

بسمه تعالي



دانشگاه علم و فرهنگ

پروژه مهندسي نرم افزار

سيستم مديريت کارمندان

استاد: خانم دکتر زماني فر

اعضاي گروه:

مينا بماني زاده

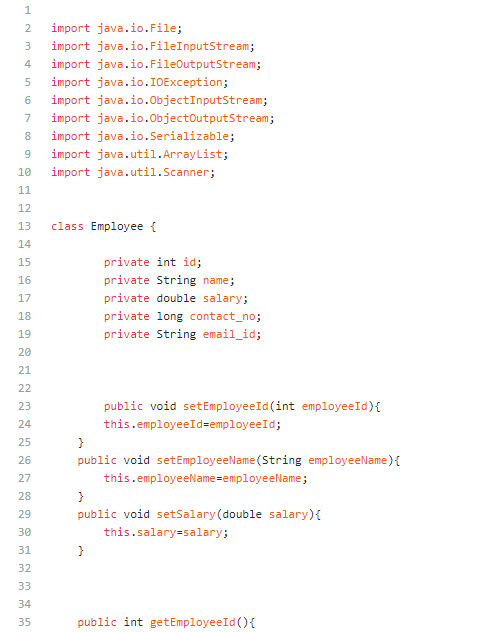
941113009

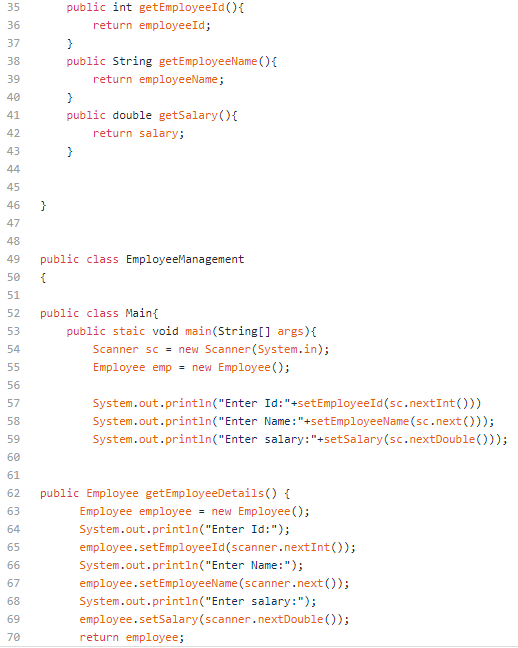
و

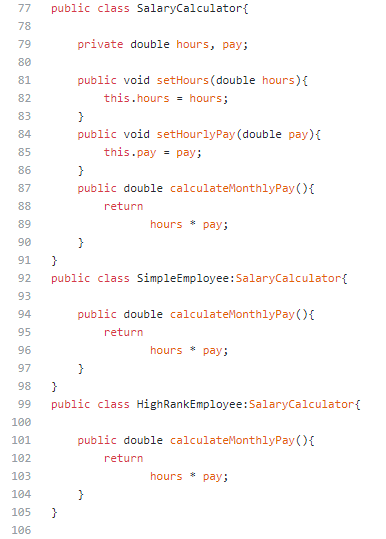
مژده ارداني

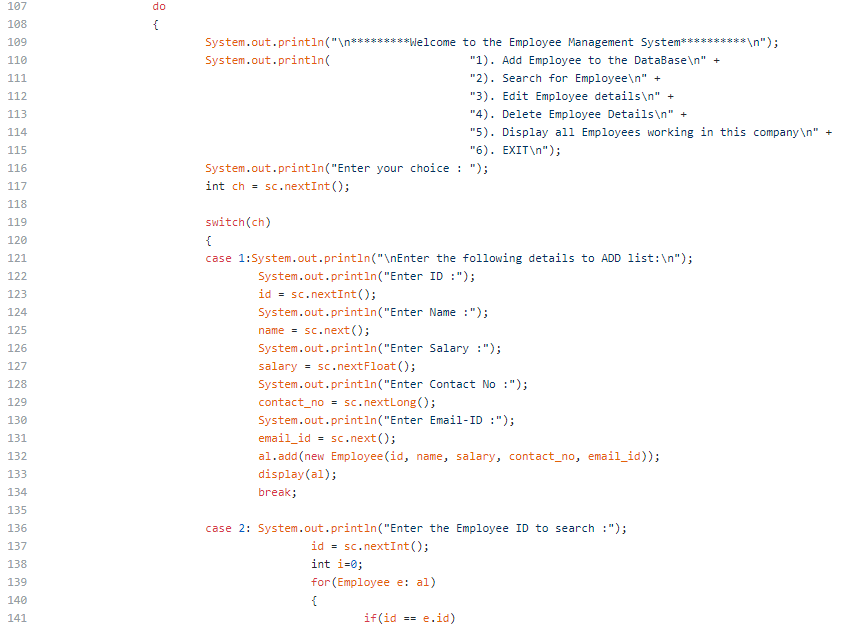
941113003

معرفي سيستم مديريت کارمندان:

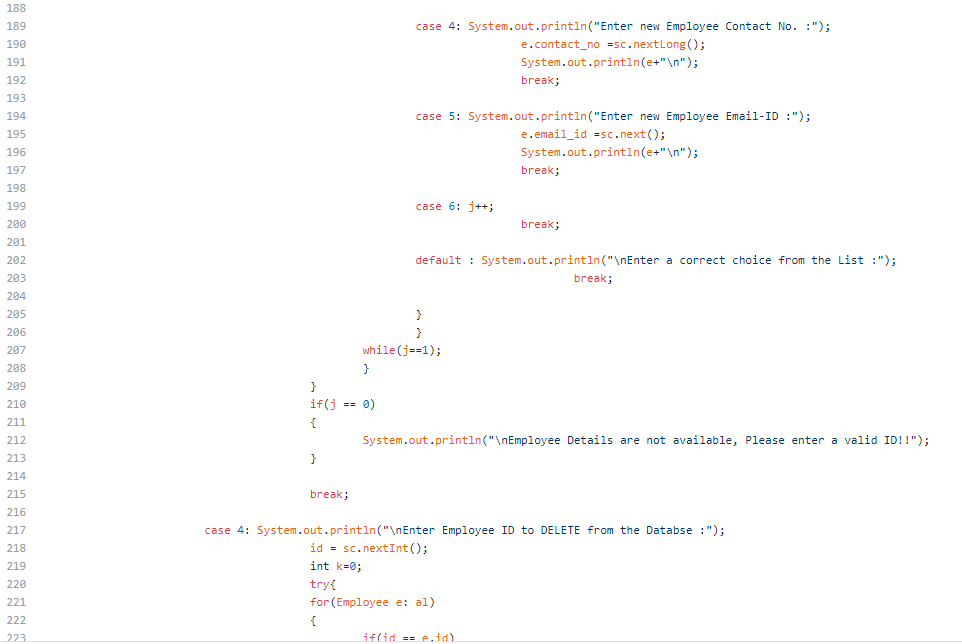


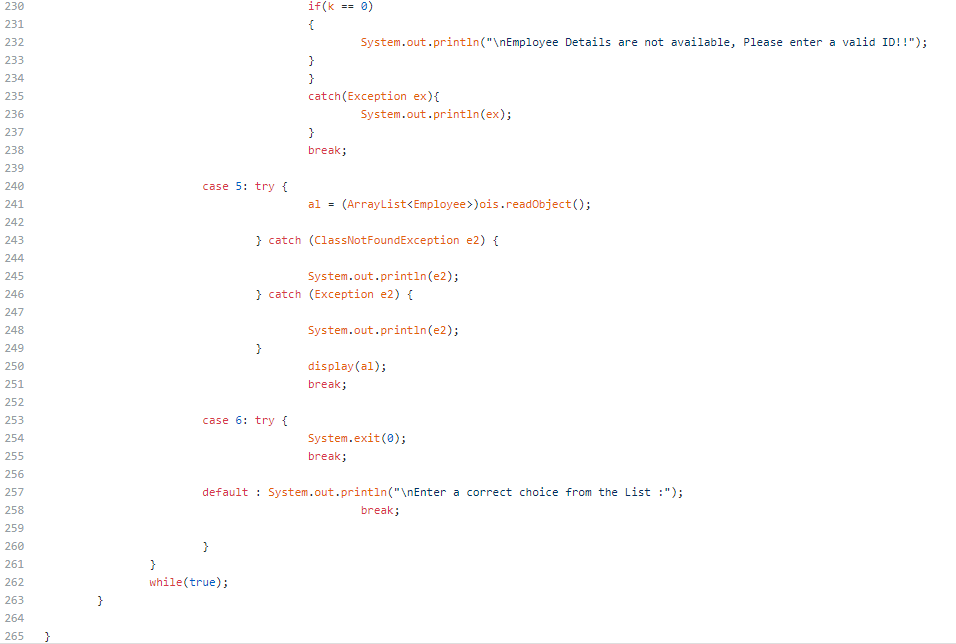












ما سعي کرده ايم که solid principals را در کدنويسي اين سيستم رعايت کنيم براي مثال از open/closed principal استفاده کرده ايم تا کدهايمان نسبت به اضافه کردن متد ها و کلاس هاي جديد باز باشد و نسبت به اصلاح کدها بسته باشد. به عبارتي يعني هر گاه که نياز بود ويژگي جديدي به برنامه مان اضافه کنيم مجبور به اصلاح و تغيير ساير قسمت ها نشويم.

مثلا در قسمت محاسبه حقوق کارمندان، کارمندان به دو دسته کارمند ساده و کارمند داراي مقام بالاتر تقسيم شده اند که اين دو کلاس از کلاس کارمند ارث بري کرده اند. حالا هر زماني که بخواهيم نوع ديگري از کارمند را تعريف کنيم نيازي به دست زدن به ساير کلاس ها و تغيير آن ها نيست و بدون ايجاد تغيير در ساير قسمت ها به راحتي ميتوانيم نوع جديدي از کارمندان را اضافه کنيم.

همچنين از قوانين single responsibility سعي کرده ايم استفاده کنيم تا هر کلاسي تنها يک مسئوليت داشته باشد.

از refactoring براي بازيابي کد و از بين بردن bad smell هايي مانند کد هاي تکراري(duplicated code) و يا متد ها و کلاس هاي طولاني(large class و long method) و کامنت ها استفاده کرده ايم تا در حد ممکن کدها تميز باشند.

از الگوهاي طراحي(design patterns) سعي کرديم براي بهتر شدن کد ها کمک بگيريم. از singleton کمک گرفتيم تا هر کلاس يک نوع آبجکت/اينستنس داشته باشد و نيز از factory method براي اينکه نقش کارخانه را درايجاد آبجکت کارمند ايفا کند.

معماري استفاده شده در سيستم مديريت کارمندان :

معماري که ما در سيستم مديريت کارمندان استفاده کرده ايم معماري MVC (Model-View-Controller) مي باشد.

زيرا اين سيستم به صورت تعاملي است و به سه بخش زير تقسيم شده است:

1.model: که شامل قابليت هاي اصلي برنامه و داده هاست.

2.view: وظيفه این بخش نشان دادن اطلاعات کارمند به کاربر است.

3.controller: این بخش مدیریت داده‌های ورودی را که از سوی کاربران وارد می‌شود، برعهده دارد. همچنین وظیفه برقراری ارتباط ميان model و view نيز بر عهده controller است.

ما برای طراحی سيستم مديريت کارمندان از چنین الگویی استفاده کرديم تا فرآیند به‌کارگیری داده‌ها درون برنامه کاربردی و فرآیند نشان دادن اطلاعات و دریافت داده‌ها از کاربران از یکدیگر تفکیک ‌شود. همین موضوع باعث می‌شود تا پیچیدگی طراحی به میزان قابل‌توجهی کاهش پیداکرده و فرآیند توسعه نرم‌افزار به‌راحتی انجام شود.

سيستم از سه مولفه model و view و controller تشکيل شده است. View همان واسط کاربري (user interface) است، model همان داده هاي مربوط به کارمندان است و controller همان منطق برنامه ميباشد که داده ها را از model گرفته و در view به کاربر نمايش مي دهد.



رويه تست(testing):

پروسه ي ارزيابي توانايي سيستم

بررسي اينکه چقدر نيازمندي ها را پاسخ ميدهد

اطمینان از صحت عملکرد سيستم

پيدا کردن عيوب سيستم (هميشه اگر عيوب را زودتر پيدا کنيم هزينه برطرف کردن آن کمتر است)

بررسي مزايا و ريسک هاي سيستم

بررسي هزينه هاي احتمالي ديرکرد درانتشار سيستم

اطمینان از اينکه آیا نرم افزار با توجه به انتظار ما رفتار می کند یا رفتار دیگری از خود نشان می دهد.

ما از دو رويکرد براي تست سيستم استفاده کرديم:

BlackBox Testing

WhiteBox Testing

در تست جعبه سياه تست مکانيسم هاي داخلي سيستم را ناديده گرفتيم و روي خروجي توليد شده تمرکز کرديم.

سپس در مرحله بعد تست جعبه سفيد انجام داديم يعني متد هاي سيستم و در کل مکانيسم داخلي سيستم را تست کرديم.

تست هاي acceptance testing و performance testing و system testing را انجام داديم.

در performance testing عملکرد و کارايي (blackbox) سيستم را تست کرديم تا متوجه شويم برنامه مان عملکرد و کارايي لازم را داراست.

در system testing عملکرد نرم افزار را روي پلتفرم هاي مختلف امتحان کرديم تا مطمئن شويم که روي سيستم عامل هاي مختلف به خوبي کار ميکند.ممکن است کاربري از سيستم عامل ويندوز استفاده کند و ديگري از مک يا غيره براي همين بايد مطمئن شد که روي تمامي سيستم عامل ها به درستي کار ميکند.

و مرحله بعد acceptance testing براي اينکه مشخص کنيم آیا برنامه نیازهای مشتری را پاسخ می دهد و آیا برنامه همان چیزی که مشتری می خواهد هست یا خیر و اين تست پذيرش تست نهايي است که روي سيستم انجام داديم.

مديريت پيکربندي (configuration management) :

مدیریت پیکربندی نرم افزار (SCM) مجموعه ای از فعالیت هايي است که برای کنترل تغییرات طراحی شده اند. يکي از بخش هاي ضروري مديريت کيفيت نرم افزار است. در هر جای چرخه حیات سیستم که باشیم امکان اينکه سیستم تغییر کند وجود دارد. مدیریت پیکربندی نرم افزار(SCM) مجموعه ای از فعالیت هاست که برای مدیریت تغییرات در سرتاسر چرخه حیات نرم افزار کامپیوتری توسعه یافته اند.

چون امکان تغییر در هر زمانی وجود دارد، فعالیت های SCM به دلایلی که به دنبال خواهد آمد اجرا میشوند:

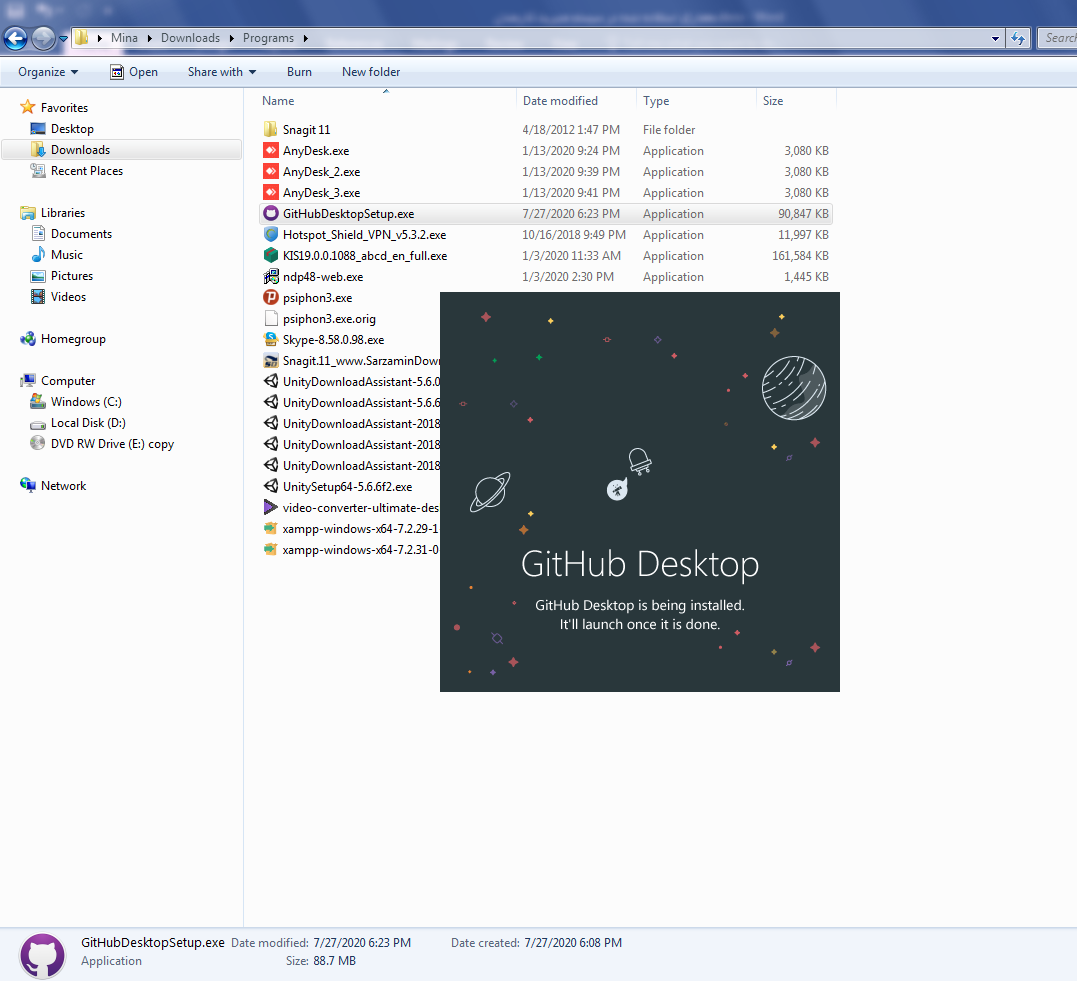
1- شناسایی تغییر

2- کنترل تغییر

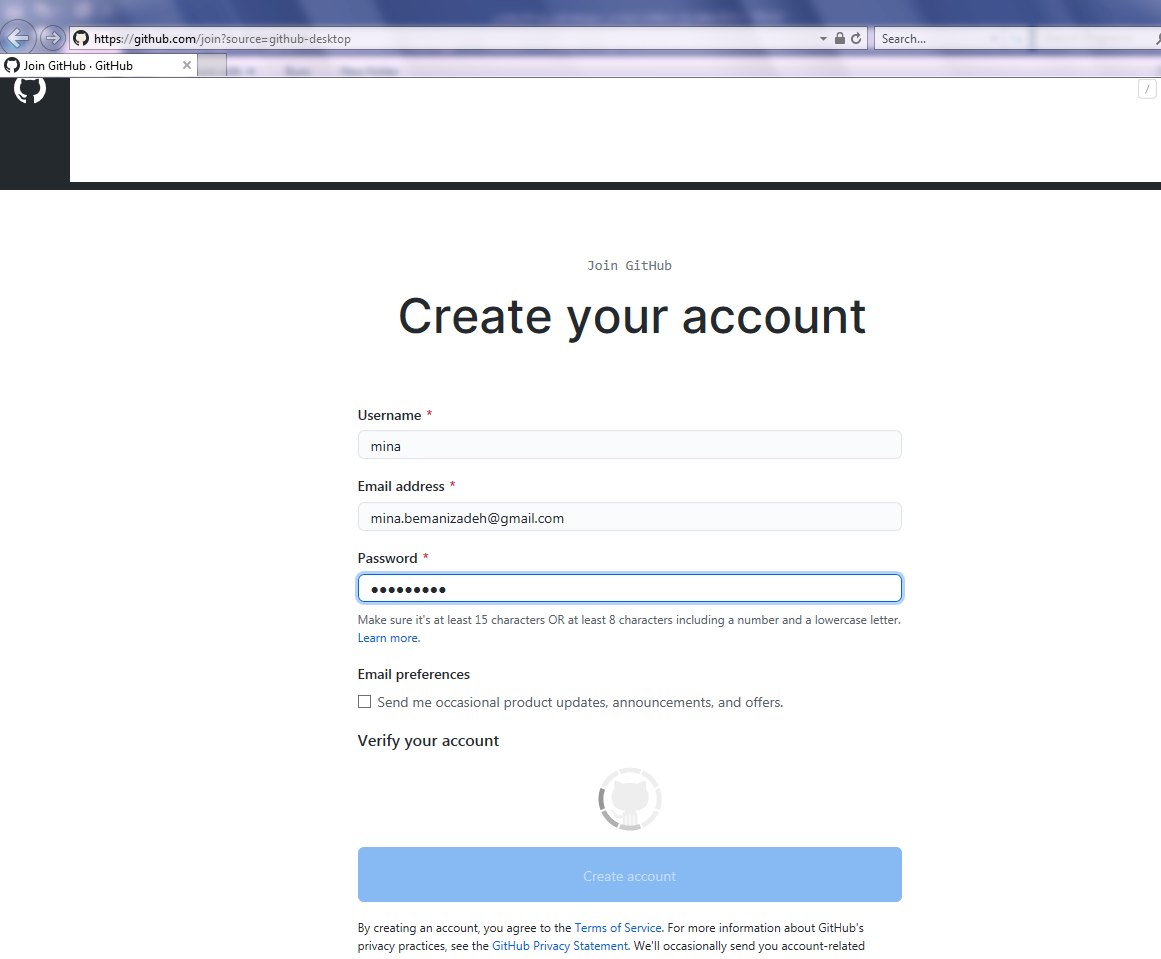
3- حصول اطمینان از پیاده سازی مناسب تغییر

4- گزارش تغییر به دیگران

مديريت پيکربندي در GitHub:

ابتدا GitHub Desktop را نصب کرديم.

سپس در آن اکانت ساختيم.



و بعد از منوي file قسمت add local repository کد ها را اضافه کرديم.

