移动网上营业厅设计与实现

（多媒体大作业）

1. 作业简介

设计并模拟了移动网上营业厅的基本功能。使用的技术包括编程语言C#，数据库SQL Server、ASP.NET技术。不过在网站页面设计方面用到了HTML、CSS、Javascript技术，并且为了美化整体界面效果，使用了前端框架Semantic UI，这些知识都很多媒体技术课程联系紧密。

2. 相关技术简介

2.1 ASP.NET

服务器端动态脚本语言有很多种,ASP.NET 便是其中之一,在当前诸多动态网页开发技 术中占有重要地位。ASP.NET 和 PHP、Python 一样,均能在服务器端高效地处理用户通过 HTTP 协议􏰀交的请求,经过后台信息处理后将数据反馈到用户机。在微软庞大的平台体系 中,ASP.NET 一直是其 Web 平台最重要的部署之一。在微软 IIS 服务器上,ASP.NET 具有 极为优异的表现,无论是并发数量还是数据共享方面相比于其他服务器端脚本语言毫不逊色。

2.2 C#编程语言

C#程序设计语言和现在最流行的Java程序设计语言有很多相似的地方,或者说 C#语言就是微软为了抗衡Java而设计的面向对象的程序设计语言。在最新几个月的Tiobe世界编程语言排行榜中长期霸占前五的位置,可见C#程序设计语言的影响力和普及度还是很高的。首先,C#程序设计语言和 Java 的语法上极为相似,都需要先编译生成 MSIL 中间语言,当程序运行的时候,再由这些中间代码在.NET平台下生成相应的二进制可执行代码,这些二进制代码会被写进代码缓冲区内，所以当这些代码被再次执行时，程序的速度会大大加快。将代码转为字节码的目的就是为了进一步提高可移植性。C#的一个很明显的特点就是其将组件和对象模型直接集成，所以说C#是高效的、稳定的、安全的。

2.3 SQL Server数据库

SQL Server可以在几乎所有Windows操作系统上运行，这使得SQL Server成为在微软平台下数据库的第一选择。SQL Server首先基本兼容ANSI SQL数据库语言，使得软件开发工程师几乎不需要付出额外学习SQL语句的时间就可以快速上手。另外SQL Server在数据安全性方面也远远优于市面上其他同类数据库，可以有效防止SQL注入等问题。经过几代更新，现在发生在SQL Server上的安全问题已经越来越少了。

SQL Server 还有一个很大的优势就是其支持可视化界面的操作，程序员可以在可视化界面下通过简单的操作完成表单的建立、数据的查找和信息的更新。而且 SQL 语句可以内嵌到网页代码中，当网页执行到相关SQL语句时，只需要进行响应的数据库操作就可以得到数据库反馈的信息。

2.4 前端框架Semantic

Semantic UI是最新语义化设计的前段框架，目前项目托管在开源社区上，这使得有更多的人可以参与到该框架的设计与开发中来，也保证了该框架的质量。Semantic UI可以使用任何HTML标签来表现UI控件，这就为前端工程师提供可以通过标准化语句使用框架的方法, 既保证了美观，又保证了代码的可读性。目前Semantic UI能够得到诸如主流浏览器的支持, 所以在适用性方面Semantic UI完全可以使用在当前的网页上。

3. 系统模块及功能

3.1 模块设计

本系统功能模块由以下几部分组成：账户登录模块、超级管理员功能模块、柜台营业员功能模块、用户功能模块。其中登录模块实现了对三种不同用户个人信息的验证，超级管理员模块实现了新业务的增删和柜台营业员信息的增删，柜台营业员模块实现了新用户的创建和用户基本信息修改。除此之外，用户功能模块实现了诸如用户缴费、套餐变更、业务办理、个人信息查询等功能，是整个系统中最复杂的部分。

3.1.1 登录模块

登录模块主要实现了对超级管理员、营业厅员工和普通用户登录系统的功能。三种不同 身份的用户使用相同的登录界面进行登录,但是通过三种不同长度的帐来区分。一旦完成对使用者身份的确认,就可以转跳 到登录界面,否则系统通过弹窗􏰀示用户用户名或密码错误,并且会重新自动转跳到用户登录界面。

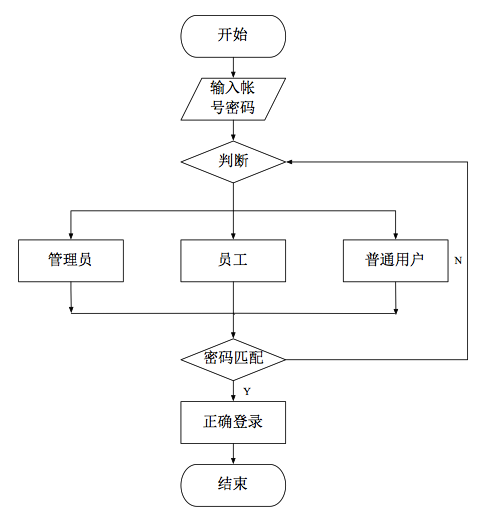


图1：用户登录模块

3.1.2 超级管理员模块

超级管理员模块实现了查找和创建新员工的功能。超级管理员可以通过输入营业厅员工的工号实现对员工基本信息的调取，如果查找不到此工号的员工还可以创建新员工，填写新员工必要信息。超级管理员还可以对能够提供给用户开通的套餐进行增删，如果删除的套餐已有用户开通，则对开通次套餐的用户进行提醒。另外，超级管理员可以管理可供使用的充值卡信息，录入充值卡卡号。

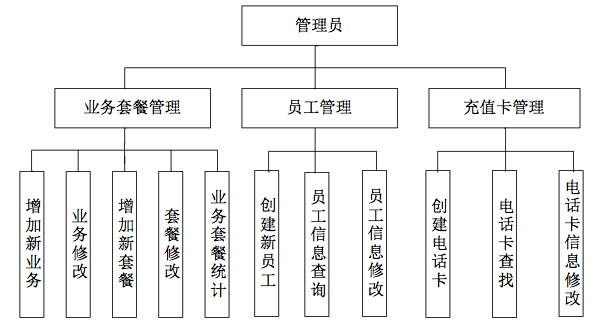


图2：超级管理员模块

3.1.3 营业厅员工模块

营业厅员工模块提供给营业厅员工查找和创建新用户的功能。营业厅员工可以通过输入用户的手机号对用户基本信息进行查询和修改，可以帮助员工开通任何想开通的业务或套餐。如果查询的手机号不存在还可以创建新手机用户，填写新用户必要的信息。

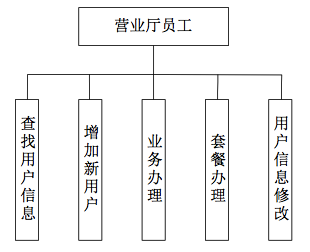


图3：营业厅员工信息

3.1.4 普通用户模块

普通用户模块实现了用户对个人信息查询和修改功能，还提供给用户快捷的支付方式。其中可供查询的信息有当前话费余额、剩余流量、剩余积分、缴费记录和号码归属地等。话 费充值功能提供了三种方式，分别是网银充值、积分兑换和充值卡充值。另外，用户还可以对开通的初始套餐根据个人使用情况进行变更，在套餐之外用户还可以选择额外的业务开通。

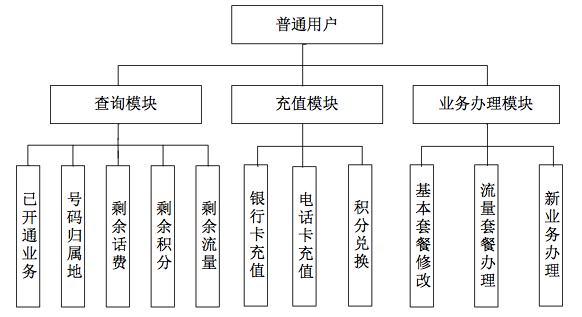


图4：普通用户模块

4. 系统实现与界面展示

4.1 登录界面

本系统三种权限的用户都使用相同的登录界面。当使用本系统时，各种权限的用户都需要先到登录界面输入帐号密码进行登录。



图5：登录界面

4.2 管理员模块

管理员模块主要实现的是查找员工信息、创建新用户、创建新套餐、修改套餐信息、创 建新业务、修改业务信息和开通情况统计。一级功能目录下分为用户管理、套餐业务管理电话卡管理。用户管理模块下包括用户查找、用户基本信息显示和创建新员工的功能。套餐业务管理模块下包括新套餐创建、修改套餐信息、创建新业务、修改业务信息和开通情况统计功能。电话卡管理包括显示当前已有电话卡和创建新电话卡。

4.2.1 员工管理功能

管理员可以通过员工工号查找员工的信息,点击按钮后系统会在数据库中通过员工工号检索用户信息,当查找成功时返回用户的工号、登录密码、工作位置和职位等基本信息。



图6：查询用户信息界面

4.2.2 创建员工功能

创建员工时需要在表单中填写员工工号、员工登录密码、所在营业厅地址和职位信息。其中员工工号必须是9位数，一旦系统检测到工号长度不为9为则会进行提醒。



图7：新员工创建界面

4.2.3 套餐管理功能

套餐管理功能包括对基础套餐和流量套餐信息的编辑和删除功能。



图8：套餐信息编辑界面

管理员还可以创建尚不在套餐列表中的套餐。



图9：套餐创建界面

4.2.4 业务管理功能

业务管理功能包括对可开通业务的编辑和删除功能。管理员还可以创建尚不在业务列表中的业务。



图10：业务编辑界面



图11：业务创建界面

4.2.5 充值卡管理

管理员可以创建尚不存在的存储卡，存储卡被创建之后可以可以被用户用于充值话费，每张充值卡有设置一位有效位，有效位用于标记充值卡是否被使用。管理员可以通过后台管理所有已存在的充值卡，并且可以实现对已使用充值卡的快速删除。



图12：充值卡创建



图13：充值卡管理

4.3 营业厅员工模块

营业厅员工模块主要是实现对用户的管理，实现的功能包括用户查找、创建新用户、用户基本信息修改、用户已开通套餐业务修改。

4.3.1 用户查询功能

营业厅员工可以通过手机号查找用户的信息，点击按钮后系统会在数据库中通过用户手机号检索用户信息，当查找成功时返回用户的手机号、登录密码、号码归属地和当前基本等基本信息。



图14：查询用户信息模块



图15：用户基本信息模块

4.3.2 创建新用户功能

创建新用户时需要在表单中填写手机号、用户登录密码、号码归属地、个人邮箱等基本信息。在所有属性中，用户手机号、登录密码、号码归属地、初始积分、个人邮箱和初始话费是必填部分，这些部分用于初始信息的查询,如果不填写则会提示错误。而剩下的银行卡 卡号、支付密码、套餐和业务则是选填部分，即使不填写也不会有影响。



图16：创建新用户

4.4 普通用户模块

普通用户模块主要是实现用户基础套餐、流量套餐和基本业务的办理功能。其中基础套餐和流量套餐是单选的，业务可以多选。除此之外，用户还可以对一些基本信息进行查询。



图17：基础套餐办理

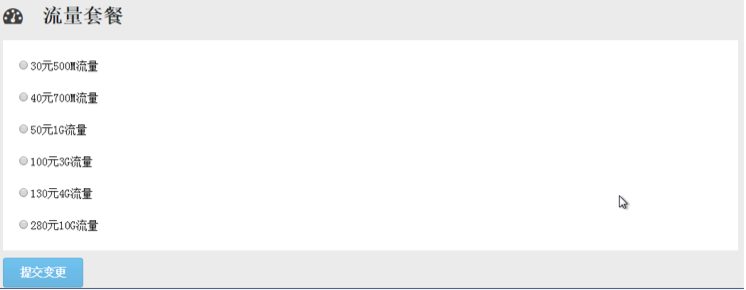


图18：流量套餐办理



图19：业务办理

4.4.1 信息查询功能

用户可以对个人及一些基本信息进行查询，在此不做一一展示，只展示用户查询已开通业务的查询。



图20：业务查询界面

4.4.2 话费充值

用户可以通过积分兑换、手机充值卡和银行卡三种方式充值。



图21：话费充值界面

5. 心得与体会

本次多媒体大作业基于之前的数据库大作业，但是进行了很多改进，尤其是在界面方面花了很多心思去设计和优化。另外，进一步完善了功能，加入了用户模块以及充值卡功能，这在之前的版本中并没有出现。对于界面的布局借鉴了当前比较流行的设计样式，最上端为一级导航栏，在左侧的导航是在一级导航下的二级导航，使得全站的功能进一步清晰。除此以外，在使用简单的HTML、CSS、Javascript以外，还特地学习了前端最近比较流行的框架Semantic UI做了进一步美化，可用性和美观性都大大提高。