

گزارش کار سیستم انتخاب واحد با معماری سه لایه اعضای گروه: مریم ابراهیم نژاد ، ریحانه حسینی نژاد

در این پروژه سیستم انتخاب واحد دانشگاه را شامل دو جدول برای دانشجو و دروس ارائه شده با معماری سه لایه را پیاده سازی کردیم. که در آن از بانک اطلاعاتی Access استفاده کردیم.

ابتدا نرم افزار Access را اجرا کردیم و یک بانک اطلاعاتی بنام select unit ایجاد کردیم و دو جدول شامل course, Studen ایجاد نمودیم.

در ادامه وارد visual studio شدیم و یک برنامه جدید بنام N_Layer_Course ایجاد کردیم و بر روی نام برنامه در پنجره Solution Explorer کلیک راست کرده و برای هر یک از لایه ها یک پوشه ساختیم.

لایه DataAccess

با انتخاب گزینه Class در پنجره باز شده کلاس های زیر را در پوشه DAL ایجاد کنید.

۱- یک کلاس به نام ConnectionManager به این پوشه اضافه کردیم. این کلاس برای مدیریت شی connection مورد استفاده قرار می گیرد.

کار این کلاس ایجاد شی اتصال، باز کردن شی اتصال و بستن آن می باشد. تنها به ازاء تمام اشیائی که از این کلاس تعریف می شوند تنها یک شی اتصال موجود باشد.

یک کلاس به نام BaseAdapter به پوشه DAL اضافه کنید و کلاس ایجاد شده را به شکل زیر برنامه نویسی کردیم. این کلاس پایه ذخیره سازی در بانک اطلاعاتی و واکنشی داده هاست و به نحوی که برای ذخیره سازی و واکنشی داده های هر موجودیت مناسب می باشد و بدون تغییر در هر برنامه ای می تواند در لایه DAL قرار گیرد.

این کلاس شامل متدهای زیر می باشد:

- متد Select_All: این متد یک رشته پرسوجو از پارامتر ورودی دریافت کرده، داده های معادل رشته پرس و جو را واکنشی کرده و بصورت یک DataTable برمی گرداند.

- **Select_By**: این متد یک رشته پرسوجو و یک مجموعه از پارامترهای مورد نیاز رشته پرسوجو را از پارامتر ورودی دریافت کرده، داده‌های معادل رشته پرسوجو را واکنشی کرده و بصورت یک **Datatable** برمی‌گرداند.

- متد **Execute_Command_NonQuery**: این متد برای اجرای تمام دستورات **Access** که خروجی ندارند طراحی شده است. این متد یک رشته پرسوجو و یک مجموعه از پارامترهای مورد نیاز رشته پرسوجو را از پارامتر ورودی دریافت کرده، رشته پرسوجو را اجرا کرده و نتیجه اجرا بر روی داده‌ی بانک اطلاعاتی اعمال می‌گردد. از این متد برای درج، حذف، و بروزرسانی داده‌ها در بانک اطلاعاتی استفاده می‌گردد. یک کلاس به نام **courseAdapter** به پوشه **DAL** اضافه کردیم. این کلاس را برای مدیریت ذخیره و بازیابی اشیائی از نوع کلاس **course** پیاده‌سازی کردیم، کلاس **BaseAdapter** را به ارث می‌برد و دارای تمام اجزاء کلاس **BaseAdapter** است. این کلاس شامل ۶ متد به شرح زیر است:

- متد **Insert**: این متد اطلاعات یک درس را از پارامتر ورودی دریافت کرده و اطلاعات دریافت شده را به مجموعه‌ای از نوع اشیاء پارامتر تبدیل می‌کند. سپس به کمک متد **Execute_Command_NonQuery** که از کلاس پایه به ارث برده شده است، اطلاعات را در بانک اطلاعاتی درج می‌کند.

- متد **Update_By_Course**: این متد اطلاعات جدید یک درس را از پارامتر ورودی دریافت کرده، اطلاعات دریافت شده را به مجموعه‌ای از نوع اشیاء پارامتر تبدیل می‌کند. سپس به کمک متد **Execute_Command_NonQuery** که از کلاس پایه به ارث برده شده اطلاعات را در بانک اطلاعاتی بروزرسانی می‌کند. در این متد ویرایش اطلاعات براساس شماره درس دانشجو یکتا می‌باشد انجام میشود.

- متد **Delete_By_Course**: این متد شماره درس را از پارامتر ورودی دریافت کرده، اطلاعات دریافت شده را به مجموعه‌ای از نوع اشیاء پارامتر تبدیل می‌کند و به کمک متد **Execute_Command_NonQuery** که از کلاس پایه به ارث برده شده است، اطلاعات را از بانک اطلاعاتی حذف می‌کند. حذف اطلاعات بر حسب شماره درس انجام می‌پذیرد.

- **Select_Allcourse**: این متد اطلاعات تمامی رکوردهای موجود در بانک اطلاعاتی را بصورت یک شی جدول داده‌ای برمیگرداند. این متد به کمک متد **Select_All** به ارث برده شده از کلاس پایه تمامی رکوردهای موجود در بانک اطلاعاتی را بصورت یک شی **DataTable** برمیگرداند.

- **Select_Allcourse_By_Course**: این متد شماره درس را از پارامتر ورودی دریافت می‌کند و اطلاعات دریافت شده را به مجموعه‌ای از نوع اشیاء پارامتر تبدیل می‌کند. سپس به کمک متد **Select_By** به ارث برده شده از کلاس پایه، مشخصات درس ورودی را از بانک اطلاعاتی بازیابی کرده و بصورت یک شی **DataTable** برمی‌گرداند.

- **Select_Allcourse_Like_By**: این متد نام فیلد مورد جستجو و مقدار آن را از پارامتر ورودی دریافت کرده، اطلاعات دریافت شده را به مجموعه‌ای از نوع اشیاء پارامتر تبدیل می‌کند. سپس به کمک متد **Select_By** که از کلاس پایه به ارث برده شده است، نتایج جستجو را از بانک اطلاعاتی بازیابی کرده و بصورت یک شی **DataTable** برمی‌گرداند. در این متد از ویژگی **Like** زبان پرس‌وجوی **Access** استفاده شده است. لایه **Data** با ایجاد این کلاس پایان یافت.

لایه Business

فرمان تغییرات در داده‌ها نظیر ذخیره کردن، حذف کردن، بروز رسانی و نظایر آن توسط این لایه انتقال داده می‌شود. پیامی در رابطه با اجرای موفقیت آمیز یا عدم اجرای فرمان کاربر توسط لایه **UI** به کاربر نمایش داده می‌شود. برای پیاده‌سازی این لایه دو مجموعه کلاس در نظر گرفتیم:

- مجموعه اول کلاس‌هایی برای ارتباط با لایه بالاتر یعنی لایه **UI**. وظیفه این کلاس‌ها دریافت داده‌های ورودی از لایه **UI**، و ارسال داده‌ها و نتایج به عنوان خروجی به لایه نمایش است.

- مجموعه دوم کلاس‌هایی برای ارتباط با لایه پایین‌تر یعنی لایه **DAL** می‌باشد. وظیفه این کلاس‌ها ارتباط با لایه پایین‌تر است. این کلاس‌ها را در زیر پوشه‌ای بنام **Providers** در لایه **BL** قرار داده ایم.

در این مرحله بر روی پوشه **Provider** کلیک راست کرده و با انتخاب گزینه **New Item** و با انتخاب گزینه **Interface** و **Class** در

پنجره باز شده اجزاء زیر را در پوشه Provider ایجاد کردیم. یک Interface به نام IProvider به این پوشه اضافه کردیم. این واسط کمک می‌کند تا تمام موجودیت‌های این لایه به یک صورت مدیریت شوند. واسط افزوده شده برنامه‌نویسی کردیم. یک Class به نام CourseProvider به زیرپوشه Provider در لایه BL اضافه کردیم.

در این کلاس به کمک اشیایی از نوع کلاس CourseAdapter با لایه DAL ارتباط برقرار می‌شود و همچنین با اشیایی از نوع کلاس CourseEntity با لایه UI ارتباط برقرار می‌شود.

آخرین کلاس لایه BL کلاسی بنام CoursetEntity می باشد که اشیایی از نوع این کلاس برای ارتباط با لایه بالاتر یعنی لایه UI بکار می‌روند. ابتدا یک کلاس بنام CourseEntity به زیرپوشه Entities در لایه BL اضافه کردیم و سپس آن را برنامه‌نویسی کردیم.

ایجاد لایه Peresentaion

لایه کاربر در اصل همان ظاهر برنامه است که کاربران با آن ارتباط برقرار می‌کنند تا نیازهای خود را به سیستم برای اعمال تغییرات یا گرفتن گزارش ارائه دهند. این لایه بطور کلی از قوانین موجود در سیستم مجزا می‌باشد. کاربران تنها لیستی از داده‌ها را می‌بینند و یا داده‌های جدید را به این لایه ارائه می‌دهند. نمایش داده‌ها به کاربر نهایی اجازه دادن به آن‌ها برای ارتباط داشتن با داده‌ها، اصلی‌ترین وظیفه این لایه است. در برنامه انتخاب واحد لایه UI تنها یک فرم است که کاربر بوسیله آن می‌تواند با عملیات انتخاب واحد و داده‌های درون آن ارتباط برقرار کند. در لایه UI نیاز داریم تا به کلاس‌های موجود در لایه BL دسترسی داشته باشیم. برای ارتباط لایه UI با لایه BL باید در لایه کاربر یک شی از نوع کلاس CourseProvider موجود در لایه BL تعریف کنیم. برای کدنویسی این فرم 4 زیربرنامه داریم.

زیربرنامه delete:

زیربرنامه insert

زیربرنامه update

