف هر یک از فایل‌ها می‌توان از طریق Initial Size مقدار اولیه حجم فایل را تخمین زد. معمولاً می‌گویند سائز اولیه Logfile برابر ۳۵ درصد فایل mdf باشد که البته با توجه به نوع دیتابیس میتواند متفاوت باشد.

در قسمت AutoGrowth می‌توان مقدار افزایش حجم فایل را پس از رسیدن به حداکثر حجم تعریف شده، تعیین کرد. ضمناً در همین قسمت می‌توان حداکثر حجم مجاز فایل را تعیین کرده و یا اینکه حداکثر حجم مجاز را نامحدود تعریف نمود.

پس از آن می‌توان محل ذخیره کردن فایل را در قسمت Path تعیین کرد.

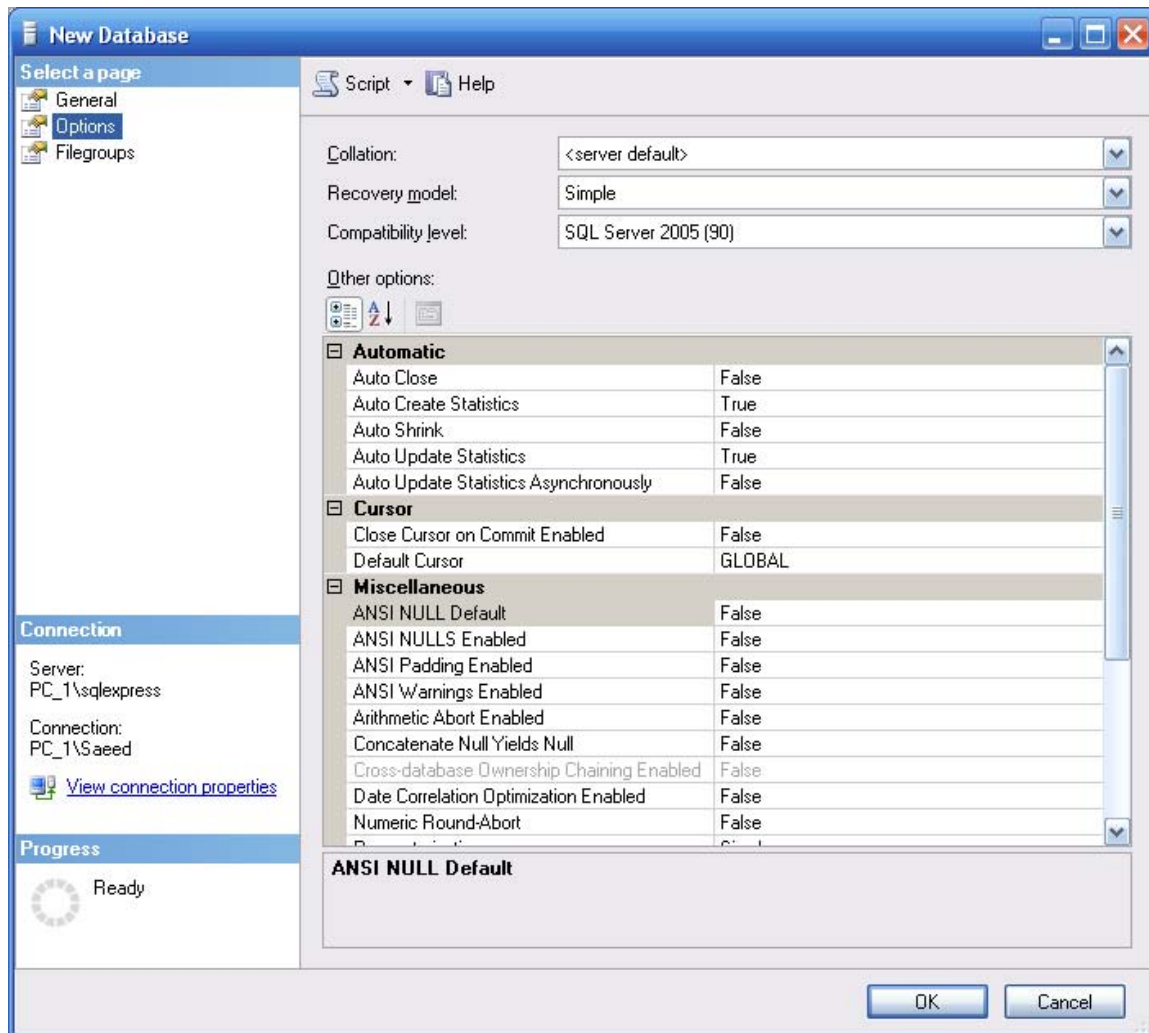
برای انجام هر کاری در SQL Server یک دستور به نام T-SQL صادر و اجرا می‌شود. مثلاً در زمان ایجاد دیتابیس جدید که از طریق منوی New Database انجام می‌شود، یک سری دستور TSQL اتوماتیک ایجاد شده و برای سرور ارسال می‌گردد. برای دیدن و یا احتمالاً اصلاح این دستورات، میتوانید روی دکمه Script که در بالای اکثر پنجره‌های SQL Server 2005 وجود دارد کلیک کنید.

در SQL Server در هر لحظه فقط قادر به اجرای یک دستور است. پس از اجرای دستور، History آن دستور در LogFile ذخیره می‌شود. این فایل دارای کاربردهای زیادی است که مختصراً به چند مورد اشاره می‌شود: فرض کنید در پایان هر روز از اطلاعات Backup گیری می‌شود. ولی در وسط روز هارد دیسک با مشکل مواجه می‌شود. اگر فایل‌های Master و Log روی دو هارد دیسک جداگانه نگهداری شده باشند، میتوان ابتدا فایل اصلی را از Backup شب گذشته بازخوانی کرد و سپس با کمک دستورات موجود در LogFile از زمان Backup تا کنون، نسبت به بازسازی فایل اصلی اقدام کرد.

یکی دیگر از کاربردهای LogFile در Replication است. مثلاً فرض کنید بنا به دلیلی لازم است دو نسخه از فایل اصلی بر روی دو سرور جداگانه در شهرهای مختلف نگهداری شود. به روز رسانی همزمان و Online فایل‌های اصلی دو سرور نیاز به پهنای باند وسیعی دارد. در حالی که میتوان در پایان روز، تنها دستورات اجرا شده روزانه، از روی LogFile برای سرور مقابل ارسال شده و در آنجا دستورات اجرا شده و فایل اصلی بروزرسانی شود. از این روش می‌توان برای ساخت فایل Mirror در دو مکان مختلف استفاده کرد.

بنابراین توصیه میشه محل نگهداری فایل‌های mdf و ldf جداگانه و بر روی دو هارد دیسک جدا باشد، مگر اینکه از تکنیک Raid برای نگهداری اطلاعات استفاده شود که توضیح آنها در سواد بنده نمی‌گنجد!

در قسمت صفحه New Database منویی به نام Option وجود دارد (شکل زیر) که بعضی از آنها را به اختصار توضیح میدهم.



شکل ۳-۷: پنجره خصوصیات ایجاد پایگاه جدید

### ۳-۳-۱۰: Recovery Model

اگر حالت Simple انتخاب شود فایل Log بصورت مینیموم ثبت خواهد شد و اکثر دستورات Log نخواهند شد. در این حالت سرعت عملیات افزایش یافته ولی در صورت خرابی اطلاعات، امکان بازیابی وجود ندارد. اگر-Bulk Logged انتخاب گردد، در این حالت از تمامی دستورات Log گرفته خواهد شد. اما در دستورات Bulk بصورت مینیموم Log گرفته می‌شود. در صورت انتخاب Full، از تمام دستورات حتی Bulk ها بصورت کامل Log می‌گیرد.

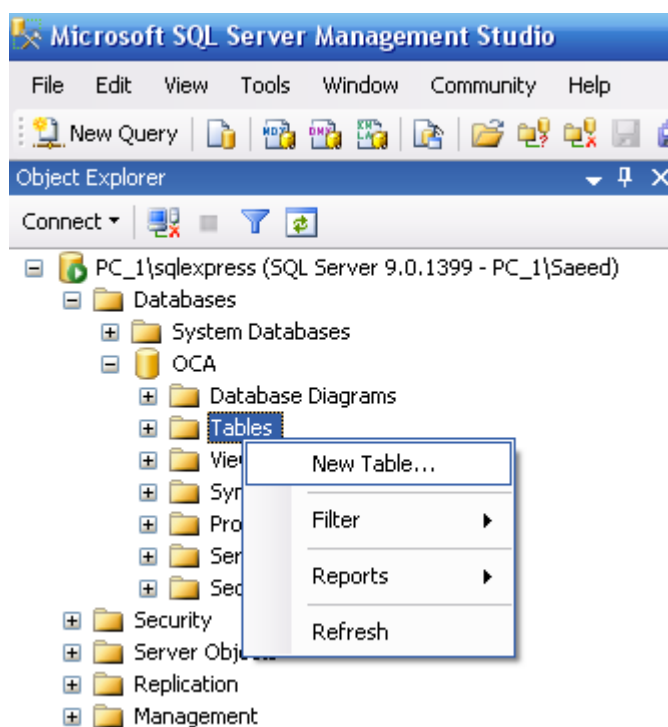
### ۱۱-۳-۳: Compatibility Level

خود SQL Server 2005 بصورت اتوماتیک فایل‌های SQL Server 2000 را به 2005 تبدیل می‌کند. ولی تمام امکانات SQL Server 2005 بر روی فایل‌های تبدیل شده قابل استفاده نمی‌باشد. مگر اینکه در این قسمت، حالت SQL Server 2005(90) انتخاب گردد.

بقیه‌ی موارد را می‌توانید با زدن دکمه HELP در MSDN مطالعه کنید.

### ۱۲-۳-۳: Table ها و DataType ها :

برای ساخت یک جدول باید ابتدا Database مربوطه انتخاب شده و سپس بر روی قسمت Table کلیک راست کرده و گزینه New Table انتخاب شود. (شکل زیر)



شکل ۳-۸: ایجاد پایگاه جدید

سپس در سمت راست صفحه پنجره ساخت Table ظاهر می‌شود. (شکل زیر) که من در آن سه فیلد با Datatype های مختلف ساختم.

برای ساخت جدول باید ابتدا فیلدها یا همان ستون‌های جدول تعریف شود. فیلدهای دارای **DataType** های مختلفی هستند که به اختصار آنها رو توضیح میدم:

## Binary Data

این نوع فیلدها برای نگهداری اطلاعات بصورت بایناری مانند تصاویر مناسب هستند و شامل چهار نوع به شرح زیر می‌باشد:

### Binary

این نوع فیلدها، از ۱ تا ۸۰۰۰ بایت را در خود جای می‌دهند.

### Var Binary

این نوع فیلدها هم از ۱ تا ۸۰۰۰ بایت را در خود جای می‌دهند. (متغیر)

## Image

این نوع فیلدها از ۱ تا حداکثر ۲ گیگابایت را می‌توانند ذخیره کنند. فرق این نوع داده‌ها با دو نوع قبلی این است که در دو نوع قبلی، اطلاعات در خود رکورد ثبت می‌شوند ولی در این نوع داده‌ها، اطلاعات در یک **Page** ذخیره می‌شود و به جایش در رکورد، یک پوینتر ۱۶ بیتی ذخیره می‌شود. این نوع فیلدها در **SQL 10** حذف شده و به جایش باید از **VarBinary** استفاده کرد.

### VarBinary(Max)

این نوع **Datatype** در **SQL 2005** معرفی شده و تقریباً همانند داده‌های **Image** هستند.

## Character Data

این نوع فیلد برای نگهداری عبارات و یا حروف **ASCII** می‌باشد. در این نوع فیلدها، برای نگهداری هر حرف، یک بایت اشغال می‌شود و لذا نیاز به **Collation** برای تعیین زبان اطلاعات می‌باشد. این نوع فیلدها هم چهار نوع هستند:

### Char

این نوع فیلدها، اطلاعات متنی با طول ثابت از ۱ تا حداکثر ۸۰۰۰ حرف را در خود ذخیره می‌کنند.

### VarChar

این نوع فیلدها اطلاعات متنی با طول متغیر از ۱ تا حداکثر ۸۰۰۰ حرف را در خود ذخیره می‌کنند. فرق بین **Char** و **VarChar** در این است که در **Char**، طول رشته ثابت است. یعنی اگر یک فیلد را از

نوع (20)Char معرفی کرده و در آن کلمه Orion را قرار دهیم، عین ۲۰ حرف استفاده خواهد شد. یعنی ۵ کاراکتر اول را کلمه مربوطه اشغال کرده و ۱۵ کاراکتر باقی مانده، Blank خواهند بود. اما در VarChar اینگونه نیست.

## Text

در این نوع داده‌ها، اطلاعات از ۱ تا ۲ مگا حرف ذخیره می‌شود. این نوع داده همانند Image و VarBinary(MAX) در خود رکورد ذخیره نمی‌شوند. بلکه توسط یک پوینتر به جای دیگری اشاره می‌کنند. این نوع داده در SQL 10 حذف شده و بجای آن‌ها از VarChar(MAX) استفاده می‌شود.

## VarChar(Max)

در این نوع داده‌ها، اطلاعات از ۱ تا ۲ مگا حرف ذخیره می‌شود و مکانیزم آن هم بصورت پوینتری می‌باشد.

## Unicode Data

این نوع فیلدها برای نگهداری متون Unicode بوده و برای نگهداری هر حرف، از دو بایت استفاده می‌شود. پس مسلماً نسبت به نوع داده‌های کاراکتری، حافظه بیشتری را به خود اختصاص می‌دهد و در ضمن کمی هم کندتر است. این نوع فیلدها، احتیاج به Collation ندارند.

## nChar

در این نوع داده، اطلاعات از ۱ تا حداکثر ۴۰۰۰ حرف با طول ثابت ذخیره می‌شود.

## nVarChar

در این نوع داده، اطلاعات از ۱ تا حداکثر ۴۰۰۰ بایت با طول متغیر ذخیره می‌شود.

## nText

در این نوع داده، از ۱ تا ۱ مگا حرف ذخیره می‌شود. مکانیزم آن هم بصورت Pointer می‌باشد.

## nVarChar(MAX)

در این نوع داده از ۱ تا ۱ مگا حرف ذخیره می‌شود. مکانیزم آن هم بصورت Pointer می‌باشد.

## Integer Data

این نوع فیلد برای نگهداری اعداد صحیح و بدون اعشار استفاده می‌گردد و دارای ۴ نوع به شرح زیر است. در ضمن این نوع فیلدها رتبه یک سرعت در نوع فیلدهای عددی را دارد.

## TinyInt

یک بایت را اشغال می‌کند و می‌تواند از ۰ تا ۲۵۵ را در خود ذخیره کند.

## SmallInt

یک عدد دو بایتی است و می‌تواند از ۳۲۷۶۷ منفی تا ۳۲۷۶۷ مثبت را در خود ذخیره کند.

## Int

یک عدد چهار بایتی است که می‌تواند اعداد بین مثبت و منفی ۲ میلیارد را در خود ذخیره کند.

## BigInt

یک عدد ۸ بایتی است که می‌تواند اعداد بین مثبت و منفی ۴ میلیارد را در خود ذخیره کند.

## Money Data

این نوع فیلد برای نگهداری اعداد غیر صحیح با تعداد ارقام اعشار ثابت ۴ رقمی استفاده می‌شود. این نوع فیلدها، دارای رتبه دوم سرعت در بین انواع داده‌ی عددی است.

## SmallMoney

یک عدد ۴ بایتی است که می‌تواند ۶ رقم صحیح و ۴ رقم اعشار را در خود ذخیره کند.

## Money

یک عدد ۸ بایتی است که می‌تواند ۱۵ رقم صحیح و ۴ رقم اعشار را در خود ذخیره کند.

## Approximate Data

این نوع فیلدها برای نگهداری اعداد غیر صحیح با تعداد ارقام اعشار متغیر و یا تخمینی استفاده می‌شود. این نوع داده‌ها رتبه سوم سرعت در بین انواع داده‌های عددی دارند و استفاده از آنها به دلیل کندی، توصیه نمی‌گردد. مگر در مواقع لزوم.

## Real

یک عدد ۴ بایتی است که اعداد بصورت توانی از ۱۰ نگهداری می‌شوند. (تعداد اعشار مشخص نیست)

## Float

یک عدد ۸ بایتی است که اعداد بصورت توانی از ۱۰ نگهداری می‌شوند. (تعداد اعشار مشخص نیست)



## Decimal

این نوع فیلد برای نگهداری اعداد اعشاری با تعداد اعشار مشخص استفاده می‌گردد. این نوع فیلدها بسیار کند بوده و استفاده از آنها توصیه نمی‌گردد. فرمت اینگونه داده‌ها به شکل زیر است:

### Decimal (Precision,Scale)

که در آن Precision به معنای تعداد کل رقم‌های عدد و Scale تعداد ارقام اعشار را مشخص می‌کند. مثلاً اگر فیلد بصورت Deciaml(6,2) تعریف شود، حداکثر آن برابر ۹۹۹۹.۹۹ می‌باشد.

## Date/Time

این نوع فیلدها برای نگهداری تاریخ میلادی و ساعت استفاده می‌شود و برای تاریخ شمسی کاربرد ندارد.

### SmallDateTime

این نوع فیلد، ۴ بایتی است و از سال ۱۹۰۰ تا ۲۰۷۹ را با دقت هزارم ثانیه ذخیره می‌کند.

### DateTime

این نوع فیلد، ۸ بایتی است و از سال ۱۷۰۰ تا ۹۹۹۹ را با دقت هزارم ثانیه ذخیره می‌کند.

## Other DataTypes

برخی از انواع فیلدهای خاص را در این قسمت به اختصار توضیح میدم:

### Bit

یک فیلد دو بیتی است و می‌تواند ۰ و ۱ و Null را ذخیره کند. کاربرد آن در زمان‌هایی است که دو حالت وجود داشته باشد. مانند جنسیت زن و مرد.

### TimeStamp

در این فیلد ۸ بایتی، تایم لحظه‌ای اجرای دستور نگهداری می‌شود و کاربرد آن کنترل بروزرسانی همزمان (Concurrency) اطلاعات توسط چند کاربر است. البته در تعداد رکوردهای پایین به کار نمی‌آید و بیشتر زمانی مورد نیاز است که تعداد رکوردها خیلی زیاد باشد مثلاً ۱۰۰ میلیون رکورد!

## UniqueIdentifier

این فیلد ۱۶ بایتی، به ما کدی Unique یا تک می‌دهد که به اصطلاح GUID می‌گویند. یکی از کاربردهای آن در Replication است.

### SQL\_Variant

این نوع فیلد برای نگهداری انواع داده استفاده می‌شود و نوع آن با توجه به اولین مقداری که در آن قرار می‌گیرد تعیین خواهد شد. چون نوع و حجم فیلد مشخص نیست، لذا تنها یک اشاره‌گر ۱۶ بایتی در آن قرار گرفته و داده اصلی در فایل جداگانه نگهداری می‌شود. استفاده از این نوع فیلد، توصیه نمی‌گردد.

### Cursor

این فیلد مربوط به کنترل Cursor است که در آینده توضیح میدم.

### Table

این نوع فیلد را هم در زمان آموزش Function ها خواهید آموخت

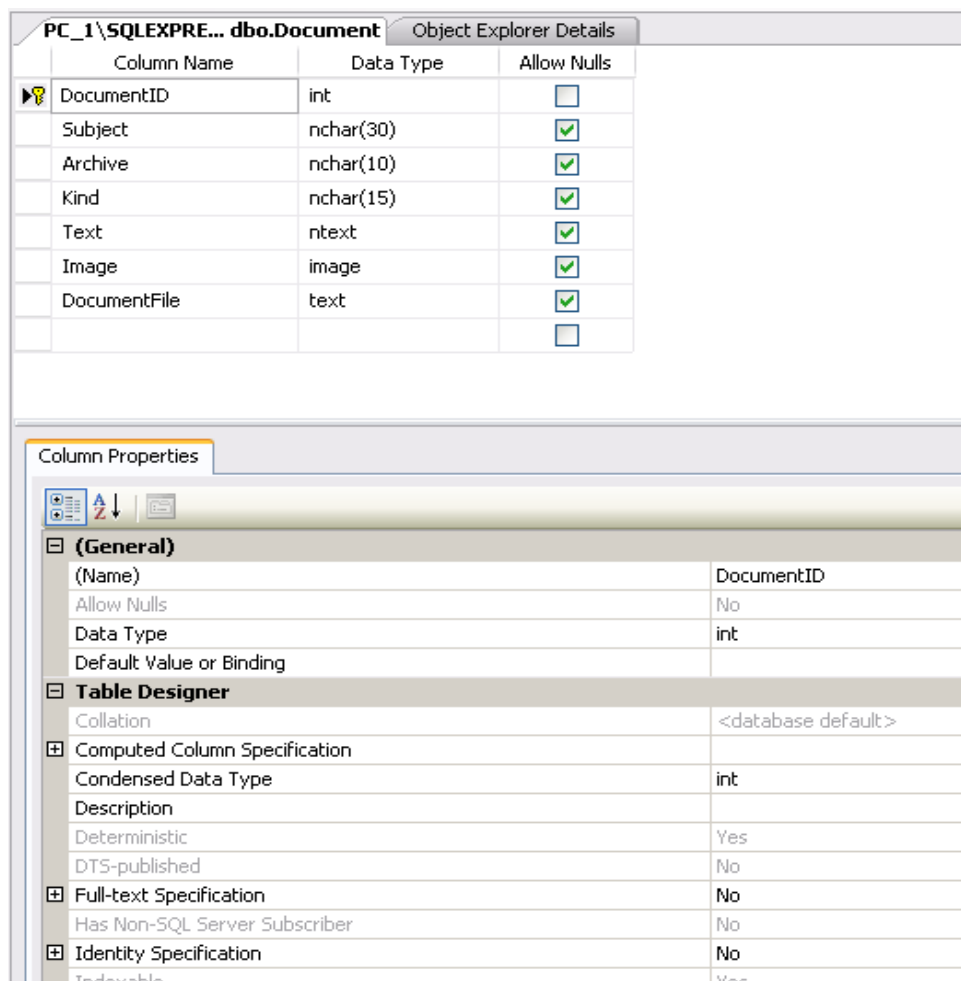
### XML

این فیلد بیشتر برای انتقال اطلاعات و دستورات تحت web استفاده می‌شود و شامل انواع MetaData های مختلف است. این فیلد در SQL 2005 معرفی گردید.

## ۳-۳-۱۳: سایر مشخصات Table

### ۳-۳-۱۳-۱: تعریف سایر مشخصات جدول

در هنگام تعریف فیلدهای (ستون‌های) یک جدول، صفحه‌ای با عنوان Column Properties در قسمت پایین صفحه ظاهر می‌شود (شکل زیر) که از طریق این فرم، میشه مشخصات بیشتری از اون فیلد خاص رو تعریف کرد که من در اینجا پر کاربردترین‌ها رو توضیح میدم.



شکل ۳-۹: ستون های جدول

گزینه Allow Nulls که البته در کنار Data Type هم می توان آنرا تعیین کرد، مشخص می کند که آیا آن فیلد خاص می تواند مقدار Null یا همان هیچی! را بپذیرد یا خیر.

یکی دیگر از این موارد، مشخص کردن مقدار اولیه ی فیلد مورد نظر هستش. برای این کار در مقابل عبارت Default Value or Binding مقدار اولیه رو وارد می کنید. مشخص کردن مقدار اولیه به این درد می خورد که مقدار فیلد ما مثلاً در هنگام Insert یک رکورد جدید به جدول، در صورت مشخص نبودن مقدار، Null نشود.

اگر با بانک اطلاعاتی Access آشنا باشید، یکی از انواع داده ها، Auto Number بود که در Data Type های SQL Server وجود نداشت. برای پیاده سازی AutoNumber به این صورت عمل می کنیم: اگر روی علامت مثبت کنار Identity Specification کلیک کنیم، سه گزینه ی دیگر نمایش داده خواهند شد.

اگر برای فیلدهای عددی غیر اعشاری، Is Identity برابر Yes شود، در اینصورت این فیلد بصورت اتوماتیک در هنگام درج رکوردهای جدید، اضافه خواهد شد. مقدار شروع در قسمت Identity Seed و میزان افزایش در هر دفعه، در قسمت Identity Increment قرار خواهد گرفت. ضمناً برای استفاده از این امکان، باید در قسمت Default Value مقدار پیش فرض تعیین نگردد و حتماً Allow Null هم نباشد.

چنانچه فیلد از نوع Unicode مثل nchar استفاده می شود، باید کد استاندارد زبان آن مشخص شود. برای این کار کلید انتخاب Collation را زده و از دو حالت SQL Collation و Windows Collation بهتر است Windows Collation را انتخاب کرده و بعد زبان Arabic و حالت Dictionary Sort انتخاب شود

اگر ما ۲ فیلد a و b از نوع int تعریف کنیم و فیلد سومى به نام C را هم تعریف کنیم و سپس بر روی دکمه مثبت Computed Column Specification کلیک کنیم، میتوانیم در قسمت Formula عبارت  $a+b$  را وارد کنیم (شکل زیر). با این کار مقدار C همیشه برابر با  $a+b$  خواهد بود. اگر گزینه Is Persisted برابر No باشد، اصلاً فضایی برای فیلد C در جدول در نظر گرفته نخواهد شد و همواره این مقدار محاسبه می گردد. اما اگر Is Persisted برای Yes باشد، مقدار C در جدول ذخیره می شود.

	Column Name	Data Type	Allow Nulls
	a	int	<input checked="" type="checkbox"/>
	b	int	<input checked="" type="checkbox"/>
▶	c		<input type="checkbox"/>
			<input type="checkbox"/>

☐	Computed Column Specification	a+b
	(Formula)	a+b
	Is Persisted	No

شکل ۳-۱۰ : Is Persisted

حالا برای ذخیره جدول، کافیه فرم فعلی رو ببندید. سپس از تون سؤال میکنه که آیا میخواهید جدول ذخیره بشه؟ و اگر شما Yes رو بزنید، از شما اسم جدول پرسیده می شود. اسم جدول را وارد کرده و Ok را بزنید.

قبلاً هم به این مورد اشاره کردم که SQL برای انجام هر کاری یک دستور T-SQL را تولید و اجرا می کند. مثلاً برای ساخت یک جدول (یا هر چیز دیگر) می توان مستقیماً از دستورات SQL استفاده کرد. برای اینکار در محیط Management Studio روی دکمه New Query کلیک کرده (سمت چپ، بالا) و دستورات را وارد کنید و سپس Execute را کلیک کنید.

اما حالا که شما جدول رو بصورت Visual ساختید، باز هم می تونید ببینید که خود SQLServer از چه

دستوراتی برای ساخت جدول استفاده کرده. برای این کار پس از اینکه جدول رو ذخیره کردید، در قسمت Object Browser روی آن کلیک راست کرده و مسیر زیر را انتخاب کنید:

Script Table as > CREATE to > New Query Editor Window

پس از آن، کدی که خود SQL Server برای ساخت جدول مورد نظر، تولید کرده را می‌توانید دیده و یا ذخیره کنید.

### ۲-۳-۳: محدود کننده‌های فیلد (Constraint)

باید برای بعضی از فیلدها، بر اساس نیاز، تعاریف و محدود کننده‌های خاصی در نظر گرفته شود که در این قسمت، به شرح آنها می‌پردازم.

#### Unique Constraint

بعضی از فیلدها، کلید نیستند (کلیدها را کمی پایین‌تر توضیح میدهم)، اما باید غیر تکراری باشند. مثلاً فرض کنید آدرس ایمیل، جزو کلید نیست اما باید Unique یا واحد باشد. برای این کار از Unique Index ها استفاده می‌کنیم که در درس‌های بعدی به اونها می‌پردازم.

#### Default Constraint

وقتی در هنگام ساخت جدول و تعیین نوع فیلدها، مقدار Default Value به یک فیلد بدهیم، در حقیقت داریم نوعی محدودیت اعمال می‌کنیم. محدودیت ما این است که اگر کاربر در هنگام درج یا به روزرسانی مقدار آن فیلد را وارد نکرد، مقدار Default Value به جای آن قرار بگیرد.

#### Null ability Constraint

اگر در هنگام ساخت جدول و تعیین نوع فیلدها، مقدار Allow Null آن را غیر فعال کنیم، در حقیقت به SQL Server می‌گوییم که این مقدار نمی‌تواند مقدار Null بگیرد. پس در هنگام درج یا به روزرسانی، اگر کاربر مقدار فلان فیلد خاص را وارد نکرد، جلوی انجام عملیات گرفته می‌شود.

#### Primary Key Constraint

تعریف فیلد کلید اصلی، یکی از مهمترین محدودیت‌ها می‌باشد. فیلد کلید، مقدار تکراری قبول نمی‌کند و بیشتر برای تفکیک و جستجوی رکوردها مورد استفاده قرار می‌گیرد. برای تعریف یکی از فیلدها به عنوان کلید اصلی، باید در زمان ایجاد جدول، بر روی نام فیلد کلیک راست کرده و گزینه Set Primary Key زده شود. بعد از این کار، علامت کلید در کنار فیلد مورد نظر ظاهر می‌شود. ضمناً این

امکان وجود دارد که همزمان دو یا چند فیلد بعنوان کلید اصلی تعریف شوند. برای اینکار دقیقاً مثل وقتی که می‌خواهید در ویندوز دو یا چند فایل یا فولدر را انتخاب کنید، کلید Ctrl را نگه داشته و روی فیلدهای مورد نظرتون کلیک کنید و سپس روی یکی از آن‌ها کلیک راست کرده و گزینه Set Primary Key را انتخاب کنید.

ضمناً توجه داشته باشید که فیلد کلید اصلی هیچ وقت Null نمی‌پذیرد. پس ستون Allow Nulls را برای این فیلد علامت نزنید.

## Foreign Key Constraint

برای ارتباط بین جداول استفاده می‌شود و در اصل Relational Integrity را فراهم می‌کند. به عبارت ساده‌تر، کلید خارجی همان مقدار کلید اصلی از جدول دیگر است. توجه داشته باشید که مقدار یک کلید خارجی میتواند تکراری باشد. به عنوان مثال دو جدول Titles و Publishers که یکی مشخصات کتاب‌ها و دیگری مشخصات ناشران را ذخیره می‌کند را فرض کنید. در جدول Titles (کتاب‌ها) یک فیلد به نام pub\_id وجود دارد که در اصل کلید خارجی برای فیلد pub\_id موجود در جدول Publishers می‌باشد. وجود این کلید به این خاطر است که ما بتوانیم ناشر یک کتاب را پیدا کنیم.

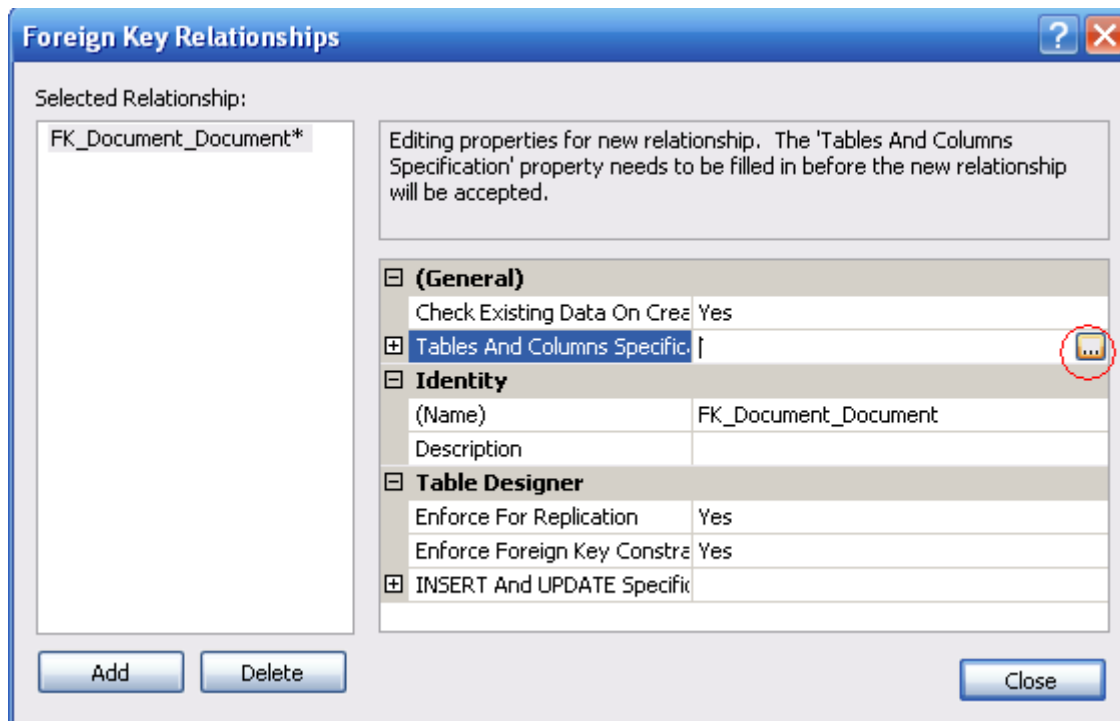


شکل ۳-۱۱: کلید خارجی

فیلدهای کلید خارجی، سه نوع محدودیت را ایجاد می‌کنند:

- اجازه حذف رکوردی از جدول پدر نداریم که فرزندی داشته باشد.
- اجازه insert رکورد در جدول فرزند با کد پدری که در جدول پدر موجود نیست را نداریم.
- مقدار کلید پدری را که دارای فرزند هست را نمی‌توانیم تغییر دهیم

برای ساختن کلید خارجی، پس از ساخت Table ها، بر روی Table فرزند و در قسمت Keys کلیک راست کرده و گزینه New Foreign Key انتخاب گردد. سپس فرمی همانند زیر ظاهر می شود:



شکل ۳-۱۲: فرم ساخت کلید خارجی جدید

حالا قبل از اینکه مشخص کنیم، کلید خارجی ما از کدام فیلد و از کدام جدول به وجود می آید، به بعضی از مواردی که در این فرم هست، می پردازم .

اگر **Enforce For Replication** فعال باشد، تمام محدودیت هایی که بالاتر به آن ها اشاره کردم، در زمان **Replication** هم رعایت خواهد شد .

چنانچه گزینه **Enforce Foreign Key Constraint** فعال باشد، محدودیت جلوگیری از اضافه شدن رکورد به جدول فرزند در صورتی که مقدار آن در جدول پدر نباشد، رعایت می گردد. در صورت غیر فعال بودن هم، این محدودیت به حساب نخواهد آمد.

در قسمت **INSERT and UPDATE Specification** میتوان قوانین و محدودیت های حذف و اضافه و بروزرسانی را تغییر داد. به عنوان مثال، **Delete Rules** دارای چهار حالت زیر است:

- 1- اگر **No Action** باشد، اجازه حذف رکورد پدر را در صورتی که فرزندی از آن پدر موجود باشد، نمی دهد.
- 2- اگر **Cascade** باشد، در این حالت، می توان رکورد پدری را حذف نمود. اما بصورت اتوماتیک تمام فرزندهای آن هم حذف خواهند شد.
- 3- اگر **Set Null** باشد، باز هم اجازه حذف رکورد پدر داده می شود، اما فیلد مرتبط در جدول فرزند، در صورتی که دارای شرط **Allow Null** باشد، **Null** خواهد شد.

۱۴- اگر Set Default باشد، باز هم اجازه حذف رکورد پدر داده می‌شود. اما فیلد مرتبط در جدول فرزند، در صورتی که مقدار Default داشته باشد، برابر مقدار پیش فرض خواهد شد.

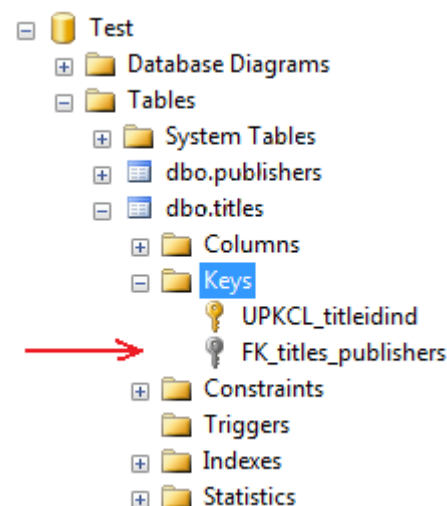
حالا باید مشخص کنیم که پدر از کدام جدول و از کدام فیلد خواهد آمد. برای این کار بر روی دکمه کنار گزینه Tables and Columns Specification کلیک کنید. با این کار فرمی همانند شکل زیر ظاهر می‌شود:

Relationship name:	
FK_Document_Document	
Primary key table:	Foreign key table:
Document	Document
DocumentID	DocumentID

شکل ۳-۱۳: انتخاب رابطه ها

بعد از اینکه اطلاعات را تکمیل گردید، فرم‌ها را ببینید تا از شما در مورد ذخیره کردن تغییرات سؤال پرسد. سپس روی گزینه Keys در قسمت Object Browser کلیک راست کرده و گزینه Refresh را کلیک کنید تا آخرین تغییرات را برای شما نمایش دهد. اگر کارها درست پیش رفته باشد باید کلید خارجی بصورت زیر ساخته شده باشد:



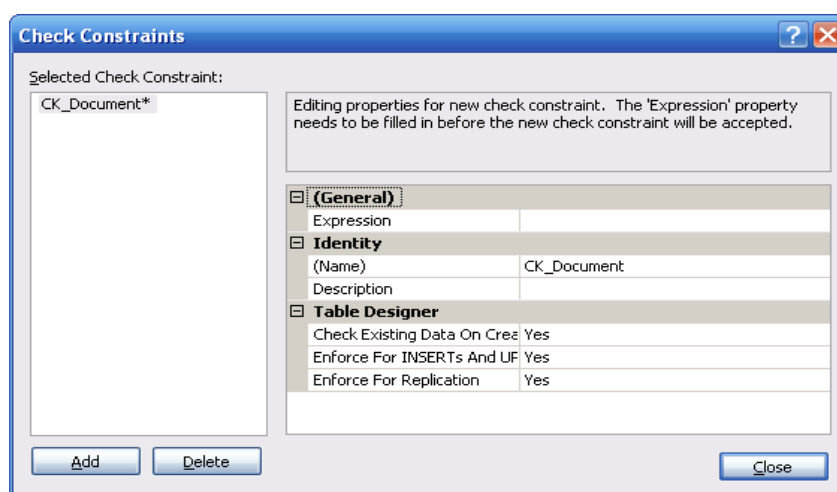


شکل ۳-۱۴: نمای کلید ساخته شده

### ۳-۱۳-۳-۳: Check Constraint

کاربردش برای وقتی است که بخواهیم برای یک فیلد، یک سری محدودیت خاص تعریف کنیم. مثلاً بگوییم فلان فیلد عددی، باید حتماً از ۱۰ بزرگتر باشد و در غیر اینصورت در هنگام ورود اطلاعات، کاربر با خطا مواجه گردد. برای به وجود آوردن یک Check Constraint روی علامت Plus جدول مورد نظرتون کلیک کنید و روی گزینه Constraints کلیک راست کرده و گزینه New Constraint را انتخاب کنید.

سپس فرمی مانند زیر باز می‌شود.



شکل ۳-۱۵: ساخت محدودیت

در اینجا و در قسمت Expression، باید عبارت محدود کننده رو وارد کنید. مثلاً در این مثال می‌گیم که فیلد Price بزرگتر از ۱۰ باشه. پس باید در قسمت Expression عبارت  $Price > 10$  رو وارد کرده و دکمه Close رو بزنید و تمام فرم‌های باز غیر از Object Browser را ببندید تا از ذخیره شدن تغییرات اطمینان حاصل بکنید. حالا اگر در Object Browser، بر روی Constraints کلیک راست کرده و گزینه Refresh را انتخاب کنید.

## فصل چهارم : پیاده سازی

## ۴-۱: اصول برنامه نویسی چند لایه

در این قسمت می خواهیم با مفاهیم Design Pattern آشنا شویم. در صورتیکه که با زبان های NET. به عنوان زبانهای کاملاً شیء گرا آشنایی داشته باشید، مطمئناً با کلاس های متعددی آشنا شده و یا شخصاً نسبت به ایجاد آنها اقدام کرده اید. حال اگر به زبان ساده Member های یک کلاس را Fields, Property, Method, Event بنامیم، شاید شما تاکنون با صدها و یا هزاران کلاسی که با یک و یا ترکیبی از این Member ها ایجاد می شوند، آشنا شده و یا ایجاد کرده باشید. حال با توجه به مطالب عنوان شده، سوال این است که یادگیری Pattern ها چه جایگاهی در طراحی و یا پیاده سازی کلاس ها دارد؟

اجازه دهید که با مثال ساده، این مساله را بررسی نماییم:

شاید در زمان طراحی و یا پیاده سازی برنامه های خود، با کلاسهایی روبرو شده اید که اصطلاحاً کلاسهای کلاسیک نبوده اند! و در زمان طراحی و یا پیاده سازی آنها، شما را کمی به فکر فرو برده و یا بعضاً دچار سردرگمی کرده باشند! در اینگونه موارد Pattern ها به یاری شما آمده و مشکلاتتان را تا حجم قابل قبولی حل می نمایند. اجازه دهید نمونه ای را با هم بررسی نماییم. تصور کنید که تصمیم دارید در پروژه خود، کلاسی طراحی نمایید که بتوان از آن تنها یک شیء ایجاد نمود! ممکن است کمی تعجب کنید!! ولی باور کنید که در یک پروژه واقعی به کرات اتفاق می افتد که شما تمایل به ایجاد چنین کلاس هایی داشته باشید. اجازه دهید نمونه هایی از این دست را برای شما ذکر نمایم:

۱- کلاس مدیرعامل. (در یک سازمان بیش از یک مدیرعامل وجود ندارد)

۲- کلاس رئیس هیات مدیره. (در یک سازمان بیش از یک رئیس هیات مدیره وجود ندارد)

۳- کلاس Supervisor.

۴- کلاس Connection. با توجه به اینکه در اکثر موارد پیشنهاد می شود که در هر پروژه، خصوصاً در

پروژه های Windows Based بیش از یک Connection به بانک اطلاعاتی نداشته باشیم، ایجاد کلاسی که تنها امکان یک شیء Connection داشته باشد، بسیار اهمیت خواهد داشت.

با توجه به نمونه های فوق، ایجاد کلاسی که بتوان از آن فقط یک شیء ایجاد نمود، اهمیت زیادی پیدا می کند. ولی چگونه چنین کلاسی ایجاد نماییم:

۱- در صورتیکه خودمان می خواهیم از چنین کلاسی استفاده نماییم، تا آخر پروژه حواسمان را جمع کنیم که خدای نکرده، بیش از یک شیء از آن ایجاد نکنیم!

۲- در صورتیکه در پروژه تیمی کار می کنیم، همکارانمان را قسم دهیم! که از این کلاس بیش از یک شیء ایجاد نکنند!!

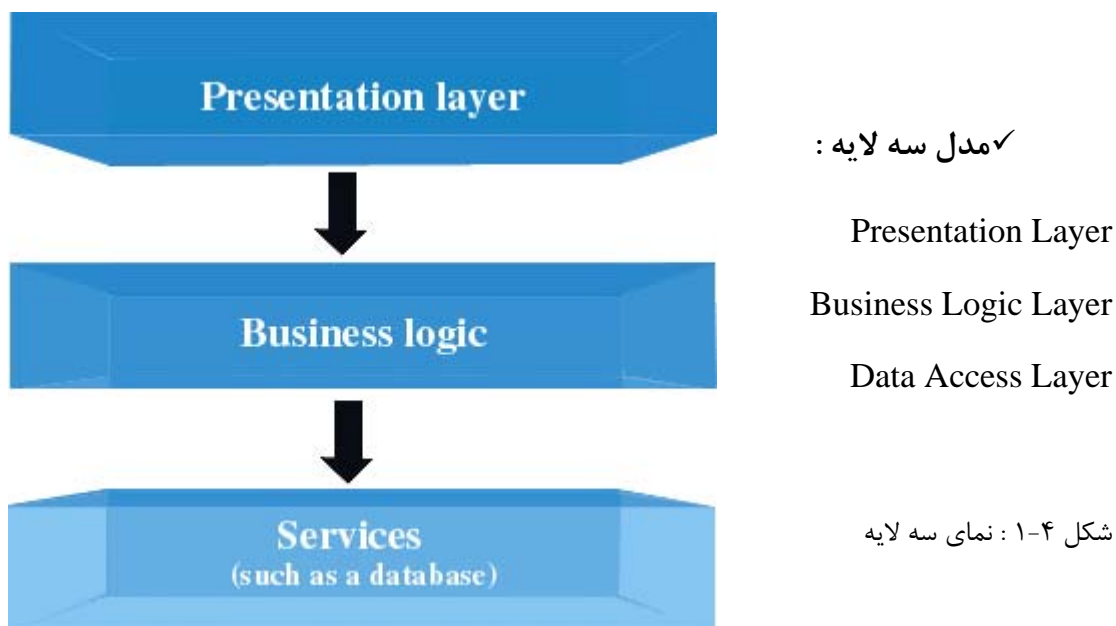
۳- ساعتها فکر کنیم تا یک راه حل نسبتاً قابل قبولی پیدا کنیم!!!

صبر کنید! این روشها را فقط برای مزاح عرض کردم!! مطمئن باشید که بکارگیری هر یک از این سه روش فوق، پروژه شما را با بحران جدی مواجه می کند. پس راه حل چیست؟

در واقع طراحی چنین کلاس هایی، به مرور زمان، گریبانگیر هر برنامه نویسی شده است و به مرور زمان، برنامه نویسان خبره با توجه به تجربیات شخصی و دیگران، تصمیم به ایجاد الگوهایی کرده اند که راه حل اینگونه مشکلات خواهد بود. در حال حاضر شاید در حدود یکصد الگو یا Pattern در این رابطه وجود داشته باشد که تنها بیست و پنج مورد از آنها استاندارد شده و مورد استفاده طراحان و برنامه نویسان قرار می گیرد. شاید یکی از ساده ترین و پرکاربرد ترین این الگوها، الگوی **Singleton** بوده که با استفاده از آن، نمونه های فوق به راحتی قابل طراحی و پیاده سازی می باشند. نکته قابل توجه این است که در برنامه Rational XDE 2003، تمامی این الگوهای معروف به صورت Built in وجود داشته و طراحان می توانند تنها با یک Drag & Drop ساده، از این الگوها در طراحی خود استفاده نمایند.

مطالبی که تا اینجا گفته شد برای این بود که در برنامه های چند لایه ساختار برنامه بسیار پیچیده و بزرگ است و بدین منظور ما ملزم هستیم که از الگوی singleton استفاده کنیم.

در معماری چند لایه تمام برنامه به چندین بخش تقسیم می شود. این بخشها می توانند فیزیکی یا منطقی باشند. هر بخش کار خاصی را انجام می دهد مثلاً نمایش interface کاربر یا دسترسی به داده ها. برنامه می تواند به هر تعداد لایه داشته باشد ولی به هر حال بیش تر برنامه ها سه یا هفت لایه ی مجزا دارند که عبارتند از:



#### ✓ مدل هفت لایه :

DataAccess Layer : ارتباط با پایگاه داده ها در این لایه انجام می شود.

IDAL Layer : ارتباط لایه DataAccess با لایه های بالاتر را برقرار می کند.

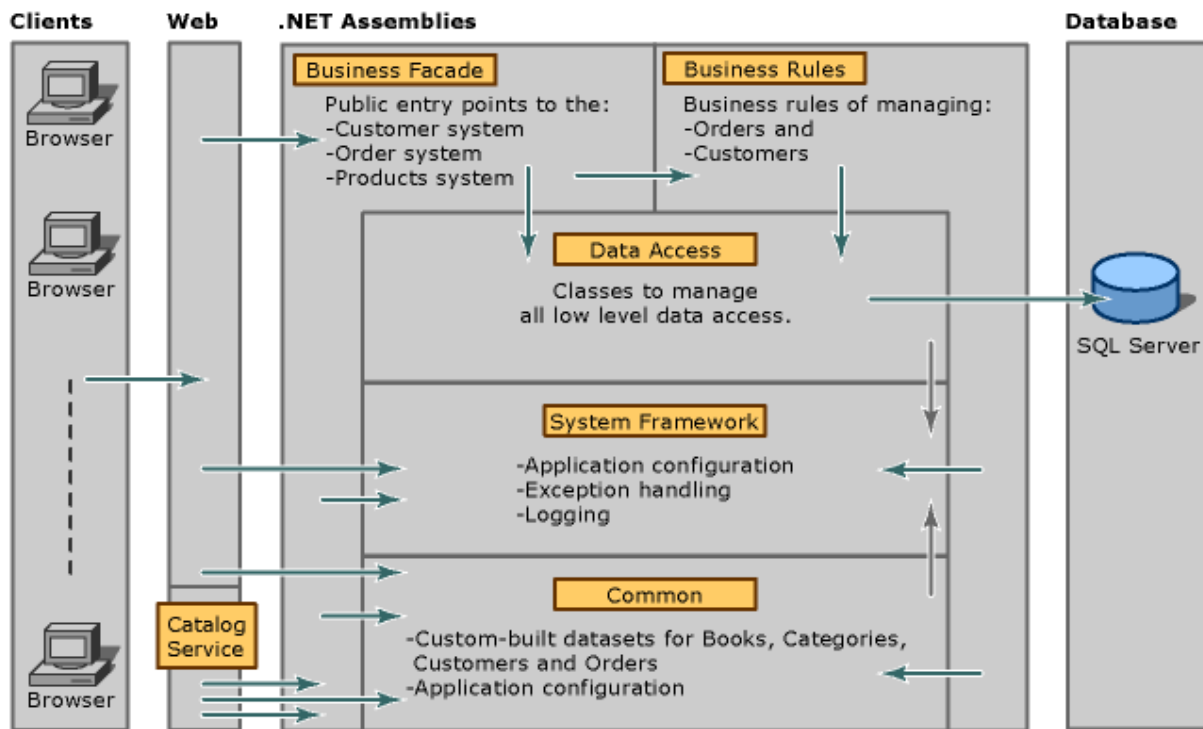
SystemFramework Layer

Common Layer : لایه برای نگهداری دیتاست ها و ثابت ها است.

BusinessRules Layer : توابع محاسباتی در این لایه قرار دارند.

BusinessFacade Layer : کار های محاسباتی در این لایه انجام می شود.

UserInterface Layer : کار این لایه دریافت ورودی و نمایش خروجی است.

**Note:**

Arrows to be interpreted as "... using ...". for example .: The Data Access subsystem is using System Framework subsystem.

شکل ۴-۲: نمای ۷ لایه

همان‌طور که احتمالاً حدس زده‌اید، لایه‌ی **Presentation** چیزی نیست به جز بخشی از نرم‌افزار که با کاربر برنامه‌ی شما ارتباط برقرار می‌کند (interface برنامه‌ی شماست). نمایش داده‌ها به کاربر نهایی و اجازه به آنان برای ارتباط داشتن با داده‌ها، اصلی‌ترین وظیفه‌ی این لایه است.

در بیش‌تر موارد، داده‌هایی که توسط کاربر وارد می‌شوند نیاز به اعتبارسنجی یا پردازش اضافی دارند. این مسئولیت لایه‌ی **Business Logic** است.

در نهایت داده‌های برنامه‌ی شما نیاز به ذخیره و بازیابی از طریق یک انبار داده دارند (مثلاً سیستم مدیریت **DataBase**‌های رابطه‌ای یا **RDBMS** و یا **XML** و ...) این وظیفه توسط لایه‌ی دسترسی به داده انجام می‌شود.

به‌طور خلاصه، فرایند مورد نظر ما این‌گونه کار می‌کند:

۱- کاربر برای داده‌های برنامه، درخواستی ارسال می‌کند.

۲- لایه‌ی Data Access داده‌های مورد نظر را بازیابی می‌کند و از طریق لایه‌ی Business Logic آن‌ها را به لایه‌ی نمایش می‌فرستد. بعضی مواقع لایه‌ی دسترسی به داده‌ها، این داده‌ها را مستقیماً به لایه‌ی نمایش ارسال می‌کند.

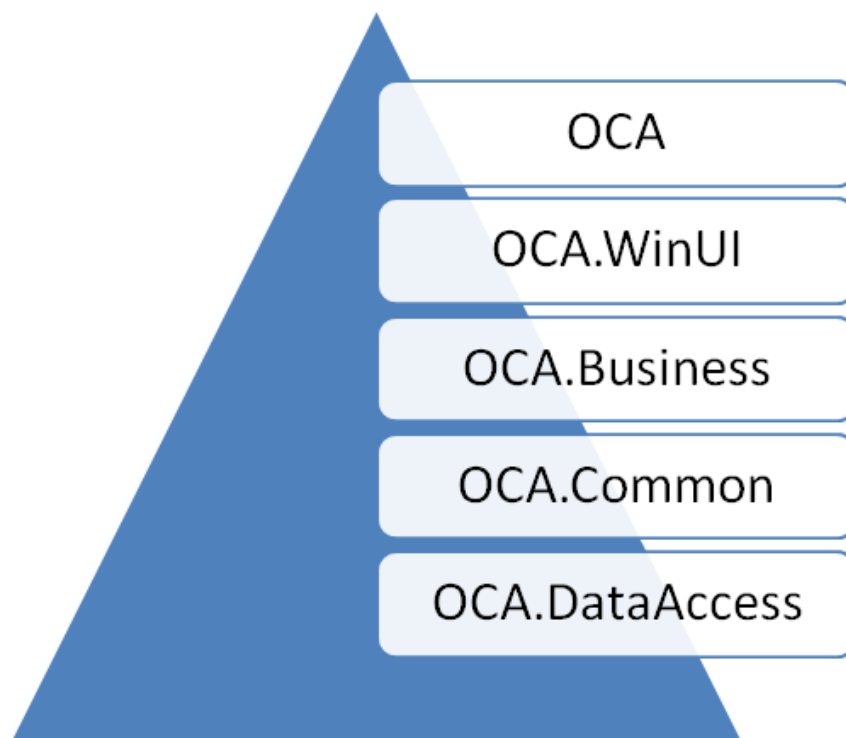
۳- لایه‌ی نمایش اطلاعاتی که باید نمایش داده شوند را از طریق لایه‌ی Business Logic دریافت می‌کند.

۴- کاربر داده‌ها را تغییر می‌دهد و عمل مناسب در مورد آن‌ها را اجرا می‌کند (مثل اضافه یا به‌روز کردن داده‌ها)

۵- لایه‌ی Business Logic صحت داده‌های وارد شده توسط کاربر را بررسی می‌کند (داده‌ها را اعتبارسنجی می‌کند)

۶- اگر داده‌ها معتبر باشند آن‌ها را برای به‌روز رسانی در بانک اطلاعاتی به‌دست لایه‌ی دسترسی به داده می‌سپارد.

### ساختار لایه‌ای سیستم ارتباط و اتوماسیون اداری سبا



شکل ۴-۳: نمای ۳ لایه برنامه اتوماسیون سبا



#### ۴-۱-۱: مزیت‌های برنامه‌های چند لایه

- برنامه‌ها به چند بخش منطقی جدا از هم تقسیم می‌شوند و اتصال میان UI (رابط کاربری)، پردازش‌ها و بانک اطلاعاتی کم می‌شود.
- تغییر در بانک اطلاعاتی یا روال‌های دسترسی به داده‌ها، تاثیری در لایه‌ی نمایش یا برنامه‌ی کلاینت نخواهد گذاشت.
- برنامه‌ی کلاینت با عبارات SQL آمیخته نخواهد شد.
- نام جداول و ستون‌ها به‌طور مؤثری از برنامه‌ی Client حذف می‌شوند.
- برنامه‌ی Client نمی‌فهمد که داده‌ها از کجا آمده‌اند (چیزی که به آن Location Transparency گفته می‌شود)
- تغییر یا گسترش برنامه بسیار ساده‌تر خواهد شد، بدون نیاز به تغییر یا کامپایل مجدد برنامه‌ی Client.
- نکته‌ی منفی در معماری چند لایه این است که شما باید تعداد زیادی بخش‌ها و کلاس‌های ازهم جدا در نرم‌افزار بسازید. اما به هر حال مزایای این روش بیش‌تر و برتر از معایب آن است.

#### ۴-۱-۲: انتخاب‌های لایه‌ی Presentation

- دو انتخاب اصلی برای ساخت یک لایه‌ی نمایش در دات نت وجود دارد. آن‌ها فرم‌های ویندوزی یا فرم‌های وبی ASP.NET هستند.
- با استفاده از ویندوز فرم‌ها شما می‌توانید برنامه‌های دسکتاپ فرم محور (Form Base) معمول را بسازید. برنامه‌های ویندوز فرمی می‌توانند المان‌های رابط کاربری بسیار غنی به‌کاربر پیشنهاد کنند. آن‌ها کم و بیش شبیه به فرم‌های ویژوال بیسیک هستند.
- جذاب‌ترین گزینه برای توسعه‌ی لایه‌ی نمایش استفاده از وب فرم‌های ASP.NET است. کنترل‌هایی مثل: دیتاگرید، دیتالیست و تقویم (Calendar) یک رابط کاربری قدرتمند را با مقدار کمی کد فراهم می‌کنند.
- انتخاب‌هایی که در بالا برای ساخت یک لایه‌ی نمایش بررسی کردیم می‌توانند توسط زبان‌های مختلفی مثل C#.net پیاده‌سازی شوند.

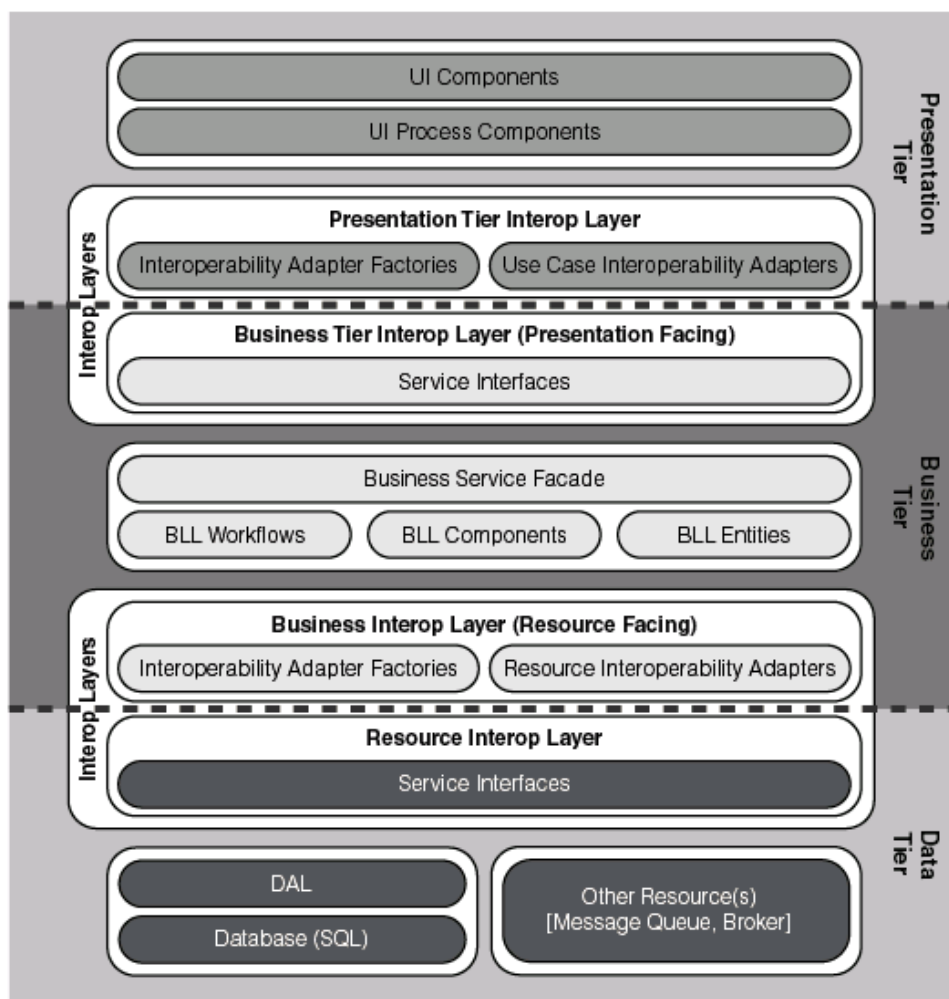
#### ۳-۱-۴: انتخاب‌های لایه‌ی Business Logic

لایه‌ی Business Logic از چندین بخش که کارهایی نظیر اعتبار سنجی کار، گردش کار یا کارهای مشابه را انجام می‌دهند تشکیل شده است.

Component های Net. این لایه را شکل می‌دهند. شما می‌توانید با Interop از Component های COM استفاده کنید ولی این کار کارایی را پایین خواهد آورد.

وب‌سرویس‌های ASP.NET هم می‌توانند به‌عنوان یک Business Logic عمل کنند. اما به هر حال آن‌ها را نمی‌توان در همه‌ی شرایط به‌عنوان جایگزین Component ها به کار برد. وب‌سرویس‌ها تنها زمانی قابل استفاده خواهند بود که اعتبارسنجی در جایی بیرون از شبکه‌ی شما اتفاق افتاده باشد.

Component هایی که توسعه می‌دهید به ماندن روی ماشین‌های مشابه نیازی ندارند. با استفاده از دات NET Remoting می‌توانید آن‌ها را روی چندین ماشین توزیع کنید.



شکل ۴-۴: معماری لایه ای برنامه سه لایه

#### ۴-۱-۴: انتخاب‌های لایه‌ی Data Access

این لایه با دستکاری داده‌ها مثل اضافه، حذف و به‌روز رسانی آن‌ها سر و کار دارد. داده‌هایی که به آن‌ها اشاره کردیم می‌توانند در RDBMS یا XML قرار داشته باشند. شما باید لایه‌ی دسترسی به داده را چنان طراحی کنید که دیگر لایه‌ها نیازی به دانستن وضعیت انبار داده‌ها نداشته باشند.

ADO.NET فناوری دسترسی به داده‌ی تحت دات Net. است. اگر چه ADO.NET از طریق کلاس‌های **DataReader** اجازه‌ی دسترسی به داده‌های در هنگام اتصال را می‌دهد ولی بیش‌ترین تمرکز روی دسترسی به داده‌ها در زمان متصل نبودن است. **DataSet** نقش کلیدی را در این مورد بازی می‌کند. در بعضی موارد می‌توانید ADO را هم برای دسترسی به داده‌ها استفاده کنید ولی استفاده از آن باید دلیل معتبری داشته باشد. از ADO استفاده نکنید فقط به خاطر این که **RecordSet**‌ها را دوست دارید!

این جا هم Component های Net. لایه را تشکیل می دهند. همان طور که قبلا گفته شد می توانید از Component های کلاسیک COM هم استفاده کنید.

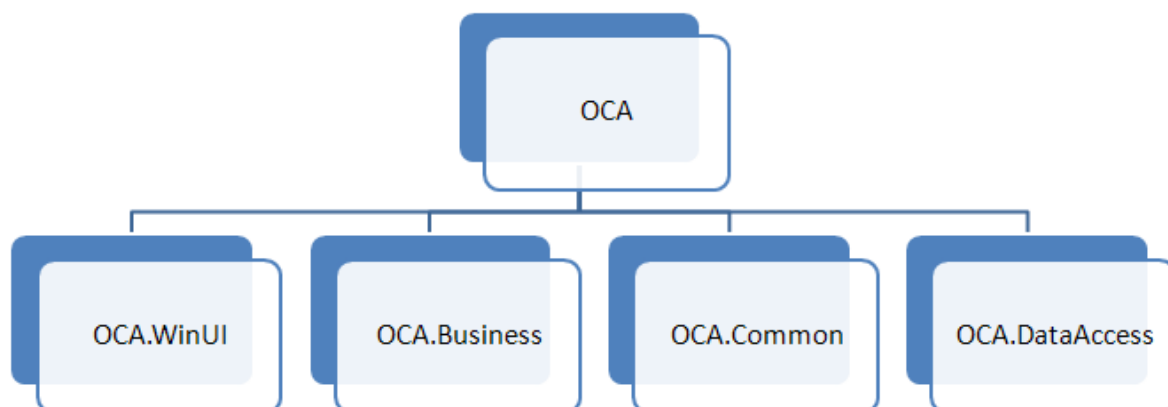
همچنین وب سرویس ها هم می توانند لایه ی دسترسی به داده را شکل دهند. این مخصوصا زمانی درست است که DataBase شما فراهم کننده ی (Provider) داده ندارد. در این گونه موارد شما می توانید مقداری کد برای اتصال به داده ها و پر کردن DataBase ها و بازگرداندن نتایج درون DataSet به درخواست کننده ی داده بنویسید.

علاوه بر ADO.NET شما می توانید از امکانات سیستم مدیریت DataBase خود مثل توابع و یا روال های ذخیره شده (Stored Procedures) استفاده کنید.

#### ۴-۱-۵: ارسال داده از یک لایه به لایه ی دیگر :

در تمام موارد به ارسال اطلاعات از یک لایه به لایه ی دیگر نیاز است؛ به طور معمول برنامه نویسان از رشته ها، آرایه ها، RecordSet های غیرمتصل برای رسیدن به این هدف استفاده می کنند. در .Net، DataSet ها یک راه فوق العاده برای انتقال اطلاعات میان لایه ها فراهم می کنند. شما حتی می توانید با برنامه نویسی یک DataSet بسازید و آن را با داده های خودتان پر کنید. اگر اشیا را خیلی دوست دارید می توانید از Typed DataSets استفاده کنید که در واقع کلاسی مشتق شده از کلاس DataSet است که جداول و سطرها را به شکل یک شی معرفی می کند.

### ساختار لایه ای سیستم ارتباط و اتوماسیون اداری سبا



شکل ۴-۵: ساختار درختی کامپوننت های تشکیل دهنده نرم افزار

#### ۴-۱-۶: ساختار لایه ای نرم افزار ارتباط و اتوماسیون اداری سبا :

این نرم افزار از ۴ لایه که در شکل زیر می بینید تشکیل شده است. هر یک از این لایه به صورت یک DLL Library هستند ، که به قسمت اصلی OCA اضافه شده اند.

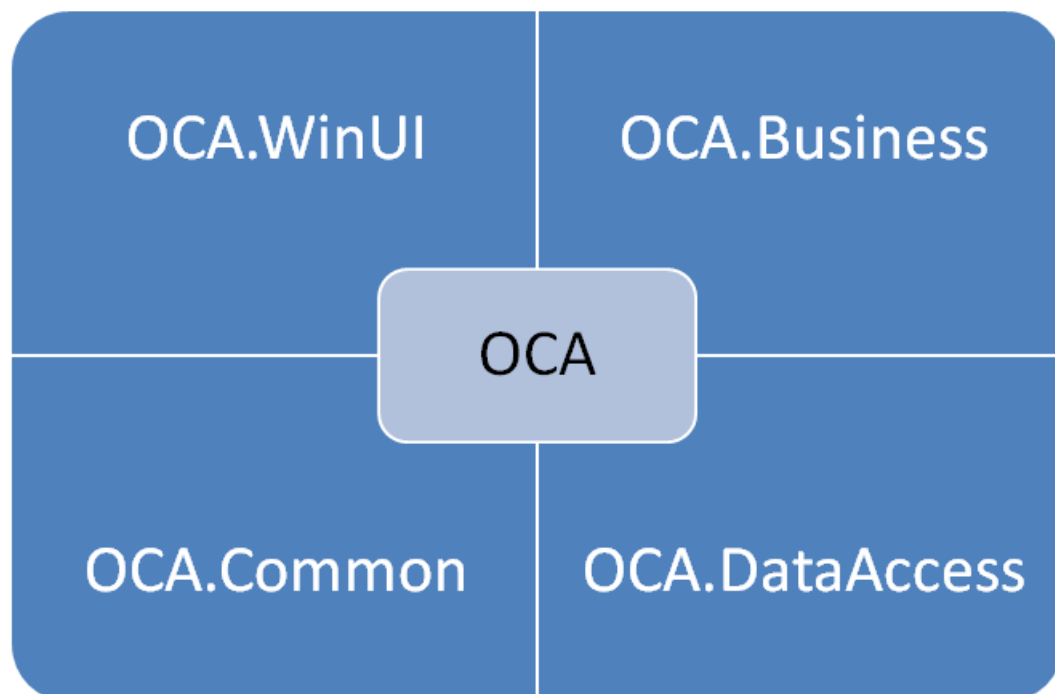
وظیفه لایه Business انجام امور محاسباتی و توابع محاسباتی و منطقی است.

وظیفه لایه WinUI پیاده سازی واسط گرافیکی کاربر است.

وظیفه لایه DataAccess پیاده سازی ارتباط با بانک اطلاعاتی است.

وظیفه لایه Common نگهداری DataSet ها و اطلاعات ثابت های برنامه و چیز هایی که همه لایه به آن احتیاج دارند ، است.

#### ساختار لایه ای سیستم ارتباط و اتوماسیون اداری سبا

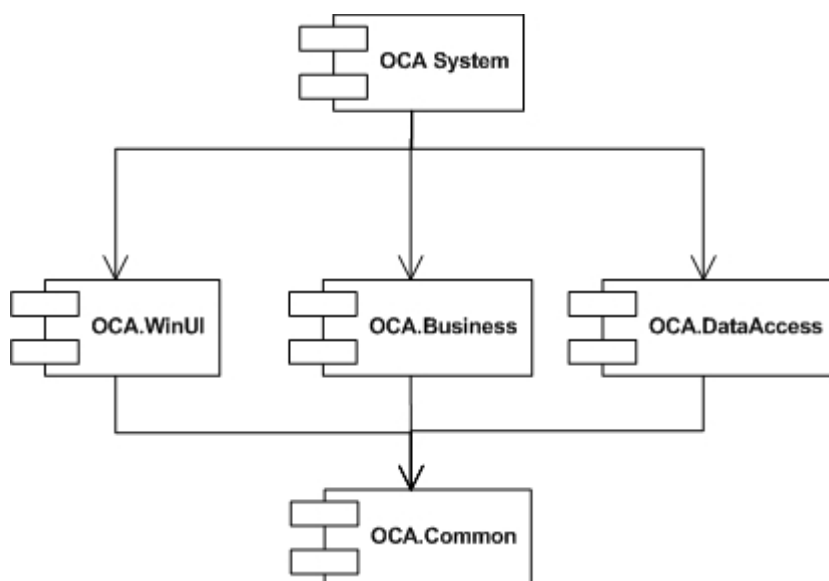


شکل ۴-۶: اجزای سازنده برنامه اتوماسیون سبا

## ۴-۲: نمودار های UML پیاده سازی

### ۴-۲-۱: نمودار جزء (Component Diagram):

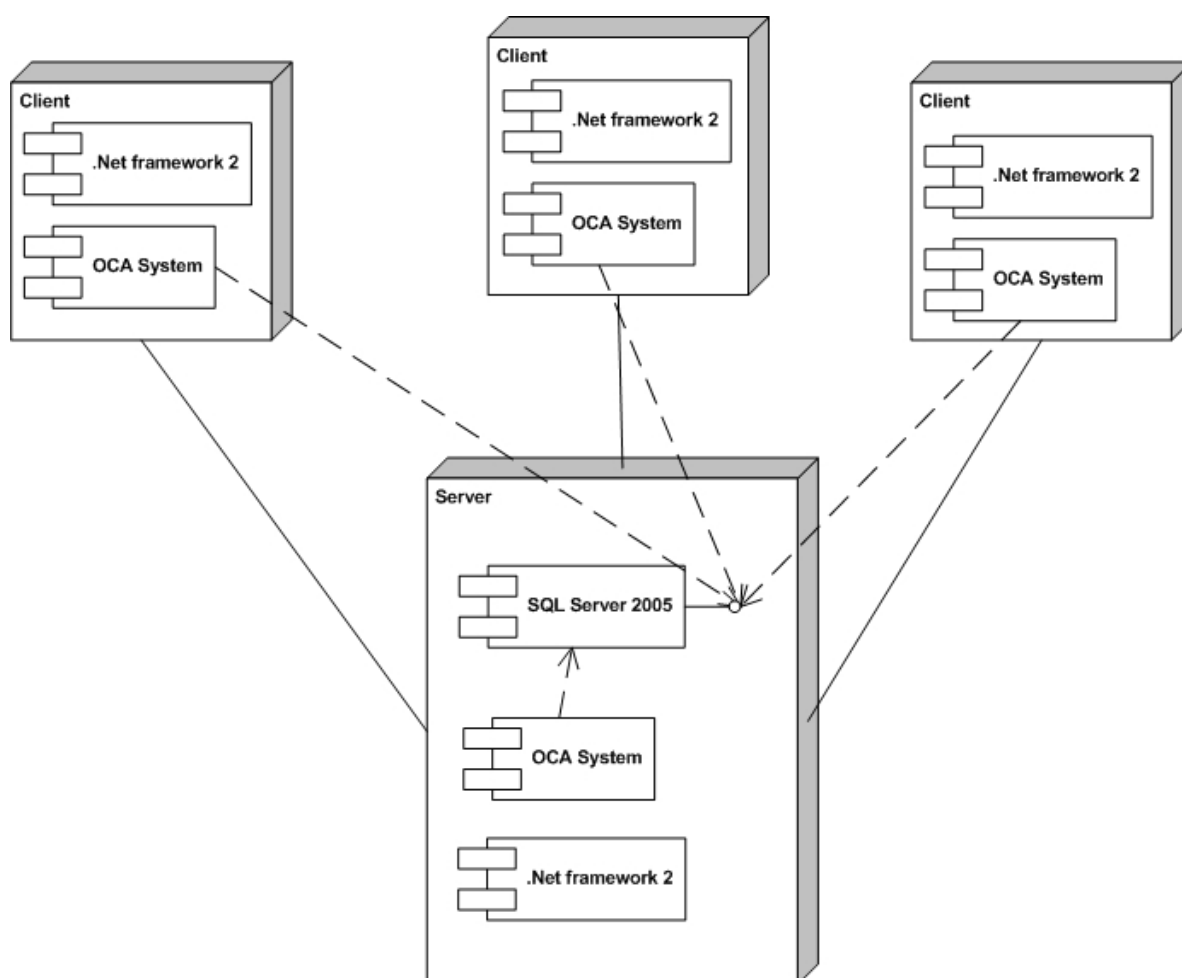
این نمودار اجزا و کامپوننت ها و dll های برنامه و ارتباط بین آنها را به نمایش درمی آورد.



شکل ۴-۷: نمودار اجزای نرم افزار اتوماسیون اداری سبا

#### ۴-۲-۲: نمودار استقرار (Deployment Diagram) :

این نمودار اجزا و طراحی سخت افزاری و شبکه ای که نرم افزار روی آن نصب است را به نمایش در می آورد.



شکل ۴-۸ : نمودار استقرار سیستم اتوماسیون سبا

### ۳-۴: واسط گرافیکی کاربر و کامپوننت های آن و راهنمای استفاده

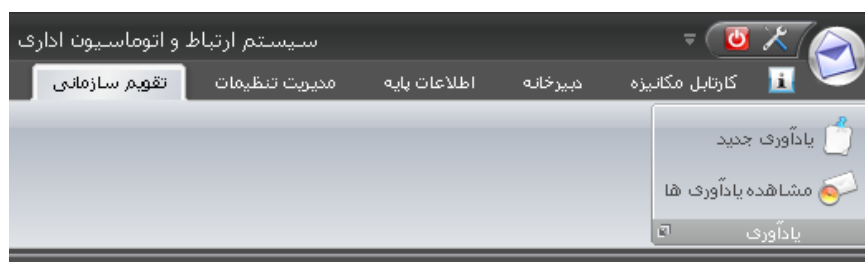
در طراحی واسط گرافیکی این نرم افزار از دو کامپوننت Krypton Toolkit و SandRibbon استفاده شده است ، که واسط این نرم افزار را به ظاهری مشابه ویندوز Vista می رساند، امکان استفاده از کنترل پیشرفته Ribbon Office 2007 را در اختیار ما می گذارد.



شکل ۳-۴ : صفحه اصلی نرم افزار



هر یک از بخش های نرم افزار یکی از بخش های منوی Ribbon است.





شکل ۴-۱۰: بخشهای مختلف منوی ریون

در شکل زیر اطلاعات فعال سازی نرم افزار نمایش داده شده است ، این نرم افزار از سیستم امنیتی سبا (ورژن ۲/۰) استفاده می کند.



شکل ۴-۱۱: پنجره در باره نرم افزار

## ۴-۳-۱: کار تبیل مکانیزه :

در این قسمت هر کاربر می تواند به مدیریت نامه های خود پردازد.

## ۴-۳-۱-۱: فرم اطلاعات کاربری:

در این فرم کاربر می تواند اطلاعات شخصی خود از قبیل نام ، گروه ، نام کاربری ، کلمه عبور ، عکس ، امضا و ... را ویرایش نماید.

شکل ۴-۱۲ : صفحه اطلاعات کاربر

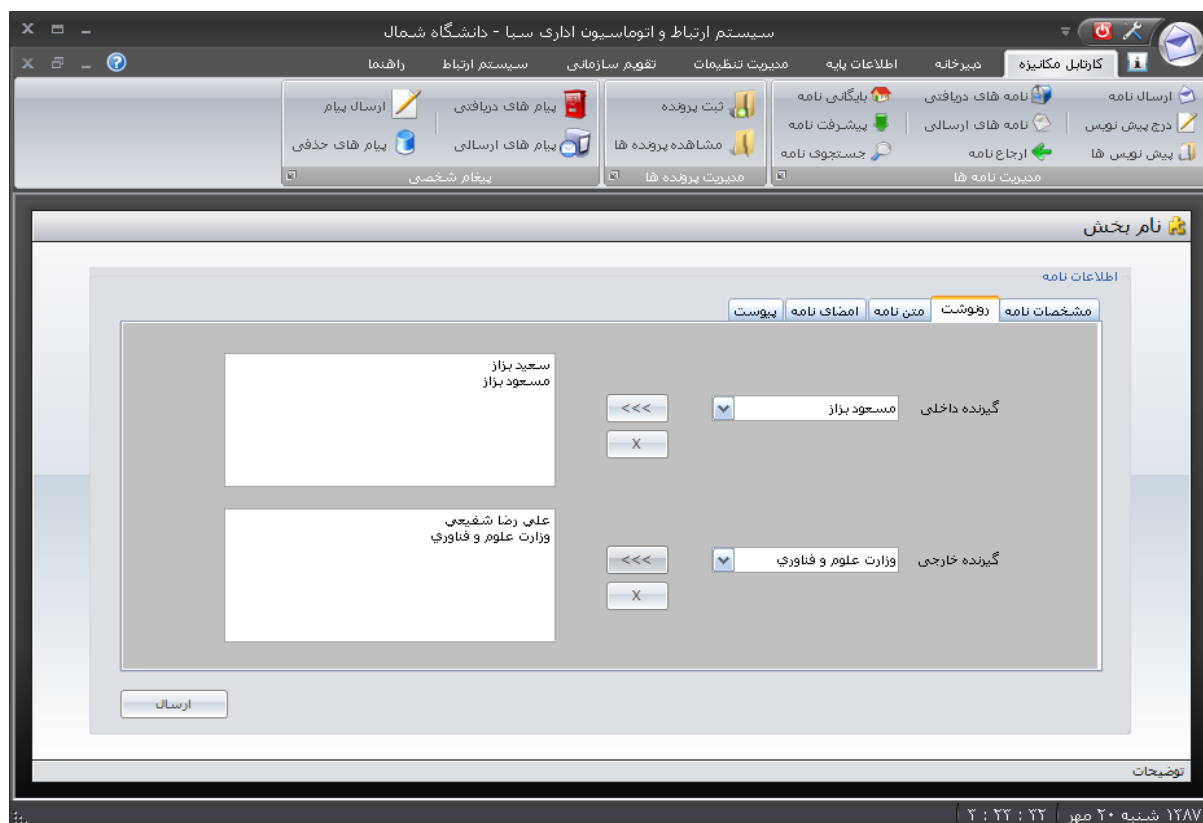
## ۴-۳-۱-۲: فرم ارسال نامه :

این فرم از ۵ بخش تشکیل شده است ، عبارتند از : مشخصات نامه ، رونوشت ، متن نامه ، امضای نامه و پیوست

**مشخصات نامه :** در این قسمت عنوان نامه ، تاریخ نامه ، اولویت ، امنیت ، وضعیت ، موضوع و گیرنده خارجی- داخلی مشخص می شوند.

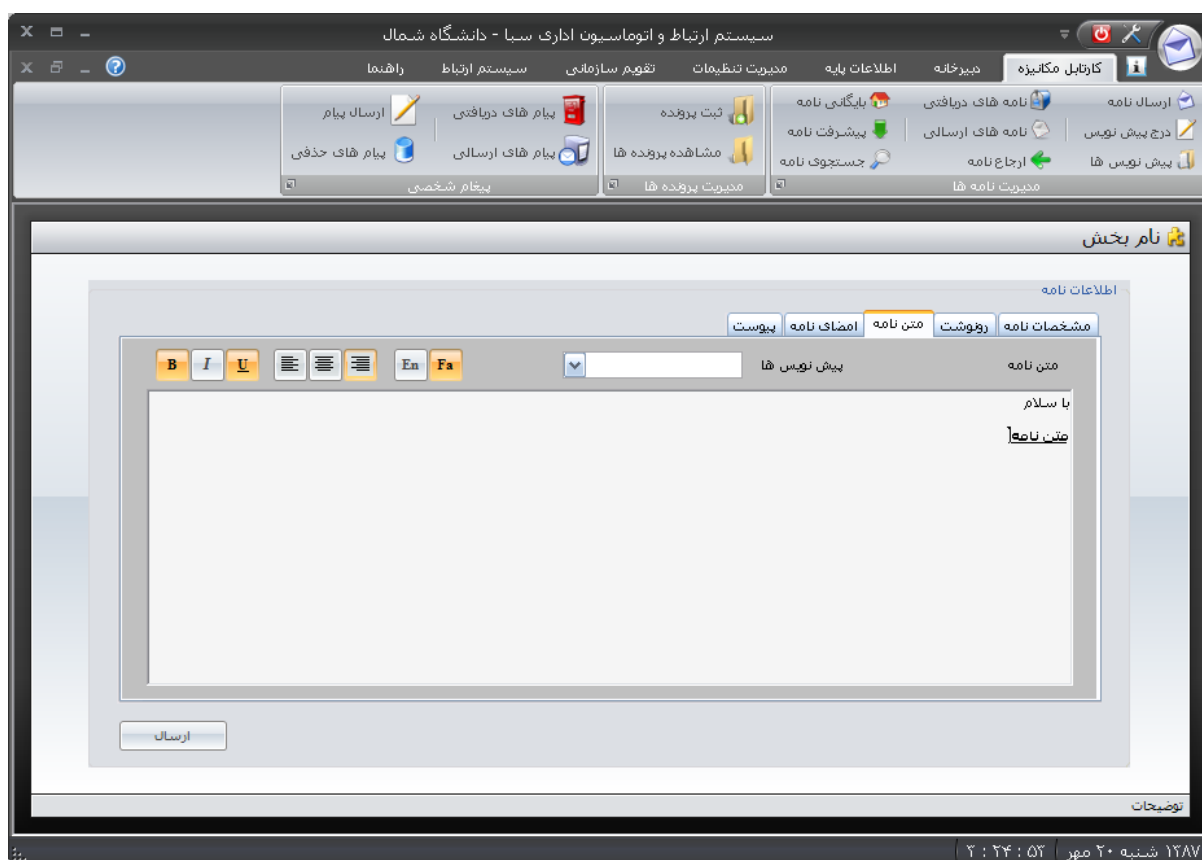
شکل ۴-۱۳ : فرم ارسال نامه

**رونوشت :** در این قسمت می توان رونشت نامه را برای سازمان خارجی یا کاربران داخلی ارسال کرد.



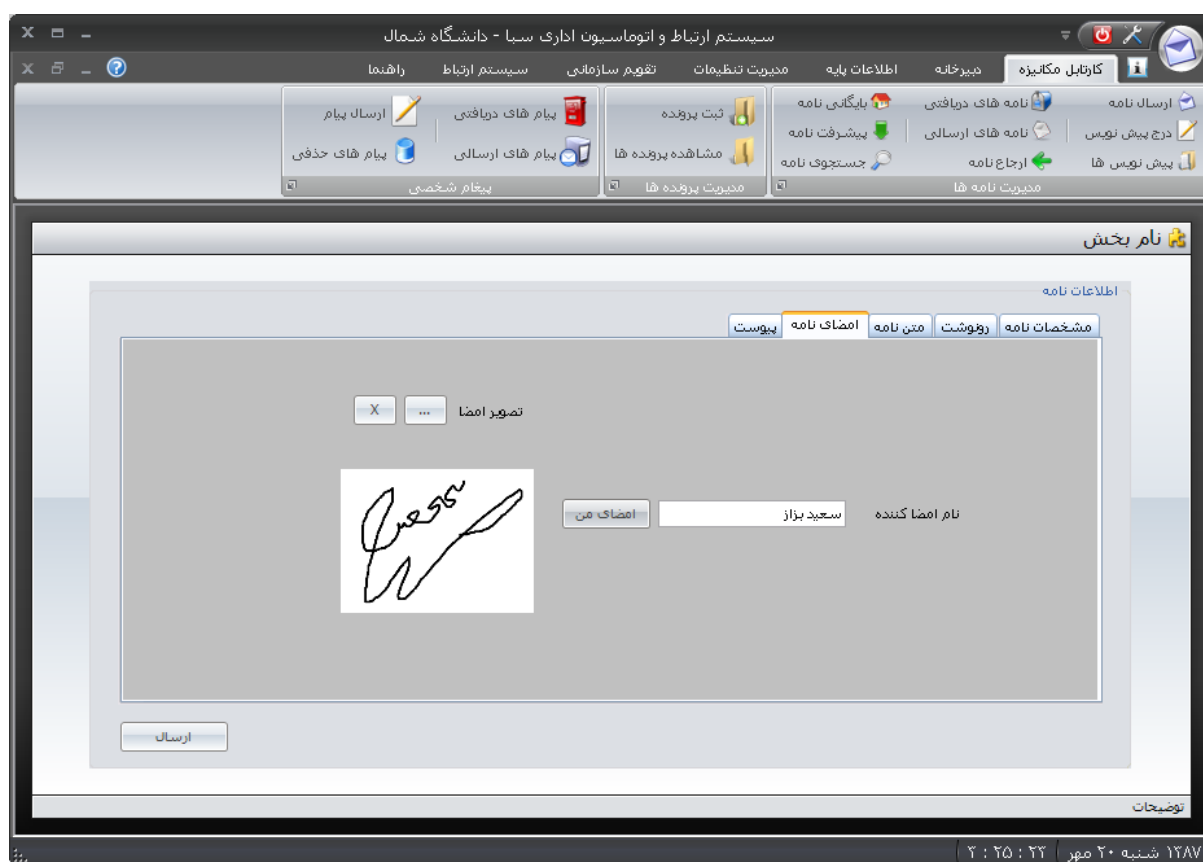
شکل ۴-۱۴ : فرم ارسال نامه – بخش رونوشت

**متن نامه :** در این قسمت متن نامه را تایپ می کنید و می توانید از پیش نویس های خود در نامه استفاده کنید.



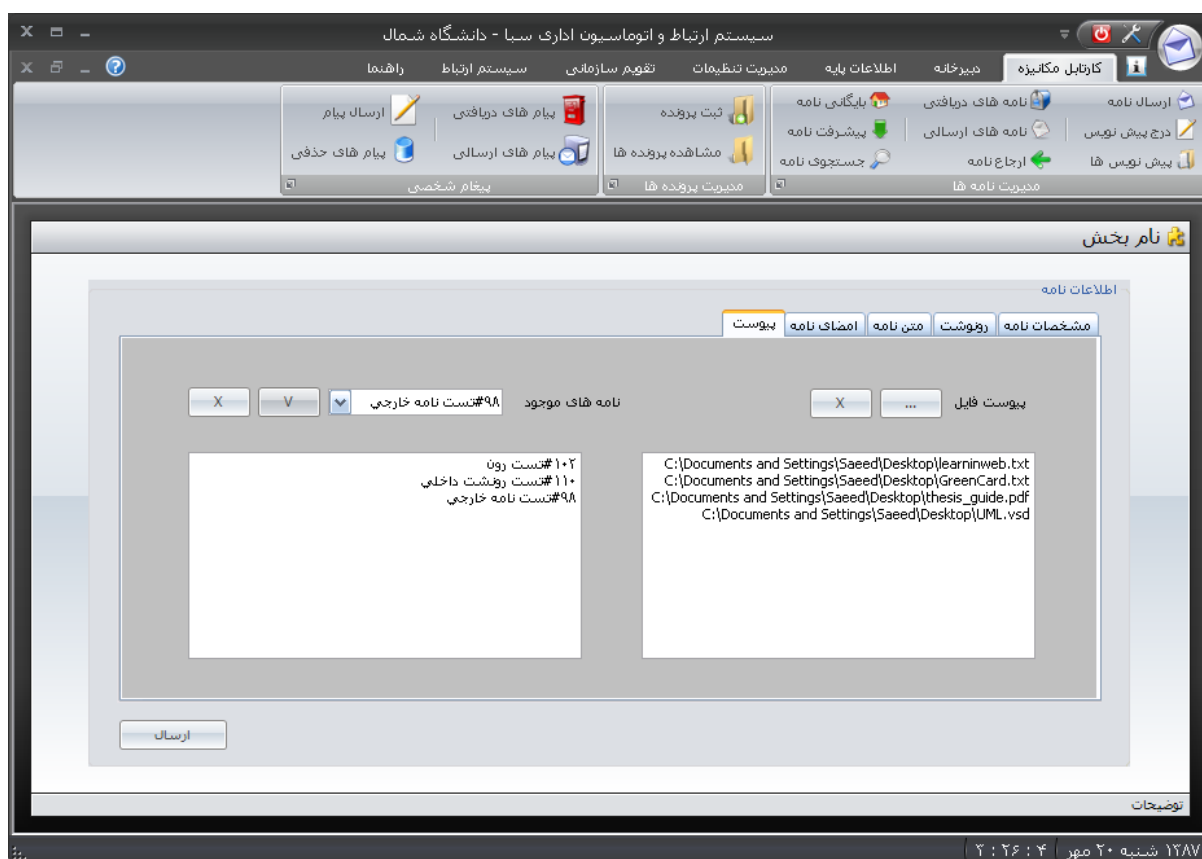
شکل ۴-۱۵ : فرم ارسال نامه – بخش متن نامه

**امضای نامه :** در این قسمت می توانید امضای خود یا شخص دیگری را در نامه قرار دهید.



شکل ۴-۱۶ : فرم ارسال نامه – بخش متن امضای نامه

**پیوست :** در این قسمت می توانید فایل یا نامه های موجود در سیستم را به عنوان پیوست ، به این نامه اضافه کنید.

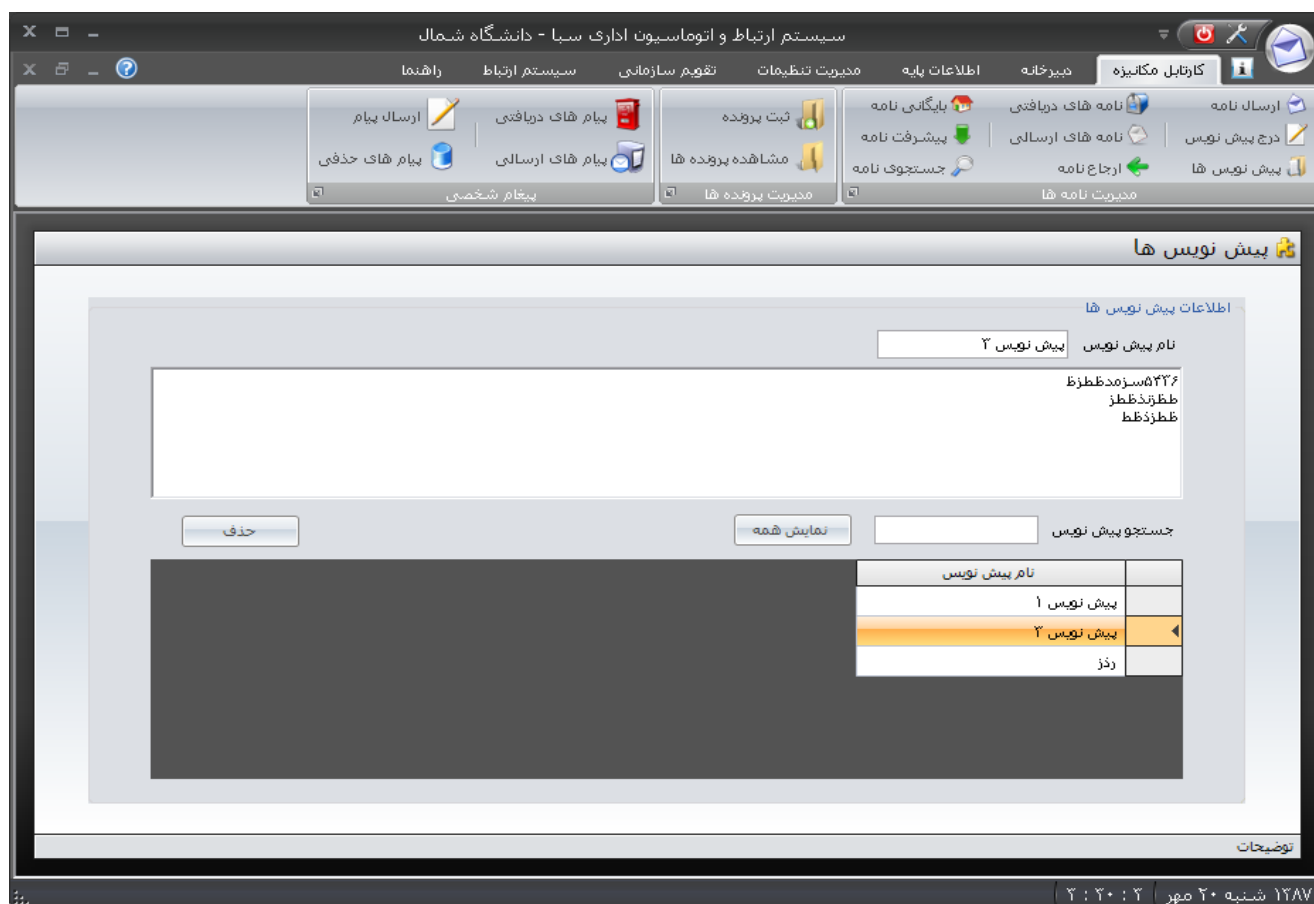


شکل ۴-۱۷ : فرم ارسال نامه – بخش پیوست



## ۴-۳-۱: فرم ویرایش پیش نویس ها :

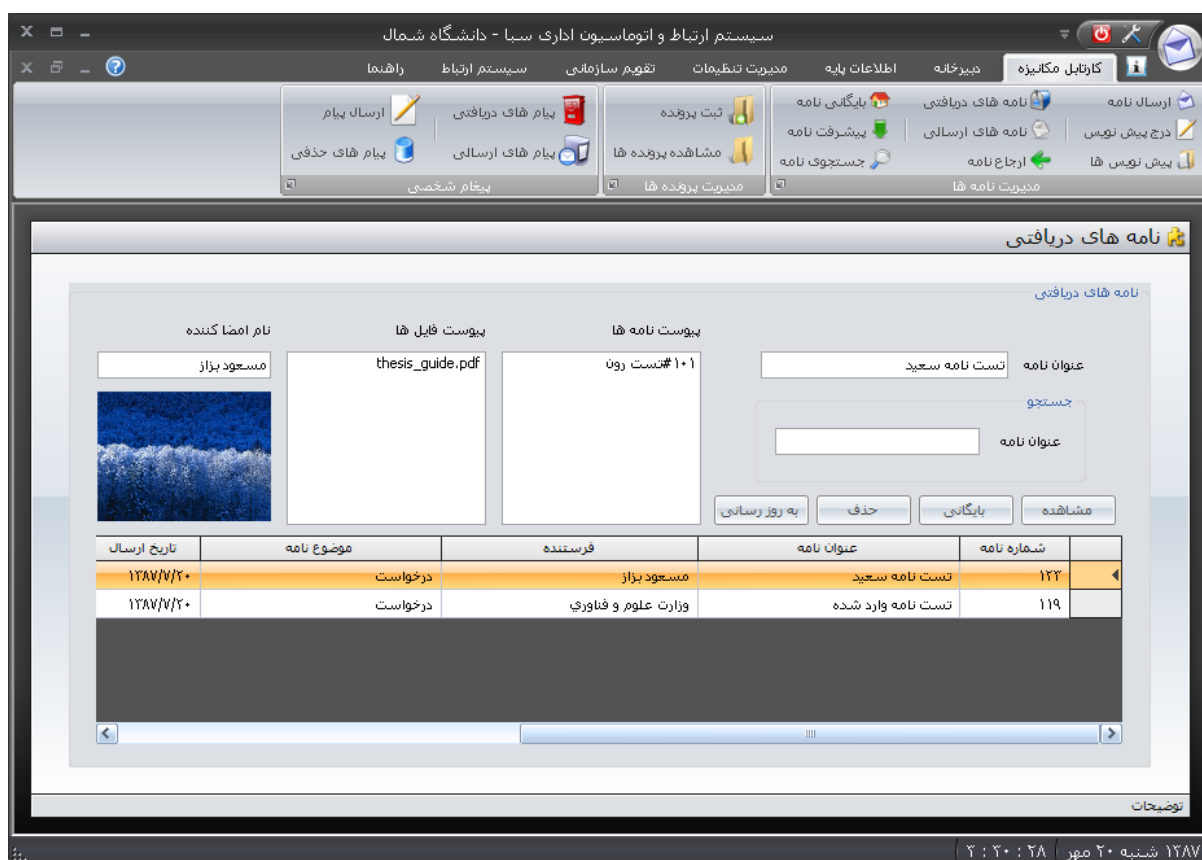
در این فرم می توانید به مشاهده و حذف پیش نویس ها بپردازید.



شکل ۴-۱۸ : فرم پیش نویس ها

## ۴-۳-۱-۴: فرم نامه های دریافتی :

در این قسمت می توانید نامه رسیده برای شما را مشاهده کنید ، متن نامه با کلیک روی دگمه مشاهده ، در فرم نمایش نامه به نمایش در می آید. و پیوست ها نیز با دابل کلیک روی آنها قابل دستیابی اند.



شکل ۴-۱۹ : فرم نامه دریافتی

به نام خدا

عنوان نامه : تست نامه سعید

تاریخ : ۱۳۸۷/۷/۲۰

شرح ارجاع :

متن نامه :

باسلام

چاکریم داداشی

شکل ۴-۲۰: فرم نمایش نامه

ذخیره فایل پیوست

Save in: Desktop

File name: Make PDF مستندات

Save as type: فایل پیوست

شماره نامه	عنوان نامه	گیرنده	موضوع نامه	تاریخ
۱۱۲	تست رون	وزارت علوم و فناوری	درخواست انتقالی	۱۳۸۷/۷/۲۰
۱۱۲	تست رون	مسعود بزاز	درخواست انتقالی	۱۳۸۷/۷/۲۰
۱۱۰	تست روشت داخلی	علی رضا شفیعی	درخواست انتقالی	۱۳۸۷/۷/۲۰
۱۰۲	تست رون	علی رضا شفیعی	درخواست انتقالی	۱۳۸۷/۷/۲۰
۱۰۰	تست رون	مسعود بزاز	درخواست انتقالی	۱۳۸۷/۷/۲۰
۹۸	تست نامه داداشی	تست نامه داداشی	درخواست انتقالی	۱۳۸۷/۷/۲۰

شکل ۴-۲۱: طریقه ذخیره پیوست

## ۴-۳-۱-۵: فرم ارجاع نامه:

این فرم مشابه فرم ارسال نامه می باشد با این تفاوت که نامه های موجود را در یک نامه جدید ارجاع می دهیم و متن نامه تغییر نمی کند ولی شرح ارجاع به نامه اضافه می شود.

The screenshot shows the 'ارجاع نامه' (Letter Referral) form within the 'سیستم ارتباط و اتوماسیون اداری سبا' (Saba Office Automation and Communication System) interface. The form is titled 'ارجاع نامه' and contains the following fields and options:

- تاریخ نامه** (Letter Date): ۱۳۸۷/۷/۲۰
- عنوان نامه** (Letter Title): تست نامه سعید
- موضوع نامه** (Letter Subject): درخواست
- اولویت نامه** (Letter Priority): بسیار فوری
- امنیت نامه** (Letter Security): فوق سری
- وضعیت نامه** (Letter Status): برگشت خورده
- گزینه** (Option): داخلی (selected), خارجی
- گزینه** (Option): ارسال (selected), ارجاع

The form is part of a larger application window with a menu bar and a toolbar. The status bar at the bottom indicates the date and time: ۱۳۸۷ شنبه ۲۰ مهر ۴ : ۵۲ : ۲۷.

شکل ۴-۲۲: فرم ارجاع نامه

## ۴-۳-۱-۶: فرم بایگانی نامه:

در قسمت بایگانی نامه هایی که در بخش نامه دریافتی و ارسالی بایگانی شده اند ، به نمایش درمی آید و می توانید نامه های موجود را در پرونده خاصی ثبت نمایید.

نام بایگانی شده

نام امضا کننده: سعید بزاز

پیوست فایل ها: Make PDF مستندات.txt

پیوست نامه ها: ۹۸# تست نامه خارجی

پرونده: نام پرونده: ذخیره

جستجو: نام های آموزش، نام های ارتباط با صنعت، نام های حسابداری

بر اساس: نام پرونده

شماره نامه	عنوان نامه	نام پرونده	تاریخ ارسال	فرستنده
۱۰۸	تست رویش داخلی		۱۳۸۷/۷/۲۰	سعید بزاز
۱۰۶	تست رویش داخلی	نامه های حسابداری	۱۳۸۷/۷/۲۰	سعید بزاز
۱۰۲	تست رون	نامه های آموزش	۱۳۸۷/۷/۲۰	سعید بزاز

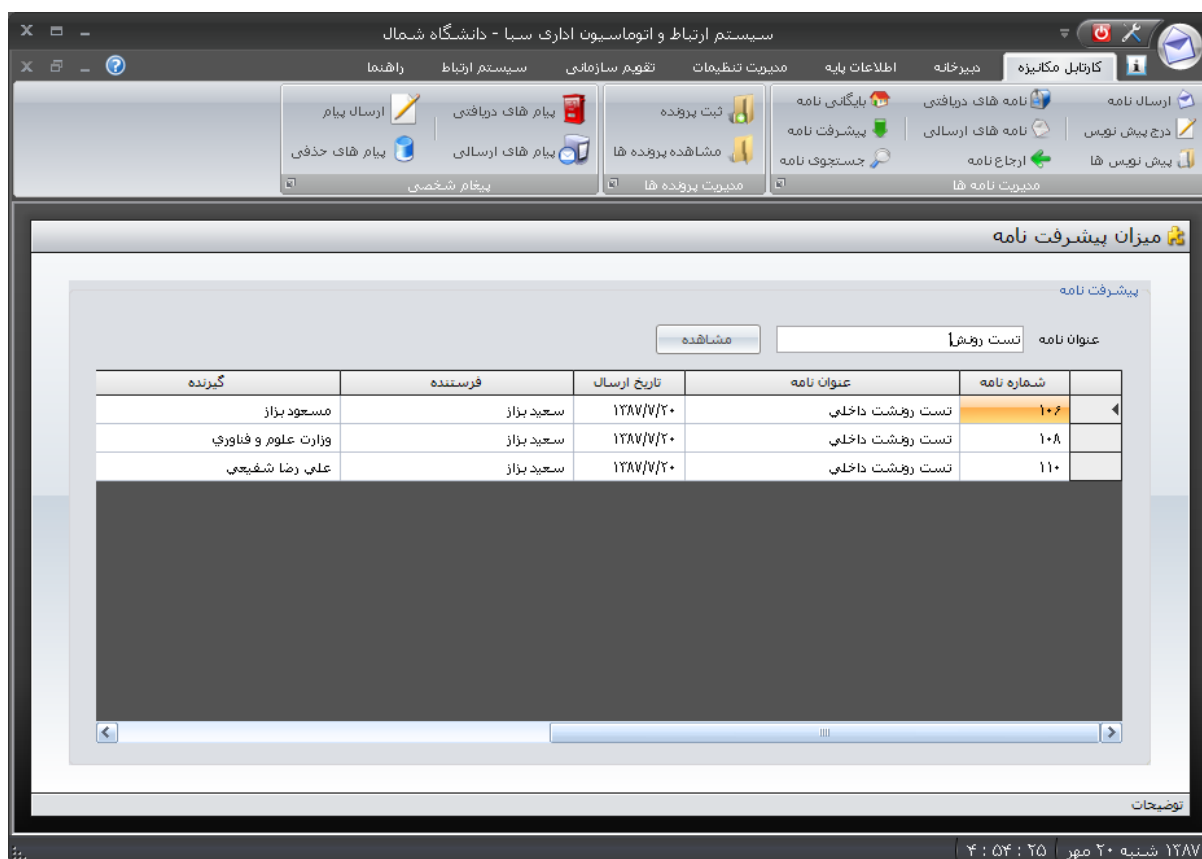
توضیحات

۱۳۸۷ شنبه ۲۰ مهر ۴۲ : ۵۲ : ۴

شکل ۴-۲۳: فرم بایگانی نامه

## ۴-۳-۱-۷: فرم پیشرفت نامه:

در این فرم روال اداری که یک نامه در سازمان طی کرده است را به نمایش در آورده ام.



شکل ۴-۲۴: فرم پیشرفت نامه

## ۴-۳-۱-۸: فرم جستجو نامه:

در این فرم شما می توانید تمام نامه های موجود در سیستم را مشاهده کنید و با تمام پارامتر ها در بین آنها جستجو کنید.

The screenshot shows the 'جستجو نامه' (Search Letter) interface. At the top, there are several tabs: 'کارتابل مگانیزه', 'اطلاعات پایه', 'مدیریت تنظیمات', 'تقویم سازمانی', 'سیستم ارتباط', and 'راهنما'. Below these are icons for various functions like 'ارسال نامه', 'درج پیش نویس', 'پیش نویس ها', 'مدیریت نامه ها', 'جستجوی نامه', 'ارجاع نامه', 'مدیریت پرونده ها', 'ثبت پرونده', 'مشاهده پرونده ها', 'پیام های دریافتی', 'پیام های ارسالی', 'پیام های حذفی', and 'پیام شخصی'.

The main search area is titled 'جستجو نامه' and contains several input fields and buttons:

- عنوان نامه:** Search by title.
- تست رون:** Search by copy test.
- جستجو نامه:** Search by letter number.
- بر اساس:** Search by basis.
- مشاهده:** View button.

Below these fields is a table with the following columns: 'گیرنده' (Receiver), 'فرستنده' (Sender), 'تاریخ ارسال' (Send Date), 'مه' (Priority), and 'شماره نامه' (Letter Number). The table contains several rows of data, with the row for 'مسعود بزاز' (Masoud Bazar) highlighted in orange.

On the right side of the table, there is a sidebar with a list of search criteria: 'شماره نامه', 'عنوان نامه', 'صاحب نامه', 'فرستنده', 'گیرنده', 'وضعیت نامه', 'موضوع نامه', 'اولویت نامه', and 'تست نامه خارجی'.

At the bottom of the window, there is a status bar showing the date and time: '۱۳۸۷ شنبه ۲۰ مهر ۴:۵۵:۱'.

شکل ۴-۲۵: فرم جستجو نامه

#### ۴-۳-۱-۹: فرم ثبت پرونده:

در این فرم می توان پرونده جدید تعریف کرد.

The screenshot displays the 'ثبت پرونده' (Register Case) form within the 'سبا' (Saba) system. The interface includes a sidebar with various navigation options such as 'ارسال پیام' (Send Message), 'پیام های دریافتی' (Received Messages), and 'ثبت پرونده' (Register Case). The main content area features a table for selecting a case name, with 'نامہ های آموزش' (Educational Cases) highlighted. A modal dialog is open, prompting the user to enter the 'نام پرونده' (Case Name) and select a date from a calendar.

نام پرونده
نامہ های آموزش
نامہ های ارتباط با صنعت
نامہ های حسابداری

اطلاعات پرونده

نام پرونده:

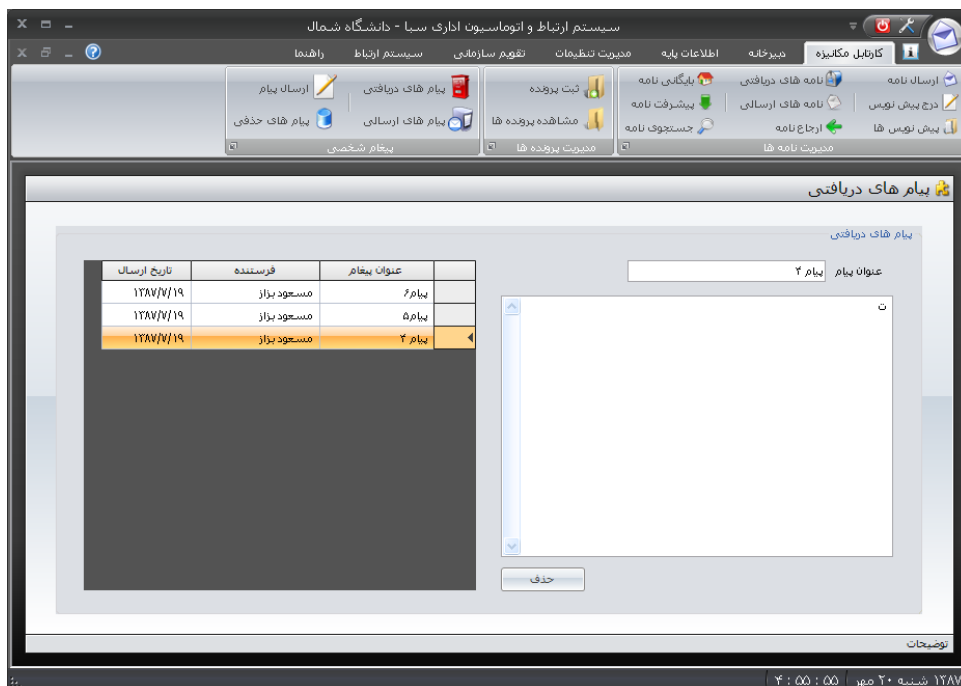
به روز رسانی ثبت

شکل ۴-۲۶: فرم ثبت پرونده

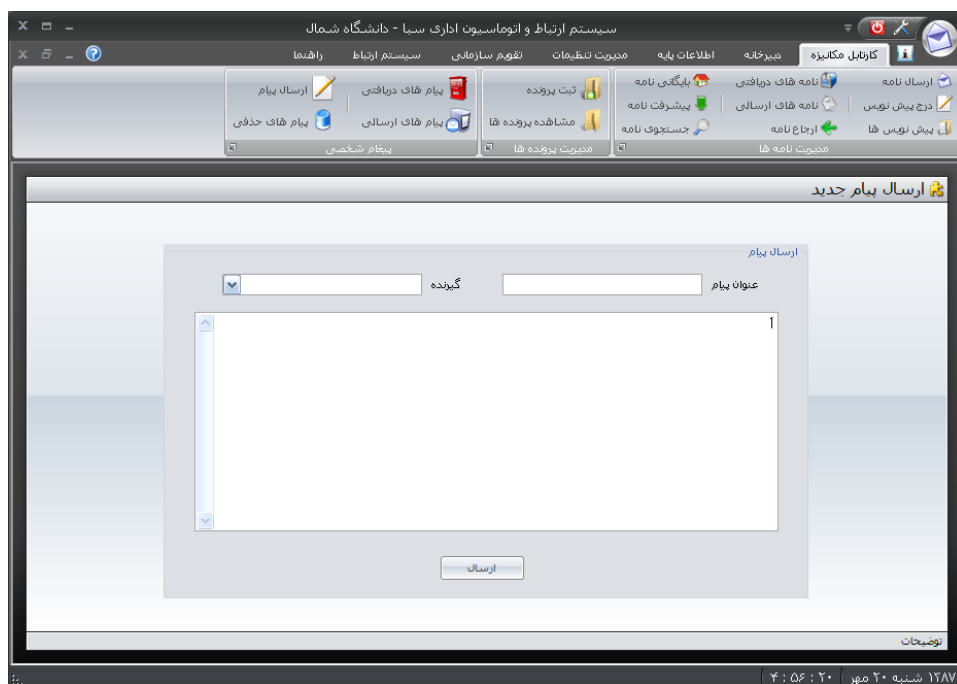


## ۴-۳-۱۰: فرم دریافت و ارسال پیام شخصی :

در این فرم ها می توانید به ارسال و دریافت پیام شخصی پردازید.



شکل ۴-۲۷ : فرم پیام های دریافتی



شکل ۴-۲۸ : فرم ارسال پیام

## ۴-۳-۲: دبیرخانه :

در این بخش از نرم افزار کاربران دبیرخانه می توانند به مدیریت و ثبت نامه های صادره و وارده بپردازند.

## ۴-۳-۲-۱: فرم مشاهده نامه های صادره :

در این فرم نامه هایی که کاربران داخلی برای سازمان های خارجی ارسال کرده اند ، به نمایش در می آید.

شماره نامه	عنوان نامه	فرستنده	گیرنده
۱۲۹	تست ۲ به سعید	مسعود بزاز	علی رضا شفیعی
۱۲۷	تست ۲ به سعید	مسعود بزاز	وزارت علوم و فناوری
۱۲۵	تست نامه سعید	مسعود بزاز	وزارت علوم و فناوری
۱۱۷	تست متن نامه	سعید بزاز	علی رضا شفیعی
۱۱۵	تست رون	سعید بزاز	وزارت علوم و فناوری
۱۱۱	تست رونشت داخلی	سعید بزاز	علی رضا شفیعی

شکل ۴-۲۹: مشاهده نامه های صادره

## ۴-۳-۲: فرم مشاهده نامه های داخلی :

در این فرم تمام نامه هایی که بین کاربران داخلی مکاتبه شده است به مشاهده در می آید.

نام امضا کننده

پیوست فایل ها

پیوست نامه ها

عنوان نامه

جستجو نامه

بر اساس شماره نامه

مشاهده

به روز رسانی

شماره نامه	عنوان نامه	تاریخ ارسال	فرستنده	گیرنده
۹۶	تست نامه داخلی	۱۳۸۷/۷/۲۰	سعید بزاز	مسعود بزاز
۱۰۰	تست رون	۱۳۸۷/۷/۲۰	سعید بزاز	مسعود بزاز
۱۰۶	تست رونشت داخلی	۱۳۸۷/۷/۲۰	سعید بزاز	مسعود بزاز
۱۱۲	تست رون	۱۳۸۷/۷/۲۰	سعید بزاز	مسعود بزاز
۱۲۲	تست نامه سعید	۱۳۸۷/۷/۲۰	مسعود بزاز	سعید بزاز

توضیحات

۱۳۸۷ شنبه ۲۰ مهر ۴: ۵۷

شکل ۴-۳۰: فرم مشاهده نامه داخلی

## ۴-۳-۳: اطلاعات پایه :

در این بخش تنظیمات پایه و اولیه برای شروع به کار سیستم انجام می شود که در بخش های کارتابل و دبیرخانه از آنها استفاده می شود.

## ۴-۳-۳-۱: فرم تعریف واحد سازمانی :

در این فرم می توانید به تعریف واحد های سازمان و ساختار آنها بپردازید.

شکل ۴-۳۱: فرم تعریف واحد سازمانی

## ۴-۳-۱: فرم تعریف سمت سازمانی :

در این فرم می توانید به تعریف سمت ها و ساختار آنها بپردازید.

سمت جدید

نام سمت

نام سمت بالایی

گروه سمت

ثبت

به روزرسانی

تعریف سمت

رئیس دانشگاه

معاونت آموزشی

مدیر گروه کامپیوتر

مدیر گروه عمران

مدیر گروه صنایع

مدیر گروه تربیت بدنی

معاونت فرهنگی

مدیر خراست

معاونت مالی

مدیر حسابداری

معاونت خدمات

مدیر سرویس

راندنده اتوبوس

مدیر آشپزخانه

آشپز

توضیحات

شکل ۴-۳۲: فرم تعریف سمت

## ۴-۳-۲: فرم تعریف سطح دسترسی :

در قسمت می توانید سطح دسترسی های مختلف را تعریف کنید و در قسمت ویرایش کارمندان می توانید دسترسی را به کاربر اختصاص داد. هر کاربر فقط به قسمتهایی که در سطح دسترسی اش تیک خورده دسترسی دارد.

شکل ۴-۳ : فرم تعریف سطح دسترسی

## ۴-۳-۳-۳: فرم مدیریت کارمندان :

در قسمت مدیر و یا کارمندانی که به این قسمت دسترسی دارند ، می توانند کارمندان سیستم را ویرایش کنند و اطلاعات آنها را تغییر دهند.

The screenshot shows the 'Management of Employees' (مدیریت کارمندان) form. The form has several input fields and buttons. Below the form is a table with the following data:

شماره پرسنلی	نام	نام خانوادگی	نام کاربری	سمت	واحد سازمانی	دسترسی
۱	سعید	بزاز	admin	رئیس دانشگاه	دانشگاه شمال	مدیر ارشد
۷	مسعود	بزاز	masoud	مدیر گروه عمران	مالی	کارمند معمولی

شکل ۴-۳۴ : فرم مدیریت کارمندان

## ۴-۳-۳-۴: فرم مدیریت اشخاص حقیقی و حقوقی :

در این قسمت می توانید اشخاص حقیقی و حقوقی را به عنوان سازمان های ثبت و ویرایش کرد.

**مدیریت سازمان های خارجی**

اطلاعات سازمان خارجی

نوع شخص:  شخص حقیقی / حقوقی

نام شخص:  نام شخص

آدرس:  آدرس

تلفن:  تلفن

فکس:  فکس

ایمیل:  ایمیل

وب سایت:  وب سایت

دفعه ثبت:  دفعه ثبت

نمایش همه

نوع شخص	نام شخص	تلفن	فکس	ایمیل	وب سایت
شخص حقیقی	علی رضا شفیعی	۷۷۵۶۰۰۲۱	۶۶۶۸۹۲۱۲	m.shafie@gmail.com	شیراز، خیابان ولی عصر، روبر
شخص حقوقی	وزارت علوم و فناوری	۶۶۵۹۸۲۷۱	۶۶۶۸۹۲۱۲	olom@info.ir	تهران، بلوار میر داماد، خیابان

توضیحات

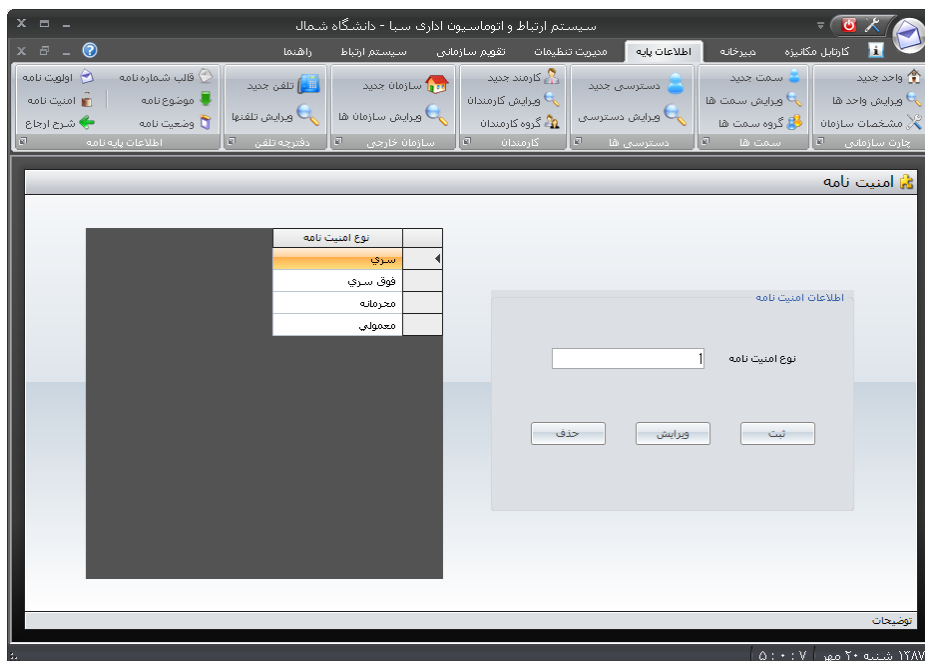
۱۲۸۷ شنبه ۲۰ مهر ۴ : ۵۹ : ۴۰

شکل ۴-۳۵: فرم مدیریت اشخاص حقیقی و حقوقی

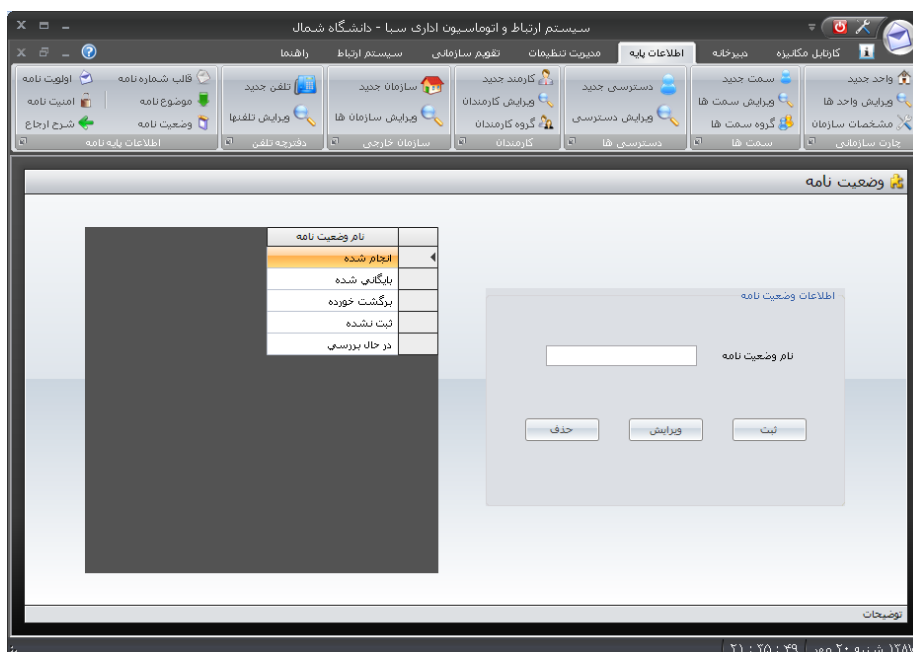


#### ۴-۳-۵: فرم های مدیریت اطلاعات نامه :

در این قسمت می توانید می توانید اطلاعاتی از قبیل امنیت ، اولویت ، موضوع و ... را برای نامه ها تعریف کنید.



شکل ۴-۳۶ : فرم امنیت نامه



شکل ۴-۳۷ : فرم وضعیت نامه

## فصل پنجم : نتیجه گیری

با وجود بررسی بالا تا حالا دیگر باید خوب با سیستم اتوماسیون اداری آشنا شده باشید ، این سیستم وظیفه خطیر و مهمی را در بهبود امر مکاتبات اداری به عهده دارد و با طراحی درست آن متوان تا حد قابل قبولی امور مکاتبات داخلی و خارجی سازمان ها را سرعت داد.

در این پروژه تا آنجایی زمان به من اجازه داد و در حیطه محدوده پروژه ام بود ، تلاش در هر چه بهتر و کامل تر نمودن این سیستم نمودم ، با این حال هیچ چیزی خارج از ایراد نیست و همواره آماده پذیرش راهنمایی دوستان در مورد این سیستم هستم. اما برای ادامه سیر طراحی این نرم افزار در آینده و بهبود و هر چه کامل تر شدن و رسیدن به خط مصرف باید خط مشئی هایی را تعیین کرد ، که در زیر به اختصار به شرح آن می پردازم:

امکان استفاده از سیستم ارتباط که به دلایل زمانی در این ورژن از نرم افزار به بهره برداری نرسید، باید در ورژن هایی بعدی محیا شود و همان طوری از اسم نرم افزار مشخص است یکی از قابلیت های مهم این نرم افزار می باشد . در ضمن قابلیت ویدئو کنفرانس نیز در آینده نزدیک باید در دسته امکانات نرم افزار قرار گیرد. همچنین یک امکان موثر دیگر که می تواند به سیستم اضافه شود ، استفاده از صندوق صوتی است و هر کاربر می تواند برای خود یک صندوق صوتی داشته باشد.

امکان ارتباط مستقیم نرم افزار با فکس و دریافت نامه ها و قرار دادن تصویر آنها در یک نامه وارده با توجه به شماره تلفن سازمان خارجی ، باید به نرم افزار اضافه شود و همچنین قابلیت ارسال نامه های موجود در سیستم توسط فکس.

امکان ارتباط مستقیم با سخت افزار اسکنر و قرار دادن مستقیم نامه های اسکن شده در داخل نامه ها، به عنوان متن و یا پیوست نامه. نامه های دریافتی با استفاده از تکنولوژی OCR پردازش شده و متن آنها در سیستم به عنوان متن نامه ذخیره می شود.

امکان استفاده از سخت افزار قلم نوری که از طریق آن می توان به تولید نامه با قلم نوری نمود و فرآیند تولید نامه را سرعت بخشید.

امکان ارتباط مستقیم با ایمیل سرور ( نظیر Microsoft Exchange Server ) که از این طریق بتوان به ورود و خروج نامه ها از طریق پست الکترونیکی دست یابی پیدا کرد.

اضافه شدن لایه OCA.WebUI به نرم افزار، که قابلیت استفاده در شبکه جهانی وب را به ما بدهد و کاربران بتوانند از هر نقطه ای به سیستم دسترسی داشته باشند.

اضافه شدن لایه OCA.MobileUI به نرم افزار ، که قابلیت مدیریت از طریق تلفن همراه را در اختیار کاربران قرار می دهد و کاربران می توانند از طریق تلفن همراه نامه خود را دریافت و ارسال نمایند.

سعید بزاز

s.bazzaz@yahoo.com

## مراجع :

۱- [www.roshd.ir](http://www.roshd.ir)

۲- [www.barnamenevis.org](http://www.barnamenevis.org)

۳- [www.rayvarz.com](http://www.rayvarz.com)



Shomal University  
Engineering Faculty  
Computer Department

# Official Communication & Automation Systems (OCA Systems)

In Partial Fulfillment of the Requirement for the degree of  
Bachelor of Science in  
Computer – Software

By:  
Saeed Bazzaz Abkenar

Adviser :  
Mr Hamed Rajabi Ghomi

October 2008