

## چکیده:

بروز خطا یا استثنا در برنامه های رایانه ای اجتناب ناپذیر است، برای کاهش خطا در برنامه باید شرایطی که در آن برنامه با مشکل مواجه میشود را به دقت بررسی نمود. به همین منظور باید سازکار مشخصی برای ثبت و گزارشگیری از خطا ها وجود داشته باشد. در این مقاله روشهای مطلوب گزارش خطا باختصار بیان میشود.

هر برنامه ای در زمان اجرا با شرایطی مواجه میشود که اجرای درخواست قابل قبول کاربر در آن شرایط امکانپذیر نمی باشد. حتی اگر برنامه فاقد اشکالات ساختاری و منطقی باشد ممکن است در حین اجرا بدلیل قطع شدن شبکه و یا خارج شدن سرویس دهنده از مدار اجرای برنامه با مشکل مواجه شود. هر برنامه بر اساس قواعد و قوانین خاصی کار میکند که اصطلاحاً به آن Business Logic می گویند مواردی که در آن عدم اجرای درخواست کاربر بدلیل عدم انطباق آن با آن قوانین باشد جزو موارد خطا تلقی نمی شود به طور مثال برنامه ها از اجرای دستورات زیر جلوگیری میکنند زیرا بر خلاف قواعد کار هستند (خطا نیستند):

- برداشت از حساب بانکی مسدود شده.
- برداشت دو میلیون ریال از حساب بانکی به موجودی یک میلیون ریال.
- حذف کاربر SysAdmin از فهرست کاربران.
- دسترسی ویرایش همزمان به یک رکورد پایگاه داده.
- برداشت مبلغ منفی از یک حساب بانکی.
- صورتحساب از اول مهر تا اول شهریور (تاریخ اتمام مقدم بر تاریخ شروع).

برخی از موارد فوق قابل پیشگیری است. مثلاً به محض انتخاب شدن کاربر SysAdmin باید کلید «حذف» غیر فعال شود و یا باید امکان ورود عدد منفی در فیلد مبلغ منتفی گردد. اما موارد اول و دوم قابل پیشبینی نیستند لذا تا قبل از بررسی شرایط مشخص نیست که آیا درخواست قابل قبول است یا خیر. موارد فوق درخواست های غیر قابل قبول تلقی میگردند و خطا نیستند.

خطا یا استثنا شرایطی است که در آن اجرای دستور قابل قبول کاربر میسر نیست. به طور مثال به دلیل قطع شدن شبکه امکان ارسال تراکنش نباشد و یا به

دلیل خارج بودن سرویس دهنده پایگاه داده از سرویس دهی اجرای جستجو امکانپذیر نباشد.

اگر در برنامه از بلوکهای try and catch استفاده شده باشد تمام موارد استثنا بدون اینکه به سیستم عامل ارجاع شود در خود برنامه قابل پیگیری است به سازوکار پیگیری استثنا Exception Handling گویند و آن عبارتست از بکارگیری درست بلوکهای Try-Catch-Finally در برنامه و اتخاذ تصمیم درست در مواقع وقوع استثنا.

هرگاه یا استثنا رخ دهد برنامه باید به طور مشخص وقوع استثنا را به اطلاع کاربر برساند و نیز اطلاعات دیگری در خصوص شرایطی که در آن خطا رخ داده را ذخیره کند.

در اینجا چند مثال از گزارش خطا در برنامه های مختلف نشان داده می شود موارد از بد به خوب مرتب شده اند و همه برنامه ها واقعی هستند:



شکل یک: بدترین روش اعلان خطا به دلیل نداشتن پیام



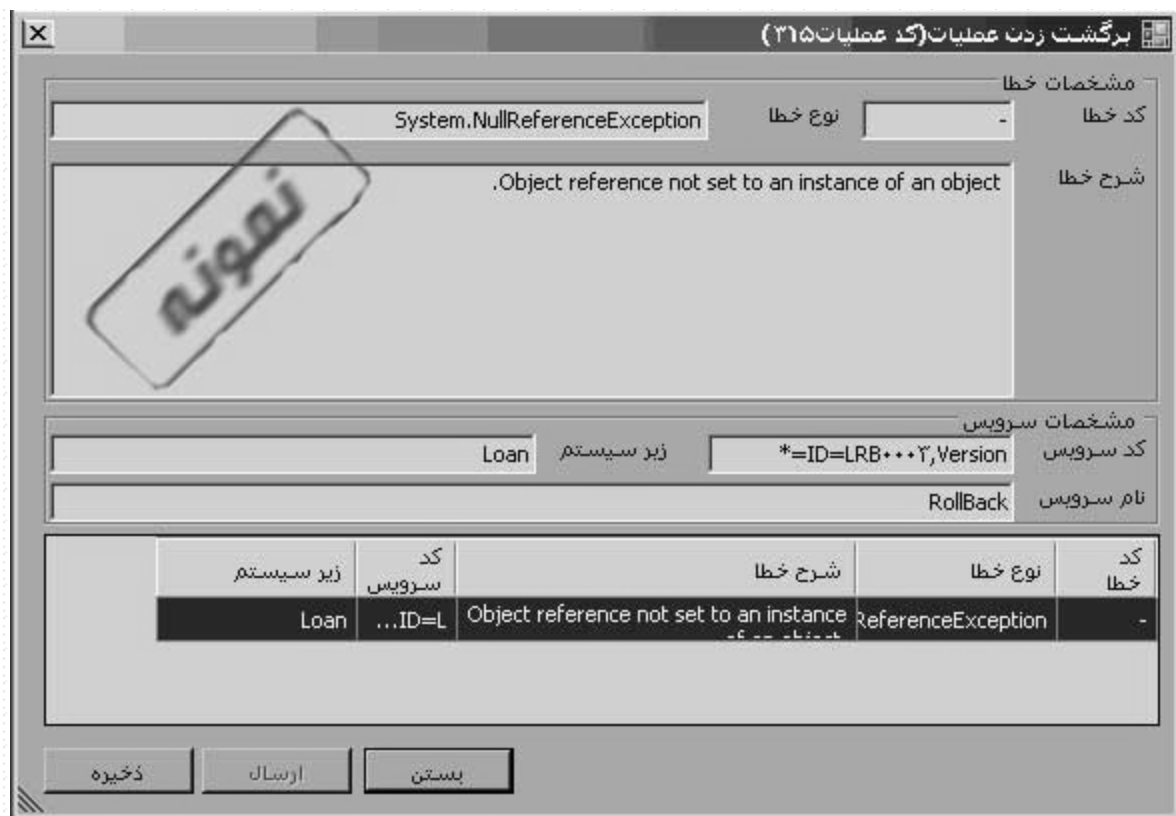
شکل دو : اطلاعات ارائه شده کلی و غیر قابل رهگیری است




شکل سه: جزئیات فنی ارائه شده اما کامل نیست.



شکل چهار: پیاده سازی مناسب از گزارش خطا با امکان ذخیره.



شکل پنج: پیاده سازی کامل و مطلوب با جزئیات کامل و امکان ذخیره و ارسال.

طرح مسئله: 

می خواهیم یک utility داشته باشیم که پس از وقوع استثنا آن را به کاربر اطلاع داده و دارای امکانات زیر باشد:

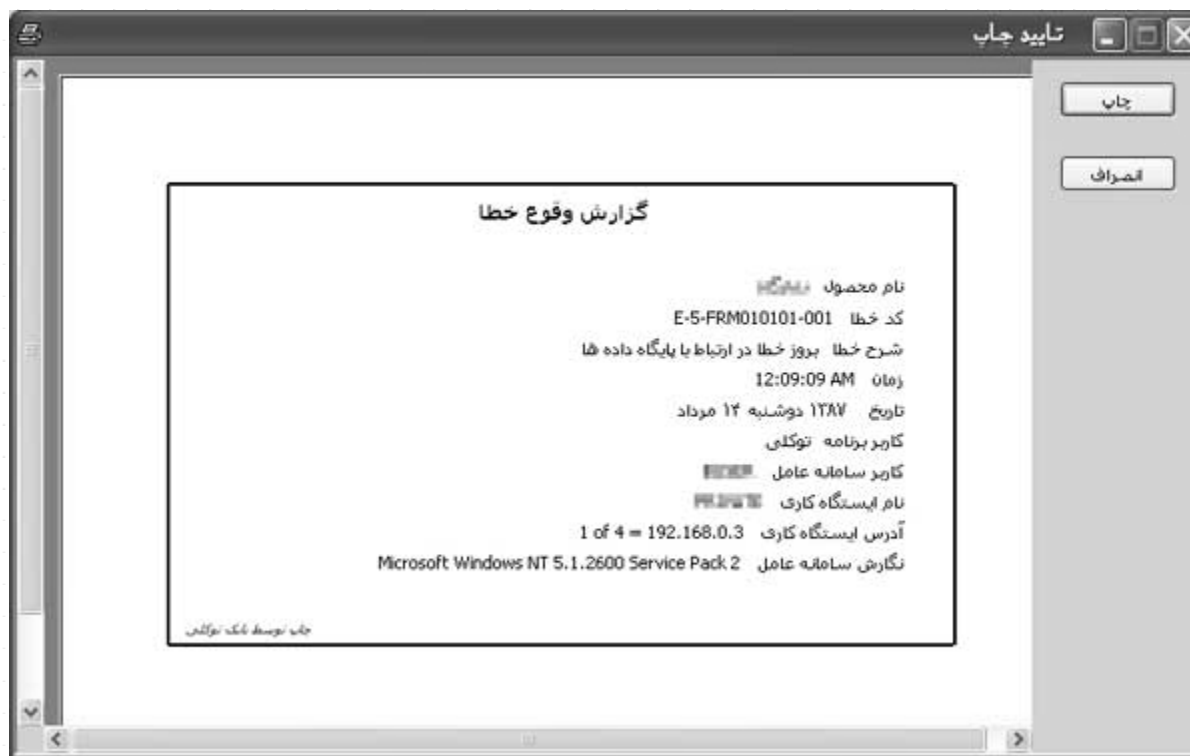
ارائه راه حل.  امکان انتقال به کلیپ برد.  امکان چاپ.



کاربر میتواند روی راه حل کلیک نموده و آن را مطالعه کند:



و یا از آن چاپ بگیرد:



همانطور که مشاهده می گردد اطلاعات کامل و دقیقی از خطا ارائه میشود. در حین نمایش خطا سازکار رویدادنگاری محلی نیز یک رکورد از خطا ذخیره میکند. در مواقعی که بدلیل data inconsistency خطا رخ میدهد، یک رونوشت از داده های ورودی نیز رمز شده و ذخیره میگردد. علت ذخیره در پایگاه داده محلی آنست که احتمالاً در مواقع وقوع خطا، دسترسی به پایگاه داده مرکزی مقدور نیست. همزمان با ذخیره سازی رکورد خطا در پایگاه داده محلی یک رکورد نیز در زمانبند برنامه Job Scheduler ذخیره میشود تا به محض برقراری ارتباط با مرکز رکورد یا رکوردهای گزارش خطای محلی را به مرکز ارسال نماید و بدین ترتیب امکان رهگیری خطا با داشتن اطلاعات کافی میسر میشود.

دقت فرمایید که بین کاربر برنامه و کاربر سیستم عامل تمایز قائل شده ایم و علاوه بر نام و آدرس سیستم، نگارش سیستم عامل را از جهت ملاحظات compatibility مورد توجه قرار می دهیم.

■ برای بدست آوردن نام کاربر و سیستم و نگارش سیستم عامل از شی Environment استفاده شده است

```
ErrDlg.OSUSER = Environment.UserName.ToString();  
  
ErrDlg.MACHINENAME = Environment.MachineName;  
  
ErrDlg.OSVER = Environment.OSVersion.ToString();
```

■ برای بدست آوردن آدرس IP از این کد استفاده شده است:

```
public static string FindIP()  
{  
    try  
    {  
        IPEndPoint ipEntry = Dns.GetHostByName(Environment.MachineName);  
  
        IPAddress[] addr = ipEntry.AddressList;  
  
        addr = System.Net.Dns.GetHostAddresses(Environment.MachineName);  
  
        return "1 of " + addr.Length.ToString() + " = " + addr[1].ToString();  
    }  
    catch  
    {  
        return "Invalid IP Address";  
    }  
}
```