



گزارش کار پروژه‌ی هتل

امیررضا دلاوران

معصومه میرزایی

اساتید محترم : مینوفام و فرشچی

در ابتدا UI برنامه را تحلیل کرده و سپس به تحلیل بخش کد میپردازیم:

در پروژه‌ی هتل ابتدا صفحه اولیه (intro) را مشاهده میکنیم که طبق خواسته‌های پروژه شامل دو گزینه میباشد:

1. گزینه login: مختص ورود کارکنان دارای مجوز ورود و یا مدیریت هتل میباشد.
 2. گزینه reservation: کاربر بدون نیاز به ورود میتواند کارهای رزرو خود را انجام دهد.
- علاوه بر این موارد، در سمت چپ صفحه intro میتوان زمان دقیق و در سمت راست آن تاریخ میلادی را مشاهده کرد که طریقه‌ی قرارگیری و نوشتن آن به صورت کامل در کد توضیح داده میشود. در قسمت مرکزی (center) تصویر یک icon را مشاهده میکنیم و نیز در صفحه‌ی intro از دو panel برای رنگ‌آمیزی استفاده شده است.


دکمه‌ی reservation:

در ابتدا، مانند صفحه‌ی intro، متشکل شده از دو panel برای رنگ‌آمیزی و جلوه‌آرایی به برنامه:

- Panel بالایی: دو label در آن قرار گرفته که label سمت راست برای قرار دادن تصویر و label سمت چپ برای نوشتن موضوع صفحه میباشد.
- Panel پایینی:

طبق خواسته‌های صورت پروژه، هشت فیلد داریم که اطلاعات رزرو را از مشتری (client) گرفته و به Database برنامه میفرستد. به منظور سهولت و شبیه‌سازی بیشتر برنامه به واقعیت، دو بخش زمان و تاریخ رزرو به صورت اتوماتیک این اطلاعات را از خود سیستم میگیرند.

در بخش Room number مشتری فقط میتواند اتاق‌هایی که:

- 
1. رزرو نشده‌اند
 2. وجود دارند و مدیریت از قبل ساخته است
- در این بخش سه دکمه (button) داریم:
1. Submit: برای ثبت اطلاعات
 2. Edit: برای تغییر اطلاعات با کد ملی
 3. Delete: برای حذف مشتری که فقط با کد ملی میسر میباشد.

دکمه‌ی login:

در این بخش هم مانند بخش‌های قبلی، دو panel برای رنگ‌آمیزی برنامه داریم. در سمت چپ تصویر هتل و ساعت قرار گرفته. در قسمت دیگر، فیلد وارد کردن username و password قرار دارند که اگر هرکدام از آنها خالی باشند، پیغامی مبتنی بر کامل نبودن اطلاعات وارد شده از طرف برنامه به مشتری داده میشود. اگر اطلاعات وارد شده اشتباه باشند و یا وجود نداشته باشند نیز همین اتفاق رخ میدهد. دکمه‌ی submit برای ورود و دکمه‌ی cancel برای بازگشت به صفحه‌ی intro میباشد.

صفحه‌ی منو(menu):

این صفحه نیز از دو panel تشکیل شده. در سمت راست از Icon های مرتبط با دکمه‌ها استفاده شده و 4 دکمه برای دسترسی به امکانات و بخش های مختلف تعبیه شده‌اند:

1. Passengers:

دارای امکاناتی از قبیل radiobutton و combobox میباشد. قسمت جالب این صفحه دکمه‌ی show payment است که با وارد کردن تعداد مسافران، نوع اتاق و تعداد شب‌های اقامت کل مبلغ پرداختی را به مشتری نشان میدهد. مشتری میتواند پیش‌پرداختی با حداقل مقدار 20 هزار تومان انجام دهد.

2. Employees: طراحی این صفحه مانند صفحه‌ی پیش انجام شده و تنها option

خاصی که دارد این است که اگر فرد در قسمت job-title گزینه‌ی hotel general manager را انتخاب کند، میتواند پس از آن به عنوان یک کاربر جدید ورود کند.

3. Rooms: در این قسمت، که برای ساخت اتاق‌های جدید میباشد، میتوان نوع و

تعداد اتاق را مشخص کرد. در قسمت آخر reservation اگر گزینه‌ی No انتخاب شود، به قسمت reservation هتل میرود و در غیر این صورت اتاق رزرو شده و امکان انتخاب آن وجود ندارد.

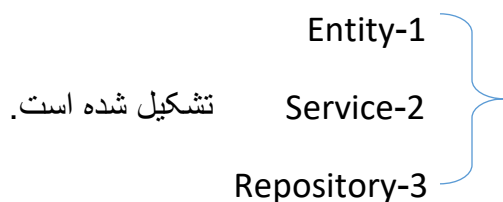
4. Hotel income: در بخش درآمد هتل 5 بخش وجود دارد. بخش اول یا همان

hotel_customer_workingday وظیفه‌ی جست و جوی مشتری‌هایی که در تاریخ مشخص رزرو خود را انجام داده‌اند را دارد. در بخش دوم (show customer payment) مجموع پرداخت‌هایی که در واقع همان پیش‌پرداخت‌های مشتری بوده را جمع بسته و محاسبه میکند. با انتخاب هر یک از دو گزینه‌ی exit و show payment یک dialog باز شده و پیغامی نشان داده میشود. در

بخش سوم (payment for each client) با توجه به کدملی و اسم مسافران، تمام پرداخت‌های اولیه را نمایش داده میشود. قسمت چهارم (Hotel working day) پیغامی مبنی بر باز بودن 24 ساعته‌ی هتل نمایش میدهد. قسمت پنجم و آخر درآمد هتل را براساس پرداخت‌های مسافران نمایش میدهد.

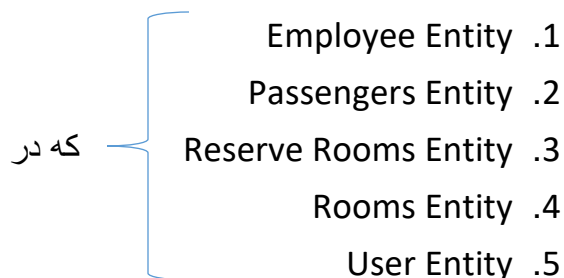
قسمت کد:

در قسمت کد پروژه، UI برنامه با swing و معماری برنامه با MVC است که مخفف Model-View-Control میباشد که بخش Model از سه بخش:



• بخش Entity:

این بخش از پنج کلاس تشکیل شده است:



هرکدام از آنها properties هایی برای ذخیره اطلاعات از repository مخصوص به هر بخش وجود دارد که با استفاده از الگوی builder ساخته شده‌اند.

• بخش Repository:

بخش Repository نیز مانند بخش Entity از 5 قسمت تشکیل شده که در Repository از دیتابیس Oracle استفاده شده.

1. Employees_Repository:

در این کلاس از پنج متد استفاده شده که متد اول برای وصل شدن به دیتابیس میباشد و نام آن Employees_Repository میباشد.

الف) Insert:

متد دوم Insert میباشد که کار آن دریافت اطلاعات و ذخیره سازی آنها در جدول Employees_Info میباشد.

ب) Update:

متد سوم، وظیفه‌ی این قید بروزرسانی اطلاعات از طریق کدملی میباشد.

ج) Delete:

Delete که برای پاک کردن اطلاعات از جدول با استفاده از کدملی میباشد. در این سه متد ذکر شده (insert,update,delet) به منظور امنیت بیشتر، دریافت اطلاعات از طریق Sql Injection انجام میشود.

د) Select:

متد آخر، Select میباشد که وظیفه‌ی آن گرفتن اطلاعات از طریق کوئری زدن به دیتابیس میباشد که با ArrayList نوشته شده است و برای ذخیره سازی اطلاعات username و password، در لیست آرایه یی با نام passwords,username به کار میرود.

2. Passengers_Repository:

مثل کلاس قبلی از چهار متد اصلی Insert, Update, Delete و وصل شدن به دیتابیس تشکیل شده و متد Select برای بازگردانی اطلاعات از جدول دیتابیس و ذخیره سازی آنها در جدول Passengers به کار میرود. این کلاس همچنین شامل متدهای زیر میباشد:

الف) Income:

برای به دست آوردن درآمد هتل از طریق جمع کردن تمام پرداخت های مشتری به کار میرود.

ب) First_Payment:

وظیفه‌ی آن، برگرداندن اطلاعات تمام پرداخت‌های اولیه‌ی مشتری میباشد.

ج) List:

این متد برای بدست آوردن اطلاعات پرداختی، نام و کدملی مشتری و نمایش آن در قسمت Payment for each client میباشد که در قسمت UI پروژه نحوه نمایش آن توضیح داده شد.

3. Reserve_Rooms_Repository

این کلاس نیز مانند کلاس‌های قبلی شامل متدهای اصلی بوده و شامل متدهای دیگری همچون List میباشد که وظیفه‌ی آن search کردن اطلاعات از طریق گرفتن تاریخ میباشد که نام رزروهای آن تاریخ را در صفحه‌ی Hotel customers on working days نمایش میدهد.

4. Rooms_Repository

این کلاس نیز همچون کلاس‌های قبلی دارای متدهای Insert, Update, Delete و متد اتصال به Database میباشد. همچنین دارای یک متد Select میباشد که برای چک کردن وضعیت اتاق‌ها ساخته شده است (در بخش Reservation توضیح داده شد که اگر وضعیت رزرو منفی باشد، اتاق در صفحه‌ی Reservation دیده میشود). اطلاعات در جدولی مشترک با کلاس Reservation_Repository ذخیره میشود.

5. User_Repository

این کلاس که آخرین کلاس این بخش میباشد از چهار متد اصلی که کلاس‌های قبلی را نیز تشکیل داده بودند تشکیل شده است. در قسمت Insert، بخشی که با نوشتن کامنت نیز مشخص شده است برای ورود اولیه به صورت دستی کاربری را میسازد. متد بعدی، متد Select میباشد که اطلاعاتی که در از Database گرفته شده است را برمیگرداند میکند و نقش بسزایی در ورود کاربر دارد.

نکته: در تمام کلاس‌های Repository متدهای Commit, Rollback و CreateTable نیز نوشته شده است.

• بخش Service:

در این بخش نیز پنج کلاسی که در بخش Repository بود وجود دارد که به منظور جلوگیری از اجرای چندین کار به صورت همزمان با الگوی Singleton نوشته شده است.

- **بخش Hotel_Income:**

کلاسی است برای به دست آوردن درآمد هتل، با استفاده از متد Income از کلاس Passengers_Repository.

- **بخش View & Controller:**

همانطور که در اول گزارش نوشته شد، View با Swing به وسیله برنامه‌ی JForm Designer نوشته شده است، به همین دلیل بخش Controller برنامه هم در پکیج View قرار دارد. کلاس‌های View با پسوند jfd و کلاس‌های Controller با پسوند java ذخیره شده است.

تمامی کلاس‌های EmployeeForm, ReservationForm, PassengersForm و کلاس‌هایی از این قبیل، که دارای دکمه‌های Submit, Edit و Delete میباشند با استفاده از ActionListener و ActionPerformed از طریق Service و Repository به دیتابیس وصل میشوند.

درکنترلر کلاس EmployeeForm اینگونه نوشته شده است که اگر شغل کارمند برابر با Hotel general manager باشد علاوه بر جدول Employee_Information در جدول Users نیز ذخیره میشود که بتواند با اطلاعات جدید هم وارد شود.

در همه‌ی بخش‌های View سعی بر این بوده که با استفاده از Image Icon تصاویری را قرار داد. همچنین در بخش View کلاس

Hotel_customer_workingday_form با استفاده از Jlist و ListModel لیست مشتری‌ها را پس از جست و جوی تاریخ به نمایش میگذارد و دکمه‌ی Exit با استفاده از متد Dispose(); یا SetVisible(False); موجب بسته شدن صفحه میشود. در تمامی بخش View، DialogMessage‌هایی برای نمایش اخطار، پیغام و اعلامیه‌ها تعبیه شده است.

پروژه به صورت تیمی و تقسیم کار تهیه شده است به طوری که UI پروژه و لایه‌ی Controller توسط امیررضا دلاوران و لایه‌ی Model به صورت کامل توسط معصومه میرزایی نوشته شده است.

نکته: در بخش‌هایی از پروژه Thread‌هایی برای عدم تداخل و مکث لحظه‌ای پردازش در برنامه نوشته شده است.

Diagram for Package Mo...


 Package Repository


 Package Service


 Package Entity


 Hotel_income

Powered by yFiles

 Reserve_Rooms_Entity

 Passengers_Entity

 Employee_Entity


 Rooms_Entity


 User_Entity


Powered by yFiles

 Reserve_Rooms_Repository

 Employee_Repository


 Rooms_Repository


 Passengers_Repo


 user_repo


Powered by yFiles

 Reserve_Rooms_Service


 Passengers_service


 Employee_Service


 Rooms_Service


 User_Service


Powered by yFiles


 Hotel_customer_workingday_form


 Payment_per_customers


 Hotel_income_form

 Customers_payment

 ReservationForm


 PassengersForm


 EmployeesForm

 Hotel_menu

 IntroForm

 RoomsForm

 LoginForm

 MenuForm

Powered by yFiles

Package groovyjarjarcommonscli

Package toolbarButtonGraphics

Package groovyjarjarpicocli

Package groovyjarjarantlr

Package groovyjarjarasm

Package netscape

Package META-INF

Package images

Package oracle

Package groovy

Package apple

Package javax

Package Model

Package View

Package java

Package com

Package org

Package sun

Package jdk

Package net

 Main

Powered by yfiles