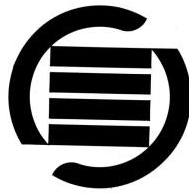


بنام خداوند جان و



وزارت علوم، تحقیقات، و فناوری

پژوهشگاه علوم و فناوری اطلاعات ایران

طرح پژوهشی

بازطراحی و روزآوری سامانه عضویت فراگیر کتابخانه‌ها

مجری

محمد مهدی آقالویی

محمد هادی آقالویی

علی اصغر حجت پناه

مجتبی زالی

آذر ۱۳۹۵

حقوق: پژوهشگاه علوم و فناوری اطلاعات ایران

طرح پژوهشی «بازطراحی و روزآوری سامانه عضویت فراگیر کتابخانه‌ها» پیرو قرارداد شماره ۳۳/۹۱۷۶ مورخ ۱۳۹۴/۱/۲۶ میان پژوهشگاه علوم و فناوری اطلاعات ایران (کارفرما) و آقای محمد مهدی آقالویی (مجری) اجرا شده است. گزارش حاضر مستند این طرح است. این گزارش و تمامی حقوق مادی آن بر اساس «قانون حمایت از حقوق مؤلفان و مصنفان و هنرمندان» مصوب سال ۱۳۸۴ و اصلاحیه‌های بعدی آن و همچنین آیین‌نامه‌های اجرایی این قانون، متعلق به پژوهشگاه علوم و فناوری اطلاعات ایران است و هرگونه استفاده از تمام یا پاره‌ای از آن شامل نقل قول، تکثیر، انتشار، کاربرد نتایج، تکمیل و مانند آن به صورت چاپی الکترونیکی یا وسایل دیگر تنها با اجازه کتبی پژوهشگاه امکان‌پذیر است. نقل قول در حد هزار واژه در انتشارات علمی مانند کتاب و مقاله با درج اطلاعات کامل کتابشناختی، نیازی به مجوز پژوهشگاه ندارد.

صفحه مادرک

بازطراحی و روزآوری سامانه عضویت فراگیر کتابخانه‌ها

مدیر طرح پژوهشی: محمد مهدی آقالویی

نشانی: تهران، پونک، بلوار میرزابابایی، بلوار عدل جنوبی، کوچه مرادی، پلاک ۱۸، واحد ۴

تلفن: ۰۹۱۲۳۲۵۹۸۰۵

رایانامه: Hadi.Louie@gmail.com

همکاران اصلی: محمد هادی آقالویی، کارشناس نرم‌افزار، پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)

علی اصغر حجت‌پناه، کارشناس نرم‌افزار، پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)

مجتبی زالی، کارشناس نرم‌افزار، پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)

ناظر طرح: جناب آقای دکتر سیروس علیدوستی

چکیده

طرح غدیر یا «عضویت فراگیر کتابخانه‌ها»، راهکاری برای ایجاد دسترسی مستقیم استفاده‌کنندگان به منابع کتابخانه‌های دانشگاهی است و براساس موفقیت و تجربه‌ی حاصل از «طرح تعمیم خدمات کتابخانه‌های تخصصی به افراد غیر عضو» طراحی شده است و به عنوان یک طرح دائمی به اجرا در آمده است. هدف اصلی این طرح ایجاد امکان دسترسی مستقیم و حضوری به منابع کتابخانه‌های زیرپوشش وزارت علوم، تحقیقات و فناوری برای اعضای هیات علمی و دانشجویان تحصیلات تکمیلی از طریق یک نظام رسمی و هماهنگ است. طرح حاضر به بازطراحی سامانه‌ی عضویت فراگیر کتابخانه‌ها (طرح غدیر) می‌پردازد.

کلیدواژه‌ها: طرح غدیر، عضویت فراگیر کتابخانه‌ها، مدیریت کتابخانه‌ها، خدمات کتابخانه‌ها

فهرست مطالب

عنوان	شماره صفحه
مقدمه	۱
۱. مشخصات فنی سامانه	۳
۱-۱. طرح غدیر در بستر وب	۳
۲-۱. بازطراحی سامانه غدیر	۴
۳-۱. راهکار پیشنهادی	۴
۴-۱. هدف پژوهش	۵
۵-۱. چارچوب اصلی سامانه	۵
۶-۱. زبان‌های برنامه‌نویسی	۶
۱-۶-۱. C#	۶
۲-۶-۱. Html	۷
۳-۶-۱. Css	۷
۴-۶-۱. Javascript	۸
۵-۶-۱. مفهوم JavaScript	۸
۶-۶-۱. قابلیت JavaScript	۸
۷-۱. تکنولوژی‌های مورد استفاده	۱۰
۱-۷-۱. JQuery	۱۰
۲-۷-۱. AngularJs	۱۰
۳-۷-۱. Bootstrap	۱۱
۴-۷-۱. Linq	۱۱
۵-۷-۱. Entity Framework	۱۲
۶-۷-۱. Stimul Report	۱۲

۱۳.....	Sql Server	۱-۷-۷
۱۴.....	۸- معماری سامانه	
۱۴.....	الگوی توسعه MVC	۱-۸-۱
۱۴.....	اجزای تشکیل دهنده ی MVC	۲-۸-۱
۱۴.....	مزایای برنامه های مبتنی بر MVC	۳-۸-۱
۱۵.....	مزایای برنامه های مبتنی بر وب فرم	۴-۸-۱
۱۵.....	ویژگی های ASP.NET MVC	۵-۸-۱
۱۷.....	۹-۱. ساختار کد سامانه	
۱۸.....	Models	۱-۹-۱
۱۸.....	References	۲-۹-۱
۱۹.....	Infrastructur	۳-۹-۱
۴۵.....	DAL(Data Access Layer)	۱-۹-۴
۴۶.....	تعریف Repository ها	۵-۹-۱
۵۰.....	Irandoc	۶-۹-۱
۵۳.....	Resources	۷-۹-۱
۵۶.....	ViewModels	۸-۹-۱
۵۶.....	GhadirParsa	۹-۹-۱
۶۱.....	۲. نگهداری سامانه طرح غدیر	
۶۱.....	۱-۲. صدور کارت های عضویت	
۶۱.....	۲-۲. پیش ثبت نام	
۶۸.....	۳. استقرار سامانه طرح غدیر	
۶۸.....	۱-۳. سورس سامانه	
۶۹.....	۳-۲. روش باز نمودن سورس	
۷۱.....	۳-۳. پایگاه داده سامانه	

۴-۳. نصب DOTNET FRAMEWORK نسخه ۴,۵,۱.....	۷۳
۵-۳. تنظیم وب سرور.....	۷۳
۶-۳. تنظیم MAIL SERVER برای ارسال رایانامه.....	۷۵
۷-۳. تنظیم ارسال پیامک.....	۷۷
۸-۳. تنظیم پشتیبان گیری از سامانه.....	۷۹
۹-۳. نصب فونت گزارش های سامانه.....	۸۲
منابع.....	۸۳

فهرست جدول‌ها

عنوان	شماره صفحه
جدول ۱-۱ مشخصات کلاس پرویه ایرانداک	۵۰
جدول ۲-۱ RESOURCE های شاخص در سامانه	۵۳
جدول ۳-۱ مشخصه‌های اصلی در VIEWMODEL	۵۴

فهرست شکل‌ها

عنوان	شماره صفحه
شکل ۱-۱ چارچوب اصلی NET.....	۶
شکل ۲-۱ الگوی MVC به همراه فرایندهای حاکم بر آن	۱۶
شکل ۳-۱ بخش‌های ساختاری در پروژه غدیر	۱۷
شکل ۱-۲ فرم پیش ثبت‌نام رابط	۶۲
شکل ۲-۲ گام اول در تکمیل اطلاعات کاربر	۶۳
شکل ۳-۲ گام دوم - بارگذاری تصویر	۶۴
شکل ۴-۲ گام سوم - نمایش کد فاکتور پرداخت	۶۴
شکل ۵-۲ گام چهارم - اتصال به سامانه پرداخت آنلاین	۶۵
شکل ۶-۲ گام پنجم - نمایش تمامی اطلاعات تکمیل شده و تایید نهایی	۶۶

مقدمه

طرح غدیر یا «عضویت فراگیر کتابخانه‌ها»، راهکاری برای ایجاد دسترسی مستقیم استفاده‌کنندگان به منابع کتابخانه‌های دانشگاهی است و براساس موفقیت و تجربه‌ی حاصل از «طرح تعمیم خدمات کتابخانه‌های تخصصی به افراد غیر عضو» طراحی شده است و به عنوان یک طرح دائمی به اجرا در آمده است. هدف اصلی این طرح ایجاد امکان دسترسی مستقیم و حضوری به منابع کتابخانه‌های زیرپوشش وزارت علوم، تحقیقات و فناوری برای اعضای هیات علمی و دانشجویان تحصیلات تکمیلی از طریق یک نظام رسمی و هماهنگ است. طرح حاضر به بازطراحی سامانه‌ی عضویت فراگیر کتابخانه‌ها (طرح غدیر) می‌پردازد.

فصل اول

مشخصات فنی سامانه طرح غدیر

۱. مشخصات فنی سامانه

۱-۱. طرح غدیر در بستر وب

تا کنون در پژوهشگاه تلاشهایی برای مکانیزه کردن فرایند عضویت فراگیر کتابخانه‌ها در بستر وب صورت گرفته است؛ در بهمن‌ماه سال ۱۳۹۱ سامانه‌ی هوشمند اشتراک منابع (سهام) (نرم‌افزار طرح غدیر ۲) با هدف ایجاد دسترسی وب‌محور کاربران به منابع کتابخانه‌های دانشگاهی و تخصصی طراحی و راه‌اندازی شد. اما بازطراحی این سامانه به دلایل زیر با موفقیت همراه نبوده است:

- در این سامانه بازطراحی فرآیندها در بستر وب به درستی انجام نشده است؛
 - به تمامی نیازهای فرایند عضویت فراگیر کتابخانه‌ها در این سامانه پاسخ داده نشده است؛
 - مدل داده‌ای به صورت درست طراحی نشده است؛
 - از فناوری‌های مناسب با توجه به نیازمندی‌های طرح، برای توسعه سامانه استفاده نشده است.
- در گزارشی درباره‌ی شکست نرم‌افزار طرح غدیر ۲ به برخی نتایج ناکارایی این سامانه اشاره شده است:
- نارضایتی دانشگاه‌ها و مؤسسه‌های زیرپوشش، دانشجویان تحصیلات تکمیلی و اعضای هیئت علمی، و نماینده‌های طرح غدیر از بازطراحی فرایندهای منطقی، عملیاتی، و اطلاعاتی طرح؛
 - انصراف دانشگاه‌ها و مؤسسه‌های زیرپوشش، و کتابخانه‌ها از ادامه همکاری با طرح؛
 - عدم همکاری کتابخانه‌های مقصد در امانت کتاب به اعضا؛
 - بی‌نظمی در فرایندهای منطقی، عملیاتی، و اطلاعاتی دفتر اشتراک منابع، دانشگاه‌ها و مؤسسه‌های زیرپوشش، کتابخانه‌ها، نماینده‌های طرح، و ...؛
 - بلا تکلیفی دانشگاه‌ها و مؤسسه‌های متقاضی عضویت در دوره جدید فراخوان طرح؛
 - نبود امکان گزارش‌گیری از عملکرد دانشگاه‌ها و مؤسسه‌ها، کتابخانه‌ها، نماینده‌ها، دانشجویان، و اعضای هیئت علمی در طرح؛

- نبود امکان ارائه گزارش به مقام‌های بالادستی؛
 - راهنماها، پیام‌ها، و اطلاع‌رسانی‌های نادرست، ناکافی، نارسا، درهم، و گنگ برای دانشگاه‌ها، مؤسسه‌ها، کتابخانه‌ها، نماینده‌ها، دانشجویان تحصیلات تکمیلی، و اعضای هیئت علمی؛
 - کاهش آمار پذیرش عضو، و امانت کتاب؛
- بر اساس گزارش دفتر اشتراک منابع پژوهشگاه، آمار عملکرد طرح غدیر- عضویت فراگیر کتابخانه‌ها از تیر ماه سال ۱۳۷۸ تا پایان مهر ماه ۱۳۹۳ به شرح زیر بوده است:

از تیرماه ۱۳۷۸ تا شهریور ماه ۱۳۹۱، امانت بیش از ۱۶۷ هزار جلد کتاب به نزدیک به ۷۳ هزار نفر عضو هیئت علمی و دانشجوی تحصیلات تکمیلی در شبکه‌ای به وسعت ۲۴۰ کتابخانه تخصصی و دانشگاهی کشور بوده است. اما از زمان راه‌اندازی سامانه سهام تا مهر ماه ۱۳۹۳؛ تعداد ۲ هزار و ۲۰۷ نفر پذیرش عضو صورت گرفته و ۸۵۱ جلد هم کتاب امانت داده است که در مقایسه با میانگین سالانه در زمانی که از نرم‌افزار تک‌کاربره موجود در دفتر اشتراک منابع و نظام کاملاً دستی و کاغذی بهره گرفته می‌شد؛ کاهش تعداد کتابخانه‌های زیرپوشش، کاهش میانگین سالانه پذیرش عضو، و هم چنین کاهش میانگین سالانه تعداد کتاب امانت داده شده را شاهد هستیم.

۲-۱. بازطراحی سامانه غدیر

متأسفانه روش نامناسب توسعه‌ی سامانه‌ی سهام (سامانه‌ی تحت وب)، عدم طراحی مناسب فرآیندها، نبود مستندات کافی و عدم رعایت اصول مستندسازی در کد برنامه، امکان تغییر و بهبود سامانه را غیرعملی کرده است به گونه‌ای که به نظر می‌رسد صرف وقت بیشتر در این زمینه اتلاف زمان و نیروی انسانی خواهد بود. این مسائل در کنار به وجود آمدن نیازمندی‌های تازه و همچنین توسعه چشمگیر فناوری‌های تحت وب در طول سالهای اخیر نیاز به بازطراحی آن در سطح وب را نشان می‌دهد.

۳-۱. راهکار پیشنهادی

راهکار پیشنهادی طرح حاضر، بازطراحی سامانه‌ی غدیر با استفاده از از فناوری‌های متن‌باز مایکروسافت از جمله MVC۵ و در قالب سه گام اصلی، یعنی پشتیبانی و عضویت، استفاده اعضا از خدمات کتابخانه‌ای و در نهایت تصفیه حساب و آمار است.

یکی از نکات مهم در بازطراحی سیستم‌های تک‌کاربره به سامانه‌های تحت وب و تعاملی با کاربر، لزوم اعمال تغییرات در فرم‌ها، نحوه انجام فرایندها و یا حذف برخی فرایندها، تغییر ماهیت برخی خدمات و تغییر شیوه‌نامه‌هاست. بر این اساس لازم است که همه این موارد با همکاری تنگاتنگ میان مالک و بهره‌بردار سامانه و گروه توسعه‌دهنده صورت گیرد. بر این اساس، بازطراحی و توسعه سامانه غدیر بر مبنای رکن اصلی یعنی همکاری تیم توسعه‌دهنده با دفتر اشتراک منابع صورت خواهد گرفت و اقلام فرمهای موجود در سامانه، نقشها و دسترسی‌های مورد نیاز، انواع گزارشهای مورد نیاز در سامانه، ساختارهای موجود مانند انواع منابع و پایگاه‌ها و ... همگی با هماهنگی و بر اساس نظر بهره‌بردار تعیین و پیاده‌سازی می‌شود.

۱-۴. هدف پژوهش

هدف اصلی این پیشنهادیهی بازطراحی سامانهی اشتراک منابع غدیر است که امکان دسترسی مستقیم و حضوری به منابع کتابخانه‌های زیر پوشش طرح برای استفادهی دانشگاهیان و پژوهشگران کشور را از طریق یک نظام رسمی و هماهنگ فراهم می‌کند. اهداف زیر در راستای بازطراحی سامانهی غدیر دنبال خواهد شد:

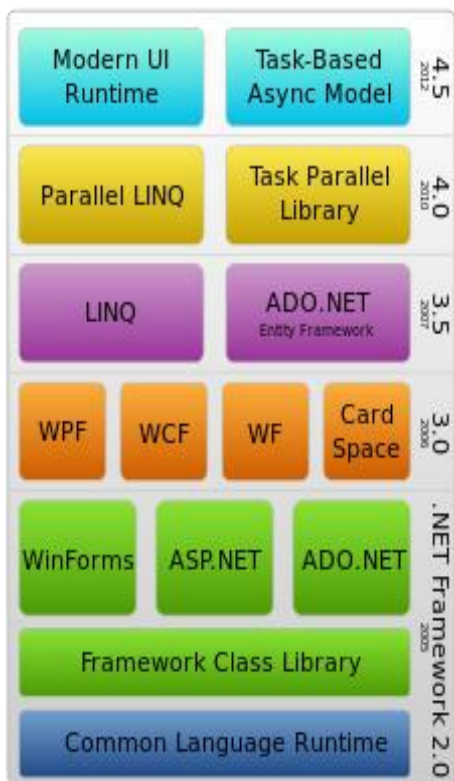
- طراحی و پیاده‌سازی مناسب و کاربرپسند طرح غدیر در بستر وب؛
- شناسایی نیازمندی‌های جدید بهره‌وران و کاربران سامانه؛
- طراحی فرآیندهای مناسب برای تعمیم و گسترش مراکز زیر پوشش به سایر سازمان‌ها؛
- استفاده از اطلاعات پایه‌ی استاندارد پژوهشگاه علوم و فناوری اطلاعات برای گزارش‌گیری‌های تلفیقی با دیگر سامانه‌ها.

۱-۵. چارچوب اصلی سامانه

چارچوب دات‌نت یا همان (به انگلیسی .NET Framework): یک فناوری نرم‌افزاری است که بر روی تمامی ویرایش‌های سیستم‌عامل ویندوز مایکروسافت قابل اجراست و در سیستم‌عامل‌های دیگر از جمله لینوکس و مکینتاش نیز وارد شده است.

این چارچوب شامل مجموعه‌ای از زبانهای برنامه‌نویسی است که سی شارپ و ویژوال بیسیک مهمترین آنها می‌باشند. مجموعه‌ای از کتابخانه‌های بسیار غنی جهت کمک به سهولت توسعه نرم‌افزار در این چارچوب قرار گرفته‌اند که در قالب بخش‌های عمده‌ای همچون فناوری‌های ASP.NET ، ADO.NET و بسیاری فناوری‌های خاص دیگر ارائه می‌شوند که تعداد آنها در نسخه‌های اخیر همگام با محدود تر شدن اهداف مایکروسافت بیشتر شده است.

مجموعه‌ای فناوری‌های موجود در این چارچوب در شکل زیر آمده است. این چارچوب از سال ۲۰۰۰ شروع به توسعه شد و تا حال به نسخه ۴,۶ .Net Framework رسیده است.



شکل ۱-۱ چارچوب اصلی .Net.

۱-۶. زبان های برنامه نویسی

در طراحی و پیاده سازی سامانه عضویت فراگیر کتابخانه ها از زبان های برنامه نویسی زیر استفاده شده است:

۱-۶-۱. C#:

سی شارپ به انگلیسی C# زبانی شیء گرا و سطح بالا از خانواده زبان های چارچوب دات نت شرکت مایکروسافت است. زبان سی شارپ، یک زبان برنامه نویسی چند الگویی و منظم شده مدل های تابعی، امری، عمومی، شیء گرا و جز گرا و در بستر دات نت می باشد. این زبان توسط مایکروسافت و جزئی از دات نت به وجود آمد و بعداً استانداردهای ECMA و ISO را نیز در بر گرفت. سی شارپ یکی از ۴۴ زبان برنامه نویسی است که توسط زمان اجرای زبان مشترک از چارچوب دات نت پشتیبانی می شوند و در همه جا به وسیله مایکروسافت ویژوال استودیو شناخته می شود. زبان سی شارپ با قدرت و در عین حال سطح بالایی خود توانسته توجه بسیاری از برنامه نویسان را به خود جلب کند.

این زبان برپایه سادگی، مدرن بودن، همه منظوره و شیء گرا بودن ساخته شد. آندرس هجلزبرگ، طراح زبان برنامه نویسی دلفی، سرپرستی تیم طراحان زبان سی شارپ را بر عهده داشت. این زبان دارای دستوری شیء گرا مشابه ++C است و به

شدت از زبان های جاوا و دلفینیاژمند مدرک تأثیر پذیرفته است. در ابتدا نام این زبان COOL بود که مخفف C like Object Oriented Language بود، هر چند در ژوئیه ۲۰۰۰، زمانی که مایکروسافت پروژه را عمومی اعلام کرد، اسم آن به سی شارپ تغییر پیدا کرد. آخرین نسخه آن نسخه ۵٫۰ است که همزمان با دات نت ۴٫۵ در اوت ۲۰۱۲ منتشر شد. در این سامانه برای توسعه سمت سرور (Back End) از این زبان برنامه نویسی استفاده شده است. این زبان در بطن چارچوب Net. قرار دارد.

۱-۶-۲. Html

عبارت HTML (اچ تی ام ال) مخفف Hyper Text Markup Language به معنی زبان نشانه گذاری فوق متن است. در حقیقت Html زبان استاندارد طراحی صفحات وب است و کلیه کدهای صفحه اعم از طرف سرور و طرف مشتری در نهایت به کدهای HTML تبدیل شده و توسط مرورگر نمایش داده می شوند.

به عبارت دیگر مرورگرها هیچکدام از کدها و کنترل های سمت سرور همچون کدهای asp و php را نمی شناسند و کد قابل فهم برای آنها اچ تی ام ال می باشد. کامپایلر های زبان های برنامه نویسی سروری در نهایت کد های خود را برای نمایش به کد اچ تی ام ال تبدیل میکنند و برای مرورگر میفرستند تا به کاربران نمایش داده شود.

زبان HTML یک زبان نشانه گذاری است، به این معنی که بخش های مختلف توسط اجزایی به نام تگ از هم جدا شده، که هر کدام دارای کاربرد و خواص مربوط خود هستند. این تگ ها به مرورگر اعلام می کنند که هر بخش از صفحه چه نوع عنصری است و باید به چه صورت نمایش داده شود.

در یک صفحه HTML می توان انواع عناصر از قبیل متن، تیتیر، عکس، جدول و ... را قرار داد، که برای هر عنصر باید از تگ مربوط به آن استفاده کرد. صفحات HTML فقط از کد ها که به صورت متن هستند تشکیل شده اند. بدین معنا که برای تصویر کد مربوط به تمایش تصویر و جدول و ... کدهای اچ تی ام ال مربوط به هر یک را باید نوشت و مرورگر با رسیدن به این کد ها و تگ ها، المنت های مرتبط با آن را نمایش می دهد.

هر یک از کدهای html، معنا و مفهوم خاصی دارند و تأثیر مشخصی بر محتوا می گذارند. مثلاً برچسب هایی برای تغییر شکل ظاهری متن، نظیر درشت و ضخیم کردن یک کلمه یا برقراری پیوند به صفحات دیگر در اچ تی ام ال تعریف شده اند.

۱-۶-۳. Css

الگوهای آبخاری یا روی انداز آبخاری سبک یا سی اس اس (به انگلیسی CSS: Cascading Style Sheets) در کنار اچ تی ام ال هسته فناوری ساخت صفحه های وب هستند. سی اس اس روشی ساده برای نمایش چیدمان و جلوه های تصویری (مانند نوع قلم، رنگ و اندازه ها) بر صفحه های وب است. الگوهای آبخاری از جنس زبان های نشانه گذاری، با ساختار متن ساده رایانه هستند و درون هر کدام، دستورهای آبخار مانند و پی در پی، برای چگونگی نمایش هر صفحه وب افزوده می شود. به گفته ای ساده تر، این دستورها روش نشان داده شدن قلم ها و اندازه شان، رنگ ها و پس زمینه ها، روش چیدمان موزاییک های دربرگیرنده داده ها (دیواره ها)، و بسیاری دیگر از عنصرهای ساختار هر صفحه وب را، درون خود جای می دهند.

کاربرد اصلی CSS چیست ؟ بیشترین استفاده ای که از این زبان در حال حاضر می شود مشخص کردن سبک صفحه وب اچ تی ام ال و اکس اچ تی ام ال است ولی آن را می توان بر هر نوع مستند اکس ام ال از جمله اس وی جی و اکس یوال اعمال کرد. نگهداری و تغییر مشخصات سی اس اس به عهده کنسرسیوم وب جهانگستر است.

یکی از مهمترین نکته ها برای طراحی سایت ، استفاده بهینه از CSS میباشد در این بخش از آموزش CSS به شرح ویژگی های اساسی و مهم در طراحی وب سایت می پردازیم و روشهای اضافه کردن CSS به صفحه را مورد بررسی قرار می دهیم.

سه روش برای اضافه کردن CSS به صفحه وجود دارد:

- CSS خارجی
- CSS داخلی
- سبک درون خطی

برای آسان کردن کار موتور های جستجو سعی کنید از انیمیشن های فلش و یا تگ های span بوسیله CSS ، کمتر استفاده کنید. با استفاده از CSS از دوباره نویسی کدهای HTML که باعث کند شدن لود صفحه می شود جلوگیری میکنیم.

۱-۶-۴. Javascript

در اولین روزهای تاسیس مجمع جهانی وب (Wide Web World) دستورات HTML بسیار ساده و آسان بود، بطوریکه فراگیری فرامین مورد نیاز بیشتر از چند روز طول نمیکشید ولی هنگامی که Web ، شروع به پیشرفت نمود، طراحان خواستار کنترل بیشتری شده و در نتیجه مجموعه دستورات مورد نیاز تهیه گردید.

از آنجایی که صفحات وب ، بهتر است بصورت دینامیک و متحرک باشند، طراحان نیز خواستار ارتباط بین بینندگان سایت و صاحبان آن شدند، در نتیجه استفاده از دستورات HTML به تنهایی نیاز آنها را برطرف نکرد و سپس شرکت Netscape برای ایجاد و کنترل ارتباط بین صاحبان سایت و بینندگان آن در صفحات وب از JavaScript کمک گرفت.

۱-۶-۵. مفهوم JavaScript

جاوا اسکریپت دقیقاً یک زبان برنامه نویسی میباشد که بوسیله آن میتوان بین کاربر و سایت ارتباط برقرار نمود. توجه داشته باشید که استفاده از JavaScript در صفحات Web نیازی به نوشتن اسکریپتها نداشته و میتوانید اسکریپتهای موجود در شبکه اینترنت را کپی و در مکان مورد نظر با ایجاد تغییرات مورد نیاز استفاده کنید.

۱-۶-۶. قابلیت JavaScript

بوسیله جاوا اسکریپت میتوان قابلیتهای بیشماری از قبیل ایجاد ارتباط بین کاربر و سایت به صفحه وب اضافه نمود. مثلاً سایتهایی را مشاهده کرده اید که با قرار گرفتن نشانگر ماوس بر روی دکمه مورد نظر بصورتی متفاوت (برجسته ، تغییر

محتویات درون دکمه) نمایش داده می‌شود، این عمل توسط جاوااسکریپت انجام گرفته است که به آن rollover گفته می‌شود.

بوسیله rollover میتوان حالتی ایجاد نمود تا اطلاعات وارد شده توسط کاربر درون فرمها بطور صحیح باشند و در نتیجه در زمان و هزینه مربوط بتوان صرفه جویی نمود. میتوان محاسبات لازم و مورد نیاز عددی را بدین وسیله در دستگاه کاربر و بدون نیاز به هیچ گونه فرآیندی در server ها انجام داد. تفاوت بین برنامه های server-side و client-side بدین گونه است که در برنامه server-side برنامه توسط CGI در خود سرورها اجرا میشوند مانند برنامه ASP ، ولی در برنامه های client-side برنامه در دستگاه کاربر اجرا خواهد شد.

از دیگر فرمتها و قابلیتهای JavaScript میتوان، صفحات HTML را بر اساس عمل کاربر نمایش داد. فرض کنید که یک سایت آژانس مسافرتی را باز کرده و مقصد خود را هاوایی انتخاب کنید، بوسیله اسکریپت میتوان آخرین اطلاعات مسافرتی مورد نیاز برای هاوایی را در صفحه ای جدید مشاهده نمایید.

بوسیله این برنامه میتوان بر Browser کنترل انجام داد و پنجره های جدیدی را باز نمود، جعبه های پیغام را به کاربر اعلام نمود و بر روی قسمت status bar پنجره مرورگر پیغام مخصوصی را نمایش داد.

همچنین بخاطر وجود قابلیت های ویژه ای که در این نوع برنامه نویسی وجود دارد میتوان ساعت، تقویم و هر برنامه دیگری نیز تهیه کرد.

برخلاف شباهت اسمی موجود برنامه Java و JavaScript ، هیچگونه شباهتی بین این دو برنامه وجود ندارد. جاوا یک زبان برنامه نویسی بسیار کامل پیشرفته ای است که توسط شرکت sun microsystem تهیه شده و با استفاده از آنکه نسل بعدی برنامه های C و ++C میباشد، میتوان تمامی برنامه مورد نیاز را نوشته و در ضمن قطعات الکترونیکی استفاده کننده را نیز کنترل نمود. از خاصیت های دیگر جاوا اجرای برنامه نوشته شده توسط آن در تمامی مکانها با هرگونه سیستم عامل از ویندوز، یونیکس و مک میباشد.

اساس جاوا در برنامه های client-side برای ایجاد applet ها میباشد، این برنامه های کوچک توسط شبکه اینترنت دریافت شده و درون خود browser ها اجرا میشوند بطوریکه به علت وجود قابلیت همه گیر جاوا این برنامه در تمامی مرورگرهایی که قابلیت اجرای جاوا را دارند، کار میکنند.

در سرتاسر این سامانه به صورت گسترده از کدهای جاوا اسکریپت استفاده شده است. این زبان به همراه فریمورک JQuery و Angular بخش اصلی کدهای سمت کلاینت (Client Side) را تشکیل می‌دهد.

۱-۷. تکنولوژی های مورد استفاده

در سامانه عضویت فراگیر کتابخانه ها از تکنولوژی های زیر برای توسعه و پیاده سازی استفاده شده است:

۱-۷-۱. JQuery

به طور کلی JQuery یک کتابخانه جاوااسکریپتی سبک و سریع است که شعارش "کمتر بنویس و بیشتر انجام بده" و هدفش استفاده آسان و سریع از امکانات جاوااسکریپت در وب سایت است. برای یادگیری JQuery باید اطلاعات پایه ای در مورد HTML، CSS و JavaScript داشته باشید. جی کوثری کارهایی که نیاز به کدنویسی های فراوان جاوااسکریپت دارد را به گونه ای ساده می کند که می توانید با یک خط کد همه آنها را انجام دهید.

زبان برنامه نویسی JavaScript، یک زبان برنامه نویسی خیلی پیچیده و گسترده است و یادگیری کامل آن به زمان و حوصله خیلی زیادی نیاز دارد. در حقیقت JQuery فریم ورکی است که بر اساس زبان برنامه نویسی جاوا اسکریپت ایجاد شده است. به زبان ساده می توان گفت، برنامه نویس با استفاده از زبان برنامه نویسی جاوا اسکریپت، یک سری قانده و قانون تعریف کرده است و یک دستور زبان جدید ساخته که شما با آن قواعد یه کد به مراتب بسیار ساده تر می نویسید، سپس آن فایل فریم ورک دستورات شما را به دستورات استاندارد جاوا اسکریپت ترجمه می کند و سپس آن ها را اجرا می کند.

۱-۷-۲. AngularJs

پس از ظهور فن آوری AJAX شاهد پیدایش Web Application های بسیاری بودیم. بسیاری از برنامه نویسان اقدام به تولید وب سایتها و وب اپلیکیشنهای گوناگون با حال و هوای نرم افزارهای تحت ویندوز و یا دیگر سیستم های عامل نمودند. منظورم از حال و هوا refresh نشدن صفحات و یا عدم استفاده از post back ها میباشد که برای کاربران وب تجربه جدیدی بود و دیگر با هر کلیک نیاز نبود زمان زیادی برای لود شدن مجدد محتوا در بروزر صرف کنند.

اتفاقی که پشت پرده روی میداد این بود که بروزرها تنها بخشهایی از صفحه که نیاز به بروزرسانی داشتند را از سرور درخواست میکرد و پس از دریافت با استفاده از javascript محتویات صفحه را آپدیت میکردند. در ابتدا بخشهای مورد نیاز کلاینت در سرور به HTML رندر میشدند و بعد برای کلاینت فرستاده میشدند. در گام بعدی برنامه نویسان بجای اینکه محتوا را در سرور به HTML تبدیل کنند داده خام را تنها از سرور میگرفتند و در سمت کلاینت اقدام به تولید HTML با استفاده از داده دریافتی میکردند.

این روش بسیار ساده و کارآمد میباشد اما برای صفحات ساده و یا تنها بخشی از صفحات. هر چه پیچیدگی صفحات وب و منطق نرم افزاری آنها پیچیده تر میشد پیاده سازی این صفحات با استفاده از AJAX به مراتب زمانبر تر و پیچیده تر از پیاده سازی آنها به صورت سنتی میشد. بسیاری از پروژه های Single Page Application یا SPA یا برنامه های تک صفحه ای (که در آنها عملاً تنها یک صفحه توسط کلاینت لود می شود و محتوای آن یک صفحه به مرور با تعاملات گوناگون کاربر به روز رسانی میشد) به علت پیچیدگی کد سمت کلاینت و قابلیت تغییر پذیری و آپگرید پایین با شکست مواجه شدند. تمامی این دلایل باعث شد که برنامه نویسان به دنبال یک معماری جدید در سمت کلاینت باشند.

۱-۷-۳. Bootstrap

Bootstrap مجموعه ای از ابزارهای رایگان برای ایجاد صفحات وب و نرم افزارهای تحت وب است که شامل دستورات HTML، CSS و توابع جاوا اسکریپت جهت تولید و نمایش فرم ها، دکمه ها، تب ها، ستون ها و سایر المان های مورد نیاز طراحی وب می باشد.

Bootstrap در ابتدا توسط مارک اتو و جاکوب تورنتون و در جهت ایجاد یک چارچوب ظاهری مشخص و یکسان در ابزارهای تویتر طراحی و نوشته شد. قبل از شروع این پروژه نمونه های زیادی با همین رویکرد ایجاد شده بود که همگی با سرنوشتی مشابه و عدم استقبال طراحان وب دنیا مواجه شده بودند. به دلیل وجود مشکلات اساسی در نمونه های دیگر، سازنده اصلی تویتر یا همان مارک اتو تصمیم به ساخت یک سیستم داخلی و قدرتمند برای خود با نام Bootstrap گرفت.

این چارچوب یا Framework با تمامی مرورگرهای استاندارد همخوانی داشته و حتی در نسخه های قدیمی تر مانند اینترنت اکسپلورر ۸ نیز ظاهر زیبای خود را حفظ میکند. از نسخه دوم Bootstrap به بعد طراحی واکنشگرا یا رسپانسیو نیز در آن لحاظ شد که موجب نمایش مناسب صفحه در تلفن های هوشمند و تبلت ها میگردد. Bootstrap متن باز بوده و در سایت Github برای دانلود رایگان قرار گرفته است. امکان همکاری در این پروژه و شخصی سازی امکانات و ظاهر آن فراهم شده است و تاکنون مستندات مربوط به آن به چندین زبان دنیا ترجمه شده است.

۱-۷-۴. Linq

LINQ مخفف Language-Integrated Query میباشد که به مجموعه فن آوریهای گفته می شود که قابلیت query یا جستار بر روی انواع مجموعه های داده ای به C# و یا دیگر زبانهای برنامه نویسی دات نت اضافه میکند.

به عبارتی LINQ بتکنیکی است که از آن برای استخراج داده از هر Object ای که اینترفیس IEnumerable را پیاده سازی کرده باشد استفاده می شود. در LINQ آرایه ها، collection ها، پایگاه های داده و فایل های XML همگی منابع اطلاعاتی یا data source هایی هستند که میتوانیم در آنها اقدام به جستجو، فیلتر و استخراج داده با دستورات زبان C# و یا VB نماییم. در کنفرانس برنامه نویسان مایکروسافت سال ۲۰۰۵ Anders Hejlsberg و تیمش LINQ را معرفی کردند و در دات نت فریم ورک ۳٫۵ منتشر شد. LINQ در بطن زبانهای C# و VB قرار گرفته است و در محیط Visual Studio قابلیت رنگبندی کد و IntelliSense برای آن فعال هست همچنین شما میتوانید query های خود را debug کنید. مدت زمان نوشتن query با استفاده از LINQ نسبت به دیگر متدهای قدیمی بسیار کمتر میباشد.

با استفاده از LINQ به آسانی میتوانید رابطه بین جداول پایگاه داده را ببینید که این قابلیت زمانی که شما جداول مختلف را باید join کنید به شما کمک فراوانی میکند. قابلیت Transform در LINQ به شما کمک میکند که به سادگی انواع داده ای را به یکدیگر تبدیل کنید به طور مثال خیلی ساده میتوانید داده SQL را به XML تبدیل کنید.

۱-۷-۵. Entity Framework

Entity Framework یک ORM است که توسط شرکت Microsoft توسعه داده می‌شود. نسخه ۱٫۰ آن مایوس کننده بود و در موارد بسیاری کمبود داشت. اما با توجه به سرمایه‌گذاری زیادی که Microsoft روی پروژه انجام داده بود، مشخص بود که قصد جدی در توسعه این ابزار دارد. نسخه کنونی آن، ۶ هست که با ۲۰۱۵ Visual Studio روی سیستم نصب می‌شود. با توجه به اهمیت این بحث برای Microsoft، انتظار می‌رود که توسعه این ابزار به طور فعالی ادامه پیدا کند و امکانات کاملی به آن اضافه شود.

باید توجه داشت که ORM یا Object Relational Mapping ابزاری است که به وسیله آن می‌توان جداول پایگاه داده را پردازش کرد و آنها را به فرمت شیء نگاشت کرد که در کدنویسی خوانا تر باشند. در کدنویسی به جای این که به صورت مستقیم با جداول پایگاه داده کار کنیم بر روی اشیاء کار می‌کنیم. به این اشیاء موجودیت یا Entity گفته می‌شود.

ADO.NET Entity Framework به گونه‌ای طراحی شده است تا توسعه دهندگان را قادر سازد در مقابل مدل‌های مرسوم مفهومی (conceptual) از یک مدل جدید با امکان دسترسی و برنامه نویسی مستقیم بر اساس ساختار یک پایگاه داده رابطه‌ای استفاده نمایند. هدف از تولید این مدل، کاهش چشمگیر میزان کدنویسی، سهولت گسترش، ترمیم، بهبود و نگهداری برنامه‌ها بوده است.

دسترسی به داده‌ها را ساده می‌سازد و به ما اجازه می‌دهد تا مدل مفهومی مورد نیاز برای کسب و کار را ایجاد کنیم. Entity Framework اجازه می‌دهد تا شما روابط چندبه چند (n: n) را بدون join ها متعدد مدیریت نمایید. جهت کاهش پیچیدگی کد، جداول مرتبط را در یک Entity مدل می‌کند. پشتیبانی غنی و خوب از ارث‌بری‌ها برای جدول به ازای سلسله مراتب، زیرکلاس (subclass) و تایپ‌های به هم چسبیده (concrete type).

چون دارای مکانیسم بسیار قوی Query گیری از Entity هاست. کسانی که مایل به استفاده از LINQ نیستند، میتوانند از Entity Client provider همراه با EntitySQL استفاده نمایند. مستقل از نوع پایگاه داده می‌باشد و بر اساس استاندارد ADO.NET Providers کار می‌کند.

۱-۷-۶. Stimul Report

نرم افزاری است جهت تهیه گزارشات پیچیده بدون نیاز به دانش فنی و قابل استفاده برای افراد غیر متخصص. این سامانه امکان یک پارچه سازی بانک‌های اطلاعاتی مختلف جهت تهیه گزارشات مدیریتی و تصمیم‌سازی را به وجود می‌آورد. در تهیه گزارش‌های سامانه عضویت فراگیر کتابخانه‌ها از این تکنولوژی گزارش‌گیری استفاده شده است. این تکنولوژی به راحتی به منابع داده متصل می‌شود و فراداده لازم را در محیطی کاملاً User friendly به کاربر نشان می‌دهد.

۱-۲-۷. Sql Server

پایگاه داده MS SQL Server یکی از سیستم‌های مدیریت بانک‌های اطلاعاتی رابطه‌ای (Relational) است که توسط شرکت مایکروسافت ارائه شده است. این پایگاه داده از مدل سرویس دهنده - سرویس گیرنده (Client/Server) تبعیت می‌نماید. در این مدل، درخواست‌های (InQuery) سرویس گیرندگان برای سرویس دهنده ارسال و در سمت سرویس دهنده بررسی و آنالیز می‌گردند. در ادامه، پردازش‌های مورد نیاز بر روی اطلاعات ذخیره شده در بانک‌های اطلاعاتی انجام و در نهایت، نتایج برای سرویس گیرنده ارسال خواهد شد. این پایگاه داده با استفاده از مجموعه عناصری (Components) که به صورت هدفمند اجراء می‌گردند، قادر به تامین نیازها و درخواست‌ها از مخازن داده (Data Storages) می‌باشد.

سامانه عضویت فراگیر کتابخانه‌ها از طریق تکنولوژی Entity Framework در .Net. از امکانات Sql Server استفاده می‌کند. در واقع این تکنولوژی استفاده از دستورات Sql را از طریق زبان Linq در خود کد سامانه امکان پذیر می‌کند.

۸-۱. معماری سامانه

۸-۱-۱. الگوی توسعه MVC

الگوی MVC مخفف سه کلمه Model (مدل) و View (نمایشگر) و controller (کنترلگر) است. برخی از برنامه نویسان، همچنان از ASP.NET که بر مبنای فرم های وب و Postback است، استفاده می کنند، برخی از ویژگی های MVC سود می برند و بعضی ها هم دو پلت فرم را ترکیب می کنند و این موضوع بیانگر این است که هیچکدام از پلت فرم ها ناقص یکدیگر نیستند. در واقع MVC بر روی معماری های چند لایه ای جهت تفکیک بخش های مختلف برنامه (بخش های منطقی برنامه مانند داده ها، مجوزها، کنترل صحت داده ها و لایه های مرتبط با کاربر نهایی) قرار می گیرد.

۸-۱-۲. اجزای تشکیل دهنده ی MVC

- **Model (مدل):** قسمتی از برنامه کاربردی است که مسئول بازیابی داده از بانک اطلاعاتی، ذخیره آن، تبدیل آن به شی یا آبجکت ها و پیاده سازی منطق برنامه برای داده های دامنه ی مسئله است. در حقیقت بار اصلی معماری MVC بر عهده ی این بخش است. مثلاً یک آبجکت Product ممکن است اطلاعات را از بانک اطلاعاتی بازیابی کرده، بر روی آنها عملیاتی را انجام دهد و سرانجام نتیجه را در بانک اطلاعاتی یا در جدول Products ذخیره کند.
- **View (نمایشگر):** اجزایی از برنامه است که واسط کاربری برنامه (UI) را می سازد. معمولاً این UI از داده های مدل ساخته می شود. در واقع نقطه پایان برنامه کاربردی است. به کاربر نتایج عملیات و بازیابی و نمایش داده از طریق برقراری ارتباط با دو بخش دیگر؛ یعنی مدل و کنترلگر را نشان می دهد. برای مثال، هنگامی که کاربر در فرم ورود به سیستم رمز عبور خود را وارد می کند، اکثر برنامه نویسان در همان فرم اقدام به چک کردن رمز عبور می کنند که این عمل مغایر با قوانین MVC است. در MVC هنگامی که کاربر رمز عبور را کرد، رمز عبور بدون هیچگونه اعمالی به بخش های دیگر فرستاده می شود و فقط یک نتیجه ساده یا خبر از بخش های دیگر دریافت می کند که از طریق آن اجازه ی ورود به برنامه داده می شود.
- **Controller (کنترلگر):** اجزایی از برنامه هستند که مدیریت تعامل با کاربر را بر عهده دارند. می توان گفت که واسط بین مدل و نمایشگر می باشند؛ یعنی با مدل کار می کند و در انتها نمایشگری را برای نشان دادن واسط کاربری انتخاب می کند. ورودی کاربر را مدیریت کرده و به آنها پاسخ می دهد و با کاربر تعامل می کند. برای مثال، کنترلگر عبارت های پرس و جوی بانک اطلاعاتی را مدیریت کرده و آن ها را به مدل ارسال می کند، وظیفه اجرای پرس و جوها با مدل است.

۸-۱-۳. مزایای برنامه های مبتنی بر MVC

- با تقسیم یک برنامه به سه قسمت مدل، نمایشگر و کنترلگر، مدیریت برنامه یا پروژه را ساده تر می کند.
- از ViewState و فرم های سروری استفاده نمی کند و از این نظر برای برنامه نویسانی که تسلط کامل بر رفتار برنامه را می خواهند عالی است.

- از الگوی کنترلر جلو استفاده می کند که درخواست های برنامه را توسط یک کنترلر پردازش می کند. این مسئله باعث می شود تا بتوانیم برنامه هایی را طراحی کنیم که از زیر ساخت های غنی مسیر یابی پشتیبانی می کند.
- پشتیبانی بهتری از طراحی و توسعه آزمون محور دارد.
- برای برنامه های پشتیبانی شده توسط تیم های بزرگ برنامه نویسان که کنترل بسیار بر رفتار برنامه را می خواهند، بهتر کار می کنند.

۱-۸-۴. مزایای برنامه های مبتنی بر وب فرم

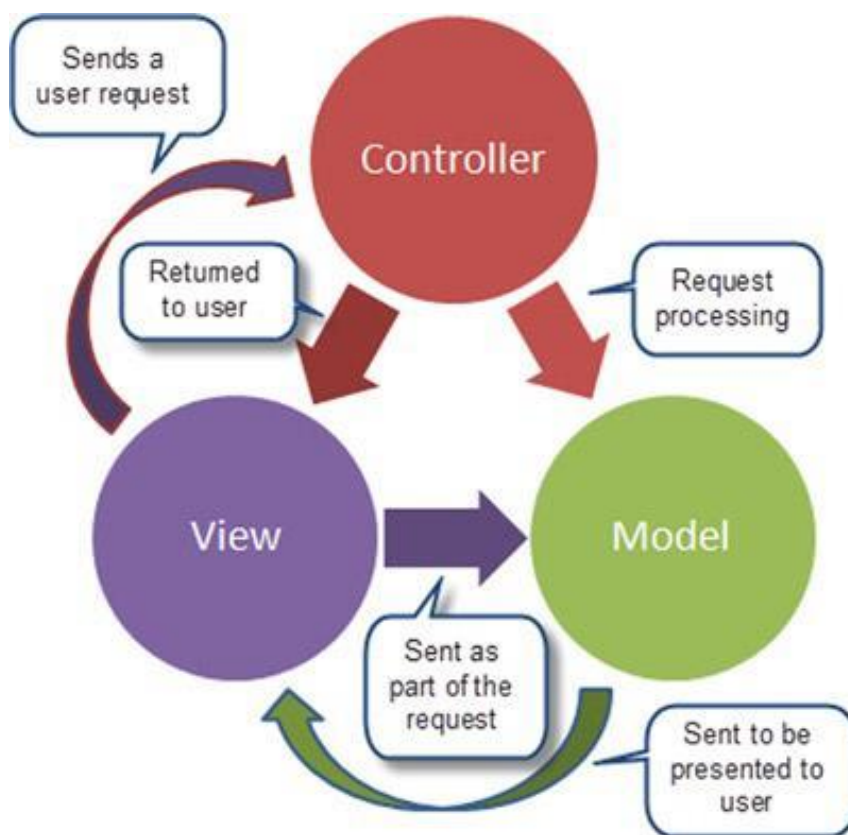
- از مدل رویداد استفاده می کند که وضعیت را روی HTTP حفظ می کند. این روش برای برنامه های وبی با منطق یک خطی مناسب است.
- از الگوی کنترلر صفحه استفاده می کند که به هر صفحه کارایی تابعی می دهد.
- از View State و فرم های سروری استفاده می کند که مدیریت اطلاعات وضعیت را ساده تر می کند.
- برای تیم های کوچک برنامه نویسی که می خواهند با استفاده از صدها کنترل موجود برای ساخت سریع برنامه وبی استفاده کنند، مناسب است.
- عموماً برای توسعه وب ساده تر است. زیرا اجرای آن (کلاس Page، کنترل ها و ...) متمرکز شده اند و معمولاً کد کمتری نسبت به MVC نیاز دارد.

۱-۸-۵. ویژگی های ASP.NET MVC

- جداسازی وظایف برنامه کاربردی (منطق ورودی، منطق کاری و منطق واسط کاربری) و امکان تست کردن برنامه و طراحی و توسعه آزمون محور به صورت پیش فرض. تمام ارتباطات اصلی در MVC بر مبنای واسط است و می توان آن ها را با اشیای ساختگی Mick تست کرد. می توانید ک. نترلرها را بدون اینکه آنها را در ASP.NET اجرا کنید، تست کنید و این باعث افزایش سرعت و انعطاف تست برنامه می شود. می توانید از هر فریم ورک تست که با فریم ورک Net. منطبق است برای این کار استفاده کنید.
- فریم ورکی توسعه پذیر و قابل اتصال. اجزای اصلی ASP.NET MVC به گونه ای طراحی شده اند که به راحتی جایگزین یا سفارشی شوند. می توانید به سادگی پیاده سازی های خودتان را جایگزین موتور نمایشگر، سیاست های آدرس، سریالی کردن پارامترهای متدهای عملیات یا سایر اجزا کنید. همچنین ASP.NET MVC از فریم ورک های تزریق وابستگی و وارونگی کنترل پشتیبانی می کند. به شما امکان می دهد به جای این که اشیاء را توسط کلاس ها بسازید، آنها را از جایی بیرون از کد مثل فایل های تنظیمات بگیرید و این مسئله تست برنامه را ساده تر می کند.
- نگاشت های قوی آدرس ها که امکان ساخت برنامه هایی با آدرس های معنی و جستجو شدنی (قابل جستجو توسط موتورهای جستجوگر) را می دهد و نیازی نیست که آدرس ها پسوند فایل داشته باشند، آنها به گونه ای

طراحی شده اند که از الگوهای نامگذاری قابل فهم برای موتورهای جستجو SEO و آدرس های انتقال وضعیت قابل نمایش REST پیروی کنند.

- پشتیبانی برای استفاده از نشانه گذاری در صفحات ASP.NET (فایل های .aspx). کنترل های کاربر (فایل های .ascx) صفحات Master (با پسوند .master) به عنوان قالب های نمایشگر. شما می توانید از برخی ویژگی های موجود ASP.NET از قبیل صفحات master تو در تو، اسکریپت درونی (<./=>)، کنترل های سرور اعلان شده، قالب، انقیاد داده ها، محلی سازی و ... در فریم ورک ASP.NET MVC استفاده کنید.
- پشتیبانی از ویژگی های موجود ASP.NET MVC : ASP.NET به ما این امکان را می دهد تا از ویژگی هایی مانند مجوز فرم و تشخیص هویت ویندوز، مجوز آدرس ها، عضویت و نقش ها، خروجی و کش کردن داده ها و Session و مدیریت وضعیت پروفایل، مانیتور کردن صحیح، سیستم پیکر بندی و معماری عرضه کننده استفاده کنید.



شکل ۱-۲ الگوی MVC به همراه فرایندهای حاکم بر آن

۹-۱. ساختار کد سامانه

پایاده‌سازی سامانه عضویت فراگیر کتابخانه‌ها در ۶ پروژه به هم متصل صورت پذیرفته است که در شکل ۳-۱ نمایش داده شده است:

۱. DAL (Data Access Layer)

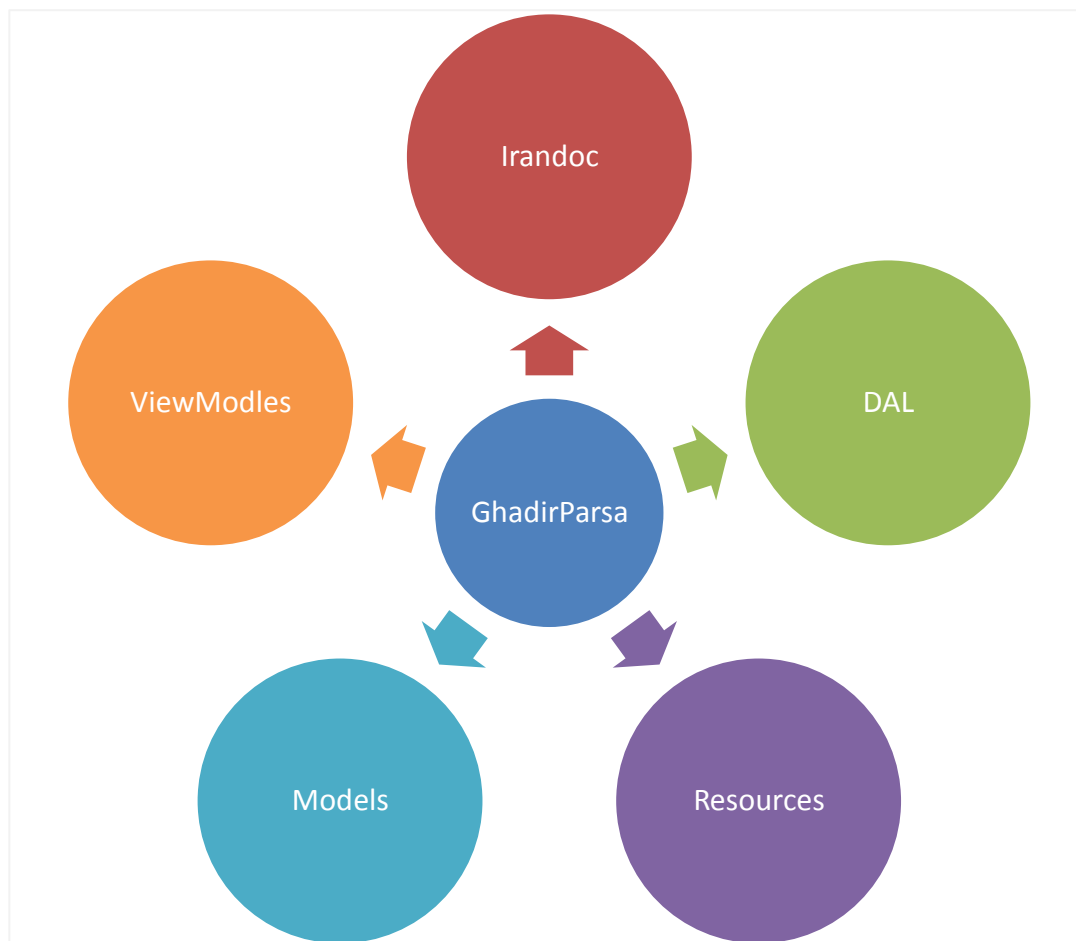
۲. GhadirParsa (Main Project)

۳. Irandoc (Utility Project)

۴. Models (Data Layer)

۵. Resources

۶. ViewModels



شکل ۳-۱ بخش‌های ساختاری در پروژه غدیر

همان طور که در شکل ۱-۳ ملاحظه می کنید، پروژه GhadirParsa پروژه اصلی سامانه است که تمام پروژه های دیگر سامانه با ارتباط دارند. پروژه Models شامل تمام مدل های سامانه است که از طریق آن پایگاه داده سامانه ساخته می شود و هرگونه تغییر پایگاه داده از طریق آن انجام می شود. پروژه Irandoc شامل تمام توابع کمی سامانه است که کارهای عمومی از جمله کار با تاریخ ها، Extension Methods، انواع تبدیل های نوع داده، انواع توابع پایه Security و ... در آن انجام می شود.

پروژه DAL که در واقع ارتباط با پایگاه داده را برای ما فراهم می کند. در این پروژه به ازای هر یک از Model ها یک Repository آماده شده است که امکان اتصال به پایگاه داده را فراهم می کند.

پروژه Resources محلی برای ذخیره Caption ها برنامه است که به صورت دوزبانه آماده شده است.

پروژه ViewModels شامل مدل هایی برای ارتباط و Bind کردن و همچنین Validation های سمت کلاینت است.

در ادامه مستند هر کدام از پروژه ها به تفصیل توضیح داده خواهند شد.

۱-۹-۱ Models

در گذشته برای اتصال به پایگاه داده معمولاً از روش Database first استفاده میشد. به این معنا که ابتدا از طریق یک Relational Database مدل داده ای طراحی و سپس از طریق Connection String، نرم افزار به پایگاه داده متصل و کوئری های لازم بروی آن زده میشد. در چند سال اخیر روشی با نام Code First مطرح شده است. در این روش به جای طراحی دیتابیس در داخل Relational Database، جداول به صورت کلاس هایی همراه با Attribute ها تعریف می شود و سپس از طریق مکانیسم هایی در داخل Visula Studio کلاس ها به جداولی در داخل Relational Database نگاشت می شوند. این روش بسیار انعطاف پذیر است و قدرت توسعه دهنده نرم افزار را برای تغییرات افزایش می دهد. در ادامه این بخش تمامی قسمت های این پروژه توضیح داده خواهد شد:

۱-۹-۲ References

۱. Resources: از پروژه Resources برای برخی Description ها و همچنین Attribute ها استفاده می شود.

۲. Irandoc: از پروژه Irandoc برای برخی کارهای General و Security استفاده می شود.

۳. مجموعه Dll های Identity و Owin: از این Dll ها برای راه اندازی Authentication و Authorization سیستم استفاده شده است.

۴. SmartStringResources: از این Dll برای متوجه کردن Attribte هایی که کار Validation را روی Model انجام می دهند و احتیاج به پیغامی سمت کلاینت دارند، استفاده می شود.

این Dll در واقع برای جلوگیری از Hard Code نوشتن در داخل Model Class ایجاد شده است.

۱-۹-۳. Infrastructur

این فولدر حاوی دو کلاس با نام های AppRoleManager و AppUserManager و Roles است.

۱. AppUserManager

این کلاس شامل متدی با نام Create است که امکان استفاده از جداول Users در تکنولوژی Identity را فراهم می کند.

در این متد با ساختن یک آبجکت از نوع PasswordVaidator می توان موارد زیر را هنگام تعریف User مشخص نمود:

- RequiredLength: مشخص کننده این است که تعداد کارکترهای گذرواژه باید حتما چه تعداد باشد.

- RequireNonLetterOrDigit: مشخص می کند که آیا گذرواژه شامل غیر حروف و غیر عدد باشد یا خیر.

- RequireDigit: آیا گذرواژه شامل عدد باشد یا خیر.

- RequireLowercase: آیا گذرواژه شامل حروف کوچک باشد یا خیر.

- RequireUppercase: آیا گذرواژه شامل حروف بزرگ باشد یا خیر.

در این متد همچنین می توان خاصیت UserLockoutEnabledByDefault را نیز تنظیم کرد. این خاصیت برای دادن توانایی برای بلاک کردن کاربران به کار می رود.

۲. AppRoleManager

این کلاس شامل متدی با نام Create است که امکان استفاده از جداول Roles در تکنولوژی Identity را فراهم می کند.

۳. Roles

این فایل شامل دو کلاس با نام های زیر است:

▪ Roles

این کلاس نقش های سامانه را Type می کند. به عنوان مثال نقش Student را به جای Hard Code نوشتن به شکل زیر در سامانه استفاده می شود:

```
User.IsInRole(Models.Infrastructure.Roles.Student)
```

▪ pRoles

Approles از نوع Enum تعریف شده است و دقیقا به مقداری عددی نقش ها اشاره دارد. برای استفاده از این Type در کدهای سامانه به شکل زیر استفاده شده است.

```
If(roleId == (int)Models.Infrastructure.AppRoles.Student)
{
    Do Something...
}
```

سامانه عضویت فراگیر کتابخانه ها شامل نقش های زیر است:

- Programmer
- Administrator
- Student
- Organization
- Irandoc
- Librarian

در بخش توضیح پروژه اصلی سامانه نحوه استفاده از این نقش ها و محدودیت کارها به تفصیل توضیح داده خواهد شد.

▪ فایل GhadirSecurity

در این فایل تمامی کلاس های لازم برای بخش مدیریت کاربران که باید در پایگاه داده ساخته شود، قرار دارد.

در این سامانه برای Authentication و Authorization از تکنولوژی ۲ Identity در چارچوب .Net استفاده شده است.

تکنولوژی Identity در سال ۲۰۱۳ معرفی شد، که دنباله سیستم ASP.NET Membership بود. سیستم قبلی گرچه طی سالیان استفاده می شد اما مشکلات زیادی هم به همراه داشت. بعلاوه با توسعه دنیای وب و نرم افزار، قابلیت های مدرنی مورد نیاز بودند که باید پشتیبانی می شدند. فریم ورک Identity در ابتدا سیستم ساده و کارآمدی برای مدیریت کاربران بوجود آورد و مشکلات پیشین را تا حد زیادی برطرف نمود. بعنوان مثال فریم ورک جدید مبتنی بر EF Code-first است، که سفارشی کردن سیستم عضویت را بسیار آسان می کند و به شما کنترل کامل می دهد. یا مثلاً احراز هویت مبتنی بر پروتوکل OAuth پشتیبانی می شود که به شما اجازه استفاده از فراهم کنندگان خارجی مانند گوگل، فیسبوک و غیره را می دهد.

لازم به توضیح است که همه کلاس های پایه Identity از نوع داده ای Guid به عنوان کلید اصلی جدول استفاده می کنند. در این سامانه با توجه به اینکه امکان ثبت کاربران با شمار زیاد دیده شده است، نیاز به تغییر این نوع داده ای به نوع Integer (به صورت ترتیبی) بود.

برای این تغییر باید تمام کلاس های پایه Identity دوباره بازنویسی می شدند. علاوه بر این برخی از این جداول پایه نیازی به افزودن متغیرهایی جدیدی نیز بودند.

در زیر تمام کلاسی‌های پایه Identity که دوباره برای این سامانه بازنویسی شده‌اند را ملاحظه می‌کنید:

● GhadirRole

این کلاس جدول Role را در پایگاه داده می‌سازد. افزودن

`IdentityRole<int, GhadirUserRole>`

به عنوان کلاس پایه‌ای که GhadirRole از آن به ارث می‌رود، باعث می‌شود که بتوان کلید اصلی این جدول را به صورت Integer تعریف کرد.

این کلاس به صورت Default خود Identity دارای دو فیلد است:

- Id
- Name
- Description

● GhadirUser

این کلاس جدول اصلی User ها را در پایگاه داده می‌سازد. افزودن

`IdentityRole<int, GhadirUserRole>`

به عنوان کلاس پایه‌ای که GhadirUser از آن به ارث می‌رود، باعث می‌شود که بتوان کلید اصلی این جدول را به صورت Integer تعریف کرد.

این جدول شامل فیلدهایی پایه‌است که تکنولوژی Identity آن را به صورت Default در نظر گرفته است:

- Id
- Email
- EmailConfirmed
- PasswordHash
- SecurityStamp
- PhoneNumber
- PhoneNumberConfirmed
- TwoFactorEnabled
- LockoutEndDateUtc
- LockoutEnabled
- AccessFailedCount
- UserName

علاوه بر این فیلدها، فیلدهای زیر نیز به صورت Explicit در این جدول تعریف شده‌اند.

- UserTypeId
- IsActive
- SourceCollageId
- RabetCollageId
- FirstName
- LastName

- FatherName
- GenderId
- IdentityNum
- BirthDay
- NationalCode
- PassportNum
- Mobile\
- Mobile۲
- PostalCode
- PostalAddress
- Image
- CreatedAt
- UpdatedAt

این نکته قابل ذکر است که Attribute با نام StringLength سبب می شود که مقدار فیلدهای کارکتری در پایگاه داده در نظر گرفته شود. در صورت عدم مشخص کردن آن، فیلد با نوع داده ای Nvarchar(max) ساخته می شود.

همچنین با مشخص کردن Attribute با نام Required این فیلد به عنوان Not Null در پایگاه داده ساخته می شود و ذخیره کردن رکورد بدون ورود اطلاعات این فیلد موجب خطا خواهد شد.

• GhadirUserRole

این کلاس در واقع ارتباط بین دو جدول GhadirUser و GhadirRole است. این جدول به صورت Default دارای دو فیلد UserId و RoleId است.

• GhadirUserLogin و GhadirUserClaim

این دو کلاس در سامانه عضویت فراگیر کتابخانه ها استفاده نشده است ولی به علت اینکه تکنولوژی Identity را باید به صورت Identity Primary Key پیاده کنیم، مجبور به دوباره نویسی این کلاس ها نیز هستیم.

• GhadirUserStore

این کلاس مشخص می کند که GhadirUser باید از چه Context ی برای اتصال به پایگاه داده استفاده کند.

از آنجاییکه تکنولوژی Identity از Context خود برای اتصال به پایگاه داده استفاده می کند و جداول دیگر پایگاه داده از Context ی دیگر باید به نوعی Identity را مجبور به استفاده از Context جداول دیگر سامانه کنیم تا در کل سامانه از یک Context استفاده شده باشد.

وجود دو Context در سامانه به معنی دو اتصال باز به سمت پایگاه داده است که می تواند موجب تداخل و کاهش Performance سامانه شود.

به همین جهت باید در این کلاس مشخص کنیم که تکنولوژی Identity حتما از Context جداول دیگر سامانه استفاده کند.

- GhadirRoleStore

این کلاس مشخص می‌کند که RoleStore باید از چه Contextی برای اتصال به پایگاه داده استفاده کند.

- مدل‌های سامانه (Database Models)

در این قسمت تمام جداول سامانه که به صورت کلاس‌هایی تعریف شده‌اند، توضیح داده خواهد شد. نکته قابل ذکر این است تمام مدل‌ها از کلاس پایه BaseEntity به ارث رفته‌اند. در صورتیکه بخواهیم یک فیلد در تمام مدل‌ها وجود داشته باشد، می‌توان آن را در این کلاس پایه تعریف کرد.

- Access

این مدل دسترسی‌های سامانه را در خود نگه می‌دارد. فیلدهای این مدل عبارتند از:

- Id
- Title
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- Permissions

از نوع ICollection است که مشخص می‌سازد که هر یک Access می‌تواند چند Permission داشته باشد. در واقع این ارتباط بین جدول Permission است.

- Permission

این مدل در واقع جدول واسطه (ارتباط چند به چند) بین Access و GhadirUser است. این مدل مشخص می‌سازد یک کاربر چه دسترسی‌هایی می‌تواند نگه دارد. فیلدهای این مدل عبارتند از:

- Id
- AccessId
- UserId
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- User

از نوع مدل GhadirUser که نشان می‌دهد یک Access خاص متعلق به چه کاربری است.

- Access

از نوع مدل Access که نشان می دهد کاربر چه Access ی دارد.

- CardLost

این مدل اعلام گم شدن کارت عضویت یا امانت از سوی کاربر را نگه داری می کند. فیلدهای این مدل عبارتند از:

- Id
- MemberId
- SourceMarkazId
- FormNumber
- FormDate
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- Member

نشان می دهد که اعلام گم شدن توسط چه کاربری انجام پذیرفته است.

- CardLostDetails

یک ارتباط یک چند است که جزییات کارت های عضویت و امانت گم شده را نشان می دهد.

- SourceMarkaz

نشان می دهد که این اعلام گم شدن کارت مربوط به کدام کتابخانه مبدأ است.

- CardType

این مدل نوع کارت گم شده را مشخص می کند. کارت گم شده می تواند عضویت یا امانت باشد. فیلدهای این مدل عبارتند از:

- Id
- Title
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- CardLostDetails

کارت هایی که با این وضعیت گم شده اند را نشان می دهد.

- CardLostDetails

این مدل در واقع جزئیات کارت عضویت یا امانت گم شده را در خود نگهداری می‌کند. فیلدهایی این مدل عبارتند از:

- Id ○
- CardTypeId ○
- CardLostId ○
- CardIdentity ○
- CardOrder ○
- IsReturnToUser ○
- ReturnToUserDate ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

CardLost •

رکورد اعلام گم شدن کارت را مشخص می‌کند. با یک اعلام گم شدن کتاب امکان اعلام چند کارت عضویت یا امانت است.

CardType •

نوع کارت عضویت یا امانت را مشخص می‌کند.

Faculty •

این مدل لیست دانشکده‌های یک سازمان را نگهداری می‌کند. فیلدهای این مدل عبارتند از:

- Id ○
- TitleFa ○
- TitleEn ○
- Neshani۱ ○
- Neshani۲ ○
- Neshani۳ ○
- Neshani۴ ○
- SazmanId ○
- MarkazId ○
- MarkazAuthorityId ○
- ShahrId ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

UserInfos •

لیست کاربرانی که در این دانشکده بوده‌اند را می‌دهد.

• Markaz

رکورد کتابخانه مبدأ که کاربران آن دانشکده برای عضویت به آن مراجعه کرده اند، را می دهد.

• Shahrha

شهری که دانشکده در آن قرار دارد را می دهد.

• FacultyType

کاربران سامانه عضویت فراگیر کتابخانه ها شامل دسته کلی هستند:

۱. وزارت علوم، تحقیقات، و فناوری اطلاعات: در حال حاضر سامانه به همین گروه در حال سرویس دهی است.

۲. سایر سازمان ها

نوع کاربران که می توانند عضو طرح شوند نیز دو دسته هستند:

۱. دانشجویان

۲. اعضای هیئت علمی

این مدل در واقع نوع استخدامی اعضای هیئت علمی را مشخص می کند که به دو نوع رسمی و پیمانی تقسیم می شوند.

فیلدهای این مدل عبارتند از:

○ Id

○ TitleFa

○ TitleEn

○ CreatedAt

○ UpdatedAt

و ارتباطات آن عبارتند از

• UserInfos

لیست کاربران با این وضعیت استخدامی مورد نظر را می دهد.

• FacultyDegree

این مدل مرتبه عضو هیئت علمی را نشان می دهد. فیلدهای این مدل عبارتند از:

○ Id

○ TitleFa

○ TitleEn

○ CreatedAt

○ UpdatedAt

و ارتباطات آن عبارتند از

• UserInfos

لیست کاربران با این وضعیت استخدامی مورد نظر را می‌دهد.

• Fine

این مدل خسارت‌های وارد را نگه می‌دارد. اعلام هر خسارت می‌تواند شامل چند خسارت باشد که جزییات آن در مدل دیگری با نام FineDetails نگه‌داری می‌شود. فیلدهای این مدل عبارتند از:

- Id
- MemberId
- RabetId
- FormNumber
- DateIssuedForm
- FineStateId
- MarkazId
- AccountingNo
- BackName
- BranchName
- BranchCode
- OwenerName
- LateFees
- PaymentCondition
- PaymentTrackNumber
- PaymentDate
- IsCompensatedByUser
- CompensatedTrackNumber
- TypeOfCompensate
- ResponseCompensate
- SourceConfirmCompensate
- DateOfResponseCompensate
- DateSourceConfirmCompensate
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

• Markaz

این ارتباط نشان می‌دهد که اعلام خسارت از جانب کدام کتابخانه صورت پذیرفته است.

• Member

این ارتباط نشان می دهد که اعلام خسارت برای کدام عضو سامانه صورت پذیرفته است.

• User

این ارتباط نشان می دهد که اعلام خسارت توسط چه کاربری و چه مشخصاتی در کتابخانه مقصد صورت پذیرفته است.

• FineDetails

این ارتباط جزئیات مربوط به این اعلام خسارت را نشان می دهد.

• FineDetails

این مدل رکوردهایی از جزئیات یک اعلام خسارت است که در مدل Fine اعلام شده است. فیلدهای این مدل عبارتند از:

- Id
- FineId
- FineMoney
- CardIdentity
- BookTitle
- BookWriter
- BookYear
- BookISBN
- Order
- BookReturned
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

• Fine

این ارتباط نشان می دهد که این رکورد خسارت مربوط به کدام رکورد اعلام خسارت در مدل Fine است.

• Gender

این مدل جنسیت را به گونه های مختلف نگهداری می کند. به عنوان مثال هم مرد، آقا و آقای در این مدل ذخیره شده است تا در زمان های مناسب بر اساس شرایط گوناگون از یکی از گونه ها استفاده شود. فیلدهای این مدل عبارتند از:

- Id
- TitleFa
- TitleFa\
- TitleFa۲
- TitleEn

و ارتباطات آن عبارتند از:

• Users

نشان می‌دهد که یک کاربر دار چه جنسیتی است.

• TerminateUserAccount

نشان می‌دهد فردی که تصفیه حساب افراد خارج از پوشش دریافت کرده است، چه جنسیتی داشته است.

• Grade

در این مدل اطلاعات پایه مقطع تحصیلی نگهداری می‌شود. فیلدهای این مدل عبارتند از:

Id ○

TitleFa ○

TitleEn ○

GradeType ○

CreatedAt ○

UpdatedAt ○

و ارتباطات آن عبارتند از:

• Users

این ارتباط نشان می‌دهد که چه کاربرانی دارای یک مقطع تحصیلی خاص هستند.

• Language

در این مدل اطلاعات پایه زبان‌ها نگهداری می‌شود. فیلدهای این مدل عبارتند از:

Id ○

TitleFa ○

TitleEn ○

CreatedAt ○

UpdatedAt ○

و ارتباطات آن عبارتند از:

• BorrowBooks

این ارتباط نشان می‌دهد که کدام کتاب‌های به امانت گرفته شده دارای این زبان هستند.

• Markaz

در این مدل اطلاعات مراکز تحت پوشش نگهداری می‌شوند. مراکز به سه دسته فقط مبدأ، فقط مقصد و مبدأ، مقصد تقسیم

بندی می‌شوند. همچنین این مراکز به دو دسته کلی تر وزارت علوم، تحقیقات، و فناوری اطلاعات تقسیم‌بندی می‌شوند.

فیلدهای این مدل عبارتند از:

Id ○
Title ○
ShahrId ○
VaziateMarkazId ○
SazmanId ○
VabasteId ○
Neshani\ ○
Neshani۲ ○
Neshani۳ ○
Neshani۴ ○
CodePost ○
SanPost ○
TelMost ○
CodeShahrestan ○
TelGhair ○
Dakheli ○
Email ○
WebSite ○
OzveMojaz ○
Zarfiat ○
MemberOfOrg ○
TarikhSh ○
TarikhPa ○
Longitude ○
Latitude ○
MapZoomLevel ○
CreatedAt ○
UpdatedAt ○

و ارتباطات آن عبارتند از:

● Sazmanha

این ارتباط نشان می دهد که این مرکز متعلق به کدام سازمان است. به عنوان مثال کتابخانه دکتر غلامحسین مصاحب متعلق به سازمان بنیاد دانشنامه نگاری ایران است.

● Shahrha

این ارتباط نشان می دهد که این مرکز در کدام شهر قرار دارد.

● Vabastegi

این ارتباط نشان می دهد که این مرکز برای وزارت علوم، تحقیقات، و فناوری اطلاعات است و یا متعلق به دیگر سازمان هاست.

• VaziateMarkaz

این ارتباط نشان می دهد که این مرکز فقط مبدأ یا فقط مقصد و یا مبدأ، مقصد است.

• SourceUsers

این ارتباط کاربران نماینده طرح غدیر را نشان می دهد.

• SourceUserInfos

این ارتباط کاربران مرتبط با این کتابخانه را نشان می دهد.

• FacultyUserInfos

این ارتباط دانشکده های مرتبط به این کتابخانه را نشان می دهد.

• Letters

نامه های ارسال مستندات مرتبط با این کتابخانه را نشان می دهد.

• SheetStatistics

برگ های آمار مرتبط با این کتابخانه را نشان می دهد.

• Fine

اعلام خسارت های مرتبط با این کتابخانه را نشان می دهد.

• TerminateAccounts

تصفیه حساب های افراد خارج از پوشش که در این کتابخانه تصفیه حساب خود را گرفته اند را نشان می دهد.

• Contracts

سایر سازمان هایی که با طرح غدیر قرار داد ببندند، از طریق ارتباط قراردادهایشان بدست می آید.

• CardLosses

کارت های عضویت و امانتی که گم شده در این کتابخانه را نشان می دهد.

• MarkazContract

در این مدل اطلاعات قرارداد کتابخانه های سایر سازمان ها نگهداری می شود. در صورتیکه سازمان در وزارت علوم نباشد برای عضویت در طرح غدیر حتما باید قراردادی با پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک) امضا کنند. اطلاعات این قرارداد در این مدل ذخیره می شود. فیلدهای این مدل عبارتند از:

○ Id

○ MarkazId

○ StartOfContract

○ EndOfContract

- OzveMojaz ○
- OzveJari ○
- Zarfiat ○
- Order ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

• Markaz

این ارتباط نشان می دهد که این قرارداد متعلق به کدام کتابخانه است.

• MarakezLetters

این مدل رکوردهای نامه های ارسال مستندات را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

- Id ○
- LetterNumber ○
- LetterDate ○
- UserInfoId ○
- MarkazId ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

• UserInfo

این ارتباط نشان می دهد که این نامه ارسال مستندات متعلق به کدام عضو طرح غدیر است.

• Markaz

این ارتباط نشان می دهد که این نامه ارسال مستندات متعلق به کدام کتابخانه است.

• Member

این مدل برای نگهداری عضوهای طرح عضویت فراگیر کتابخانه هاست. در این مدل شماره کارت عضویت فرد نگهداری می شود. فیلدهای این مدل عبارتند از:

- Id ○
- MemberIdentity ○
- Status ○
- ExpireDate ○
- ExpireDateShamsi ○
- Termination ○
- IsFine ○
- IsLost ○
- TerminationDate ○

- FineDate ○
- LostDate ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

• UserInfo

این ارتباط اطلاعات بیشتری از فرد عضو طرح غدیر می‌دهد. اطلاعاتی از قبیل مقطع تحصیلی، رشته و گرایش تحصیلی، وضعیت استخدام، مرتبه و...

• Cards

این ارتباط کارت‌های امانت فرد را نشان می‌دهد. برای هر کارت عضویت بنا بر نحوه عضویت و قطع تحصیلی، تعدادی کارت صادر می‌شود که اطلاعات آن از طریق ارتباط Cards قابل رویت هستند.

• SheetStatistics

این ارتباط برگ‌های آمار فرد عضو طرح غدیر را نشان می‌دهد.

• Fine

این ارتباط اعلام خسارت‌های مربوط به فرد عضو طرح غدیر را نشان می‌دهد.

• CardLosses

این ارتباط کارت‌های عضویت و امانتی که تا حال اعلام شده است را نشان می‌دهد.

• MemberCards

این مدل کارت‌های امانت یک فرد عضو طرح غدیر را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

- Id ○
- MemberId ○
- CardIdentity ○
- Status ○
- IsFine ○
- IsLost ○
- FineDate ○
- DateOfLost ○
- Order ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

• Member

این ارتباط اطلاعات کارت عضویت را نشان می‌دهد. از طریق این ارتباط و ارتباط Member با مدل UserInfo می‌توان تمام اطلاعات فرد عضو طرح غدیر را به دست آورد.

• NotifyContent

این مدل آگهی‌های صفحه اول سامانه را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

- Id ○
- Title ○
- Content ○
- CreatedAt ○
- UpdatedAt ○

• Radeha

در این مدل اطلاعات پایه رده‌های کتاب نگهداری می‌شود. در این سامانه دو نوع رده کنگره و رده دیویی دیده شده است. فیلدهای این مدل عبارتند از:

- Id ○
- Onvan ○
- Rade ○
- RadeType ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

• RadeType\

این ارتباط نوع رده کتاب را مشخص می‌کند.

• BorrowBooks

این ارتباط کتاب‌هایی که با این رده به امانت داده شده‌اند را مشخص می‌کند.

• RadeType

این مدل نوع رده را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

- Id ○
- TypeFa ○
- TypeEn ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

- Radeha

این ارتباط رده‌هایی که متعلق به این نوع رده هستند را مشخص می‌کند.

- RefuseState

این مدل اطلاعات پایه رد عضویت در طرح غدیر را نگهداری می‌کند. در صورتیکه فرد به هر علتی عضویتش رد شود باید علت رد عضویت به او گفته شود تا به رفع آن اقدام کند. فیلدهای این مدل عبارتند از:

- Id
- ShortTitle
- Desc
- IsShow
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- UserStatusRefeuses

این ارتباط رکوردهایی که دارای این علت رد هستند را نشان می‌دهد.

- UserStatusRefuse

این مدل اطلاعات دلایل رد عضویت کاربران را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

- Id
- RefuseId
- UserStatusId
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- RefuseState

این ارتباط دلیل رد عضویت فرد را نشان می‌دهد.

- UserStatus

این ارتباط کاربری که عضویت او رده شده است را نشان می‌دهد.

- Reshteha

این مدل اطلاعات گروه، رشته‌ها و گرایش‌ها را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

- Id
- Title
- TitleEn
- ParentId

- TypeId
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- UserInfos

این ارتباط نشان می دهد که یک رشته، گرایش مربوط به کدام کاربران است.

- Sazmanha

این مدل دانشگاه ها را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

- Id
- Onvan
- Description
- VabastegiId
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- Marakez

این ارتباط کتابخانه های مرتبط با آن دانشگاه را نشان می دهد.

- SazmanhaDetails

این ارتباط اطلاعات بیشتری از دانشگاه می دهد.

- SazmanhaDetail

این مدل اطلاعات بیشتری از دانشگاه را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

- Id
- Onvan
- SazmanId
- DateOfCreate
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

- Sazmanha

این ارتباط نشان می دهد که اطلاعات بیشتر مربوط به کدام دانشگاه است.

- Shahrha

این مدل اطلاعات مربوط به شهرها را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

Id ○
NameSh ○
CreatedAt ○
UpdatedAt ○

و ارتباطات آن عبارتند از:

Marakez •

این ارتباط کتابخانه هایی که در این شهر قرار دارند را نشان می دهد.

Faculties •

این ارتباط دانشکده هایی که در این شهر قرار دارند را نشان می دهد.

SheetStatistics •

این مدل برگه های آمار را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

Id ○
MemberId ○
DateOfComming ○
DestinationCollageId ○
BorrowBook ○
NumberOfBookBorrow ○
ReturnBorrowBook ○
UseBookInLocationFarsi ○
UseBookInLocationLatin ○
UseBookInLocationArabic ○
UseBookInLocationUknown ○
UsePublicationInLocation ○
UsePublicationInLocationFarsi ○
UsePublicationInLocationLatin ○
UsePublicationInLocationArabic ○
UsePublicationInLocationUnkown ○
UseDigitalInLocation ○
UseDigitalInLocationFarsi ○
UseDigitalInLocationLatin ○
UseDigitalInLocationArabic ○
UseDigitalInLocationUnkown ○
UseThesisInLocation ○
UseConsultation ○
UseReadingroom ○
UsePrint ○
UserDatabase ○

- UseInternetService
- OtherServices
- OtherServicesDesc
- UserInserted
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

• BorrowBooks

این ارتباط کتاب‌هایی که به امانت گرفته شده‌اند را نشان می‌دهد.

• Member

این ارتباط عضوی که برگ آمار متعلق به اوست را نشان می‌دهد.

• DestinationMarkaz

این ارتباط کتابخانه مقصدی که برگ آمار در آن پر شده است را نشان می‌دهد.

• BorrowBook

این مدل کتاب‌های به امانت گرفته شده که در یک برگ آمار آمده است را نگهداری می‌کند. فیلدهای این مدل عبارتند از:

- Id
- Isbn
- RadeId
- LanguageId
- BorrowCardNo
- SheetStatisticId
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

• SheetStatistic

این ارتباط برگ آمار مربوط به کتاب امانتی را نشان می‌دهد.

• Language

این ارتباط زبان کتاب به امانت گرفته شده را نشان می‌دهد.

• Rade

این ارتباط رده کتاب به امانت گرفته شده را نشان می‌دهد.

• SheetPrint

این مدل تعداد برگ‌های آماری که به اعضا داده شده را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

Id ○
MemberId ○
MemberOfSheets ○
CreatedAt ○
UpdatedAt ○

و ارتباطات آن عبارتند از:

Member •

این ارتباط نشان می‌دهد برگ‌های آمار صادر شده متعلق به کدام عضو است.

SmsContent •

این مدل محتوای پیامک‌های ارسالی را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

Id ○
Content ○

و ارتباطات آن عبارتند از:

SmsUsers •

این ارتباط نشان می‌دهد پیامک با این محتوا برای کدام یک از اعضا ارسال شده است.

TerminateAccount •

این مدل تصفیه حساب افراد را در خود نگهداری می‌کند. فیلدهای این مدل عبارتند از:

Id ○
FormNumber ○
DateOfForm ○
UserInfoId ○
SourceMarkazId ○
FirstName ○
LastName ○
GenderId ○
NationalCode ○
PassportNo ○
Mobile ○
TypeOfTerminate ○
IsAccepted ○
DateOfAccept ○
CreatedAt ○
UpdatedAt ○

و ارتباطات آن عبارتند از:

• UserInfo

این ارتباط نشان می دهد که این تصفیه حساب برای کدام عضو است.

• SourceMarkaz

این ارتباط نشان میدهد که این تصفیه حساب مربوط به کدام کتابخانه مبدأ بوده است.

• Gender

اگر تصفیه حساب مربوط به افراد خارج از پوشش باشد این ارتباط مشخص می کند که فردی که تصفیه حساب کرده چه جنسیتی داشته است.

• UserInfo

این مدل تعداد عضویت های یک فرد در مقاطع مختلف را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

- Id ○
- UserId ○
- userId ○
- LastStatusId ○
- SourceCollegeId ○
- FacultyId ○
- FactorNumber ○
- CreatedBy ○
- GradeId ○
- FieldId ○
- FieldName ○
- HalfYear ○
- ArrivalDate ○
- FacultyTypeId ○
- FacultyDegreeId ○
- FormNumber ○
- AllDocPrinted ○
- IsDependency ○
- DependencyCreditDate ○
- DateIssueForm ○
- RefuseStatusId ○
- RefuseDesc ○
- UserAcceptConfirmed ○
- CreatedAt ○
- UpdatedAt ○

و ارتباطات آن عبارتند از:

• User

این ارتباط اطلاعات اصلی و کاربری فرد را مشخص می کند.

• FacultyType

این ارتباط نوع استخدام عضو هیئت علمی را مشخص می کند.

• FacultyDegree

این ارتباط مرتبه عضو هیئت علمی را مشخص می کند.

• SourceMarkaz

این ارتباط کتابخانه مبدأ عضو را مشخص می کند.

• Reshteh

این ارتباط رشته، گرایش عضو را مشخص می کند.

• Grade

این ارتباط مقطع تحصیلی عضو را مشخص می کند.

• Member

این ارتباط اطلاعات عضویت از جمله شماره کارت عضویت را مشخص می کند.

• Faculty

این ارتباط اطلاعات دانشکده را در خود نگهداری می کند.

• Letters

این ارتباط نامه های ارسال مستندات را در خود نگهداری می کند.

• UserStatues

این ارتباط تاریخچه عضویت فرد را در خود نگهداری می کند.

• TerminateAccount

این ارتباط اطلاعات مربوط به تصفیه حساب فرد را در خود نگهداری می کند.

• UsersSms

این مدل پیامک های که به کاربران ارسال شده است را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

○ Id

○ UserId

○ ConditionId

○ ContentId

○ RecId

○ SmsFor

CreatedAt ○

UpdatedAt ○

و ارتباطات آن عبارتند از:

• User

این ارتباط نشان می دهد که پیامک متعلق به کدام کاربر است.

• StatusType

این ارتباط نشان می دهد که پیامک برای چه وضعیتی از کاربر ارسال شده است.

• SmsCondition

این ارتباط نشان می دهد که وضعیت پیامک ارسالی به چه شده بوده است.

• SmsContent

این ارتباط نشان می دهد که پیامک به چه محتوایی ارسال شده است.

• SmsCondition

این مدل وضعیت هایی که یک پیامک می تواند داشته باشد را در خود نگهداری می کند. فیلدهای این مدل عبارتند از:

Id ○

RealCode ○

TitleFa ○

TitleEn ○

CreatedAt ○

UpdatedAt ○

و ارتباطات آن عبارتند از:

• UsersSms

این ارتباط چه پیامک هایی برای چه کاربرانی با این وضعیت ارسال شده است.

• UserStatuses

این مدل وضعیت هایی که یک کاربر طی می کند تا به عضویت طرح غدیر برسد و در نهایت تصفیه حساب را انجام دهد در

خود نگهداری می کند. فیلدهای این مدل عبارتند از:

Id ○

UserId ○

UserInfoId ○

ByUserId ○

StatusTypeId ○

Description ○

CreatedAt ○

و ارتباطات آن عبارتند از:

• User

این ارتباط نشان می دهد که وضعیت مربوطه متعلق به کدام کاربر است.

• UserInfo

این ارتباط نشان می دهد که وضعیت مربوطه متعلق به کدام یک از عضویت های کاربر در دوره های مختلف است.

• StatusType

این ارتباط نشان می دهد که این رکورد دقیقا به کدام وضعیت اشاره دارد.

• UserStatusRefeuses

این ارتباط نشان می دهد دلیل رد شدن عضویت فرد در طرح عضویت فراگیر کتابخانه ها را نشان می دهد.

• StatusType

این مدل انواع وضعیت هایی که کاربر می تواند با آن مواجه شود را در خود نگه دارد می کند. فیلدهای این مدل عبارتند از:

○ Id

○ TitleFa

○ TitleEn

○ CreatedAt

○ UpdatedAt

و ارتباطات آن عبارتند از:

• UserStatues

این ارتباط چه تاریخچه هایی با این وضعیت ثبت شده است.

• Vabastegi

این مدل اطلاعات پایه انواع وابستگی سازمان ها را مشخص می کند. فیلدهای این مدل عبارتند از:

○ Id

○ Title

○ CreatedAt

○ UpdatedAT

و ارتباطات آن عبارتند از:

• Sazmanhs

این ارتباط سازمان هایی با وابستگی مشخص را نشان می دهد.

• Marakezs

این ارتباط کتابخانه‌هایی با وابستگی مشخص را نشان می‌دهد.

• VaziateMarkaz

این مدل اطلاعات پایه انواع وضعیت‌های کتابخانه‌ها را مشخص می‌کند. وضعیت‌ها می‌توانند فقط مقصد، فقط مبدأ و مبدأ، مقصد باشند. فیلدهای این مدل عبارتند از:

- Id
- Title
- CreatedAt
- UpdatedAt

و ارتباطات آن عبارتند از:

• Marakezs

این ارتباط کتابخانه‌هایی وضعیت مشخص را نشان می‌دهد.

• GhadirContext فایل

این فایل شامل Context اصلی سامانه برای ارتباط با پایگاه داده به همراه تعاریف ارتباطات پایگاه داده است. ارتباطات دیتابیس از طریق متد CodeFirst به سامانه منتقل می‌شود. در متد CodeFirst برای تغییرات بروی دیتابیس باید از

Update-Database -Force

دستور زیر در بخش Package Manager Console استفاده کرد.

برای دسترسی به بخش Console کافیت به منوی Tools و بعد Nuget Package Manager و بعد گزینه Package Manager Console را انتخاب کرد. کلاس GhadirContext با به ارث بردن از کلاس‌های زیر Context مورد استفاده در سامانه را با Context استفاده از تکنولوژی Identity به اشتراک می‌گذارد.

IdentityDbContext<GhadirUser, GhadirRole,
[int](#), GhadirUserLogins, GhadirUserRole, GhadirUserClaims>

• فایل Migration/Configuration

این کلاس شامل تنظیمات متد CodeFirst است. همچنین این کلاس شامل متدی با نام Seed است که از طریق آن می توان اطلاعات پایه سامانه را Initialize کرد. به عنوان مثال Role های سامانه را می توان در آن مقدار دهی و سپس به دیتابیس منتقل کرد.

تنها تنظیم صورت گرفته برای سامانه در این فایل Property با نام AutomaticMigrationEnabled است. مقدار این Property برابر با True قرار گرفته است. با True قرار دادن این Property تغییرات به صورت خودکار به پایگاه داده منتقل می شود.

۱-۹-۴. DAL(Data Access Layer)

این پروژه وظیفه ارتباط با پایگاه داده به منظور کارهای زیر را دارد:

۱. دریافت اطلاعات از پایگاه داده
۲. افزودن اطلاعات به پایگاه داده
۳. بروزرسانی اطلاعات پایگاه داده
۴. حذف اطلاعات پایگاه داده

برای انجام این چهار عمل اصلی بروی پایگاه داده نیاز به ساختاریست که عملیاتها به راحتی و بدون هزینه اضافی انجام شود. ساختاری این سامانه از آن پیروی می کند، الگوی Repository-UnitOfWork است.

• الگوی Repository

با استفاده از این الگو می توان چهار عمل اصلی Crud را به صورت Generic برای همه مدل ها تعریف کرد. برای این کار به شیوره زیر عمل شده است.:

ابتدا یک Interface با نام IRepository تعریف شده است.

در این اینترفیس متدهای اصلی عملیات Crud ایجاد شده است و سپس در کلاس Repository پیاده سازی هر یک از متدها انجام شده است. متدهای اصلی Repository عبارتند از:

Insert (T entity)

این متد جنریک برای افزودن یک رکورد اطلاعاتی به پایگاه داده به کار می رود. به طوریکه T به معنای یک مدل از پایگاه داده است. به عنوان مثال T در اینجا می تواند یک Object از مدل UserInfo باشد.

Update (T entity)

این متد جنریک برای ویرایش یک رکورد اطلاعاتی به پایگاه داده به کار می‌رود. به طوریکه T به معنای یک مدل از پایگاه داده است. به عنوان مثال T در اینجا می‌تواند یک Object از مدل UserInfo باشد.

Delete(T entity)

این متد جنریک برای حذف یک رکورد اطلاعاتی از پایگاه داده به کار می‌رود. به طوریکه T به معنای یک مدل از پایگاه داده است. به عنوان مثال T در اینجا می‌تواند یک Object از مدل UserInfo باشد.

DeleteById(int id)

این متد جنریک برای حذف یک رکورد اطلاعاتی از پایگاه داده بر اساس شناسه به کار می‌رود.

GetById (int id)

این متد برای بدست آوردن یک رکورد اطلاعاتی بر اساس شناسه آن رکورد است.

Get

این متد مجموعه‌ای از رکوردهای اطلاعاتی از یک مدل را بدست می‌دهد. خروجی این رکوردها به صورت IQueryable است که اجازه کوئری زدن دوباره بروی این مجموعه را می‌دهد.

Get(With Predicate Condition)

این متد مجموعه‌ای از رکوردهای اطلاعاتی از یک مدل را بدست می‌دهد با این تفاوت که امکان فیلتر کردن نتیجه در داخل خود این متد تعبیه شده است. خروجی این رکوردها به صورت IQueryable است که اجازه کوئری زدن دوباره بروی این مجموعه را می‌دهد.

۱-۹-۵. تعریف Repository ها

• AccessRepository

برای دسترسی به رکوردهای جدول Access انجام چهار عمل اصلی (Crud) بروی آن این Repository تعریف می‌شود. برای تعریف این Repository ابتدا یک اینترفیس از این Repository باید ساخته شود:

```
public interface IAccessRepository: IRepository<Models.Access>
{
{
```

توجه شود که Access دقیقاً همان مدل‌یست که در بخش قبل در پروژه Models تعریف شده بود. پس از تعریف اینترفیس باید خود Repository را تعریف کرد:

```
class AccessRepository : Repository<Models.Access>, IAccessRepository
{
    public AccessRepository(Models.GhadirContext databaseContext)
        : base(databaseContext)
    {
    }
}
```

با انجام این کار AccessRepository آماده استفاده در سامانه است. برای اینکه تمام Repository ها به صورت مجتمع در یک جا قرار داده شوند و به صورت کاملاً یکپارچه بتوان از آنها استفاده کرد، الگوی UnitOfWork مطرح می‌شود. کلاس UnitOfWork شامل مجموعه‌ای از Repository هاست که از تکنیک LazyLoading برای Initilize کردن Repository ها استفاده می‌کند. نمونه‌ای از این گونه Initialize کردن در زیر آمده است:

```
private IAccessRepository _accessRepository;
public IAccessRepository AccessRepository
{
    get
    {
        if (_accessRepository == null)
        {
            _accessRepository =
                new AccessRepository(DatabaseContext);
        }
        return (_accessRepository);
    }
}
```

نحوه استفاده از Repository ها به شکل زیر است:

۱. ابتدا یک Object از کلاس UnitOfWork در پروژه اصلی سامانه (در یکی از Controller ها) تعریف می‌شود.

```
DAL.UnitOfWork unitOfWork=new DAL.UnitOfWork;
```

۲. Repository مورد نظر از طریق آبجکت unitOfWork صدا زده می‌شود.

```
unitOfWork.AccessRepository
```

۳. متد لازم (Get,Delete,Insert,Update) صدا زده می‌شود.

```
unitOfWork.AccessRepository.Get()
```

این متد کل رکوردهای جدول Access را در اختیار سامانه قرار می‌دهد. همچنین برای فیلتر کردن نتایج رکورد های میتوان از فیلتر زیر استفاده کرد:

```
unitOfWork.AccessRepository.Get(p=>p.Id=۲۰)
```

نکته قابل توجه در کلاس UnitOfWork قابل Dispose بودن آن است. به عبارت دیگر پس از استفاده از Repository مورد نظر Context مورد استفاده Dispose و فضای مورد استفاده در رم آزاد می‌شود.

- PermissionRepository ○
- BorrowBookRepository ○
- CardLostRepository ○
- CardLostDetailsRepository ○
- CardTypeRepository ○
- FacultyDegreesRepository ○
- FacultyRepository ○
- FacultyTypesRepository ○
- FineRepository ○
- FineDetailsRepository ○
- FineStateRepository ○
- GenderRepository ○
- GradeRepository ○
- MarakezLettersRepository ○
- MarakezRepository ○
- MarakezContractRepository ○
- MemberRepository ○
- MemmmberCardsRepository ○
- NotifyContentRepository ○
- RadehaRepository ○
- ReshtehaRepository ○
- SazmanhaDetailRepository ○
- SazmanhaRepository ○
- ShahrhaRepository ○
- SheetStatisticRepository ○
- SheetsPrintRepository ○
- SmsConditionRepository ○
- SmsContentRepository ○
- StatusTypeRepository ○
- TerminateAccountRepository ○
- UserInfoRepository ○
- RefuseStateRepository ○
- UsersSmsRepository ○
- UserStatusesRepository ○
- UserStatusRefusesRepository ○
- VabastegiRepository ○
- VaziateMarkazRepository ○

تمام Repository های سامانه در قسمت بالا آمده است. هر کدام از Repository ها به یک مدل متصل است و امکان انجام عملیات های Crud را می دهد. باید توجه شود که این پروژه به صورت مستقیم از پروژه Models استفاده می کند و باید DLL این پروژه به بخش Reference افزوده شود.

۱-۹-۶. Irandoc

این پروژه شامل توابع و کلاس های مفیدی است که سرتاسر این سامانه از آن استفاده شده است. در ادامه به بررسی برخی از این توابع مهم پرداخته خواهد شد.

• EncryptionHelper

این کلاس از نوع Static است و شامل دو متد برای Encrypt و Decrypt کردن رشته ها است. از این متدها می توان در کد کردن پارامترهای ارسالی استفاده کرد.

جدول ۱-۱ مشخصات کلاس پروژه ایراندک

توضیحات	کلاس
این کلاس شامل Enums های تمام سامانه است.	Enums
کدهای وضعیت کارت عضویت و امانت را نشان می دهد.	MemberActivation
کد همه انواع پرینت های سامانه را نشان می دهد.	PrintType
کد انواع رده را نشان می دهد. رده کنگره یا دیویی است.	RadeType
کد جنسیت را نشان می دهد.	Gender
کد وضعیت مرکز را نشان می دهد. وضعیت مرکز می تواند فقط مقصد، فقط مبدأ و یا مبدأ، مقصد باشد.	VaziateMarkaz
کد وضعیت وابستگی سازمان را نشان می دهد. وابستگی سازمان می تواند وزارت علوم، تحقیقات، و فناوری یا سایر سازمان ها باشد.	VabastegiMarkaz
کد وضعیت مقطع تحصیلی را نشان می دهد. کد وضعیت یا وزارت علوم، تحقیقات، و فناوری یا سایر سازمان هاست.	GradeType

UserType	کدهای انواع کاربران سامانه را نشان می‌دهد.
StatusType	کدهایی شامل وضعیت‌هایی که کاربر در هنگام ثبت‌نام آن را طی می‌کند.
SmsContentType	کدهای وضعیت‌هایی که پیامک به دلیل آن ارسال می‌شود را نگهداری می‌کند.
FieldType	کدی که وضعیت گروه، رشته و گرایش بودن را نشان می‌دهد.
Grades	کدهایی که انواع مقاطع تحصیلی را نشان می‌دهد.
HalfYear	کدی که وضعیت نیمه اول یا نیمه دوم سال را مشخص می‌کند.
SearchFields	کدهایی که انواع فیلدهای قابل جست‌وجو را مشخص می‌کند.
TypeCode	کدی که شناسه ملی یا گذرواژه را مشخص می‌کند.
Language	کدهایی که شامل اطلاعات پایه زبان‌هاست.
TerminateType	کدی که نوع تصفیه حساب را مشخص می‌کند. نوع تصفیه حساب می‌تواند اعضا یا افراد خارج از پوشش باشد.
CardType	کدی که نوع کارت را مشخص می‌کند. نوع کارت می‌تواند عضویت یا امانت باشد.
TypeOfCompensate	کدهایی که وضعیت جبران خسارت را مشخص می‌کند.
Access	کدهایی که وضعیت دسترسی‌ها را مشخص می‌کند.

• File

این کلاس برای خواندن و نوشتن در فایل آماده شده است. این کلاس شامل دو متد با نام‌های

Read که عمل خواندن از فایل را انجام می‌دهد و Write که عمل نوشتن بروی فایل را انجام می‌دهد.

• Log

این کلاس شامل متدها و Propertyهایی برای لاگ کردن خطاهاییست که سامانه با آن مواجه می‌شود.

• ShortGuidId

این کلاس شامل متدها و Property هایی برای کار با نوع داده‌ای Guid است.

StaticStringMethods

این کلاس شامل متدها و Property هایی برای کار با نوع داده‌ای String است.

• Utility

این کلاس شامل متدهای و Property هایی کمکی است که در سرتاسر سامانه از آن‌ها استفاده شده است. از جمله آن‌ها می‌توان به HMAC_MD۵ اشاره کرد.

• PersianDate

این کلاس شامل تمام متدها و Property های لازم برای کار با نوع داده‌ای Date است. در این کلاس انواع تبدیلات تاریخ‌ها از شمسی به میلادی، میلادی به شمسی، قمری به شمسی و ... دیده شده است.

۱-۹-۷. Resources

این پروژه تمام عنوان‌ها، اسامی، برچسب‌ها و... که سرتاسر سامانه استفاده شده اند را نگهداری می‌کند. این پروژه باعث می‌شود که در تمام سامانه تنها از یک Resource استفاده شود. به عنوان مثال کلید FirstName به عنوان "نام" در کل سامانه استفاده می‌شود و زمان که لازم به تغییر آن باشد، کافیسست که این Resource عوض شود تا همه جایی که در سامانه به آن ارجاع داشته‌اند، تغییر پیدا کند.

استفاده از Nuget با عنوان SmartStringResource امکان استفاده از Resource ها در DataAnotation که برای Validation مدل‌ها به کار می‌رود را فراهم می‌کند. در بخش توضیحات پروژه ViewModels ها این مسئله به تفصیل توضیح داده خواهد شد. برخی از Resource های مهم در سامانه در جدول ۱-۲ آمده است:

جدول ۱-۲ Resource های شاخص در سامانه

منبع	توضیحات
Buttons	این Resource برای تمام لینک‌ها و دکمه‌های سامانه به کار رفته است.
Captions	این Resource برای تمام Label های سامانه به کار رفته است.
Messages	این Resource برای تمام پیام‌های سامانه به کار رفته است.
Title	این Resource برای تمام عناوین سامانه به کار رفته است.

• ViewModels

این مجموعه Resource ها شامل تمام Validation ها و Label هایست که مربوط به یک ViewModel می‌شود. به عنوان مثال UserViewModel که به یک View بایند (Bind) می‌شود، حاوی مجموعه‌ای کنترل‌ها و Property هاست که برای نمایش عنوان، پیام خطا و ... آن از این Resource ها استفاده می‌شود. یک مورد از استفاده Resource در ViewModels ها آمده است:

```
[RegularExpression(@"^[۰-۹,۰-۹]+$", ErrorMessageResourceType =
typeof(Resources.ViewModels.RegisterViewModels.RegisterViewModel),
ErrorMessageResourceName =
Resources.ViewModels.RegisterViewModels.Strings.RegisterViewModelKeys.Father
NameLetter )]

[StringLength(۱۵, ErrorMessageResourceType = typeof(Resources.Validations),
ErrorMessageResourceName =
Resources.Strings.ValidationsKeys.StringLengValidation)]

[Display(ResourceType =
typeof(Resources.ViewModels.RegisterViewModels.RegisterViewModel),
Name =
Resources.ViewModels.RegisterViewModels.Strings.RegisterViewModelKeys.Father
Name)]

[Required(AllowEmptyStrings = false,
ErrorMessageResourceType =
typeof(Resources.ViewModels.RegisterViewModels.RegisterViewModel),
ErrorMessageResourceName =
Resources.ViewModels.RegisterViewModels.Strings.RegisterViewModelKeys.Father
NameRequired)]

public string FatherName { get; set; }
```

برای Property با نام FatherName چهار Attribute تعریف شده است. این Attribute ها هنگامیکه ViewModel از طریق Controller به سمت View ارسال می شود، اثر خود را در نمایش و Validation ها می گذارد.

جدول ۳-۱ مشخصه های اصلی در ViewModel

مشخصه	توضیحات
RegularExpression	این Attribute ورودی هایی که یک Input از نوع Text می تواند بپذیرد را محدود می کند. عبارت منظم بالا مشخص می کند که این Input نباید کارکتهایی عددی قبول کند. در ادامه این Attribute از طریق امکاناتی که Nuget با نام SmartStringResource در دسترس قرار می دهد، می توان

عبارت خطایی را سمت کاربر نشان داد.	
این Attribute سبب می‌شود در سمت View عنوان (Lable) یک فیلد Input از طریق ViewModels تنظیم شود. در ادامه این Attribute از طریق امکاناتی که Nuget با نام SmartStringResource در دسترس قرار می‌دهد، می‌تواند عنوان Lable را از طریق پروژه Resource بخواند.	Display
این Attribute ورودی‌هایی که یک Input از نوع Text را می‌تواند بپذیرد را از نظر تعداد کارکترها محدود می‌کند. در ادامه این Attribute از طریق امکاناتی که Nuget با نام SmartStringResource در دسترس قرار می‌دهد، می‌توان عبارت خطایی را به کاربر نشان داد.	StringLength
این Attribute ورودی‌هایی که یک Input از نوع Text را می‌تواند بپذیرد را از نظر عدم وجود کارکتر محدود می‌کند و در واقع Input حتما باید دارای مقداری باشد. در ادامه این Attribute از طریق امکاناتی که Nuget با نام SmartStringResource در دسترس قرار می‌دهد، می‌توان عبارت خطایی را به کاربر نشان داد.	Required

۸-۹-۱ ViewModels

این پروژه برای کمک به نمایش اطلاعات و همچنین ذخیره اطلاعات در پایگاه داده به کار می‌رود. از طریق ViewModels ها و کمک Resource ها می‌توان اطلاعات را به صورت کاملاً ساخت‌یافته و با استفاده از متد Binding چارچوب Net. سمت کاربر نمایش داد و همچنین اطلاعات را به صورت کامل و ساخت‌یافته از کاربر دریافت کرد تا پس از Validation آن هم در سمت کلاینت و هم سمت سرور آن را در پایگاه داده ذخیره کرد.

۹-۹-۱ GhadirParsa

پروژه اصلی سامانه با نام GhadirParsa است. این پروژه از همه پروژه‌ها استفاده می‌کند. همه پروژه‌های سامانه تحت یک Solution قرار دارند. توجه شود برای اینکه پروژه اصلی سامانه این پروژه قرار بگیرد باید به Visual Studio بگوئیم که این پروژه اصلی است. با کلیک سمت راست روی پروژه GhadirParsa و انتخاب گزینه Set as Start Project این پروژه را به عنوان پروژه اصلی سامانه معرفی می‌کنیم. در ادامه تمام فایل‌های این پروژه به تفصیل توضیح داده خواهد شد.

• References

در این بخش تمام Dll‌های لازم برای اجرای این پروژه قرار گرفته است. از جمله Dll‌های مهم:

- DAL
- Models
- Resources
- ViewModels
- Irandoc

این Dll‌ها در واقع دیگر پروژه‌های سامانه هستند که به صورت Dll به پروژه اصلی سامانه افزوده شده‌اند. علاوه بر Dll‌هایی که توسط Net. به صورت پیش‌فرض به پروژه افزوده شده است، دیگر Dll‌های لازم عبارتند از:

• CaptchaMvc

برای ایجاد کپچا به کار می‌رود.

• EntityFramework

مجموعه Dll‌های EntityFramework که برای اتصال به پایگاه داده و انجام عملیات روی آن مورد استفاده قرار می‌گیرد.

• Kendo.DynamicLinq

این Dll برای نمایش اطلاعات پایگاه داده در گرید (Kendo UI) مورد استفاده قرار می‌گیرد. این Dll عمل صفحه‌بندی و ... را انجام می‌دهد.

• Microsoft.AspNet.Identity

شامل مجموعه Dll هایی که برای استفاده از تکنولوژی Identity (برای Authentication & Authorization) مورد استفاده قرار می گیرد.

• StimulSoft

این Dll برای گزارش گیری های سامانه مورد استفاده قرار می گیرد. برای گزارش گیری از مجموعه کامپوننت های Stimul استفاده شده است.

• Web References

تنها Web Reference این سامانه وب سرویس فرستادن پیامک است. شرح نحوه تنظیم این وب سرویس در بخش استقرار و پشتیبانی توضیح داده شده است.

• App_Data

در این فولدر فایل های عمومی برنامه قرار داده می شود. از جمله فایل هایی که در این فولدر قرار دارند می توان به فایل های آموزشی که توسط کاربران کتابخانه و اعضا در سامانه قرار دارد، اشاره کرد. فایل دیگری که در این فولدر قرار می گیرد، فایل Log.txt است. هر زمان سامانه با خطایی غیر قابل پیش بینی مواجه می شود، محتوای خطا توسط ماژول Log Handler در این فایل نوشته می شود تا از این طریق بتوان خطاهای سامانه را رصد کرد.

• App_Start

در این فولد تنظیمات اولیه سامانه هنگامیکه برای اولین بار سامانه بالا می آید، تنظیم می شود. هر کدام از فایل های این فولدر عبارتند از:

• BundleConfig

در این بخش تمام فایل های Css و JavaScript که باید در سامانه استفاده شوند ، تنظیم می شود. خاصیت کلاس BundleConfig این است که سبب می شود با نوشتن یک عبارت کوتاه مجموعه ای از فایل های Css و JavaScript را به Layout سامانه بیافزاییم. این کلاس شامل متدی با نام RegisterBundles است که تمام فایل های Css و JavaScript از طریق این متد به سامانه افزوده می شود. با استفاده از Bundles.Add() می توان یک Bundle از نوع ScriptBundle و یا StyleBundle به سامانه افزود.

• FilterConfig

در این کلاس Attribute هایی که می خواهیم روی Controller ها اعمال شوند را ثبت می کنیم. این کلاس شامل RegisterGlobalFilters است که در آن فیلترهایی که می خواهیم روی تمام Controller ها اعمال شوند را ثبت می کنیم. در این بخش یک Filter با نام CutomHandleErrorAttribute ثبت شده است که وظیفه آن کنترل خطاهای سامانه است. در صورتیکه سامانه به خطایی غیرقابل پیش بینی مواجه شود، سامانه به صورت خودکار به CustomHandleError منتقل می شود. در آن قسمت خطا کنترل و Log می شود.

• IdentityConfig

در این بخش تنظیمات Identity که ابزار Authentication و Authorization است، انجام می شود.

- متد ApplicationUserManager جدول User را برای استفاده در سامانه آماده می سازد.
- متد ApplicationRoleManager جدول Roles را برای استفاده در سامانه آماده می سازد.
- متد ApplicationSignInManager تنظیمات ورود به سامانه را تنظیم می سازد.

• RouteConfig

در این کلاس تنظیمات مربوط به نحوه آدرس دهی انجام می شود. این کلاس شامل متدی با نام RegisterRoute است که تنظیمات Route در آن انجام می پذیرد.

```
routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
```

```
routes.MapRoute(  
    name: "Default",  
    url: "{controller}/{action}/{id}",  
    defaults: new { controller = "Account", action = "Login", id =  
                    UrlParameter.Optional }
```

با دستور route.IgnoreRoute می توان Route های غیرمجاز را مشخص کرد. آدرس دهی اصلی سامانه به این صورت است که حتما باید ابتدا Controller و بعد Action و بعد پارامترهای درخواست بیاید. به صورت پیش فرض نیز سامانه به آدرس Account/Login هدایت می شود. همچنین وجود پارامتر در این آدرس دهی کاملاً Optional است و می توان آن را در آدرس دهی دخالت نداد.

• Startup.Auth

در این فایل تنظیمات ابتدایی ماژول Identity انجام می شود. این کلاس شامل متدی با نام ConfigureAuth است که تنظیمات اصلی در آن انجام می پذیرد. هنگامیکه سامانه برای اولین بار اجرا می شود این متد اجرا و سامانه آماده استفاده از متدهای عضویت می شود.

• WebApiConfig

در این کلاس تنظیمات مربوط به نحوه آدرس دهی Api ها انجام می شود. این کلاس شامل متدی با نام Register است که تنظیمات Route در آن انجام می پذیرد.

```
config.MapHttpAttributeRoutes();
```

```
    config.Routes.MapHttpRoute(  
        name: "DefaultApi",  
        routeTemplate: "api/{controller}/{id}",  
        defaults: new { id = RouteParameter.Optional }  
    );
```

فصل دوم

نگهداری سامانه طرح غدیر

۲. نگهداری سامانه طرح غدیر

۲-۱. صدور کارت های عضویت

عضویت در طرح غدیر شامل سه بخش می باشد که عبارتند از:

- پیش ثبت نام (توسط نماینده کتابخانه (مبدا) و در هنگام مراجعه حضوری کاربر)
- تکمیل اطلاعات و پرداخت الکترونیکی (توسط متقاضی)
- بررسی، تایید و صدور کارت های عضویت و امانت (توسط ایرانداک)

۲-۲. پیش ثبت نام

به منظور شروع فرایند عضویت در غدیر، لازم است تا متقاضی به کتابخانه مراجعه کرده و پس از ارائه اطلاعات اولیه به کارشناس و ثبت آنها در سامانه، یک نام کاربری و رمز عبور دریافت کند. نام کاربری و گذرواژه توسط سامانه ایجاد و به رایانامه و تلفن همراه کاربر پیامک می شود.

پس از افزودن، امکان ویرایش نیست.

وضعیت

☒ دانشجو ☐ هیئت علمی ☐ سایر

شماره تلغن همراه

رایانامه

شناسه ملی

دانشکده / پژوهشکده

▼

دانشکده ادبیات و علوم انسانی

مقطع

☐ کاردانی ☐ کارشناسی ☒ کارشناسی ارشد ☐ دکترای حرفه‌ای ☐ دانشجویی ☐ دکترای تخصصی

سال ورود

▼

۱۳۹۴

نیم سال

☒ اول ☐ دوم

بازگشت

افزودن

فرم پیش ثبت نام رابط اشکل -۰۰

تکمیل اطلاعات ثبت نامی توسط کاربر:

متقاضی با استفاده از نام کاربری و رمز عبور که در مرحله قبل ایجاد شده است وارد سیستم می شود و اقدام به تکمیل اطلاعات خود می نماید. این بخش از پنج گام تشکیل شده است که کاربر می تواند در بازه های زمانی متفاوت نسبت به تکمیل آنها اقدام نماید.

• گام اول:

در گام اول کاربر اطلاعاتی را نظیر نام، نام خانوادگی و شماره ملی خود را وارد می کند. تمامی اطلاعات در شکل ۱-۲ نمایش داده شده است. لازم به توضیح است که در این فرآیند راهکارهایی طراحی شده است که در صورتی که کاربر به دلایلی موفق به ادامه تکمیل اطلاعات خود نشد، هر زمان که وارد سامانه شود می تواند از ادامه کار به تکمیل اطلاعات بپردازد و هیچ گونه از اطلاعات وارد شده توسط کاربر در هر مرحله از دست نخواهد رفت.

مقررات	درخواست عضویت	بارگذاری عکس	پرداخت آنلاین	پیش نمایش	پذیرش ایرانداک	پذیرش میدا
رایانامه	hadi.louie85@gmail.com					
شماره تلفن همراه	۰۹۳۵۴۱۳۰۶۷۵					
شناسه ملی	۱۴۵۲۱۴۵۸۹					
نام						
نام خانوادگی						
نام پدر						
تاریخ تولد	سال / ماه / روز					
شماره شناسنامه						
جنس	☒ مرد ☐ زن					
گروه	انتخاب کنید					
رشته	انتخاب کنید					
گرایش	انتخاب کنید					
	☐ رشته تحصیلی، گرایش پیدا نشد.					
دانشکده / پژوهشکده	دانشکده فنی و مهندسی					
مقطع تحصیلی	کارشناسی ارشد					
سال ورود	۱۳۹۳					
نیم سال	اول					
نشانی پستی						
کد پستی						

شکل ۲-۰ گام اول در تکمیل اطلاعات کاربر

- گام دوم

در گام دوم کاربر عکس خود را برای استفاده در کارت طرح عضویت غدیر بارگزاری می‌کند. در این گام کنترل‌هایی از قبیل اندازه، نسبت، و حجم بر روی تصویر اعمال می‌شود. به این ترتیب با بهره‌گیری از این کنترل‌ها سعی بر آن خواهد شد که تصویر ارسال‌شده از سوی کاربر دارای حداکثر کیفیت لازم برای استفاده در کارت عضویت را داشته باشد.

The screenshot shows a user profile page with a progress bar at the top. The progress bar has seven steps: 'مقررات' (Completed), 'درخواست عضویت' (Completed), 'بارگذاری عکس' (Current step), 'پرداخت آنلاین' (Pending), 'پیش نمایش' (Pending), 'پذیرش ایرادها' (Pending), and 'پذیرش میما' (Pending). Below the progress bar, there are three instructions: 1. 'اندازه عکس باید سه در چهار باشد.' (Image size should be 3x4), 2. 'فرمت مجاز Jpg,Png,Gif است.' (Allowed formats are Jpg, Png, Gif), and 3. 'حجم مجاز ۲۰۰ کیلوبایت است.' (Allowed size is 200 KB). Below these instructions is a large box for the profile picture. On the left, there is a small thumbnail of the uploaded image labeled 'man 001.jpg'. On the right, there is a text input field containing 'man 001.jpg'. Below the image box, there are three buttons: 'انتخاب عکس ...' (Select image ...), 'ذخیره' (Save), and 'حذف' (Delete). At the bottom left, there is a red button labeled 'گام پیشین' (Previous step).

شکل ۳-۰ گام دوم - بارگزاری تصویر

- گام سوم

پس از بارگذاری عکس، در این گام شماره فاکتوری برای انجام مرحله پرداخت آنلاین به کاربر نمایش داده می‌شود. در صورتی که در هر زمان از فرآیند پرداخت آنلاین مشکلی در پردازش پدید آید، از طریق این شماره فاکتور می‌توان، مشکل به وجود آمده را رهگیری نمود.

The screenshot shows the same user profile page as before, but now the 'پرداخت آنلاین' (Online payment) step is completed. The progress bar shows 'بارگذاری عکس' (Completed) and 'پرداخت آنلاین' (Completed). Below the progress bar, there are two input fields: 'شماره فاکتور' (Invoice number) with the value '۸۴۱۸۹۶۲۲' and 'مبلغ پرداخت' (Payment amount) with the value '۱۰۰۰'. Below these fields, there are two buttons: 'گام پیشین' (Previous step) and 'پرداخت آنلاین' (Online payment).

شکل ۴-۰۲ گام سوم - نمایش کد فاکتور پرداخت

- گام چهارم

می‌توانید شماره‌ی فاکتور را برای پی‌گیری پرداخت یادداشت کنید.

اطلاعات پرداخت

سامانه	سامانه غدیر پارسا
شماره‌ی فاکتور	7809407
نام کاربری	—
رایانامه	hadi.louie85@gmail.com
مبلغ (ریال)	1000
انتخاب بانک	بانک ملی ایران ▼

تأیید و پرداخت بازگشت و انصراف

شکل ۵-۰۰ گام چهارم - اتصال به سامانه پرداخت آنلاین

- گام پنجم

در این مرحله نمایی از اطلاعاتی که کاربر ثبت کرده است نشان داده می‌شود و اطلاع‌رسانی نیز می‌گردد که درخواست عضویت وی به ایرانداک ارسال شده است و باید منتظر تایید ایرانداک باشد. کاربر می‌تواند تا قبل از تایید ایرانداک، اطلاعات ثبت شده خود را ویرایش نماید.

مقررات درخواست عضویت بارگذاری عکس پرداخت آنلاین پیش نمایش پذیرش ایرانداک پذیرش مبدأ

	
نام	علی اصغر
نام خانوادگی	حجت پناه
نام پدر	محمد
رایانامه	hadi.louie85@gmail.com
شماره تلفن همراه	۰۹۳۵۴۱۳۰۶۷۵
شناسه ملی	۱۴۵۲۱۴۵۸۹
دانشکده / پژوهشکده	دانشکده فنی و مهندسی
مقطع تحصیلی	کارشناسی ارشد
گروه	-
رشته	
گرایش	-
سال ورود	۱۳۹۳
نیم سال	اول
روز	۱۳۷۴/۳/۲
کد پستی	۱۱۱۱۱
نشانی پستی	تهران - شریعتی

۱ ثبت نام تکمیل شد. تا قبل از پذیرش ایرانداک امکان ویرایش مراحل قبل وجود دارد.

شکل ۶۰۰ گام پنجم - نمایش تمامی اطلاعات تکمیل شده و تایید نهایی

فصل سوم

استقرار سامانه طرح غدیر

۳. استقرار سامانه طرح غدیر

۳-۱. سوری سامانه

سوری سامانه شامل ۶ پروژه از نوع Class Library است که با یکدیگر ارتباط دارند و تحت یک پروژه Solution با نام Ghadir جمع شده اند.

- Class Library Project با نام (DAL)Day Access Layer

این پروژه نقش ارتباط کلاس ها با پایگاه داده را بر عهده دارد.

- Class Library Project با نام GhadirParsa

این پروژه اصلی سامانه است که تمام کدهای کاربردی سامانه در آن قرار دارد. فایل Web.Config که شامل تنظیمات سامانه است در این پروژه قرار گرفته است.

- Class Library Project با نام Irandoc

این پروژه شامل تمام متدها و توابع کمکی در کل سامانه است. از این پروژه در تمام پروژه های دیگر سامانه استفاده شده است.

- Class Library Project با نام Models

این پروژه نقش نگاشت کلاس ها به جداول پایگاه داده را دارد. تغییرات در این پروژه به صورت مستقیم در پایگاه داده اعمال می شود. پروژه DAL از طریق این پروژه درخواست های داده ای، ذخیره ای و بروزرسانی در پایگاه داده را انجام می دهد.

- Class Library Project با نام Resources

این پروژه شامل تمام Caption های سامانه است که به صورت دو زبانه طراحی شده است.

- Class Library Project با نام ViewModels

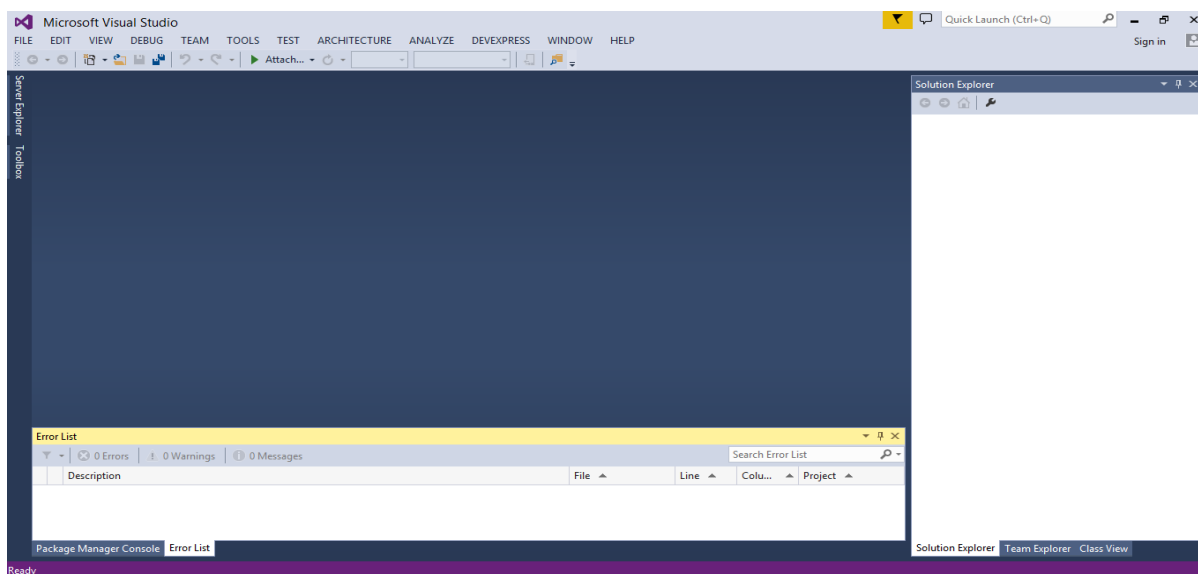
این پروژه شامل تمام کلاس‌هاییست که برای ارتباط با بخش View در پروژه اصلی (Ghadir) به کار می‌رود.

۲-۳. روش باز نمودن سورس

سورس به پیوست مستندات آمده است. برای باز نمودن سورس برنامه نیاز به برنامه ۲۰۱۳ Visual Studio است.

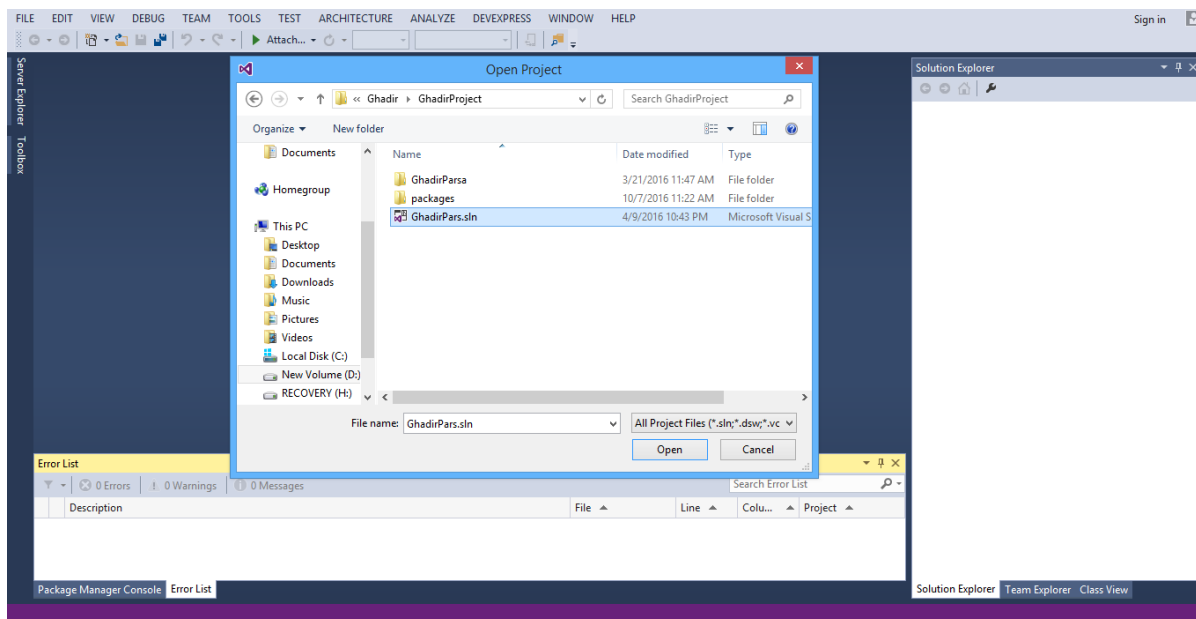
برای باز کردن سورس کفایت مراحل زیر انجام گیرد:

۱. باز کردن برنامه ۲۰۱۳ Visual Studio

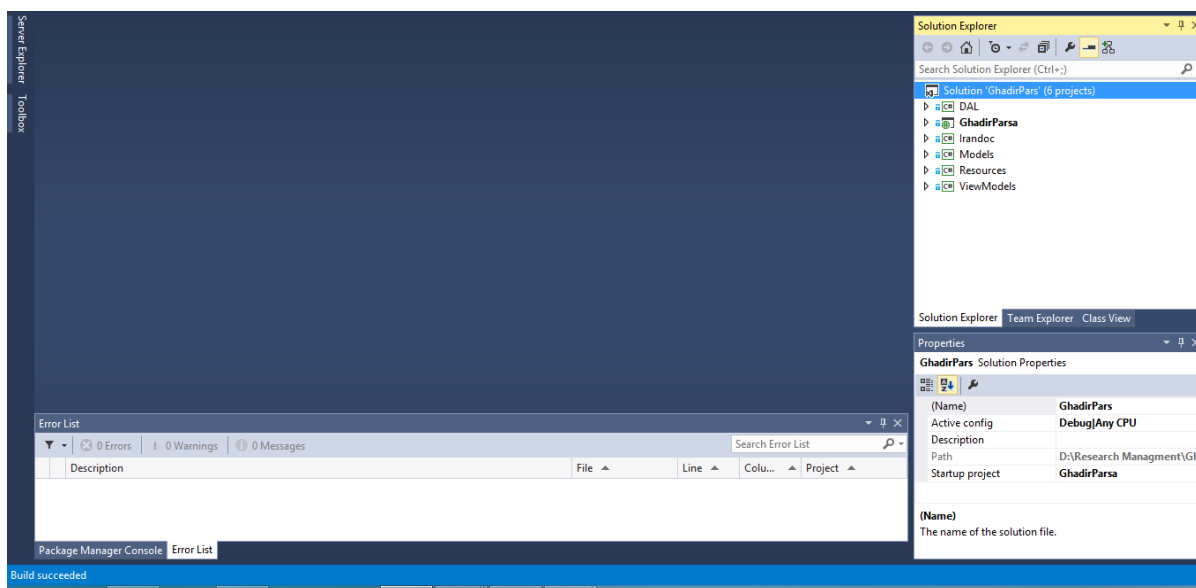


۲. باز کردن فایل Solution سامانه از طریق منوی File و زدن گزینه Open Project

بازطراحی و روزآوری سامانه عضویت فراگیر کتابخانه ها □ ۷۰



۳. کلیک سمت راست بروی Solution سامانه و زدن گزینه Build Solution



• فایل Web.Config

تمام تنظیمات سامانه در این فایل قرار دارد. این تنظیمات شامل کلیدهای مورد استفاده در سامانه، تنظیمات پایگاه داده، ارسال رایانامه، ارسال پیامک و ... است.

مسیر این فایل در ریشه فولدر اصلی سامانه با نام GhadirPars قرار داد که نمایی از آن نمایش داده شده است.

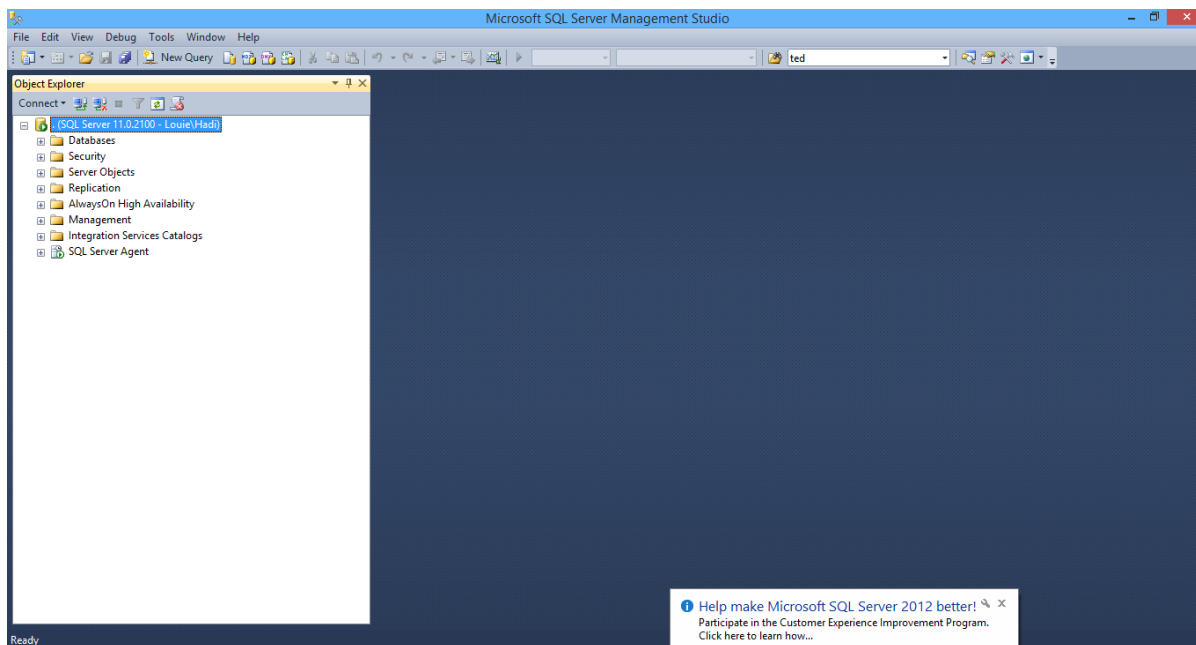
بازطراحی و روزآوری سامانه عضویت فراگیر کتابخانه ها □ ۷۱



۳-۳. پایگاه داده سامانه

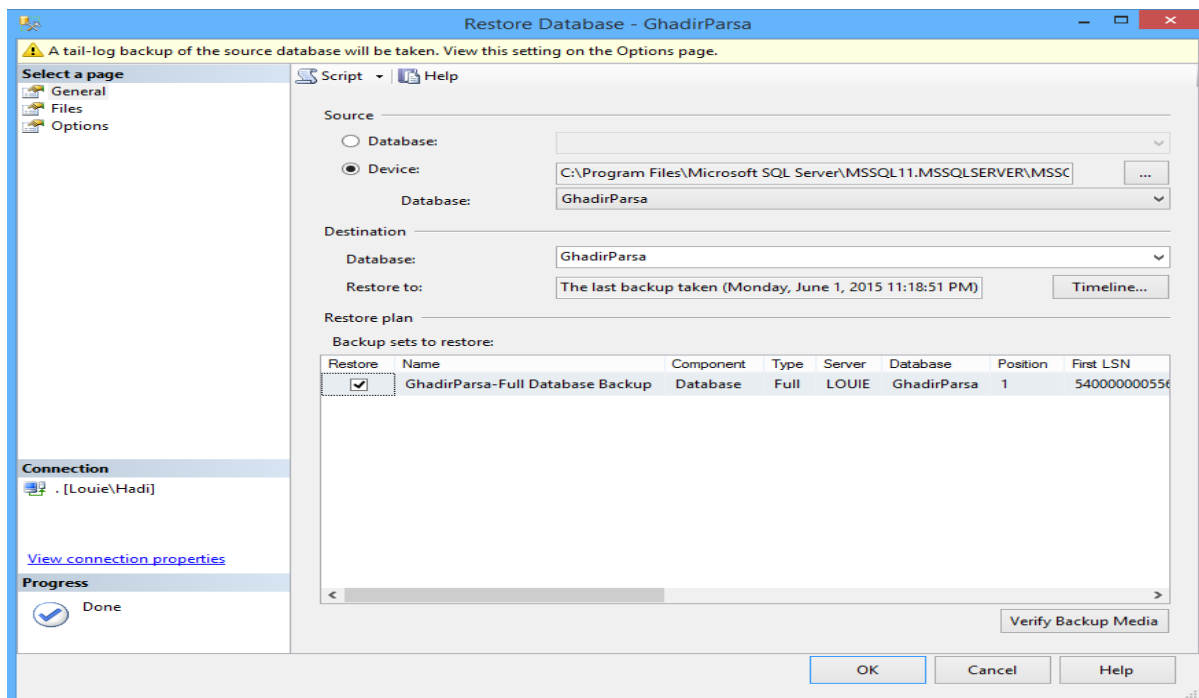
این سامانه از ۲۰۱۲ Sql Server به عنوان موتور پایگاه داده استفاده می کند. برای راه اندازی پایگاه داده کافیت از فایل پشتیبانی که در پیوست آمده است استفاده و مراحل زیر را طی کرد.

۱. باز کردن برنامه Sql Server Management Studio



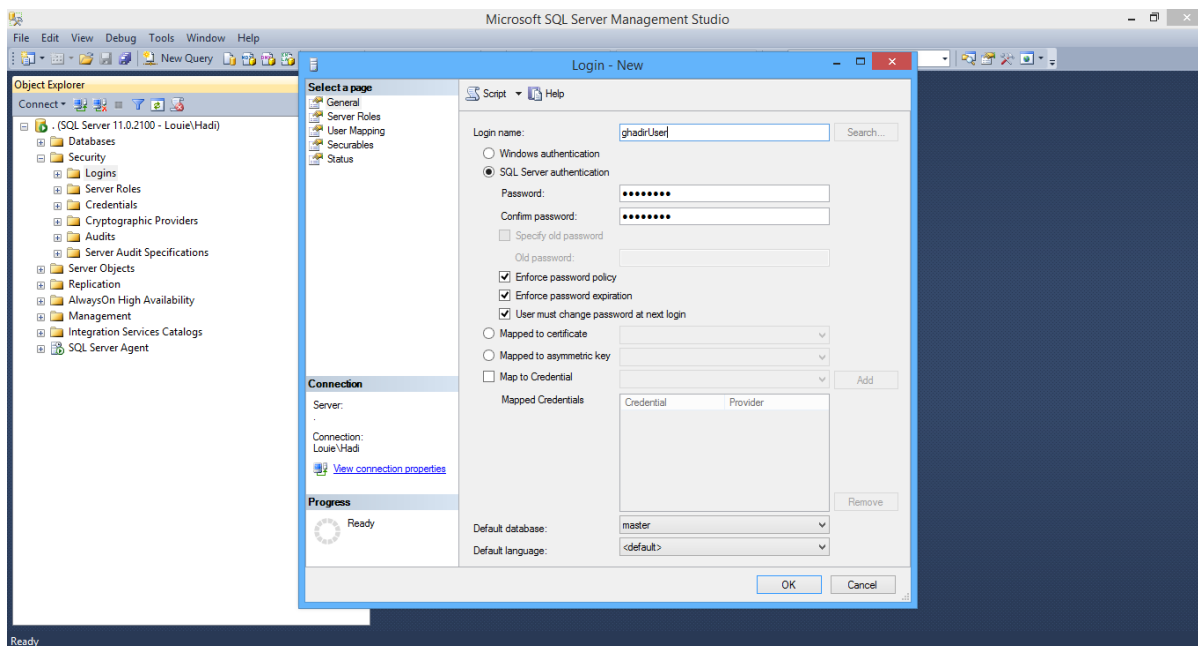
۲. کلیک سمت راست بروی Databases و زدن گزینه Restore Databases

بازطراحی و روزآوری سامانه عضویت فراگیر کتابخانه ها □ ۷۲



۳. ساختن کاربر اتصال به پایگاه داده از طریق زدن گزینه Security و سپس Logins و در نهایت

New Login



توجه شود که در قسمت Server Roles باید دسترسی های متناسب با موتور پایگاه داده و سروری که سامانه در آن قرار گیرد، داده شود.

پس از Restore کردن پایگاه داده کافیت در فایل Web.Config سامانه که توضیح آن در قسمت قبل داده شد، تنظیم شود.

در فایل Web.Config بخشی با عنوان connectionString وجود دارد که وظیفه تنظیم نحوه اتصال به پایگاه داده را بر عهده دارد. این تنظیمات بر اساس Restore کردن پایگاه داده و کاربری که تعریف شده است، امکان اتصال سامانه به پایگاه داده را فراهم می‌کند. این تنظیم برای نمونه در زیر آمده است:

```
<connectionStrings>
    <add name="GhadirContext" connectionString="data source=.;initial
catalog=GhadirParsa;
User
ID=ghadirUser;password=*****;App=EntityFramework;MultipleActiveResultSets=
true" providerName="System.Data.SqlClient" />
</connectionStrings>
```

۳-۴. نصب DotNet Framework نسخه ۴,۵,۱

به طور کلی Net Framework مجموعه‌ای از فایل‌های مورد نیاز سیستم عامل (شامل فایل‌های DLL و رجیستری و واسطه‌های استاندارد ارتباط برنامه‌ها بایکدیگر) است که برای اجرای برنامه‌های نوشته شده تحت دات نت ضروری می‌باشد. یعنی یک پکیج کامل از تمام dll های مورد نیاز برنامه‌هایی که با خود دات نت نوشته شده اند. برای نصب این Framework کافیت فایل Exe آن را اجرا کرده و مراحل نصب تا انتها انجام شود. پس از نصب تمام فایل‌های مورد نیاز برای استفاده در سامانه ایجاد می‌شود.

۳-۵. تنظیم وب سرور

این سامانه بروی وب سرور Internet Information Service(IIS ۷) قرار می‌گیرد.

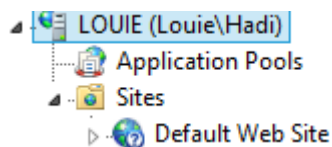
سرویس IIS که مخفف Internet Information Services می‌باشد سرویس‌های اطلاعات اینترنتی که توسط شرکت Microsoft عرضه شده و یک سرور برای کنترل کردن محتویات و دسترسی به سایت‌های وب یا FTP شما بر روی هارد ایجاد می‌کند. برای مثال هنگامی که شما می‌خواهید سایتتان را منتشر کنید قبل از upload کردن آن می‌خواهید آن را آزمایش کنید و اگر با asp طراحی می‌کنید قبل از نصب Visual Studio.Net بهتر است این سرویس را نصب کنید و گرنه مشکلاتی را برای شما به همراه خواهد داشت.

برای گذاشتن سورس برنامه روی وب سرور کافیت مراحل زیر را انجام شود:

۱. رفتن به قسمت Control Panel

۲. انتخاب گزینه Administrator Tools

۳. انتخاب گزینه Internet Information Services (IIS) Manager
۴. کلیک سمت راست بروی Sites و انتخاب Add New website
۵. در این قسمت نام سایت مورد نظر وارد شده و بعد مسیر فیزیکی سورس برنامه انتخاب شود و در نهایت پورتهی که سامانه قرار است بروی بالا بیاید، انتخاب می شود.
۶. با زدن روی گزینه Ok سایت ایجاد می شود و می توان از طریق کلید Browse سایت را بروی مرورگر مشاهده کرد.



The screenshot shows the 'Add Website' dialog box in IIS Manager. The 'Site name' is 'GhadirParsa', the 'Application pool' is 'GhadirParsa', and the 'Physical path' is 'D:\Research Management\Ghadir\GhadirProject\GhadirPi'. The 'Binding' section shows 'Type' as 'http', 'IP address' as 'All Unassigned', and 'Port' as '80'. The 'Host name' field is empty, and an example is provided: 'www.contoso.com or marketing.contoso.com'. The 'Start Website immediately' checkbox is checked.

۳-۶. تنظیم Mail Server برای ارسال رایانامه

برای ارسال رایانامه‌ها در سامانه باید دو مورد را در نظر داشت:

- تنظیم اطلاعات Mail server در Web.Config برنامه
- دریافت ناگت (Nuget) با نام Postal برای ارسال رایانامه

برای تنظیم اطلاعات Mail Server باید به شکل زیر عمل شود:

```
<system.net>  
  <mailSettings>
```

```
<smtp from="ghadir@email.irandoc.ac.ir">
    <network      host="*.irandoc.ac.ir"      password="*****"
      userName="**@*.irandoc.ac.ir" port="25" />
</smtp>
</mailSettings>
</system.net>
```

برای ارسال رایانامه کفایت در کلاس CommonClasses.Utility یک تابع برای ارسال رایانامه تعریف شود که نمونه آن در زیر آمده است:

```
public void SendEmailForAllDocPrinted(string to, string dest, string pathAttach1,
string pathAttach2)
{
    dynamic email = new Email("AllDocPrinted");
    email.To = to;
    email.Dest = dest;
    email.Attach(new Attachment(pathAttach1));
    email.Attach(new Attachment(pathAttach2));
    email.Send();
}
```

در این تابع اطلاعات لازم برای ارسال رایانامه به همراه View مورد نظر برای ارسال رایانامه مشخص می‌شود.

لازم است که ذکر شود در پوشه Views فولدیری با نام Emails وجود دارد که وظیفه تولید محتوای ارسال رایانامه را بر عهده دارد.

می‌توان یک نمونه View ارسال رایانامه را در زیر مشاهده کرد.

نکته: برای Attach کردن فایل به رایانامه ارسالی کفایت از دستور زیر به هنگام ارسال رایانامه استفاده شود:

```
email.Attach(new Attachment(pathAttach1));
```

بازطراحی و روزآوری سامانه عضویت فراگیر کتابخانه‌ها □ ۷۷

To: @ViewBag.To

From: Ghadir@irandoc.ac.ir

Subject: عضویت کارت دریافت

<div>

.بروید @ViewBag.Dest به دیگر روز سه غدیر طرح عضویت کارت دریافت برای

</div>

۳-۷. تنظیم ارسال پیامک

برای ارسال پیامک‌ها در سامانه باید دو مورد را در نظر داشت:

- تنظیم اطلاعات لازم برای ارسال پیامک در Web.Config سامانه
- افزودن وب سرویس ارسال پیامک

ابتدا باید اطلاعاتی که از Provider ارسال پیامک گرفته شده است در داخل Web.Config تنظیم شود. این اطلاعات به صورت کلیدهایی در این قسمت تنظیم می‌شوند

```
<add key="SmsUsername" value="irandoc" />
<add key="SmsPassword" value="*****" />
<add key="SmsSender" value="300079334432263" />
```

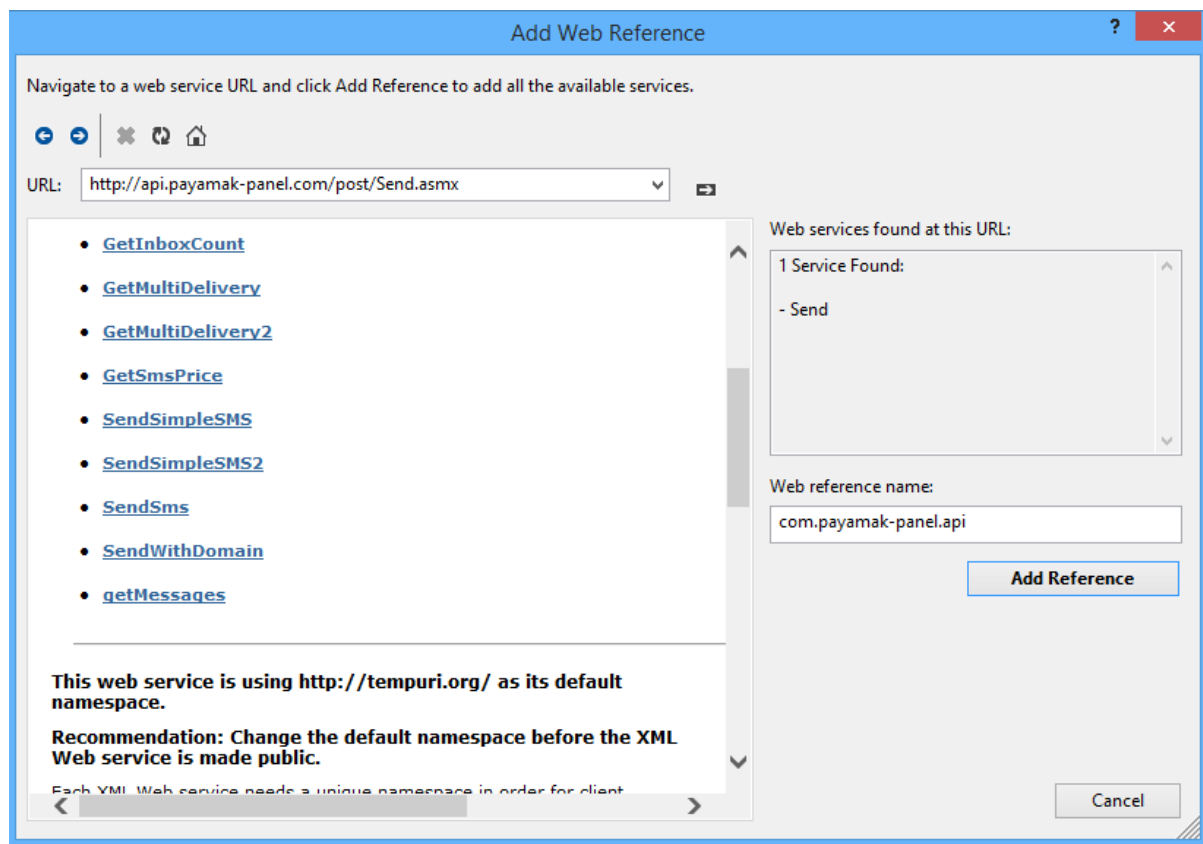
پس از این تنظیمات باید وب سرویس ارسال پیامک را طریق مراحل زیر فعال نماییم:

۱. کلیک سمت راست بروی پروژه اصلی برنامه
۲. انتخاب گزینه Add References
۳. انتخاب گزینه Add Web References
۴. ورود آدرس وب سرویس در قسمت URL و زدن گزینه Go

<http://api.payamak-panel.com/post/Send.aspx>

۵. ظاهر شدن تمام متدهای لازم برای ارسال پیامک از جمله متد مهم SendSms

۶. تنظیم تابع مناسب برای استفاده از این وب سرویس در قسمت CommonClasses.Utility



نمونه‌ای از کد ارسال پیامک را می‌توان در زیر مشاهده کرد:

```
public int SendSms(string username,string password,
                  string[] phones, string sender ,string text,bool
                  isFlash,ref long[] recIn,ref byte[] statusIn)
{
    WebServices.Send sms = new WebServices.Send();
```

```
long[] rec = null;
byte[] status = null;

int condition= sms.SendSms(username, password, phones,sender,
text, isFlash, "", ref rec, ref status);

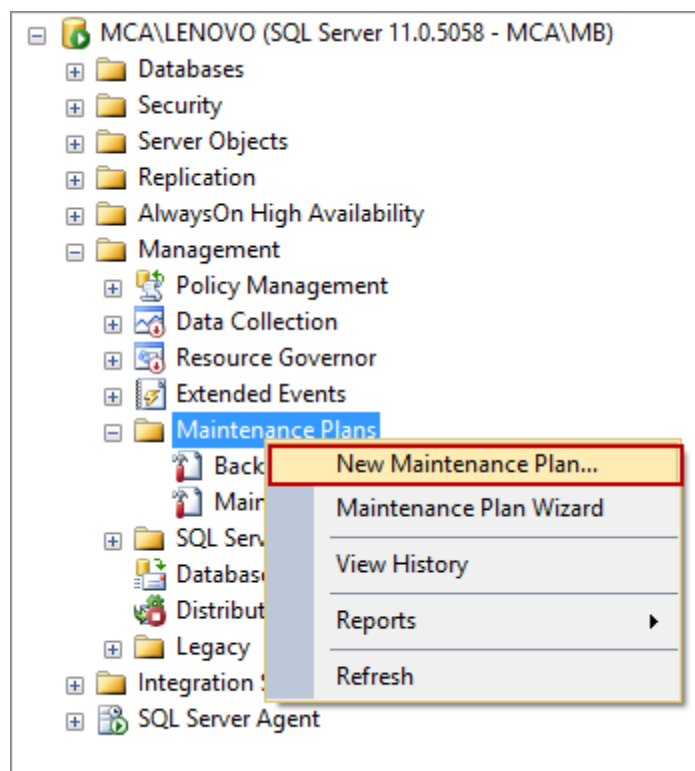
recIn = rec;
statusIn = status;

return condition;
}
```

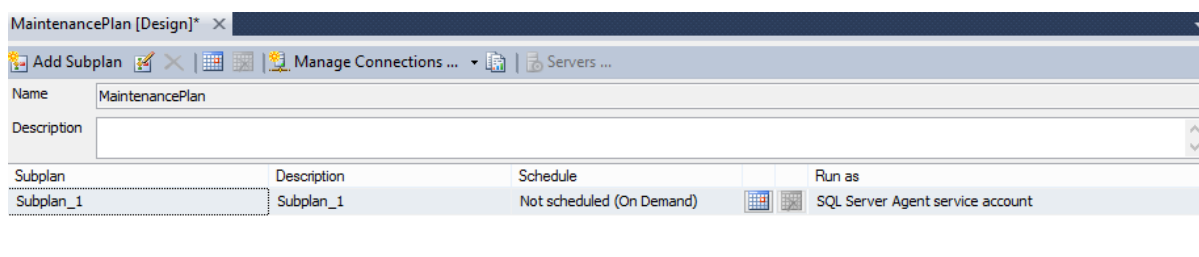
۳-۸. تنظیم پشتیبان‌گیری از سامانه

برای پشتیبان‌گیری خودکار پایگاه داده سامانه باید مراحل زیر را طی کرد.

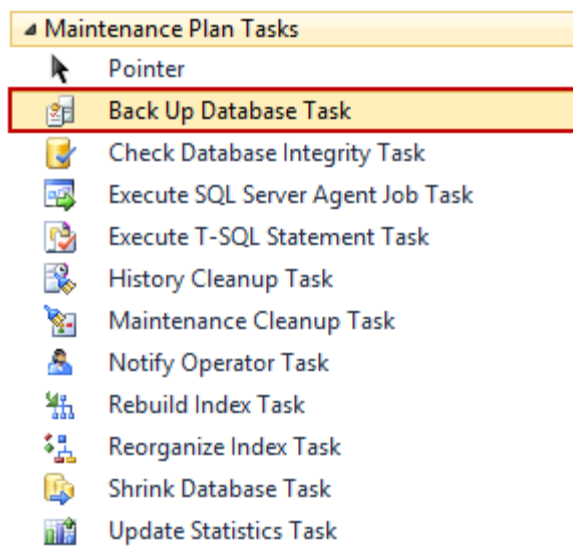
۱. پس از باز کردن Sql Server Management Studio و رفتن به قسمت Object Explore،
بر روی گزینه Management کلیک کرده و گزینه Maintenance Plans را انتخاب می‌کنیم. پس از آن روی
گزینه New Maintenance Plan کلیک می‌کنیم.



نرم‌افزار Sql Server بخشی را برای تنظیمات پشتیبان‌گیری به همراه جعبه ابزاری برای کنترل و زمانبندی در اختیار می‌گذارد.



۲. در بخش Maintenance Plan Tasks گزینه Backup Database Task را انتخاب می‌کنیم.



۳. دوباره بروی گزینه آمده روی صفحه کلیک می‌کنیم و صفحه‌ای مشابه صفحه زیر ظاهر می‌گردد:

Back Up Database Task

Connection: Local server connection New...

Backup type: Full

Database(s): Specific databases

Backup component

☒ Database

☐ Files and filegroups: ...

☐ Copy-only Backup

☐ For availability databases, ignore Replica Priority for Backup and Backup on Primary Settings

☐ Backup set will expire:

☒ After 14 days

☐ On 3/27/2015

Back up to: ☒ Disk ☐ Tape

☐ Back up databases across one or more files:

...

Add...

Remove

Contents

If backup files exist: Append

☒ Create a backup file for every database

☒ Create a sub-directory for each database

Folder: F:\Backup ...

Backup file extension: bak

☒ Verify backup integrity

Set backup compression: Use the default server setting

OK Cancel View T-SQL Help

در این قسمت محلی که باید پشتیبان گرفته شود و زمانبندی‌های لازم انجام می‌شود. به عنوان مثال می‌توان تنظیم کرد هر هفته در یک ساعت خاص پشتیبان گرفته شود. لازم به ذکر است که امکان پاک کردن پشتیبان نیز قابل زمانبندی است.

۹-۳. نصب فونت گزارش‌های سامانه

گزارش‌های سامانه از فونت B Mitra استفاده می‌کنند. برای نصب فونت بروی سرور باید به بخش Control Panel رفته و پوشه Fonts انتخاب شود. و فونت مورد نظر در آن قسمت کپی و نصب شود. لازم به ذکر است برای اینکه فونت بروی گزارش‌های اعمال شود، نیاز است که سرور Restart شود.

منابع

- بیرامی، حمیده و سیروس علیدوستی. ۱۳۹۳. نیازمندی‌های فیزیکی سیستم جدید طرح غدیر. تهران: پژوهشگاه علوم و فناوری اطلاعات ایران.
- علیدوستی، سیروس. ۱۳۷۸. گزارش مستندات نرم افزار سامانه کنونی غدیر. تهران: مرکز اطلاعات و مدارک علمی ایران.
- بیرامی، حمیده. ۱۳۹. علیدوستی، سیروس، و اصغر اسدی. ۱۳۸۵. ارزیابی کیفیت طرح غدیر - عضویت فراگیر کتابخانه‌ها. گزارش طرح پژوهشی. تهران: پژوهشگاه اطلاعات و مدارک علمی ایران.
- پناهی، سیروس. ۱۳۸۳. ارزیابی طرح غدیر از دیدگاه کاربران (تهران - ۱۳۸۳) و ارائه پیش‌نویس الگویی به وزارت بهداشت، درمان، و آموزش پزشکی. پایان‌نامه کارشناسی ارشد، دانشگاه علوم پزشکی و خدمات بهداشتی و درمانی ایران.
- بیرامی طارونی، حمیده، و سیروس علیدوستی. ۱۳۸۱. طراحی الگوی توصیف آماری عملکرد طرح غدیر - عضویت فراگیر کتابخانه‌ها. گزارش طرح پژوهشی. تهران: مرکز اطلاعات و مدارک علمی ایران. ۳. گزارش شکست نرم افزار طرح غدیر ۲. تهران: پژوهشگاه علوم و فناوری اطلاعات ایران.
- ماتسیچ، پال دلیو، و باربارا آر. مونسی. ۱۹۹۴. عوامل موفقیت همکاری. ترجمه مریم نظری و سیروس علیدوستی. ۱۳۸۱. تهران: مرکز اطلاعات و مدارک علمی ایران.
- علیدوستی، سیروس. ۱۳۸۰. اشتراک منابع و همکاری بین کتابخانه‌ها. تهران: مرکز اطلاعات و مدارک علمی ایران.
- نظری، مریم، و سیروس علیدوستی. ۱۳۸۰. مبانی و عملکرد طرح تعمیم خدمات کتابخانه‌های تخصصی به افراد غیرعضو. تهران: مرکز اطلاعات و مدارک علمی ایران.
- علیدوستی، سیروس، اصغر اسدی، و محمود خسروجردی. ۱۳۸۹. ارزیابی طرح غدیر (عضویت فراگیر کتابخانه‌ها) از دیدگاه کتابخانه‌های همکار: کاربرد «سروکوآل». علوم و فناوری اطلاعات: ویژه‌نامه مدیریت کتابخانه‌ها و مراکز اطلاع‌رسانی (پاییز): ۱۲۷-۱۴۹.

Abstract

The main objective of Redesigning Ghadir Resource Sharing System project is to establish direct and personal access to library resources covered Ministry of Science, Research and Technology for faculty and graduate students and is coordinated through a formal system. Date comprehensive plan to reform the system of libraries (Ghadir Plan) pays.

Keywords: Ghadir project, comprehensive membership libraries, library management, library services



Iranian Research Institute for Information Science and Technology

Research Project

Redesigning Ghadir Resource Sharing System

By:

Mohammad Mehdi Aghalouie

Hadi Aghalouie

Aliasghar Hojjatpanah

Mojtaba Zali

Winter ۲۰۱۶