

پروژه شبیه‌سازی عابر بانک

Automatic Teller Machine

پروژه شبیه‌سازی محیط و برخی از عملکردهای عابر بانک

پروژه شبیه سازی عابر بانک

رشته تحصیلی: مهندسی تکنولوژی نرم افزار کامپیوتر

عنوان درس: آزمایشگاه مهندسی نرم افزار

استاد راهنما: آقای مهندس احمدپور

دانشجو: دانیال کمالی، مهدی حسن شاهی

فهرست مطالب:

3 بررسی اجمالی
6 انتخاب زبان
7 بررسی کارت بانکی
8 بررسی رمز کارت
9 منو خدمات
10 موجودی حساب
11 برداشت وجه
12 انتقال وجه
16 تغییر رمز
17 پرداخت قبوض
18 نتایج
22 خروج
23 مدیریت عابر بانک
25 نحوه ارتباط فرم‌ها
27 جمع بندی
28 منابع

بررسی اجمالی:

پروژه شبیه‌سازی عابر بانک، بطور کلی شامل شبیه‌سازی محیط عابر بانک بصورت گرافیکی و پیاده‌سازی برخی از عملکردهای و وظایفی است که توسط یک عابر بانک در اختیار کاربر قرار می‌گیرد.

محیط شبیه‌سازی شده در این پروژه شامل زبان فارسی و انگلیسی است و کاربر پس از انتخاب زبان و وارد کردن مشخصات کارت بانکی شامل شماره شانزده رقمی کارت و کد اعتبارسنجی (CVV2) به صفحه وارد کردن رمز هدایت می‌شود و پس از آن قادر به انتخاب خدمت مورد نظر خواهد بود.

خدمات و عملکردهای پیاده‌سازی شده در این پروژه شامل موارد زیر است:

1. گرفتن موجودی حساب
2. برداشت وجه
3. انتقال وجه
4. تغییر رمز
5. پرداخت قبوض

و همچنین در هر مرحله از برنامه وابسته به خدمت در حال ارائه دکمه خروج، لغو یا بازگشت در دسترس است.

در ادامه مراحل اجرایی برنامه براساس نوع خدمت انتخاب شده همراه با توضیحات پیرامون نحوه عملکرد آنها به همراه عکس توضیح داده می‌شود.

پیش از توضیحات پیرامون مراحل اجرا نرم‌افزار، ابتدا وظیفه چند متغیر سراسری و تاثیر آن‌ها در اجرا نرم‌افزار را توضیح خواهیم داد. (این متغیرها در فایل Program.cs تعریف شده‌اند)

1. متغیر lang

این متغیر از نوع bool بوده و برای ذخیره زبان انتخاب شده توسط کاربر است، اگر مقدار متغیر false شود بدان معناس که کاربر زبان فارسی را انتخاب کرده و مقدار true یعنی کاربر زبان انگلیسی را انتخاب کرده. از این متغیر در همه فرم‌های برنامه استفاده می‌شود.

2. متغیر state

متغیر state از نوع char بوده و برای ذخیره نوع عملکرد انتخاب شده توسط کاربر به کار می‌رود. ذخیره شدن کاراکتر A به معنا انتخاب عملکرد انتقال پول به سایر حساب‌ها، C به معنا انتخاب عملکرد انتقال پول کارت به کارت، کاراکتر B به معنا گرفتن موجودی، W به معنا برداشت وجه، P به معنا تغییر رمز و کاراکتر R به معنا پرداخت قبوض است، هر کدام از کاراکترها هنگامی که کاربر خدمت مورد نظر را انتخاب کند در متغیر state ذخیره خواهند شد تا بعداً با بررسی این متغیر توسط دستورات شرطی if در مرحله نمایش نتایج، عملکرد مربوط به آن خدمت اجرا شود.

3. متغیر accNum

از نوع string تعریف شده و در ادامه برای ذخیره کردن شماره حساب مربوط به کارت بانکی کاربر استفاده خواهد شد.

4. متغیرهای accP1، accP2، accP3 و accP4

این چهار متغیر از نوع string بوده و برای ذخیره شماره کارت کاربر استفاده می‌شوند و در ادامه برای انتخاب شماره حساب کارت بانکی و عملکردهایی همچون کارت به کارت که به شماره کارت مبدا (فرستنده) نیاز است استفاده می‌شوند. دلیل استفاده از چهار متغیر به جای یک متغیر استفاده از چهار textBox برای گرفتن شماره کارت است به منظور شبیه‌سازی هرچه دقیق‌تر کارت بانکی و همچنین شماره کارت در دیتابیس به صورت چهار بخش مجزا ذخیره می‌شود.

همچنین نوار عنوان فرم‌های این پروژه به منظور زیبایی مجدداً طراحی شده و در قسمت کد هر فرم، کدهای مربوط به رویدادهای `panel1_MouseMove`، `panel1_MouseDown`، `panel1_MouseUp`، `minimizeBtn_Click` و `exitBtn_Click` برای حرکت دادن فرم (Move)، کمینه کردن (Minimize) و بستن (Close) فرم است.



تصویر 1: نوار عنوان

مواردی که در بین همه فرم‌ها یکسان است شامل:

1. در تمامی فرم‌ها ابتدا زبان فرم تعیین می‌شود، با توجه به مقدار متغیر `lang` که اگر `true` بود حالت نوشتن راست به چپ `label` غیرفعال شده و متن `label` و دکمه‌ها با متن انگلیسی جایگزین می‌شود، در غیر این صورت متن فارسی جایگزین خواهد شد.
 2. هر `textBox` برای خالی نماندن بررسی می‌شود و اگر کاربر `textBox` را خالی بگذارد یا آن را با `Space` پر کند پیغام هشداری نشان داده می‌شود.
 3. در هنگام نمایش `MessageBox` ابتدا توسط مقدار متغیر `lang` بررسی می‌شود زبان انتخاب شده چیست و سپس تصمیم گرفته می‌شود متن پیغام فارسی نمایش یابد یا متن انگلیسی.
 4. در طول اجرا رویداد `TextChanged` هر `textBox` برای جلوگیری از وارد کردن حروف بجای اعداد بررسی می‌شود.
- در ادامه مراحل اجرای نرم‌افزار (خدمات) براساس ترتیب و نام فرم توضیح داده می‌شود.

1. انتخاب زبان (FormSelectLanguage)

فرم انتخاب زبان دارای دو دکمه برای انتخاب زبان فارسی یا انگلیسی است، که انتخاب هرکدام از آن‌ها مقدار متغیر lang را تغییر می‌دهد. انتخاب زبان فارسی مقدار متغیر lang را false کرده و یا انتخاب زبان انگلیسی مقدار متغیر lang را true کرده و سپس به فرم بررسی کارت بانکی نمایش می‌یابد.



تصویر 2: فرم انتخاب زبان

2. بررسی کارت بانکی (FormCardCheck)

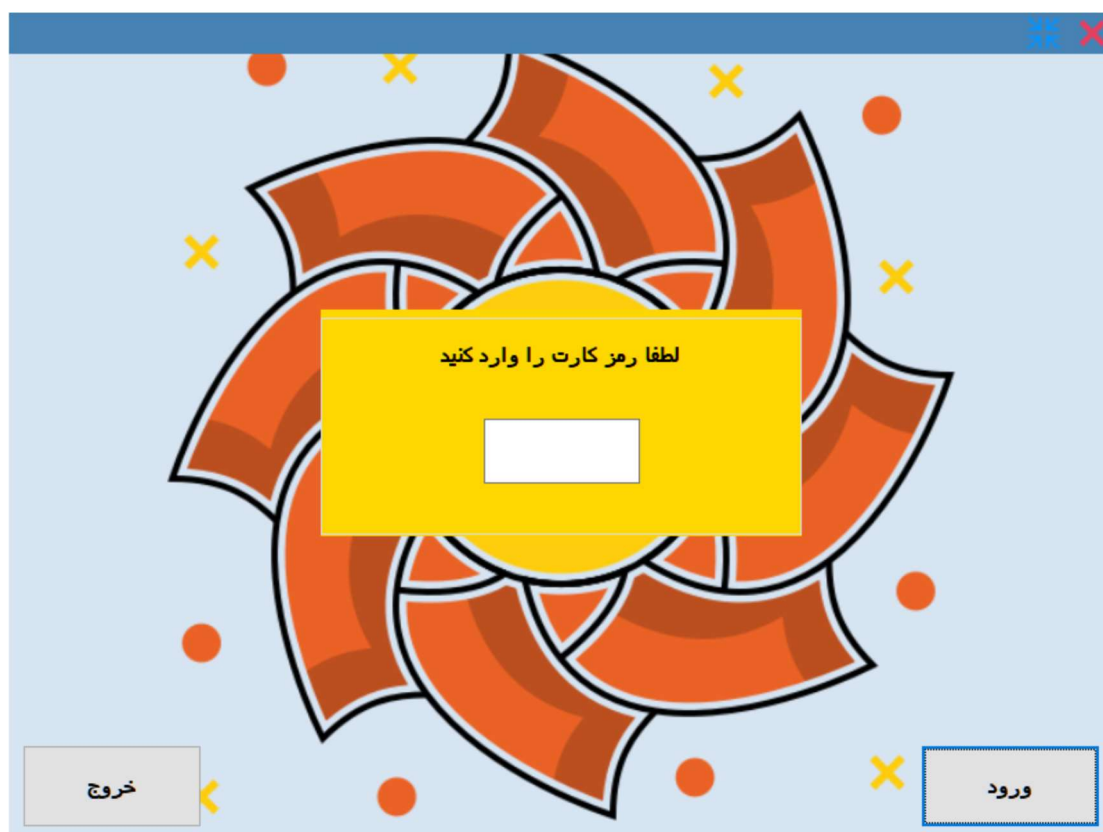
این فرم دارای محیط گرافیکی شبیه به کارت بانکی است و دارای پنج TextBox و دو دکمه است، چهار TextBox به منظور گرفتن شماره کارت و پنجمین TextBox برای دریافت CVV2 استفاده می‌شود، کلیه TextBoxها دارای محدودیت وارد کردن چهار رقم هستند.

دکمه خروج که برای برگشتن به صفحه انتخاب زبان و دکمه ورود برای وارد کردن کارت بانکی است که با اتصال به دیتابیس بررسی می‌شود کارتی با مشخصات وارد شده وجود داشته باشد، اگر وجود داشت مقادیر شماره کارت (به صورت چهار بخش چهار رقمی) و شماره حساب از دیتابیس خوانده شده و در متغیرهای سراسری accP1، accP2، accP3، accP4 (چهار بخش چهار رقمی شماره کارت) و accNum ذخیره می‌شوند و سپس فرم گرفتن و بررسی رمز کارت نمایش می‌یابد و اگر شماره کارت در دیتابیس وجود نداشت پیغام خطا نمایش داده می‌شود.

تصویر 3: فرم دریافت و بررسی کارت بانکی

3. بررسی رمز کارت (FormPasswordCheck)

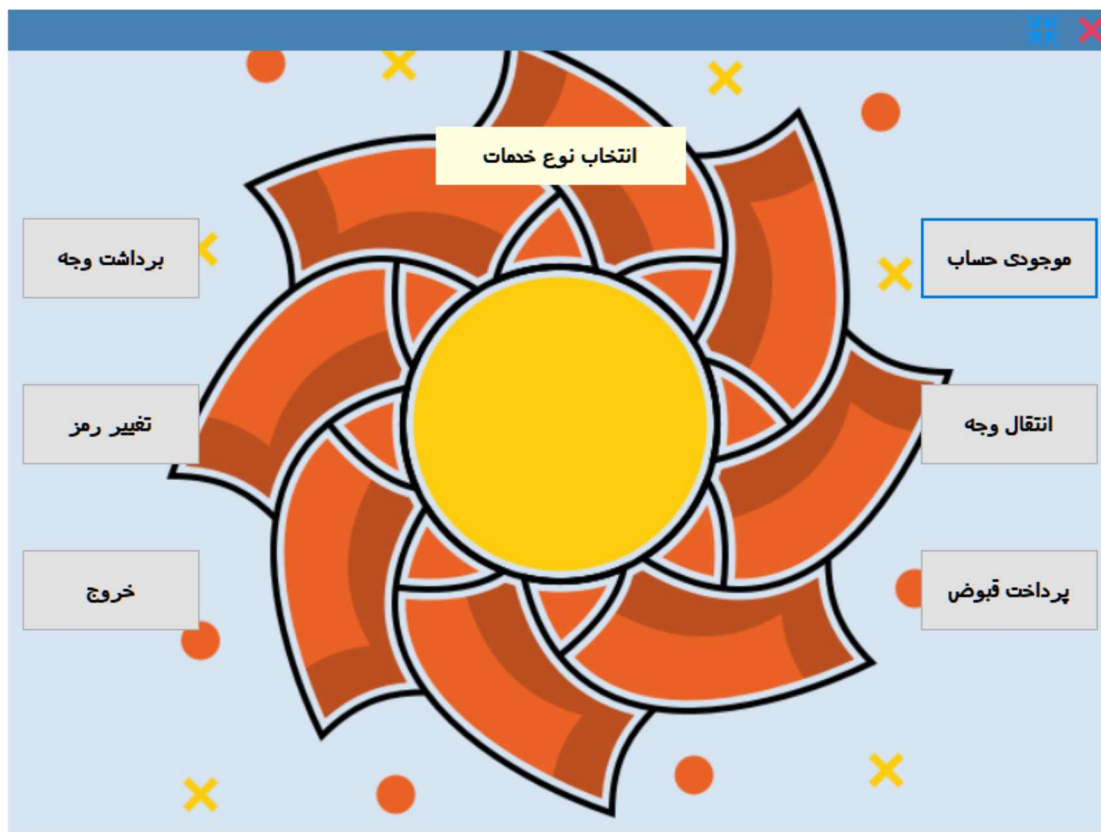
فرم دارای یک textBox با محدودیت وارد کردن چهار رقم و دکمه خروج برای برگشت به فرم انتخاب زبان و دکمه ورود است. پس از انتخاب دکمه ورود با دیتابیس ارتباط برقرار می‌شود و رمز حساب کارت بانکی خوانده شده، سپس رمز با رمزی که در textBox وارد شده مقایسه می‌شود و در صورت مشابه بودن منو خدمات نمایش می‌یابد.



تصویر 4: فرم بررسی رمز کارت بانکی

4. منو خدمات (FormMenu)

منو خدمات شامل شش دکمه موجودی حساب، برداشت وجه، انتقال وجه، تغییر رمز، پرداخت قبوض و خروج است که انتخاب هر کدام از دکمه‌ها کاربر را به خدمت مورد نظر هدایت می‌کند. همچنین در سایر فرم‌ها انتخاب دکمه بازگشت باعث برگشتن به این فرم می‌شود.

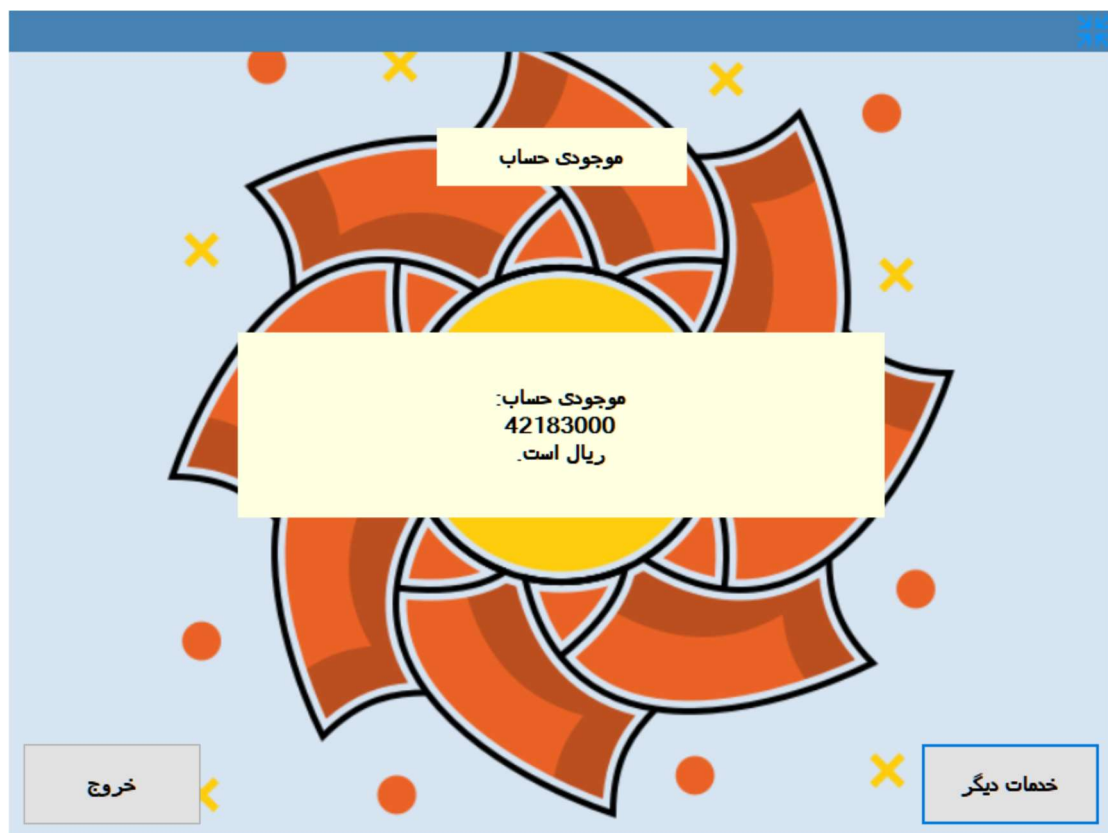


تصویر 5: فرم منو خدمات

5. موجودی حساب

انتخاب دکمه موجودی حساب در منو خدمات مقدار متغیر state را برابر با کاراکتر B می‌کند و سپس فرم نتایج نمایش می‌یابد و موجودی حساب بر روی label اصلی نمایش می‌یابد.

نحوه دریافت و نمایش موجودی حساب در بخش نتایج (FormResults) توضیح داده شده است.



تصویر 6: نمایش موجودی حساب

6. برداشت وجه (FormWithdrawMoney)

این فرم شامل هشت دکمه و یک textBox است، حداکثر تعداد ارقام قابل وارد کردن در textBox برابر با هفت رقم است. شش دکمه دارای مقادیر پیش‌فرض عددی شامل 100000 ریال، 200000 ریال، 300000 ریال، 500000 ریال، 1000000 ریال و 2000000 ریال هستند، چنانچه رقم موردنظر مقداری غیر از مقادیر دکمه‌ها بود با استفاده از textBox می‌توان آن را وارد کرد.

دکمه بازگشت برای رفتن به منو خدمات و دکمه تایید برای ثبت رقم وارد شده است. در این فرم یک متغیر سراسری به نام withdrawMoney از نوع Int64 ایجاد می‌شود و مقدار textBox یا رقم دکمه فشرده شده در آن ذخیره می‌شود و سپس فرم نتایج نمایش می‌یابد. مقدار متغیر withdrawMoney در فرم نتایج مورد استفاده قرار خواهد گرفت.

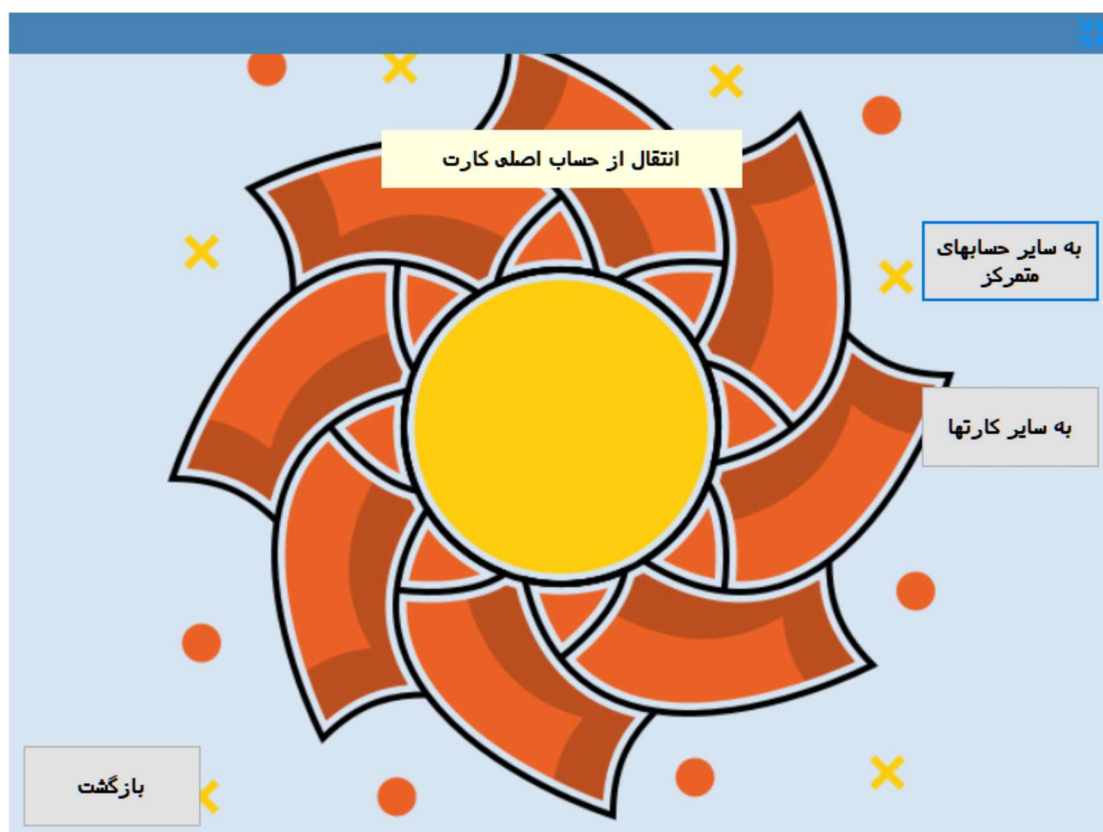
The screenshot shows a user interface for withdrawing money. At the top, there's a title bar. Below it, a yellow box contains the text 'برداشت وجه' (Withdrawal). In the center, a white box with a yellow border contains the text 'مبلغ مورد نظر را انتخاب کنید یا مبلغ را وارد کنید. (ریال)' (Select the amount you want or enter the amount. (Rial)). Below this is a white text input field. Surrounding the central area are eight buttons with the following values: 100,000, 200,000, 300,000, 500,000, 1,000,000, 2,000,000, and a button labeled 'بازگشت' (Back). At the bottom right, there is a button labeled 'تایید' (Confirm). The background is light blue with orange decorative elements.

تصویر 7: فرم برداشت وجه

7. انتقال وجه (FormTransferMoneyMainAcct)

فرم انتقال وجه شامل سه دکمه است، دکمه لغو برای بازگشت به منو خدمات و دکمه‌های به سایر حسابهای متمرکز و به سایر کارتها برای انتخاب نحوه انتقال وجه کاربرد دارند.

با انتخاب دکمه به سایر حسابهای متمرکز کاراکتر A در متغیر state ذخیره می‌شود و با انتخاب دکمه به سایر کارتها کاراکتر C در متغیر state ذخیره می‌شود.



تصویر 8: فرم انتخاب نحوه انتقال وجه از حساب اصلی کارت

در ادامه نتیجه انتخاب هر دو این دکمه‌ها توضیح داده شده است.

1. انتقال وجه به سایر حساب‌های متمرکز (FormTransferMoneyToAcct)

این فرم شامل یک textBox با محدودیت دوازده رقم (در این پروژه شماره حساب‌ها دوازده رقمی در نظر گرفته شده) و دکمه لغو برای بازگشت به منو خدمات و دکمه تایید است.

متغیر سراسری CtoA از نوع string برای ذخیره شماره حساب گیرنده ایجاد شده، پس از انتخاب دکمه تایید بررسی می‌شود حسابی با این شماره در دیتابیس وجود داشته باشد و همچنین شماره حساب وارد شده با شماره حساب کارت بانکی کاربر یکسان نباشد، اگر این شروط صحیح بودند شماره حساب مقصد (گیرنده) در متغیر CtoA برای استفاده در فرم نتایج ذخیره می‌شود و سپس فرم تعیین میزان وجه (FormMoneyAmount) نمایش خواهد یافت.



تصویر 9: فرم انتقال وجه از حساب اصلی کارت به سایر حساب‌ها

2. انتقال وجه به سایر کارت‌ها (FormTransferMoneyToCard)

این فرم شامل چهار textBox با محدودیت چهار رقم و دکمه لغو برای بازگشت به منو خدمات و دکمه تایید است.

متغیرهای سراسری CtoC1، CtoC2، CtoC3 و CtoC4 از نوع string برای ذخیره شماره کارت گیرنده ایجاد شده، پس از انتخاب دکمه تایید بررسی می‌شود کارتی با این شماره در دیتابیس وجود داشته باشد و همچنین شماره کارت وارد شده با شماره کارت کاربر یکسان نباشد، اگر این شروط صحیح بودند شماره کارت مقصد (گیرنده) در متغیرهای CtoC1، CtoC2، CtoC3 و CtoC4 برای استفاده در فرم نتایج ذخیره می‌شود و سپس فرم تعیین میزان وجه (FormMoneyAmount) نمایش خواهد یافت.

تصویر 10: فرم انتقال وجه کارت به کارت

3. تعیین میزان وجه (FormMoneyAmount)

بعد از وارد کردن شماره حساب یا شماره کارت در مرحله قبل در این فرم میزان وجه که باید منتقل شود را می‌توان تعیین کرد. این فرم شامل یک textBox با محدودیت وارد کردن 10 رقم و دکمه لغو برای بازگشت به منو خدمات و دکمه تایید است. متغیر moneyAmount از نوع Int64 در این فرم ایجاد می‌شود، با انتخاب دکمه تایید مقدار textBox در متغیر moneyAmount ذخیره شده و فرم نتایج نمایش می‌یابد. از این متغیر در فرم نتایج استفاده خواهد شد.



تصویر 11: فرم تعیین میزان وجه برای انتقال

8. تغییر رمز (FormChangePwd)

فرم تغییر رمز شامل سه textBox با محدودیت وارد کردن چهار کاراکتر و دکمه لغو برای بازگشت به منو خدمات و دکمه تایید است. با انتخاب دکمه تایید بررسی می‌شود مقدار textBox رمز فعلی برابر با دو textBox دیگر نباشد (رمز جدید نمی‌تواند همانند رمز فعلی باشد) و رمز جدید با تکرار رمز جدید یکسان باشد، اگر هر کدام از این شروط رعایت نشده باشد کاربر با پیغام خطا مواجه خواهد شد و در صورت رعایت شدن شروط و برابر بودن مقدار textBox رمز جدید و تکرار رمز جدید ارتباط با دیتابیس برقرار می‌شود و عملیات انتخاب (SELECT) در دیتابیس بر اساس مقدار متغیر accNum که حاوی شماره حساب کاربر است انجام شده، ابتدا مقدار رمز فعلی خوانده می‌شود و با مقداری که در textBox رمز فعلی وارد شده مقایسه می‌شود تا بررسی شود که کاربر رمز فعلی را به درستی وارد کرده اگر هر دو مقدار برابر بودند با استفاده از دستور به روز رسانی (UPDATE) مقدار رمز جدید در دیتابیس ذخیره می‌شود، سپس مقدار متغیر state برابر با کاراکتر P می‌شود و فرم نتایج نمایش می‌یابد.

تصویر 12: فرم تغییر رمز

9. پرداخت قبوض (FormPayBills)

فرم پرداخت قبوض شامل یک textBox با محدودیت سیزده کاراکتر (شناسه قبض سیزده رقم در نظر گرفته شده) و دکمه لغو برای بازگشت به منو خدمات و دکمه تایید است.

متغیر سراسری billID از نوع string برای ذخیره شناسه قبض ایجاد شده و با انتخاب دکمه تایید ارتباط با دیتابیس برقرار می‌شود و با دستور انتخاب (SELECT) بررسی می‌شود آیا قبضی با این شناسه در دیتابیس (جدول قبض‌ها) وجود دارد، در صورت موجود بودن قبض مقدار متغیر state برابر با کاراکتر R می‌شود و فرم نتایج نمایش می‌یابد.



تصویر 13: فرم دریافت شناسه قبض

10. نتایج (FormResults)

این فرم مهم‌ترین بخش از شبیه‌ساز عابربانک در این پروژه است به گونه‌ای که بیش‌تر عملکردهای عابربانک و ارتباط برقرارکردن با دیتابیس و اعمال تغییرات در آن در این فرم صورت می‌گیرد.

فرم نتایج شامل یک label اصلی برای نمایش متن و نتیجه تراکنش‌ها است و همچنین چهار دکمه که شامل:

1. خدمات دیگر: برای بازگشت به منو خدمات استفاده می‌شود.
 2. تایید: در هنگامی که تراکنشی احتیاج به تایید داشته باشد (مثل انتقال وجه) استفاده می‌شود.
 3. لغو: همراه با دکمه تایید نمایش می‌یابد تا کاربر پس از مطالعه label و جزئیات عملیات مورد نظر تصمیم بگیرد عملیات را لغو یا تایید کند.
 4. خروج: برای بازگشت به فرم انتخاب زبان استفاده می‌شود.
- اساس عملکرد در این فرم مقدار متغیر state است، که در ادامه به ترتیب هر شرط و عملکرد توضیح داده شده.

1. اگر مقدار متغیر state برابر با کاراکتر B باشد:

در صورت صحیح بودن این شرط ارتباط با دیتابیس برقرار می‌شود و با استفاده از دستور انتخاب (SELECT) رکورد حساب کاربری براساس شماره حسابی که در متغیر accNum ذخیره شده انتخاب می‌شود و سپس فیلد موجودی حساب همراه با متن در label اصلی نمایش می‌یابد.

2. اگر مقدار متغیر state برابر با کاراکتر W باشد:

در صورت صحیح بودن این شرط ارتباط با دیتابیس برقرار می‌شود و با استفاده از دستور انتخاب (SELECT) رکورد حساب کاربری براساس شماره حسابی که در متغیر accNum ذخیره شده انتخاب می‌شود، سپس مقدار فیلد موجودی حساب در متغیر cardMoney قرار می‌گیرد پس از آن با کم کردن مقدار برداشتی (متغیر withdrawMoney) از موجودی کل (متغیر cardMoney) بررسی می‌شود

که مقدار برداشتی از موجودی بیشتر نباشد و سپس باقی‌مانده وجه در فیلد موجودی (Balance) مربوط به شماره حساب کاربر با استفاده از دستور به روز رسانی (UPDATE) دیتابیس ذخیره می‌شود و نتیجه عملیات بر روی label نمایش پیدا می‌کند.

3. اگر مقدار متغیر state برابر با کاراکتر C باشد:

در صورت صحیح بودن این شرط ارتباط با دیتابیس برقرار می‌شود و با استفاده از دستور انتخاب (SELECT) رکورد حساب کاربری براساس شماره کارتی که در متغیرهای CtoC1، CtoC2، CtoC3 و CtoC4 ذخیره شده انتخاب می‌شود، سپس مقدار فیلد موجودی حساب در متغیر cardMoney قرار می‌گیرد پس از آن با کم کردن مقدار وجه برای انتقال (متغیر moneyAmount) از موجودی کل (متغیر cardMoney) بررسی می‌شود که مقدار برداشتی از موجودی بیشتر نباشد و سپس پیغامی بر روی label نمایش می‌یابد و با نمایش جزئیات عملیات از کاربر می‌خواهد برای ادامه دکمه تایید یا لغو را بفشارد و دو دکمه تایید و لغو نمایش می‌یابند (دکمه خدمات دیگر و دکمه خروج در این مرحله مخفی می‌باشند) با انتخاب دکمه تایید دستوراتی که در دکمه تایید نوشته شده‌اند اجرا می‌شود.

در شروط دکمه تایید دستورات شرط state اگر C بود اجرا می‌شوند و موجودی حساب کارت گیرنده با دستور به روز رسانی (UPDATE) دیتابیس برابر با موجودی فعلی به اضافه مقدار منتقل شده می‌شود و همچنین باقی‌مانده در فیلد موجودی (Balance) مربوط به شماره حساب کاربر با استفاده از دستور به روز رسانی (UPDATE) دیتابیس ذخیره می‌شود و نتیجه عملیات بر روی label نمایش پیدا می‌کند.

4. اگر مقدار متغیر state برابر با کاراکتر A باشد:

این عملیات مشابه عملیات شرط قبلی (اگر مقدار متغیر state برابر با کاراکتر C باشد) است تنها با این تفاوت که در اینجا انتخاب رکوردها براساس شماره حساب صورت می‌گیرد نه براساس شماره کارت، در صورت صحیح بودن این

شرط ارتباط با دیتابیس برقرار شده و با استفاده از دستور انتخاب (SELECT) رکورد حساب کاربری براساس شماره حسابی که در متغیر accNum ذخیره شده انتخاب می‌شود، سپس مقدار فیلد موجودی حساب در متغیر cardMoney قرار می‌گیرد پس از آن با کم کردن مقدار وجه برای انتقال (متغیر moneyAmount) از موجودی کل (متغیر cardMoney) بررسی می‌شود که مقدار برداشتی از موجودی بیشتر نباشد و سپس پیغامی بر روی label نمایش می‌یابد و با نمایش جزئیات عملیات از کاربر می‌خواهد برای ادامه دکمه تایید یا لغو را بفشارد و دو دکمه تایید و لغو نمایش می‌یابند (دکمه خدمات دیگر و دکمه خروج در این مرحله مخفی می‌باشند) با انتخاب دکمه تایید دستوراتی که در دکمه تایید نوشته شده‌اند اجرا می‌شود.

در شروط دکمه تایید دستورات شرط state اگر A بود اجرا می‌شوند و موجودی حساب کارت گیرنده با دستور به روز رسانی (UPDATE) دیتابیس برابر با موجودی فعلی به اضافه مقدار منتقل شده می‌شود و همچنین باقی‌مانده در فیلد موجودی (Balance) مربوط به شماره حساب کاربر با استفاده از دستور به روز رسانی (UPDATE) دیتابیس ذخیره می‌شود و نتیجه عملیات بر روی label نمایش پیدا می‌کند.

5. اگر مقدار متغیر state برابر با کاراکتر P باشد:

عملیات تغییر رمز هنگام انتخاب دکمه تایید در فرم تغییر رمز (FormChangePwd) صورت می‌گیرد و در فرم نتایج فقط پیغامی مبنی بر موفقیت آمیز بودن عملیات نمایش می‌یابد.

6. اگر مقدار متغیر state برابر با کاراکتر R باشد:

در صورت صحیح بودن این شرط ارتباط با دیتابیس برقرار شده و با استفاده از دستور انتخاب (SELECT) رکورد قبض (در جدول billsTable) براساس شناسه قبض (billsID) که در متغیر billID ذخیره شده انتخاب می‌شود، سپس مقدار فیلد هزینه قبض در متغیر billAmount قرار می‌گیرد و همچنین با استفاده از دستور انتخاب (SELECT) رکورد حساب کاربری براساس شماره

حسابی که در متغیر accNum ذخیره شده انتخاب می‌شود، سپس مقدار فیلد موجودی حساب در متغیر cardMoney قرار می‌گیرد پس از آن با کم کردن مقدار هزینه قبض (billAmount) از موجودی کل (متغیر cardMoney) بررسی می‌شود که هزینه قبض از موجودی بیشتر نباشد و سپس پیغامی بر روی label نمایش می‌یابد، با نمایش جزییات عملیات از کاربر می‌خواهد برای ادامه دکمه تایید یا لغو را بفشارد و دو دکمه تایید و لغو نمایش می‌یابند (دکمه خدمات دیگر و دکمه خروج در این مرحله مخفی می‌باشند) با انتخاب دکمه تایید دستوراتی که در دکمه تایید نوشته شده‌اند اجرا می‌شود.

در دستورات دکمه تایید و قسمت شرط state اگر A بود ابتدا رکورد مربوط به قبضی که پرداخت شده توسط دستور پاک کردن (DELETE) از دیتابیس حذف می‌شود، (قبض پرداخت شده از دیتابیس پاک می‌شود) سپس اگر پاک کردن قبض موفقیت آمیز بود (متغیر error برابر false بود) مقدار هزینه قبض از حساب کم می‌شود و باقی مانده موجودی حساب مجدداً توسط دستور به روز رسانی (UPDATE) در رکورد حساب کارت ذخیره می‌شود و نتیجه عملیات بر روی label نمایش پیدا می‌کند.

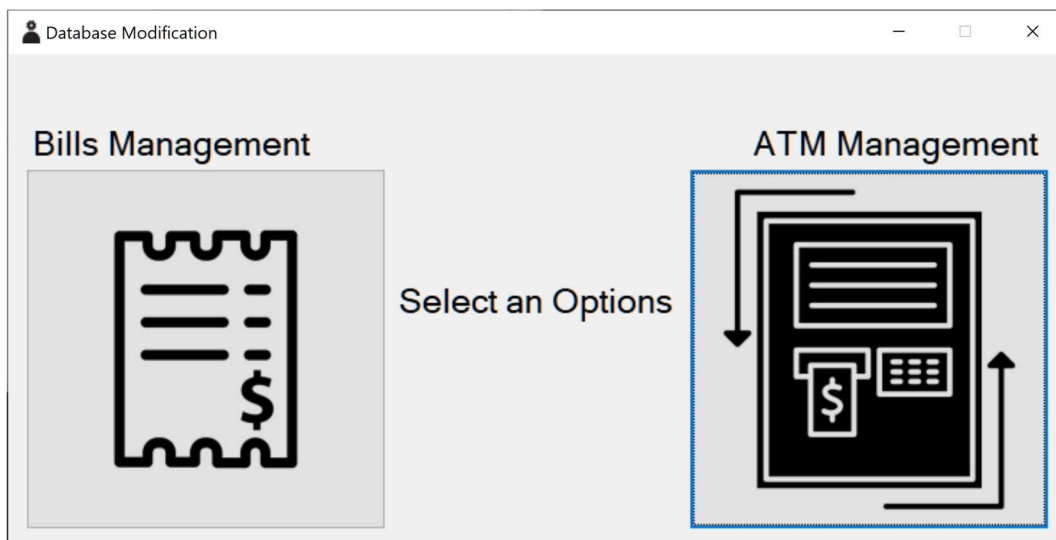
11. خروج

دکمه خروج در اکثر فرم‌ها وجود دارد و انتخاب آن موجب بازگشت به فرم انتخاب زبان و خروج کاربر از حساب کارت بانکی خود می‌شود.

جایگزینی دکمه خروج با دکمه بازگشت یا دکمه لغو در برخی از فرم‌ها به دلیل آن است که در برخی از فرم‌ها نباید در بین عملیات مربوط به یک خدمت بانکی از حساب کارت بانکی خارج شد.

مدیریت عابر بانک:

همچنین در این پروژه برای سهولت کار با جدول‌های دیتابیس، پنل مدیریت عابربانک بصورت برنامه‌ای جداگانه طراحی شده است.



تصویر 14: فرم مدیریت پایگاه داده عابر بانک

پنل مدیریت عابربانک شامل دو بخش مدیریت قبوض و مدیریت حساب کارت‌های بانکی است، قسمت مدیریت حساب کارت‌های بانکی شامل موارد زیر است:

1. افزودن (ذخیره) حساب جدید
2. حذف حساب
3. جستجو حساب براساس شماره حساب
4. به روز رسانی حساب

همچنین پنل مدیریت قبوض شامل این موارد است:

1. افزودن (ذخیره) قبض جدید
2. حذف قبض
3. جستجو قبض براساس شناسه قبض
4. به روز رسانی قبض

ATM Modification

Account Number:

Card Number:

CVV2 :

Password :

Account Balance:

Clear Save Delete Search Update Exit

تصویر 15: فرم مدیریت حساب کارت‌های بانکی

Bills Modification

Bill ID:

Bill Cost:

Clear Save Delete Search Update Exit

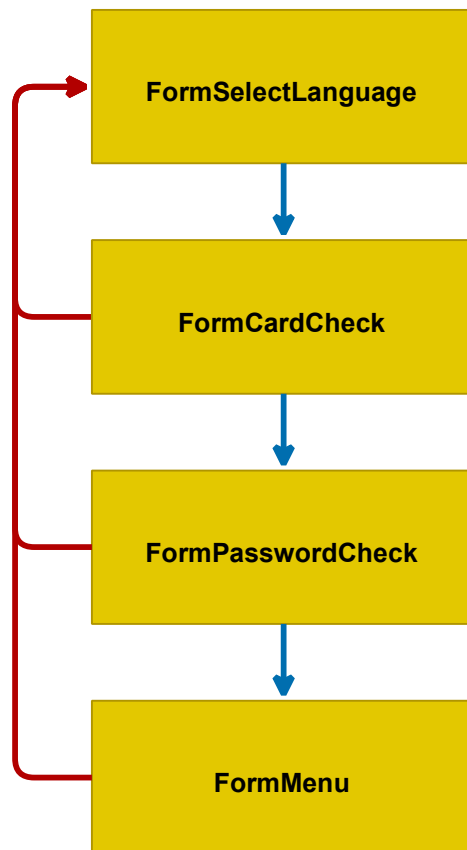
تصویر 16: فرم مدیریت قبوض

نحوه ارتباط فرم‌ها:

راهنما

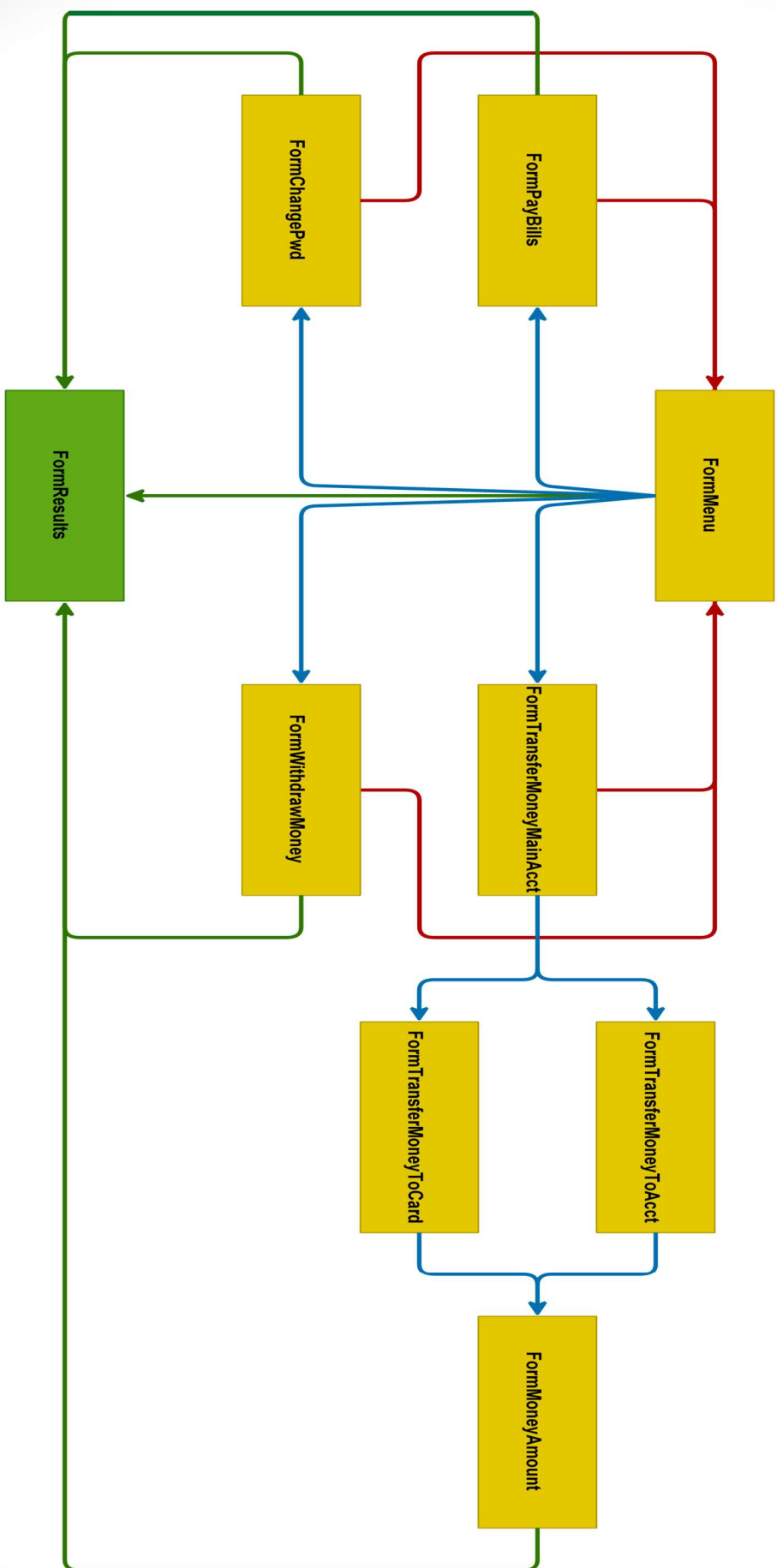
پیکان **آبی**: تایید (حرکت به جلو) پیکان **قرمز**: بازگشت، لغو پیکان **سبز**: نتیجه (پردازش و نمایش نتیجه خدمت)

(حرکت به عقب)



دیاگرام 1: نحوه ارتباط بین فرم آغازین (انتخاب زبان) تا فرم منو خدمات

نکته: تنها ارتباطات مهم بین فرم‌ها رسم شده است و ارتباطات کم اهمیت (مانند دکمه‌ی بازگشت) در فرم‌های بعد از فرم منو خدمات به دلیل ایجاد پیچیدگی زیاد در دیاگرام رسم نشده‌اند.



دیاگرام 2: نحوه ارتباط بین فرم منو خدمات با فرم های خدمات و فرم نتایج

جمع بندی:

بطور خلاصه می‌توان گفت که در این پروژه سعی بر این شده تا با ساده‌ترین روش‌ها بیشترین کارایی و همچنین محیطی که نمایان‌گر یک عابر بانک باشد را ایجاد کرد.

در کدنویسی تمام شروط و رخدادهای زمان اجرا پیشبینی شده‌اند و سعی شد تا شبیه‌ساز برای هر رخدادی آمادگی پاسخ‌گویی و راهنمایی کاربر را داشته باشد، در این مورد برای تمام بخش‌های خدمات رخداد خطاهای احتمالی پیش‌بینی شده است و پیغام‌هایی برای کاربر متناسب با خطا مواجهه شده نمایش خواهد یافت.

زبان کدنویسی این پروژه زبان C# است و برای دسترسی به داده (دیتابیس) از ADO.NET که تکنولوژی دسترسی داده از چارچوب دات‌نت شرکت مایکروسافت است که ارتباطی را میان سیستم‌های مرتبط و غیرمرتبط ایجاد می‌کند استفاده شده.

طراحی و پیاده‌سازی فرم‌ها و کدنویسی در محیط Microsoft Visual Studio صورت گرفته است و در بخش سرور پایگاه داده از سیستم مدیریت بانک‌های اطلاعاتی رابطه‌ای Microsoft SQL Server استفاده شده است.

لازم به ذکر است در ساخت این پروژه از آخرین نسخه Microsoft Visual Studio و Microsoft SQL Server استفاده شده، چنانچه در حین اجرا با خطایی مربوط به ارتباط با دیتابیس مواجه شدید به احتمال زیاد دلیل آن به روز نبودن نسخه پایگاه داده شما می‌باشد.

[1] راجر اس. پرسمن، مهندسی نرم افزار، چاپ هفتم

[2] نحوه نوشتن مستندات طراحی سایت

<https://idev-group.ir/blog/%D9%86%D8%AD%D9%88%D9%87-%D9%86%D9%88%D8%84%D8%AAD%D9%86-%D9%85%D8%B3%D8%AA%D9%86%D8%AF%D8%A7%D8%AA-%D8%B7%D8%B1%D8%A7%D8%AD%D8%8C-%D8%B3%D8%A7%D8%8C%D8%AA>

[3] عبارتهای انگلیسی در بانک

<https://fa.speaklanguages.com/%D8%A7%D9%86%D8%AF%D9%84%D8%8C%D8%B3%D8%8C/%D8%B9%D8%8C%D8%A7%D8%B1%D8%AA-%D9%B7%D8%A7%D8%AF%D8%B1-%D8%A8%D8%A7%D9%86%D8%A9>

[4] ADO.NET <https://en.wikipedia.org/wiki/ADO.NET>

[5] Microsoft SQL Server

https://en.wikipedia.org/wiki/Microsoft_SQL_Server

[6] SQL Tutorial <https://www.w3schools.com/sql/default.asp>

[7] Basics of ADO.NET

<https://www.c-sharpcorner.com/UploadFile/18fc30/understanding-the-basics-of-ado-net/>

[8] Save, Delete, Search And Update Records in ADO.NET

<https://www.c-sharpcorner.com/UploadFile/718fc8/save-delete-search-and-update-record-in-ado-net/>

[9] C# - Designing a Flat desktop Application

<https://www.youtube.com/watch?v=nLfzH4x0Vgo>

[10] Open Form2 from Form1, close Form1 from Form2

<https://stackoverflow.com/questions/2958912/open-form2-from-form1-close-form1-from-form2>

[11] C# - Winforms - Global Variables

<https://stackoverflow.com/questions/1293926/c-sharp-winforms-global-variables>

[12] Microsoft Docs <https://docs.microsoft.com/en-us/>

[13] Iconfinder <https://www.iconfinder.com/>

[14] How To Export SQL Server Database As .MDF File

<https://www.c-sharpcorner.com/UploadFile/919746/how-to-export-sql-server-database-as-an-mdf-file/>