

Tarena Teaching System

JAVA企业及互联网项目-NetCTOSS

学员用书



Java企业应用及互联网
高级工程师培训课程

目 录

系统需求说明书.....	1
1. 引言	1
2. 项目概述.....	2
3. 功能概述.....	8
4. “基础信息”模块	11
5. “角色管理”模块	17
6. “管理员管理”模块	23
7. “资费管理”模块	31
8. “帐务账号管理”模块.....	40
9. “业务账号管理”模块	52
10. “账单管理”模块	63
11. “报表”模块.....	67
数据库设计说明书.....	70
1. 概述	70
2. 数据表定义.....	71
静态结构.....	83
案例详解.....	84
1. 实现资费列表显示功能.....	84
2. 实现帐务账号的增改查以及暂停	110
3. 实现业务账号的列表显示以及开通	158

系统需求说明书

1. 引言

1.1. 编写目的

本文档的目的是分析 NetCTOSS 项目的系统需求，确定业务范围和核心功能，定义项目的专业名词，为后续的系统设计人员、开发人员和测试人员提供功能需求和非功能需求的详细定义，为测试人员提供测试用例设计的功能参考。

1.2. 阅读对象

本文档适用于 NetCTOSS 项目的需求分析人员、设计人员、开发人员、测试人员、运维人员和项目管理人员。

1.3. 术语定义

- **权限**：权限是使用者操作系统中功能模块的能力，如“角色管理”模块、“资费管理”模块和“账单管理”模块等。通过指定权限，可将使用者的操作限定在指定的范围内，以保证操作的安全。例如，一个使用者拥有“资费管理”权限，表示该使用者可以操作“资费管理”模块，以执行资费管理的操作。使用者对系统的各功能模块的访问都是受权限控制的，因此，权限决定了使用者可以执行的操作。
- **角色**：角色是多种权限的集合，可通过授予角色来简化权限的管理。比如，角色“账号管理员”同时具有“帐务账号”模块和“业务账号”模块的权限。如果设置某使用者的角色为“账号管理员”，则意味着该使用者可以同时操作“帐务账号”和“业务账号”模块。
- **“超级管理员”角色**：系统内置的角色，此角色拥有所有的功能权限，即可以操作所有的功能模块。
- **超级管理员**：系统内置的管理员，登录名为“SuperAdmin”，密码与登录名相同。该管理员拥有“超级管理员”角色，作为 NetCTOSS 系统的第一位管理员，也是第一位可以成功登录 NetCTOSS 系统的使用者。超级管理员登录后可以修改个人资料和密码。
- **管理员**：即普通管理员，由超级管理员登录后创建，并为其设置角色以配置权限。普通管理员一旦被赋予某种角色，则可以操作系统中相应的功能模块，以实现系统管理功能。
- **资费**：运营商对于客户使用服务的收费标准。运营商会定义多种收费标准由客户选择。比如，客户在 1 个月之内使用 20 小时，费用为 20.45 元；超出 20 小时的部分，以每分钟 0.01 元计费。具有资费管理权限的管理员可以修改资费。

• **帐务账号**：运营商对于客户收费的账号，对应自然人。客户如果需要使用运营商提供的服务，则首先使用身份证件开设帐务账号，客户费用结算的信息就是通过帐务账号来管理。帐务账号主要记录客户的基本信息、支付方式、所申请的各种业务账号等信息。每个客户只能开设一个帐务账号。

• **业务账号**：客户开设了帐务账号之后，就可以申请开通其所需要的业务。客户在开设帐务账号之后，可以开设多个业务账号，每个业务账号表示单独的业务。同一个客户可以使用多个 Unix 服务器上的服务，分别算作不同的业务，因此需要开设并使用不同的业务账号。比如某客户需要使用 20 服务器上的 telnet 远程登录服务，则需要开设一个业务账号；如果同时还需要使用 23 服务器上的 telnet 远程登录服务，则需要再开设一个业务账号。每个业务账号中包含资费类别，用于统计每种业务的费用。

• **账单管理**：即费用查询。客户的费用就是其帐务账号下属所有业务账号费用的总计。拥有“账单管理”权限的管理员可以查询某客户（即某业务账号）的所有业务的总费用；也可以查询该帐务账号下每种业务的费用及其明细。

• **报表**：统计报表，可以统计各种数据，如“客户使用时长”报表用于显示每位客户每月的累计时长；“时长排行榜”报表显示每台服务器上累计时长最高的前三名客户；“资费使用率”报表统计并显示每台服务器上每种资费标准的使用次数。

- **个人信息**：管理员登录后，可以修改个人的基本信息，如姓名、电话等。
- **客户**：需要使用 Unix 实验室出租业务的用户。客户通过电信公司提供的账号和密码登录指定的 Unix 机器，并使用 Unix 服务。
- **OS 账号**：客户登录所使用的账号，即 OS 账号，也是 Unix 服务器的系统区分登录用户的唯一标识。费用详单中记载的也是 OS 账号及其登录信息。

2. 项目概述

2.1. 项目背景

在电信的业务中，有一种 Unix 实验室出租业务。只要用户向电信运营商申请一个 Unix 账号（OS 账号），就可以远程登录 Unix 实验室，以使用 Unix 系统。任何用户登录电信运营商提供的 Unix 实验室的 Unix 系统时，Unix 系统都会记录该 OS 账号的登入和登出信息，这些信息都保存在 Unix 的系统日志文件中。

用户使用电信运营商提供的 Unix 实验室的服务需要缴纳一定的费用。因此，电信运营商需要一套系统，将用户登录实验室的时间长度数据采集起来，以作为对用户的收费依据。

数据采集程序采集到登录数据之后，首先需要对数据进行解析和匹配等处理，然后需要将数据发送到服务器端。服务器端得到数据之后，需要将数据保存为数据文件，同时将数据存入数据库。另外，还需要计费管理系统来实现费用的管理和查看。

整个系统的结构组成如图 - 1 所示：

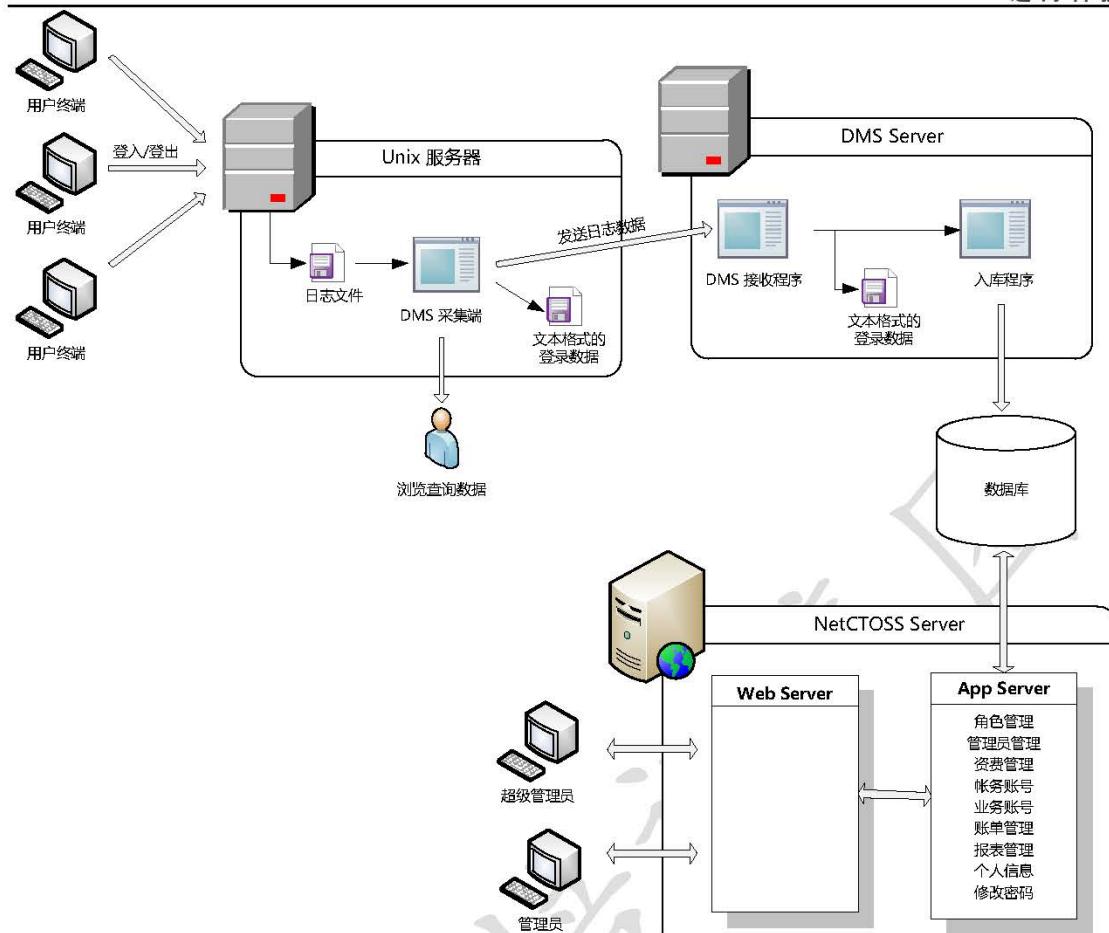


图 - 1

由图 - 1 可以看出，整个系统由 DMS 采集端、DMS 服务器端和 NetCTOSS 服务器端三部分组成。

DMS 采集端的主要任务是采集、解析、匹配和保存并发送登录数据。DMS 采集端程序通过读取 Unix 系统日志文件中记录的所有 OS 账号的登入/登出的原始记录，解析找出成对的登入/登出记录，从而得到 OS 账号登录实验室的登入时刻、登出时刻、登录时间长度等数据。采集、解析以及匹配成功后，DMS 采集端程序需要将处理好的数据以可阅读的文本形式发送到 DMS 服务器端。

DMS 服务器端的主要任务是，通过接收程序接收到日志数据，然后将日志数据备份保存为文件数据，并提供 UI 用户界面实现数据的浏览和监控等功能。另外，DMS 服务器还需要通过入库程序将日志数据存入数据库。

NetCTOSS 系统的主要任务是提供对于数据的查询和管理功能。日志数据存入数据库后，可以通过 NetCTOSS 系统来实现查询和管理功能。管理员登录 NetCTOSS 系统后，可以进行角色管理、帐务账号管理、业务账号管理、账单管理以及报表数据查询等操作。

2.2. 业务概述

2.2.1. 业务描述

NetCTOSS 系统作为整个项目的一个组成部分，是一个 B/S 结构的 Web 应用程序，其详细结构组成如图 - 2 所示：

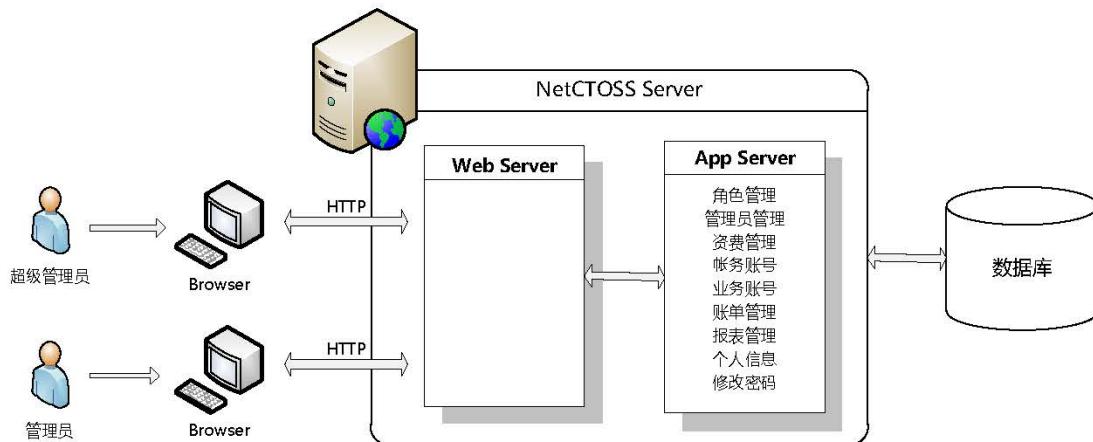


图 - 2

由图 - 2 可以看出，DMS 采集端将数据发送到 DMS 服务器端并存入数据库之后，由 NetCTOSS 系统实现对数据的管理和查询。NetCTOSS 系统作为一个后台管理系统，只有具有管理员权限的使用者通过浏览器成功登录 Web 服务器之后，才可以使用应用服务器所提供的各种业务，如“角色管理”、“资费管理”和“账单管理”等。

2.2.2. 功能描述

NetCTOSS 系统具有的功能有：“角色管理”、“管理员管理”、“资费管理”、“帐务账号管理”、“业务账号管理”、“账单管理”、“报表”、“个人信息”和“修改密码”。每个管理员登录成功后，都可以使用“个人信息”和“修改密码”功能，以实现个人信息和密码的维护，是否可以使用其他功能则取决于该管理员所拥有的权限。

NetCTOSS 系统的功能如图 - 3 所示：

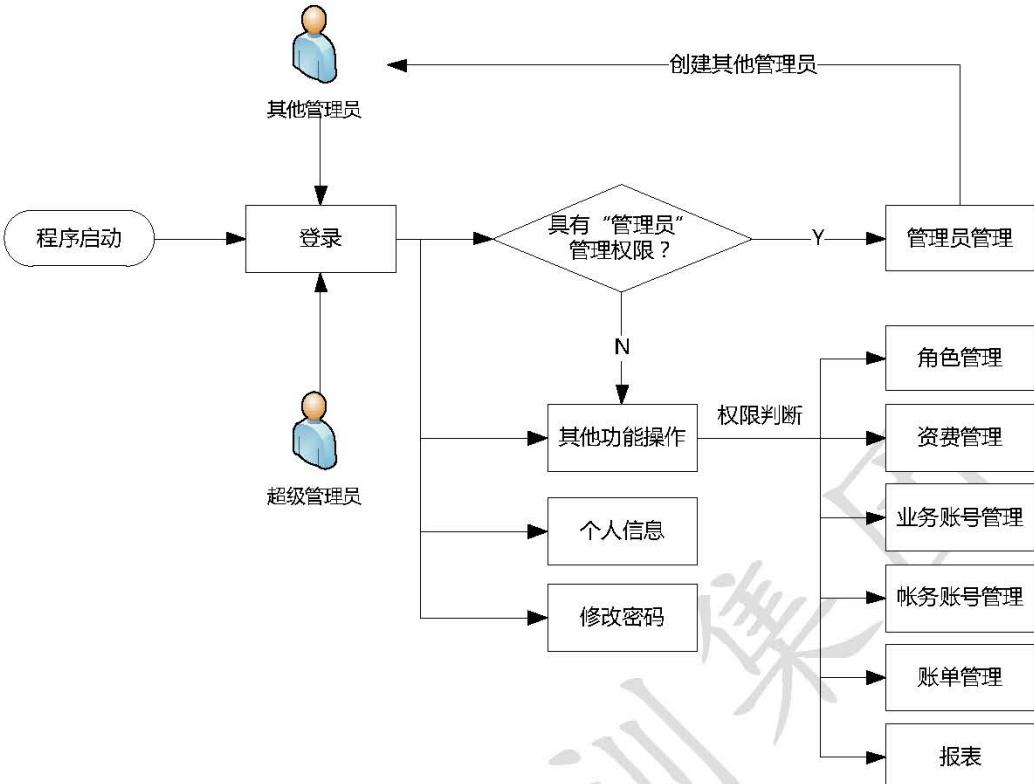


图 - 3

由图 - 3 可以看出，管理员登录成功后可以进行的操作完全取决于其所拥有的权限。NetCTOSS 系统会内置好一个“超级管理员”，作为使用该系统的第一位管理员用户。超级管理员成功登录 NetCTOSS 系统后，进入系统主页，如图 - 4 所示：



图 - 4

由图 - 4 可以看出，主页是 NetCTOSS 系统的功能导航页，根据登录的管理员的权限的不同，主页上所显示的功能操作也各不相同。如果是超级管理员登录，则可以操作系统的所有功能模块，其中包括“角色管理”和“管理员管理”模块。

超级管理员成功登录系统后，则可以通过“角色管理”和“管理员管理”模块来创建其他管理员，以共同使用 NetCTOSS 系统。

如果是权限受限的普通管理员登录，则其登录成功后的主页可能如图 - 5 所示：



图 - 5

由图 - 5 可以看出，此管理员为普通管理员，即权限受限的管理员，只能操作“账单管理”、“报表”、“个人信息”和“修改密码”4 个功能模块。

登录成功的管理员通过主页上的功能链接，则可以导航定位到相应的功能页面。

如果操作者直接在浏览器的 URL 地址栏中输入无权访问的页面地址，则将定位到无权访问的错误提示页，如图 - 6 所示：



图 - 6

图 - 6 所示的页面具有自动跳转功能，即该页面 5 秒后将自动跳转到主页，页面上表示剩余秒数的数字将自动递减。如果操作者希望立即跳转，则可以单击页面上的“返回”链接，直接定位到主页。

操作者在访问网站的过程中，如果发生异常或者访问错误，则将显示系统的错误页，如图 - 7 所示：



图 - 7

图 - 7 所示的系统错误页面也具有自动跳转功能，即该页面 5 秒后将自动跳转到登录页面，页面上表示剩余秒数的数字将自动递减。如果操作者希望立即跳转，则可以单击页面上的“返回”链接，直接定位到登录页面，重新登录。

2.2.3. 系统流程

因为不同权限的管理员登录系统后可执行的操作不同，因此，为方便系统流程的讲解，我们以超级管理员登录为例。

超级管理员登录 NetCTOSS 系统后，系统流程如图 - 8 所示：

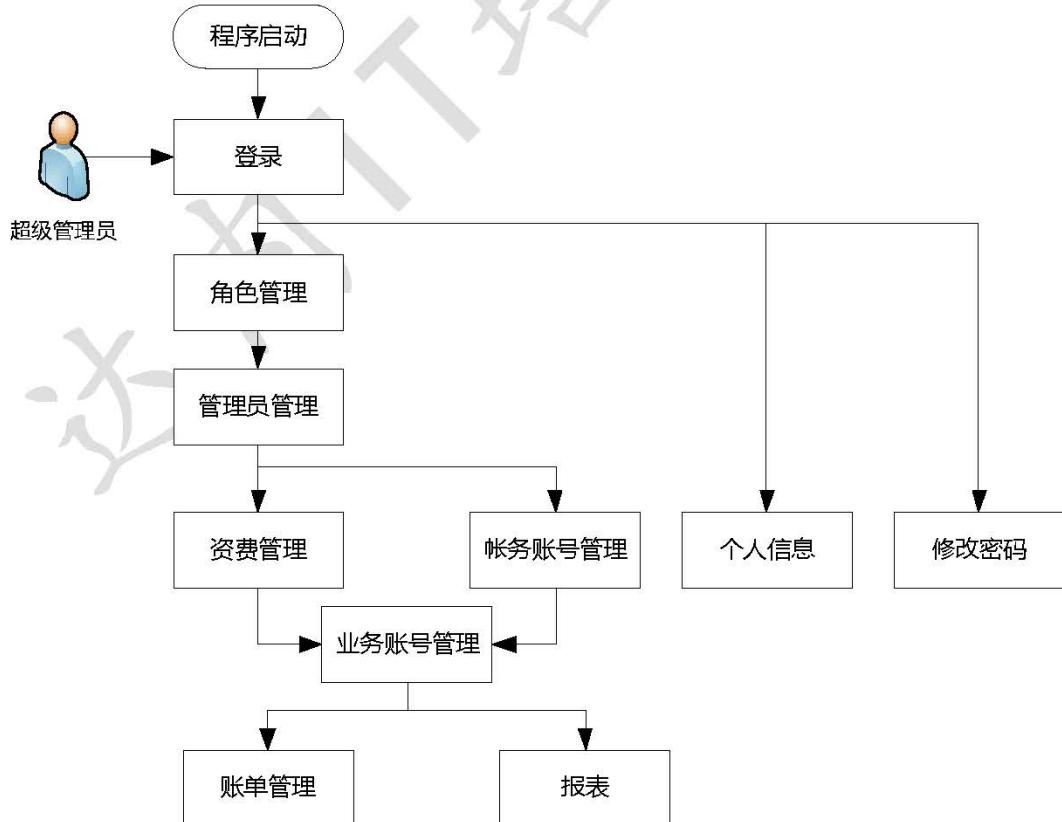


图 - 8

由图 - 8 可以看出，NetCTOSS 系统的详细流程为：

1. 超级管理员登录成功后，可以进行个人信息和密码的维护；
2. 可以进行角色管理，即创建不同的角色以操作相应的功能模块；
3. 然后进行管理员管理，即创建新的管理员，并为其赋予角色（即赋予操作相应功能模块的权限）；
4. 进行资费管理，即创建不同的资费供客户选择使用；
5. 进行帐务账号管理，即为客户开设帐务账号；
6. 进行业务账号管理，即为某帐务账号开设业务账号，并选择具体的资费标准；
7. 客户开设业务账号后，则可以使用具体的业务，管理员可以对所有客户的账单进行管理和查询；
8. 定期进行报表管理。

如果是其他管理员登录，则根据其权限的不同，进行相应的操作。

3. 功能概述

3.1. 功能模块图

NetCTOSS 系统的功能模块图如图 - 9 示：



图 - 9

3.2. 系统用例清单

编号	模块	用例定义	用例描述	重要程度
1	基础信息	登录	管理员使用用户名和密码登录系统	A
		主页	显示主页，并根据权限显示可操作的功能模块	A
		修改密码	管理员修改密码	A
		查看个人信息	查看管理员的个人信息，如角色、姓名、电话和创建时间等	A
		修改个人信息	修改管理员的个人信息中可修改的部分，如姓名和电话等	A
2	角色管理	角色浏览	查看所有的角色	A
		增加角色	增加新角色	A
		修改角色	修改某角色的信息，如角色名称和角色所拥有的权限	A
		删除角色	删除某角色	A
3	管理员管理	管理员浏览	查看所有的管理员	A
		查询管理员	根据查询条件查询管理员数据	A
		管理员重置密码	为管理员重置密码为初始密码	A
		增加管理员	增加管理员	A
		修改管理员	修改管理员的个人信息以及角色信息	A
		删除管理员	删除管理员	A
4	资费管理	资费浏览	查看所有的资费	A
		资费排序	根据月租、基费或者时长对所有的资费数据进行排序	A
		资费详细	查看资费的详细信息	A
		增加资费	增加新资费	A
		修改资费	修改没有启用的资费	A
		启用资费	启用新增加的资费	A
		删除资费	删除没有启用的资费	A
5	帐务账号管理	帐务账号浏览	查看所有的帐务账号	A
		查询帐务账号	根据查询条件查询帐务账号	A
		帐务账号详细	查看某帐务账号的详细信息	A
		增加帐务账号	增加新的帐务账号	A

		修改帐务账号	修改某帐务账号的信息，如密码、个人信息等	A
		暂停帐务账号	暂停某帐务账号	A
		开通帐务账号	开通某帐务账号	A
		删除帐务账号	删除某帐务账号	A
6	业务账号管理	业务账号浏览	查看所有的业务账号	A
		查询业务账号	根据查询条件查询业务账号	A
		业务账号详细	查看某业务账号的详细信息	A
		增加业务账号	为某帐务账号增加新的业务账号	A
		修改业务账号	修改某业务账号的信息	A
		暂停业务账号	暂停某业务账号	A
		开通业务账号	开通某业务账号	A
		删除业务账号	删除某业务账号	A
7	账单管理	账单浏览	查看所有的账单	A
		查询账单	根据查询条件查询账单	A
		账单明细浏览	查看某帐务账号账单的所有业务账单	A
		业务详单浏览	查看某业务账单的所有明细	A
8	报表	报表浏览	查看报表	A

3.3. 系统用例图

NetCTOSS 系统的用例图如图 - 10 所示：

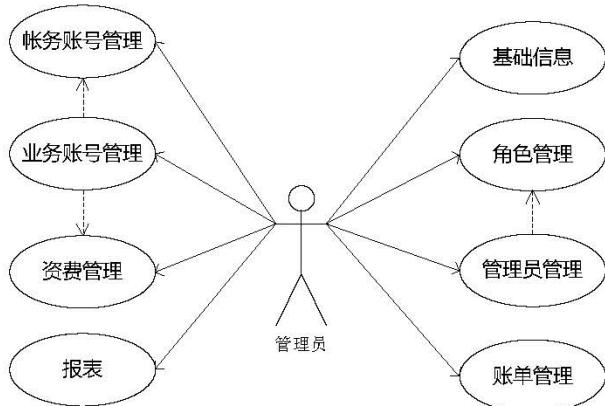


图 - 10

4. “基础信息”模块

4.1. 用例图

NetCTOSS 系统中基础信息模块的用例图如图 - 11 所示：

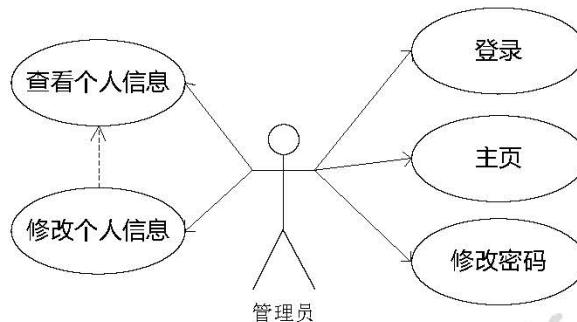


图 - 11

4.2. 用例描述

4.2.1. 登录

用例名称	登录
功能简述	管理员使用用户名和密码登录 NetCTOSS 系统，界面如图 - 12 所示
前置条件	系统成功内置超级管理员，或者超级管理员已经成功创建其他管理员
后置条件	如果登录用例成功，则进入主页
基本流	<ol style="list-style-type: none"> 录入账号、密码和验证码； 单击页面的“登录”按钮； 对录入的账号、密码和验证码进行输入合法性验证，验证规则如表 - 1 所示； 如果录入的账号、密码和验证码验证通过，则提交数据到服务器端进行登录判断，否则在界面显示录入错误的提示，如图 - 13 所示； 数据提交到服务器端后，查询数据库，检查所录入的账号和密码是否正确； 如果账号和密码正确，则成功登录，进入主页；否则，在页面提示用户名或者密码错误，如图 - 14 所示。
分支与异常	无
备注	无



图 - 12

表 - 1 “登录”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
账号	文本	非空，且为 30 长度以内的字母、数字和下划线的组合
密码	文本	非空，且为 30 长度以内的字母、数字和下划线的组合
验证码	文本	非空，必须和验证码图片上的文本相同



图 - 13



图 - 14

4.2.2. 主页

用例名称	主页
功能简述	主页是 NetCTOSS 系统的功能导航页，显示当前操作员可以操作的功能链接。根据登录的管理员的权限的不同，主页上所显示的功能操作也各不相同。如果是超级管理员登录，则可以操作 NetCTOSS 系统的所有功能模块，如图 - 4 所示；如果是权限受限的普通管理员登录，则只能操作其权限范围里所允许的功能模块，如图 - 5 所示。
前置条件	登录用例执行成功
后置条件	如果这个用例成功，则由操作者选择其他操作
基本流	<ol style="list-style-type: none"> 1. 查询当前登录的管理员的权限； 2. 根据权限显示可操作的模块链接； 3. 单击页面的“退出”链接，则退出当前系统，回到登录页面。
分支与异常	<p>第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；</p> <p>第 2 步：如果没有查询到当前管理员的权限数据（没有为其设置权限），则只显示主页、个人信息和修改密码三个功能模块的链接。</p>
备注	<p>1. 主页、个人信息和修改密码模块属于每个管理员均可操作的模块，不受权限控制。</p>

4.2.3. 修改密码

用例名称	修改密码
功能简述	管理员登录后，可以修改个人的登录密码，界面如图 - 15 所示
前置条件	管理员登录成功
后置条件	无
基本流	<ol style="list-style-type: none"> 1. 管理员登录后，单击“修改密码”的功能链接，则可以进入“修改密码”界面，界面如图 - 15 所示； 2. 在界面录入旧密码、新密码和重复新密码，其中，红色 * 号表示必须录入的数据项； 3. 单击页面的“保存”按钮； 4. 对录入的旧密码、新密码和重复录入的新密码进行输入合法性验证，验证规则如表 - 2 所示； 5. 如果录入的数据验证通过，则提交数据到服务器端，否则在界面显示录入错误的提示，如图 - 16 所示； 6. 数据提交到服务器端后，进行修改密码的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 17 所示，该提示信息 3 秒后自动消失； 7. 数据提交到服务器端后，进行修改密码的操作，如果操作失败，则在页面提示错误信息，如图 - 18 所示，该提示信息 3 秒后自动消失； 8. 如果单击页面的“取消”按钮，则回到主页。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面。
备注	<p>1. 录入的旧密码正确，才允许修改密码。</p>

The screenshot shows the 'NetCTOSS' application interface. At the top, there's a navigation bar with icons for Home, Role Management, Admin, Cost Management, Account Management, Business Account, Bill Management, Reports, Personal Information, and Modify Password. Below the navigation bar is a form titled 'Modify Password'. It contains three input fields: 'Old Password', 'New Password', and 'Repeat New Password'. Each field has a validation message next to it: 'Old Password' says '30 length of letters, numbers and underscores', 'New Password' says '30 length of letters, numbers and underscores', and 'Repeat New Password' says 'The two new passwords must be the same'. At the bottom of the form are 'Save' and 'Cancel' buttons.

图 - 15

表 - 2 “修改密码”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
旧密码	文本	非空，且为 30 长度以内的字母、数字和下划线的组合
新密码	文本	非空，且为 30 长度以内的字母、数字和下划线的组合
重复新密码	文本	非空，为 30 长度以内的字母、数字和下划线的组合，且必须和新密码相同

This screenshot shows the same 'Modify Password' page as Figure 15, but with validation errors. The 'Old Password' field is empty and has a red asterisk, indicating it is required. The 'New Password' and 'Repeat New Password' fields both contain three dots ('...') and have red asterisks, indicating they are required and contain invalid characters. The validation messages are identical to those in Figure 15: '30 length of letters, numbers and underscores' for the new password and 'The two new passwords must be the same' for the repeat password. The 'Save' and 'Cancel' buttons are at the bottom.

图 - 16

This screenshot shows the 'Modify Password' page after a successful save. A green success message box at the top left says '保存成功!' (Save successful!). The main form below has the same structure as Figures 15 and 16, with fields for 'Old Password', 'New Password', and 'Repeat New Password'. Each field has its own validation message: 'Old Password' (30 length of letters, numbers and underscores), 'New Password' (30 length of letters, numbers and underscores), and 'Repeat New Password' (The two new passwords must be the same). The 'Save' and 'Cancel' buttons are at the bottom.

图 - 17

保存失败，旧密码错误！

旧密码: * 30长度以内的字母、数字和下划线的组合

新密码: * 30长度以内的字母、数字和下划线的组合

重複新密码: * 两次新密码必须相同

保存

取消

图 - 18

4.2.4. 查看个人信息

用例名称	查看个人信息
功能简述	管理员登录后，可以查看管理员的个人信息，界面如图 - 19 所示
前置条件	管理员登录成功
后置条件	管理员可以修改个人信息
基本流	<ol style="list-style-type: none"> 管理员登录后，单击“个人信息”的功能链接，则进入“个人信息”界面，界面如图 - 19 所示； 界面上显示管理员的个人信息，其中，管理员账号、角色以及创建时间三项为只读项。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面。
备注	无



The screenshot shows the NetCTOSS system's personal information editing interface. At the top, there is a navigation bar with links for Home, Role Management, Admin, Consumption Management, Account Number, Business Account, Bill Management, Reports, Personal Information, and Modify Password. On the right side of the header is an 'Logout' button.

The main content area contains a form for updating personal information:

- 管理员帐号:
- 角色:
- 姓名: * 20长度以内的汉字、字母的组合
- 电话: 电话号码格式: 手机或固话
- Email: 50长度以内, 符合 email 格式
- 创建时间:

At the bottom of the form are two buttons: '保存' (Save) and '取消' (Cancel).

At the very bottom of the page, there is a footer note: [源自北美的技术, 最优秀的师资, 最真实的企业环境, 最适用的实战项目] 版权所有(C)加拿大达内IT培训集团公司

图 - 19

4.2.5. 修改个人信息

用例名称	修改个人信息
功能简述	管理员登录后，可以修改个人信息，并保存
前置条件	“查看个人信息”用例成功
后置条件	无
基本流	<ol style="list-style-type: none"> 管理员登录后，单击“个人信息”的功能链接，则可以进入“个人信息”界面，以查看管理员的个人信息，界面如图 - 19 所示； 在界面上修改录入个人信息，其中，红色 * 号表示必须录入的数据项； 单击“保存”按钮； 对录入的数据进行输入合法性验证，验证规则如表 - 3 所示； 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 20 所示； 如果录入的数据验证通过，则提交数据到服务器端，进行修改个人信息的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 21 所示，该提示信息 3 秒后自动消失； 数据提交到服务器端后，进行修改密码的操作，如果操作失败，则在页面提示错误信息，如图 - 22 所示，该提示信息 3 秒后自动消失； 如果单击页面的“取消”按钮，则回到主页。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面。
备注	无



The screenshot shows a form for modifying personal information. The fields and their current values are:

- 管理员帐号: admin1
- 角色: 账单管理员, 业务帐号
- 姓名: a1234 * 20长度以内的汉字、字母的组合
- 电话: 138923456782345 * 电话号码格式: 手机或固话
- Email: aaaaaaaaaaaaaaaa * 50长度以内, 符合 email 格式
- 创建时间: 2012/12/12 13:10:34

At the bottom are two buttons: 保存 (Save) and 取消 (Cancel).

图 - 20

表 - 3 “修改个人信息”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
姓名	文本	非空，20 长度以内的汉字、字母的组合
电话	文本	电话号码格式，符合手机号码或固定电话号码格式
Email	文本	50 长度以内，符合 email 格式

✓ 保存成功！

管理员帐号:	admin1	
角色:	账单管理员, 业务帐号	
姓名:	李四 *	20长度以内的汉字、字母的组合
电话:	13692314567	电话号码格式: 手机或固话
Email:	bb@bb.com	50长度以内, 符合 email 格式
创建时间:	2012/12/12 13:10:34	

保存 **取消**

图 - 21

✗ 保存失败, 数据并发错误!

管理员帐号:	admin1	
角色:	账单管理员, 业务帐号	
姓名:	张三 *	20长度以内的汉字、字母的组合
电话:	13892345678	电话号码格式: 手机或固话
Email:	aa@aa.com	50长度以内, 符合 email 格式
创建时间:	2012/12/12 13:10:34	

保存 **取消**

图 - 22

5. “角色管理”模块

5.1. 用例图

NetCTOSS 系统中“角色管理”模块的用例图如图 - 23 所示：

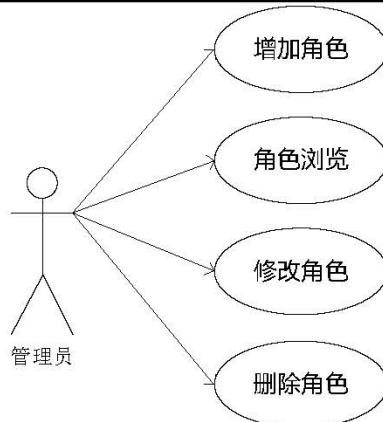


图 - 23

5.2. 用例描述

5.2.1. 角色浏览

用例名称	角色浏览
功能简述	具有“角色管理”权限的管理员登录后，可以查看所有角色信息
前置条件	具有“角色管理”权限的管理员登录成功
后置条件	可以执行增加角色、修改角色或者删除角色的用例
基本流	<ol style="list-style-type: none"> 具有“角色管理”权限的管理员登录后，单击“角色管理”的功能链接，则可以进入“角色管理”界面，以查看所有的角色信息，界面如图 - 24 所示； 界面上将分页列出所有的角色信息，每页显示 10 条数据； 如果数据小于等于 10 条，则不显示分页的页码； 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并用超级链接的方式显示其他页码； 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示； 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化； 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化； 单击“增加”按钮，则执行“增加角色”用例； 单击表格中某行中的“修改”按钮，则执行“修改角色”用例； 单击表格中某行中的“删除”按钮，则执行“删除角色”用例。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“角色管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	无



图 - 24

5.2.2. 增加角色

用例名称	增加角色
功能简述	具有“角色管理”权限的管理员登录后，可以增加新角色
前置条件	具有“角色管理”权限的管理员登录成功，且“角色浏览”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“角色管理”权限的管理员登录后，单击“角色管理”的功能链接，则可以进入“角色管理”界面，然后单击界面上的“增加”按钮，则进入“增加角色”界面，界面如图 - 25 所示； 在界面上录入角色名称，并为该角色至少选择一个权限，其中，红色 * 号表示必须录入的数据项； 单击“保存”按钮； 对录入的数据进行输入合法性验证，验证规则如表 - 4 所示； 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 26 所示； 如果录入的数据验证通过，则提交数据到服务器端，进行增加角色的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 27 所示，该提示信息 3 秒后自动消失； 数据提交到服务器端后，进行增加角色的操作，如果操作失败，则在页面提示错误信息，如图 - 28 所示，该提示信息 3 秒后自动消失； 如果单击页面的“取消”按钮，则回到“角色浏览”界面。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“角色管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	1. 角色名称唯一，不能重复

角色名称: * 不能为空，且为20长度的字母、数字和汉字

设置权限: 管理员管理 角色管理 资费管理 * 至少选择一个权限
 帐务帐号 业务帐号 账单
 报表

保存 取消

[源自北美的技术，最优秀的师资，最真实的企业环境，最适合的实战项目]
版权所有(C)加拿大达内IT培训集团公司

图 - 25

表 - 4 “增加角色”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
角色名称	文本	非空，且为 20 长度以内的字母、数字和汉字的组合 角色名称唯一，不能重复
设置权限	文本	选择录入，至少选择一个权限

角色名称: * 不能为空，且为20长度的字母、数字和汉字的组合

设置权限: 管理员管理 角色管理 资费管理 * 至少选择一个权限
 帐务帐号 业务帐号 账单
 报表

保存 取消

图 - 26

✓ 保存成功！

角色名称: 角色管理员 * 不能为空，且为20长度的字母、数字和汉字的组合

设置权限: 管理员管理 角色管理 资费管理 * 至少选择一个权限
 帐务帐号 业务帐号 账单
 报表

保存 取消

图 - 27



图 - 28

5.2.3. 修改角色

用例名称	修改角色
功能简述	具有“角色管理”权限的管理员登录后，可以修改已有角色的数据
前置条件	具有“角色管理”权限的管理员登录成功，且“角色浏览”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“角色管理”权限的管理员登录后，单击“角色管理”的功能链接，则可以进入“角色管理”界面，然后单击界面上某行数据上的“修改”按钮，则进入“修改角色”界面，界面如图 - 29 所示； 界面上将显示所修改角色的数据； 在界面上修改角色名称，并修改角色的权限，其中，红色 * 号表示必须录入的数据项； 单击“保存”按钮； 对录入的数据进行输入合法性验证，验证规则如表 - 4 所示； 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 26 所示； 如果录入的数据验证通过，则提交数据到服务器端，进行修改角色的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 27 所示，该提示信息 3 秒后自动消失； 数据提交到服务器端后，进行修改角色的操作，如果操作失败，则在页面提示错误信息，如图 - 28 所示，该提示信息 3 秒后自动消失； 如果单击页面的“取消”按钮，则回到“角色浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“角色管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页； 第2步：如果没有查询到相应的角色数据，则直接返回到图 - 24 所示的“角色浏览”界面。
备注	1. 角色名称唯一，不能重复

角色名称: 超级管理员 * 不能为空, 且为20长度的字母、数字和汉字的组合

设置权限:

<input checked="" type="checkbox"/> 管理员管理	<input checked="" type="checkbox"/> 角色管理	<input checked="" type="checkbox"/> 资费管理	* 至少选择一个权限
<input checked="" type="checkbox"/> 帐务帐号	<input checked="" type="checkbox"/> 业务帐号	<input checked="" type="checkbox"/> 账单	
<input checked="" type="checkbox"/> 报表			

保存 取消

图 - 29

5.2.4. 删除角色

用例名称	删除角色
功能简述	具有“角色管理”权限的管理员登录后，可以删除已有的角色
前置条件	具有“角色管理”权限的管理员登录成功，且“角色浏览”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“角色管理”权限的管理员登录后，单击“角色管理”的功能链接，则可以进入“角色管理”界面，然后单击界面上某行数据上的“删除”按钮，则显示提示信息，询问是否确定删除该角色数据，界面如图 - 30 所示； 如果单击“取消”按钮，则取消删除操作，回到“角色浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行删除角色的操作，如果操作成功，则在界面提示删除成功，界面如图 - 31 所示； 单击图 - 31 所示界面上的“X”图标关闭提示信息框之后，将刷新“角色浏览”界面上的数据显示； 如果删除角色的操作失败，则在界面提示删除失败的信息，界面如图 - 32 所示； 单击图 - 32 所示界面上的“X”图标关闭提示信息框之后，将回到“角色浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“角色管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	1. 目前被使用的角色不能删除

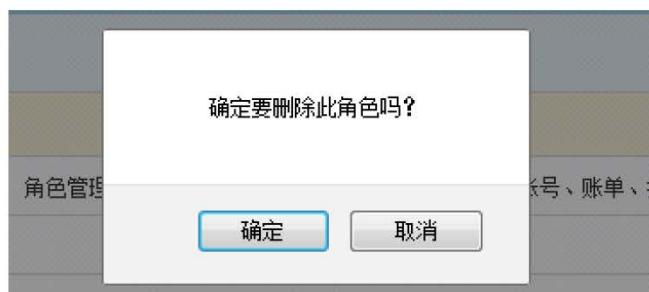


图 - 30

角色名称	拥有的权限	
贾强		<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
贾强	✓ 删除成功!	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
贾强	超级管理员、账单管理员	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
贾强	超级管理员、账单管理员	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除

图 - 31

角色名称	拥有的权限	
贾强		<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
贾强	⚠ 删除错误! 该角色被使用, 不能删除。	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
贾强	超级管理员、账单管理员	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
贾强	超级管理员、账单管理员	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除

图 - 32

6. “管理员管理”模块

6.1. 用例图

NetCTOSS 系统中“管理员管理”模块的用例图如图 - 33 所示：

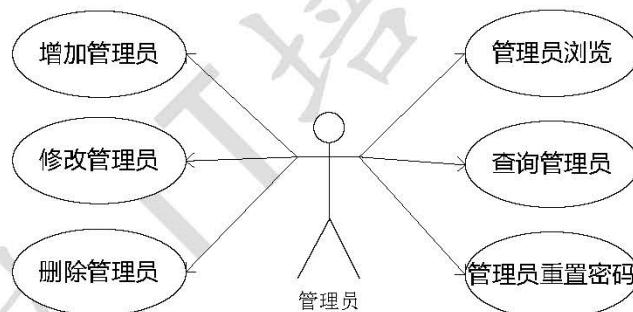


图 - 33

6.2. 用例描述

6.2.1. 管理员浏览

用例名称	管理员浏览
功能简述	具有“管理员管理”权限的管理员登录后，可以查看所有管理员的信息
前置条件	具有“管理员管理”权限的管理员登录成功
后置条件	可以执行增加管理员、查询管理员、管理员重置密码、修改管理员或者删除管理员的用例
基本流	<ol style="list-style-type: none"> 具有“管理员管理”权限的管理员登录后，单击“管理员”的功能链接，则可以进入“管理员管理”界面，以查看所有的管理员信息，界面如图 - 34 所示； 界面上将分页列出所有的管理员信息，每页显示 10 条数据；

	<p>3. 表格中的数据列中，“拥有角色”列只显示当前数据行中管理员的第一个角色名称，其他数据使用省略号替代；</p> <p>4. 当鼠标移入当前行的“拥有角色”列时，使用浮动框显示所有角色的详细信息；</p> <p>5. 如果数据小于等于 10 条，则不显示分页的页码；</p> <p>6. 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并用超级链接的方式显示其他页码；</p> <p>7. 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示；</p> <p>8. 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化；</p> <p>9. 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化；</p> <p>10. 单击“搜索”按钮，则执行“查询管理员”用例；</p> <p>11. 单击“密码重置”按钮，则执行“管理员重置密码”用例；</p> <p>12. 单击“增加”按钮，则执行“增加管理员”用例；</p> <p>13. 单击表格中某行中的“修改”按钮，则执行“修改管理员”用例；</p> <p>14. 单击表格中某行中的“删除”按钮，则执行“删除管理员”用例。</p>
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	无

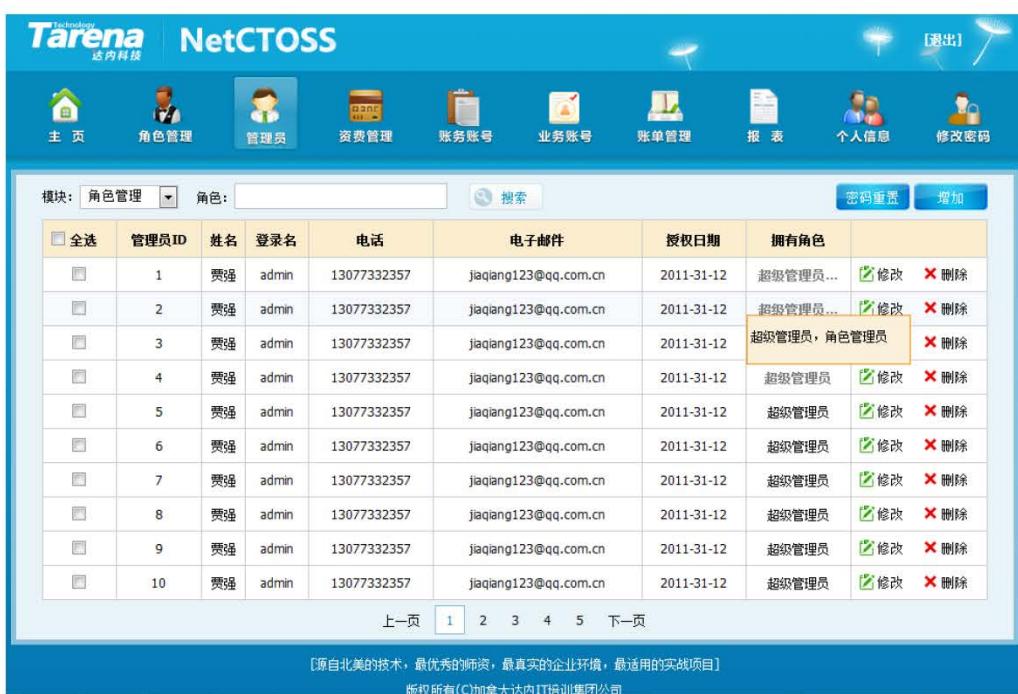


图 - 34

6.2.2. 查询管理员

用例名称	查询管理员
功能简述	具有“管理员管理”权限的管理员登录后，可以查询管理员信息
前置条件	具有“管理员管理”权限的管理员登录成功，且“管理员浏览”用例执行成功
后置条件	“查询管理员”用例执行成功后，可以执行管理员浏览、增加管理员、管理员重置密

	码、修改管理员或者删除管理员的用例
基本流	<ol style="list-style-type: none"> 具有“管理员管理”权限的管理员登录后，单击“管理员”的功能链接，则可以进入如图 - 34 所示的“管理员浏览”界面； 在界面上选择某功能模块，或者录入角色名称，单击“搜索”按钮； 将查询条件数据提交到服务器端，查询管理员数据； 如果查询成功，则将查询出的数据显示在图 - 34 所示的表格中； 如果没有查询到相应的数据，则只显示表格的表头。
分支与异常	第 1 步 :没有登录的用户访问此页面，则直接定位到登录页面 ;已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	无

6.2.3. 管理员重置密码

用例名称	管理员重置密码
功能简述	具有“管理员管理”权限的管理员登录后，可以为管理员重置密码
前置条件	具有“管理员管理”权限的管理员登录成功，且“管理员浏览”或“查询管理员”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“管理员管理”权限的管理员登录后，执行“管理员浏览”或“查询管理员”用例得到管理员的列表数据； 单击“全选”复选框，则选中表格中所有的数据，再单击“全选”复选框，则取消选中表格中所有的数据； 使用复选框单条选择表格中的数据，或者使用“全选”按钮选中当前表格中的所有数据； 单击界面上的“密码重置”按钮，如果界面上没有选择表格中的任何一行数据，则提示用户必须选择数据，如图 - 35 所示； 如果已经选择数据，并单击界面上的“密码重置”按钮，则发送数据到服务器端进行密码重置操作，服务器端将所选中的管理员的密码重置为与登录名相同的初始密码； 如果密码重置操作成功，则在界面提示操作成功，界面如图 - 36 所示； 单击图 - 36 所示界面上的“X”图标关闭提示信息框之后，将刷新“管理员浏览”界面上的数据显示； 如果密码重置操作失败，则在界面提示操作失败的信息，界面如图 - 37 所示； 单击图 - 37 所示界面上的“X”图标关闭提示信息框之后，将回到“管理员浏览”界面。
分支与异常	第 1 步 :没有登录的用户访问此页面，则直接定位到登录页面 ;已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	无



图 - 35

全选	管理员ID	姓名	登录名	电话	电子邮件	授权日期	拥有角色	
<input type="checkbox"/>	1	贾强	admin				超级管理员...	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input checked="" type="checkbox"/>	2	贾强	admin				超级管理员...	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input checked="" type="checkbox"/>	3	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员...	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input checked="" type="checkbox"/>	4	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input type="checkbox"/>	5	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员	<input checked="" type="button"/> 修改 <input type="button"/> 删除

图 - 36

全选	管理员ID	姓名	登录名	电话	电子邮件	授权日期	拥有角色	
<input type="checkbox"/>	1	贾强	admin				超级管理员...	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input checked="" type="checkbox"/>	2	贾强	admin				超级管理员...	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input checked="" type="checkbox"/>	3	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员...	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input checked="" type="checkbox"/>	4	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员	<input checked="" type="button"/> 修改 <input type="button"/> 删除
<input type="checkbox"/>	5	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员	<input checked="" type="button"/> 修改 <input type="button"/> 删除

图 - 37

6.2.4. 删除管理员

用例名称	删除管理员
功能简述	具有“管理员管理”权限的管理员登录后，可以删除已有的管理员
前置条件	具有“管理员管理”权限的管理员登录成功，且“管理员浏览”用例或“查询管理员”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“管理员管理”权限的管理员登录后，执行“管理员浏览”或“查询管理员”用例得到管理员的列表数据； 单击界面上某行数据上的“删除”按钮，则显示提示信息，询问是否确定删除该管理员数据，界面如图 - 38 所示； 如果单击“取消”按钮，则取消删除操作，回到“管理员浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行删除管理员的操作，如果操作成功，则在界面提示删除成功，界面如图 - 39 所示； 单击图 - 39 所示界面上的“X”图标关闭提示信息框之后，将刷新“管理员浏览”界面上的数据显示； 如果删除管理员的操作失败，则在界面提示删除失败的信息，界面如图 - 40 所示； 单击图 - 40 所示界面上的“X”图标关闭提示信息框之后，将回到“管理员浏

	览“界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图-6所示的无权访问的错误提示页。
备注	无

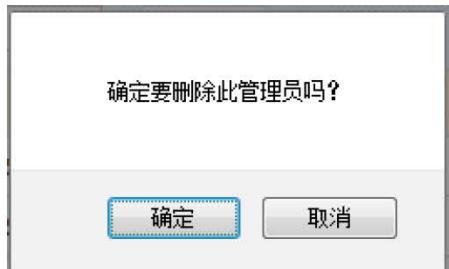


图 - 38



全选	管理员ID	姓名	登录名	电话	电子邮件	授权日期	拥有角色	操作
<input type="checkbox"/>	1	贾强	admin				超级管理员...	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
<input checked="" type="checkbox"/>	2	贾强	admin				超级管理员...	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
<input checked="" type="checkbox"/>	3	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员...	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除

图 - 39



全选	管理员ID	姓名	登录名	电话	电子邮件	授权日期	拥有角色	操作
<input type="checkbox"/>	1	贾强	admin				超级管理员...	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
<input checked="" type="checkbox"/>	2	贾强	admin				超级管理员...	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
<input checked="" type="checkbox"/>	3	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员...	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
<input checked="" type="checkbox"/>	4	贾强	admin	13077332357	jiaqiang123@qq.com.cn	2011-31-12	超级管理员	<input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除

图 - 40

6.2.5. 增加管理员

用例名称	增加管理员
功能简述	具有“管理员管理”权限的管理员登录后，可以增加新的管理员
前置条件	具有“管理员管理”权限的管理员登录成功，且“管理员浏览”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“管理员管理”权限的管理员登录后，单击“管理员”的功能链接，则可以进入“管理员浏览”界面，然后单击界面上的“增加”按钮，则进入“增加管理员”界面，界面如图-41所示； 在界面上录入各项数据，其中，红色*号表示必须录入的数据项； 单击“保存”按钮； 对录入的数据进行输入合法性验证，验证规则如表-5所示； 如果录入的数据验证不通过，则在界面上显示录入错误的提示，如图-42所示； 如果录入的数据验证通过，则提交数据到服务器端，进行增加管理员的操作，如果操作成功，则在界面上显示操作成功的提示，如图-43所示，该提示信息3秒后自动消失；

	<p>7. 数据提交到服务器端后，进行增加管理员的操作，如果操作失败，则在页面提示错误信息，如图 - 44 所示，该提示信息 3 秒后自动消失；</p> <p>8. 如果单击页面的“取消”按钮，则回到“管理员浏览”界面。</p>
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	1. 管理员账号唯一，不能重复；且至少拥有一个角色

图 - 41

表 - 5 “增加管理员”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
姓名	文本	非空，20 长度以内的汉字、字母的组合
管理员账号	文本	非空，且为 30 长度以内的字母、数字和下划线的组合 管理员账号唯一，不能重复
密码	文本	非空，且为 30 长度以内的字母、数字和下划线的组合
重复密码	文本	非空，为 30 长度以内的字母、数字和下划线的组合，且必须和第一次录入的密码相同
电话	文本	电话号码格式，符合手机号码或固定电话号码格式
Email	文本	50 长度以内，符合 email 格式
角色	文本	选择录入，至少选择一个权限



The screenshot shows a registration form with the following validation errors:

- 姓名:** 张三 * 20长度以内的汉字、字母、数字的组合
- 管理员账号:** admin * 30长度以内的字母、数字和下划线的组合
- 密码:** * 30长度以内的字母、数字和下划线的组合
- 重复密码:** * 两次密码必须相同
- 电话:** * 正确的电话号码格式: 手机或固话
- Email:** * 50长度以内, 正确的 email 格式
- 角色:** 至少选择一个
 - 超级管理员
 - 帐务账号管理员
 - 业务账号管理员
 - 帐务账号管理员
 - 业务账号管理员
 - 帐务账号管理员
 - 业务账号管理员

Buttons at the bottom: 保存 (Save) and 取消 (Cancel).

图 - 42



保存成功!

姓名: 张三 * 20长度以内的汉字、字母、数字的组合

管理员账号: admin_zhang * 30长度以内的字母、数字和下划线的组合

图 - 43



保存失败, 管理员账号重复!

姓名: 张三 * 20长度以内的汉字、字母、数字的组合

管理员账号: admin * 30长度以内的字母、数字和下划线的组合

图 - 44

6.2.6. 修改管理员

用例名称	修改管理员
功能简述	具有“管理员管理”权限的管理员登录后，可以修改已有管理员的数据
前置条件	具有“管理员管理”权限的管理员登录成功，且“管理员浏览”用例或者“查询管理员”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“管理员管理”权限的管理员登录后，执行“管理员浏览”或“查询管理员”用例得到管理员的列表数据； 单击界面上某行数据上的“修改”按钮，则进入“修改管理员”界面，界面如图 - 45 所示； 界面上将显示所修改管理员的数据； 在界面上修改管理员的数据，其中，管理员账号信息不允许修改，红色 * 号表示必须录入的数据项；

	<p>5. 单击“保存”按钮；</p> <p>6. 对录入的数据进行输入合法性验证，验证规则如表 - 5 所示；</p> <p>7. 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 46 所示；</p> <p>8. 如果录入的数据验证通过，则提交数据到服务器端，进行修改管理员的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 43 所示，该提示信息 3 秒后自动消失；</p> <p>9. 数据提交到服务器端后，进行修改管理员的操作，如果操作失败，则在页面提示错误信息，如图 - 44 所示，该提示信息 3 秒后自动消失；</p> <p>10. 如果单击页面的“取消”按钮，则回到“管理员浏览”界面。</p>
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页； 第2步：如果没有查询到相应的管理员数据，则直接返回到图 - 34 所示的“管理员浏览”界面。
备注	1. 管理员账号唯一，不能重复；且至少拥有一个角色。

The screenshot shows a registration form with the following fields and validation messages:

- 姓名:** 张三 * 20长度以内的字母、数字和下划线
- 管理员账号:** admin1
- 电话:** 13111111111 * 30长度以内，电话号码格式：手机或固话
- Email:** aa@aa.com * 50长度以内，错误的 email 格式
- 角色:** (checkboxes) 超级管理员, 帐务账号管理员, 业务账号管理员, 帐务账号管理员, 业务账号管理员, 帐务账号管理员, 业务账号管理员 * 至少选择一个
- Buttons:** 保存, 取消

图 - 45

The screenshot shows a registration form with validation errors displayed in red boxes:

- 姓名:** 张三 * 20长度以内的汉字、字母、数字的组合
- 管理员账号:** admin1
- 电话:** 13111111111 * 正确的电话号码格式：手机或固话
- Email:** aa@aa.com * 50长度以内，正确的 email 格式
- 角色:** (checkboxes) 超级管理员, 帐务账号管理员, 业务账号管理员, 帐务账号管理员, 业务账号管理员, 帐务账号管理员, 业务账号管理员 * 至少选择一个
- Buttons:** 保存, 取消

图 - 46

7. “资费管理”模块

7.1. 用例图

NetCTOSS 系统中“资费管理”模块的用例图如图 - 47 所示：

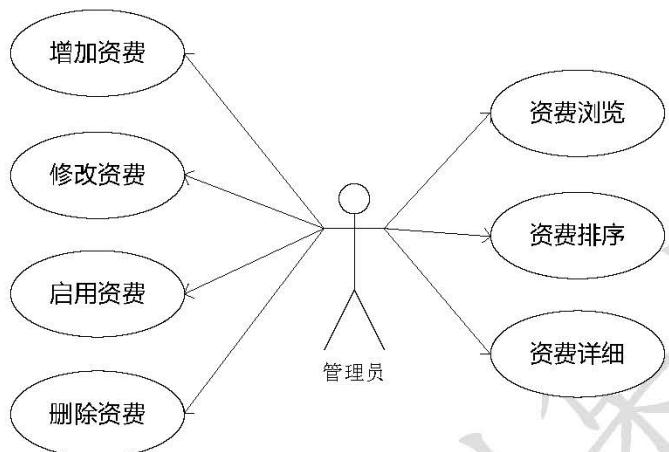


图 - 47

7.2. 用例描述

7.2.1. 资费浏览

用例名称	资费浏览
功能简述	具有“资费管理”权限的管理员登录后，可以查看所有资费的信息
前置条件	具有“资费管理”权限的管理员登录成功
后置条件	可以执行资费排序、查看资费详细、启用资费、增加资费、删除资费或者修改资费的用例
基本流	<ol style="list-style-type: none"> 具有“资费管理”权限的管理员登录后，单击“资费管理”的功能链接，则可以进入“资费管理”界面，以查看所有的资费信息，界面如图 - 48 所示； 界面上将分页列出所有的资费信息，每页显示 10 条数据； 如果数据小于等于 10 条，则不显示分页的页码； 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并用超级链接的方式显示其他页码； 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示； 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化； 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化； 表格中的数据列中，“资费名称”列为超级链接列，单击则链接到“资费详细”页面，以执行“资费详细”用例； 单击页面上的“月租”、“基费”或者“时长”按钮，则执行“资费排序”用例； 单击“增加”按钮，则执行“增加资费”用例； 资费状态为“暂停”的数据行，将显示“启用”、“修改”和“删除”按钮；

	<p>12. 单击表格中资费状态为“暂停”的数据行中的“启用”按钮，则执行“启用资费”用例；</p> <p>13. 单击表格中资费状态为“暂停”的数据行中的“修改”按钮，则执行“修改资费”用例；</p> <p>14. 单击表格中资费状态为“暂停”的数据行中的“删除”按钮，则执行“删除资费”用例。</p>
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“资费管理”权限的管理员访问此页面，则定位到图-6所示的无权访问的错误提示页。
备注	<p>1. 资费有两种状态：暂停和开通；</p> <p>2. 新创建的资费，状态为暂停，记载创建时间；</p> <p>3. 暂停状态的资费，可修改，可删除；</p> <p>4. 开通状态的资费，不能修改、不能再停用、也不能删除，可以通过增加资费创建新的资费。</p>



图 - 48

7.2.2. 资费排序

用例名称	资费排序
功能简述	具有“资费管理”权限的管理员登录后，可以对资费数据进行排序
前置条件	具有“资费管理”权限的管理员登录成功，且“资费浏览”用例执行成功
后置条件	此用例执行成功后，可执行查看资费详细、增加资费、启用资费或者修改资费的用例
基本流	<ol style="list-style-type: none"> 具有“资费管理”权限的管理员登录后，单击“资费管理”的功能链接，则可以进入如图-48所示的“资费浏览”界面； 单击“月租”按钮，则对资费数据根据月租进行升序排列，并将排序后的数据显示在图-48中的表格里，同时，“月租”按钮上的箭头显示变为降序排列标识，如图-49所示； 单击图-49中所示的“月租”按钮，则对资费数据根据月租进行降序排列，并显示数据，同时，“月租”按钮上的箭头显示变回图-48中所示的升序排列标识； 单击“基费”或者“时长”按钮，则分别对资费数据根据基费或者时长进行升

	序/降序排列，并显示数据，同时修改排序按钮的升序/降序标识。
分支与异常	无
备注	无



图 - 49

7.2.3. 启用资费

用例名称	启用资费
功能简述	具有“资费管理”权限的管理员登录后，可以启用暂停状态的资费
前置条件	具有“资费管理”权限的管理员登录成功，且“资费浏览”用例或“资费排序”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“资费管理”权限的管理员登录后，执行“资费浏览”或“资费排序”用例得到资费的列表数据； 单击界面上某行数据上的“启用”按钮，则显示提示信息，询问是否确定启用该资费，界面如图 - 50 所示； 如果单击“取消”按钮，则取消启用操作，回到“资费浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行启用资费的操作，如果操作成功，则在界面提示启用成功，界面如图 - 51 所示； 单击图 - 51 所示界面上的“X”图标关闭提示信息框之后，将刷新“资费浏览”界面上的数据显示； 如果启用资费的操作失败，则在界面提示启用失败的信息，界面如图 - 52 所示； 单击图 - 52 所示界面上的“X”图标关闭提示信息框之后，将回到“资费浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“资费管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 只能启用状态为“暂停”的资费； 资费启用后，不能修改，不能停用，也不能删除。



图 - 50

基本费用	单位费用	创建时间	开通时间	状态
24.50 元				
40.50 元		✓ 资费成功启用!		X

图 - 51

基本费用	单位费用	创建时间	开通时间	状态
24.50 元				
40.50 元		⚠ 资费启用失败! 数据并发错误。		X

图 - 52

7.2.4. 删 除 资 费

用例名称	删除资费
功能简述	具有“资费管理”权限的管理员登录后，可以删除暂停状态的资费
前置条件	具有“资费管理”权限的管理员登录成功，且“资费浏览”用例或“资费排序”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“资费管理”权限的管理员登录后，执行“资费浏览”或“资费排序”用例得到资费的列表数据； 单击界面上某行数据上的“删除”按钮，则显示提示信息，询问是否确定删除该资费，界面如图 - 53 所示； 如果单击“取消”按钮，则取消删除操作，回到“资费浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行删除资费的操作，如果操作成功，则在界面提示删除成功，界面如图 - 54 所示； 单击图 - 54 所示界面上的“X”图标关闭提示信息框之后，将刷新“资费浏览”界面上的数据显示； 如果删除资费的操作失败，则在界面提示删除失败的信息，界面如图 - 55 所示； 单击图 - 55 所示界面上的“X”图标关闭提示信息框之后，将回到“资费浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	1. 只能删除状态为“暂停”的资费。

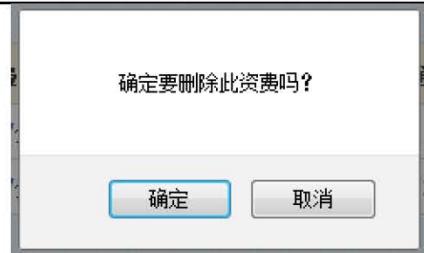


图 - 53



图 - 54

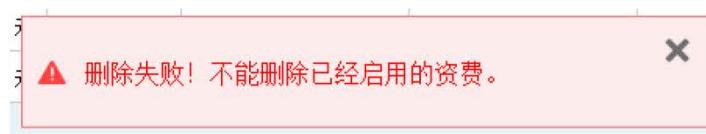


图 - 55

7.2.5. 资费详细

用例名称	资费详细
功能简述	具有“资费管理”权限的管理员登录后，可以查看某资费的详细信息
前置条件	具有“资费管理”权限的管理员登录成功，且“资费浏览”用例或“资费排序”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“资费管理”权限的管理员登录后，执行“资费浏览”或“资费排序”用例得到资费的列表数据； 单击界面上某行数据上的资费名称链接，则进入“资费详细”页面查看该资费的详细信息，界面如图 - 56 所示； 界面上将显示某资费的所有详细信息，且所有数据项不能修改； 单击“返回”按钮将返回到“资费浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“管理员管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页； 第2步：如果没有查询到相应的资费数据，则直接返回到“资费浏览”界面。
备注	无

资费ID:	1
资费名称:	包 20 小时
资费状态:	开通
资费类型:	<input type="radio"/> 包月 <input type="radio"/> 套餐 <input checked="" type="radio"/> 计时
基本时长:	20 小时
基本费用:	24.5 元
单位费用:	3.00 元/小时
创建时间:	2013/1/1 00:00:00
启动时间:	2013/1/1 00:00:00
资费说明:	包 20 小时，20 小时以内 24.5 元，超出部分 3.00 元/ 小时

[返回](#)

图 - 56

7.2.6. 增加资费

用例名称	增加资费
功能简述	具有“资费管理”权限的管理员登录后，可以增加新资费
前置条件	具有“资费管理”权限的管理员登录成功，且“资费浏览”用例或“资费排序”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“资费管理”权限的管理员登录后，单击“资费管理”的功能链接，则可以进入“资费管理”界面，然后单击界面上的“增加”按钮，则进入“增加资费”界面，界面如图 - 57 所示； 如果选择“包月”资费类型，则只需要录入“基本费用”；如果选择“计时”资费类型，则只需要录入“单位费用”；如果选择“套餐”资费类型，则需要录入“基本时长”、“基本费用”和“单位费用”。界面如图 - 58 所示； 在界面上录入各项数据，其中，红色 * 号表示必须录入的数据项； 单击“保存”按钮； 对录入的数据进行输入合法性验证，验证规则如表 - 6 所示； 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 59 所示； 如果录入的数据验证通过，则提交数据到服务器端，进行增加资费的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 60 所示，该提示信息 3

	秒后自动消失； 8. 数据提交到服务器端后，进行增加资费的操作，如果操作失败，则在页面提示错误信息，如图 - 61 所示，该提示信息 3 秒后自动消失； 9. 如果单击页面的“取消”按钮，则回到“资费浏览”界面。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“资费管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	1. 资费名称唯一，不能重复； 2. 新创建的资费，状态为暂停。



The screenshot shows a modal dialog for adding a service fee. It contains fields for 'Service Fee Name' (必填, 50-length alphanumeric and underline), 'Service Type' (radio buttons for 包月, 套餐, 计时, with 套餐 selected), 'Basic Duration' (必填, integer between 1-600 hours), 'Basic Cost' (必填, float between 0-99999.99 yuan), 'Unit Cost' (必填, float between 0-99999.99 yuan/hour), and 'Description' (必填, 100-length alphanumeric and underline). Below the fields are 'Save' and 'Cancel' buttons.

图 - 57



The screenshot shows a modal dialog for adding a service fee, identical to Figure 57 but with two instances side-by-side. Both instances show validation errors for the 'Service Fee Name' field (必填, 50-length alphanumeric and underline).

图 - 58

表 - 6 “增加资费”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
资费名称	文本	非空，50 长度的字母、数字、汉字和下划线的组合 资费名称唯一，不能重复
资费类型	文本	非空，选择“包月”、“套餐”或者“计时”
基本时长	数值	“计时”类型时必须录入，且为 1-600 之间的整数

基本费用	数值	非空，0-99999.99 之间的数值
单位费用	数值	"计时"类型时必须录入，且为 0-99999.99 之间的数值
资费说明	文本	100 长度的字母、数字和下划线的组合

资费名称: * **50长度的字母、数字、汉字和下划线的组合**

资费类型: 包月 套餐 计时

基本时长: 小时 * **1-600之间的整数**

基本费用: 元 * **0-99999.99之间的数值**

单位费用: 元/小时 * **0-99999.99之间的数值**

资费说明: * **100长度的字母、数字、汉字和下划线的组合**

保存 **取消**

图 - 59

保存成功！

图 - 60

保存失败，资费名称重复！

图 - 61

7.2.7. 修改资费

用例名称	修改资费
功能简述	具有“资费管理”权限的管理员登录后，可以修改已有资费的数据
前置条件	具有“资费管理”权限的管理员登录成功，且“资费浏览”用例或者“资费排序”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“资费管理”权限的管理员登录后，执行“资费浏览”或“资费排序”用例得到资费的列表数据； 单击界面上某行数据上的“修改”按钮，则进入“修改资费”界面，界面如图 - 62 所示； 界面上将显示所修改资费的数据，其中，资费 ID 项不能修改，且如果选择“包月”资费类型，则只能修改“基本费用”；如果选择“计时”资费类型，则只能修改“单位费用”；如果选择“套餐”资费类型，则可以修改“基本时长”、“

	基本费用 “和” 单位费用 “； 4. 在界面上修改资费的数据，其中，红色 * 号表示必须录入的数据项； 5. 单击“保存”按钮； 6. 对录入的数据进行输入合法性验证，验证规则如表 - 6 所示； 7. 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 63 所示； 8. 如果录入的数据验证通过，则提交数据到服务器端，进行修改资费的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 60 所示，该提示信息 3 秒后自动消失； 9. 数据提交到服务器端后，进行修改资费的操作，如果操作失败，则在页面提示错误信息，如图 - 61 所示，该提示信息 3 秒后自动消失； 10. 如果单击页面的“取消”按钮，则回到“资费浏览”界面。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“资费管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页； 第 2 步：如果没有查询到相应的资费数据，则直接返回到“资费浏览”界面。
备注	1. 资费名称唯一，不能重复； 2. 只能修改暂停状态的资费。



资费ID: 1

资费名称: 包 20 小时 * 50长度的字母、数字、汉字和下划线的组合

资费类型: 包月 套餐 计时

基本时长: 小时 * 1-600之间的整数

基本费用: 元 * 0-99999.99之间的数值

单位费用: 元/小时 * 0-99999.99之间的数值

资费说明: 没有启用的资费, 可以修改除 ID 以外的所有信息 100长度的字母、数字、汉字和下划线的组合

保存 取消

图 - 62

资费ID:	1
资费名称:	包 20 小时 * * 50长度的字母、数字、汉字和下划线的组合
资费类型:	<input type="radio"/> 包月 <input checked="" type="radio"/> 套餐 <input type="radio"/> 计时
基本时长:	<input type="text"/> 小时 * * 1-600之间的整数
基本费用:	<input type="text"/> 元 * * 0-99999.99之间的数值
单位费用:	<input type="text"/> 元/小时 * * 0-99999.99之间的数值
资费说明:	没有启用的资费，可以修改除 ID 以外的所有信息 * * 100长度的字母、数字、汉字和下划线的组合
<input type="button" value="保存"/> <input type="button" value="取消"/>	

图 - 63

8. “帐务账号管理”模块

8.1. 用例图

NetCTOSS 系统中“帐务账号管理”模块的用例图如图 - 64 所示：

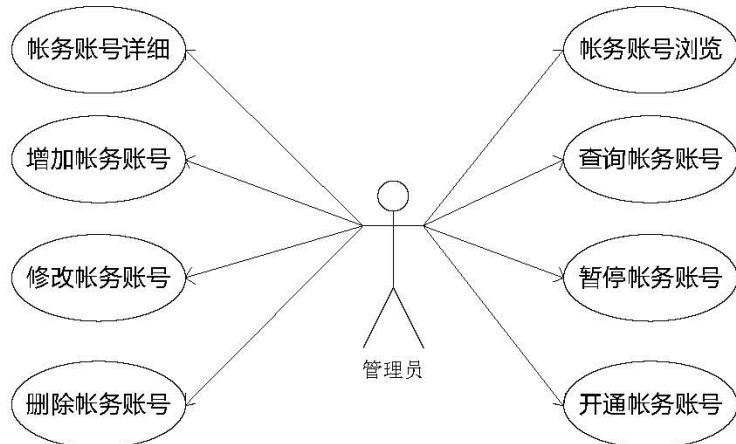


图 - 64

8.2. 用例描述

8.2.1. 帐务账号浏览

用例名称	帐务账号浏览
功能简述	具有“帐务账号管理”权限的管理员登录后，可以查看所有帐务账号的信息
前置条件	具有“帐务账号管理”权限的管理员登录成功

后置条件	可以执行查询帐务账号、查看帐务账号详细、暂停帐务账号、开通帐务账号、增加帐务账号、删除帐务账号或者修改帐务账号的用例
基本流	<ol style="list-style-type: none"> 1. 具有“帐务账号管理”权限的管理员登录后，单击“帐务账号”的功能链接，则可以进入“帐务账号浏览”界面，以查看所有的帐务账号信息，界面如图 - 65 所示； 2. 界面上将分页列出所有的帐务账号信息，每页显示 10 条数据； 3. 如果数据小于等于 10 条，则不显示分页的页码； 4. 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并显示与当前页临近的 5 个页码的链接； 5. 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示，并刷新临近 5 个页码的显示，如图 - 66 所示； 6. 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化； 7. 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化； 8. 单击“首页”链接，将显示第一页的数据和页码，如果当前页已经是第一页，则页面数据无变化； 9. 单击“末页”链接，将显示最后一页的数据和页码，如果当前页已经是最后一页，则页面数据无变化； 10. 表格中的数据列中，“姓名”列为超级链接列，单击则链接到“帐务账号详细”页面，以执行“帐务账号详细”用例； 11. 单击“搜索”按钮，则执行“查询帐务账号”用例； 12. 单击“增加”按钮，则执行“增加帐务账号”用例； 13. 状态为“开通”的帐务账号，可以暂停；单击“暂停”按钮，则执行“暂停帐务账号”用例； 14. 状态为“暂停”的帐务账号，可以开通；单击“开通”按钮，则执行“开通帐务账号”用例； 15. 单击表格中的“删除”按钮，则执行“删除帐务账号”用例； 16. 单击表格中的“修改”按钮，则执行“修改帐务账号”用例。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 1. 帐务账号有三种状态：开通、暂停和删除； 2. 新创建的帐务账号，状态为开通，并记载创建时间； 3. 状态为“删除”的帐务账号，只能查看详细信息，不能执行暂停、开通以及修改等操作。

账号ID	姓名	身份证	登录名	状态	创建日期	上次登录时间	
1	贾强	230102197902137862	jiaqiang	开通	2013-01-23	2013-02-23 00:00:00	<input type="checkbox"/> 暂停 <input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
2	贾强	230102197902137862	jiaqiang	暂停	2013-01-23	2013-02-23 00:00:00	<input checked="" type="checkbox"/> 开通 <input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
3	贾强	230102197902137862	jiaqiang	删除	2013-01-23	2013-02-23 00:00:00	
4	贾强	230102197902137862	jiaqiang	开通	2013-01-23	2013-02-23 00:00:00	<input type="checkbox"/> 暂停 <input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
5	贾强	230102197902137862	jiaqiang	暂停	2013-01-23	2013-02-23 00:00:00	<input checked="" type="checkbox"/> 开通 <input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
6	贾强	230102197902137862	jiaqiang	删除	2013-01-23	2013-02-23 00:00:00	
7	贾强	230102197902137862	jiaqiang	开通	2013-01-23	2013-02-23 00:00:00	<input type="checkbox"/> 暂停 <input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
8	贾强	230102197902137862	jiaqiang	暂停	2013-01-23	2013-02-23 00:00:00	<input checked="" type="checkbox"/> 开通 <input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除
9	贾强	230102197902137862	jiaqiang	删除	2013-01-23	2013-02-23 00:00:00	
10	贾强	230102197902137862	jiaqiang	暂停	2013-01-23	2013-02-23 00:00:00	<input checked="" type="checkbox"/> 开通 <input checked="" type="checkbox"/> 修改 <input type="checkbox"/> 删除

图 - 65



图 - 66

8.2.2. 查询帐务账号

用例名称	查询帐务账号
功能简述	具有“帐务账号管理”权限的管理员登录后，可以查询帐务账号数据
前置条件	具有“帐务账号管理”权限的管理员登录成功，且“帐务账号浏览”用例执行成功
后置条件	此用例执行成功后，可以执行帐务账号浏览、帐务账号详细、暂停帐务账号、开通帐务账号、增加帐务账号、删除帐务账号或者修改帐务账号的用例
基本流	<ol style="list-style-type: none"> 具有“帐务账号管理”权限的管理员登录后，单击“帐务账号”的功能链接，则可以进入如图 - 65 所示的“帐务账号浏览”界面； 在界面上录入查询条件，如身份证号、姓名和登录名，并选择某种状态，如“全部”、“开通”、“暂停”或者“删除”，然后单击“搜索”按钮； 将查询条件数据提交到服务器端，查询帐务账号数据； 如果查询成功，则将查询出的数据显示在图 - 65 所示的表格中； 如果没有查询到相应的数据，则只显示表格的表头。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	无

8.2.3. 暂停帐务账号

用例名称	暂停帐务账号
功能简述	具有“帐务账号管理”权限的管理员登录后，可以暂停“开通”状态下的帐务账号
前置条件	具有“帐务账号管理”权限的管理员登录成功，且“帐务账号浏览”用例或“查询帐务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“帐务账号管理”权限的管理员登录后，执行“帐务账号浏览”或“查询帐务账号”用例得到帐务账号的列表数据； 如果某行帐务账号数据的状态为“开通”，则可以单击该行数据上的“暂停”按钮，并显示提示信息，询问是否确定暂停该帐务账号，界面如图 - 67 所示； 如果单击“取消”按钮，则取消启用操作，回到“帐务账号浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行暂停帐务账号的操作，如果操作成功，则在界面提示操作成功，界面如图 - 68 所示； 单击图 - 68 所示界面上的“X”图标关闭提示信息框之后，将刷新“帐务账号浏览”界面上的数据显示； 如果暂停帐务账号的操作失败，则在界面提示暂停失败的信息，界面如图 - 69 所示； 单击图 - 69 所示界面上的“X”图标关闭提示信息框之后，将回到“帐务账号浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 只能暂停状态为“开通”的帐务账号； 暂停某帐务账号，则同时暂停其下属的所有业务账号； 执行暂停操作时，需要记载该账号的暂停时间。

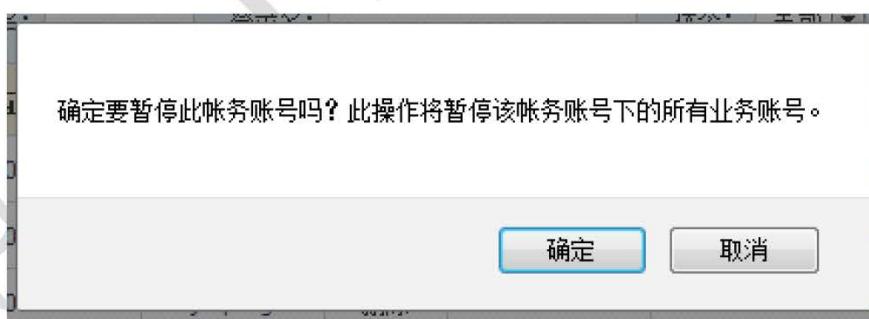


图 - 67

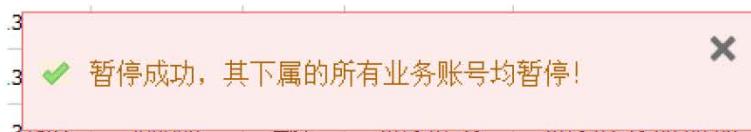


图 - 68



图 - 69

8.2.4. 开通帐务账号

用例名称	开通帐务账号
功能简述	具有“帐务账号管理”权限的管理员登录后，可以开通“暂停”状态下的帐务账号
前置条件	具有“帐务账号管理”权限的管理员登录成功，且“帐务账号浏览”用例或“查询帐务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“帐务账号管理”权限的管理员登录后，执行“帐务账号浏览”或“查询帐务账号”用例得到帐务账号的列表数据； 如果某行帐务账号数据的状态为“暂停”，则可以单击该行数据上的“开通”按钮，并显示提示信息，询问是否确定开通该帐务账号，界面如图 - 70 所示； 如果单击“取消”按钮，则取消开通操作，回到“帐务账号浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行开通帐务账号的操作，如果操作成功，则在界面提示操作成功，界面如图 - 71 所示； 单击图 - 71 所示界面上的“X”图标关闭提示信息框之后，将刷新“帐务账号浏览”界面上的数据显示； 如果开通帐务账号的操作失败，则在界面提示暂停失败的信息，界面如图 - 72 所示； 单击图 - 72 所示界面上的“X”图标关闭提示信息框之后，将回到“帐务账号浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 只能开通状态为“暂停”的帐务账号； 开通某帐务账号，不会同时开通其下属的所有业务账号，需要在“业务账号管理”模块中单独操作； 执行开通操作后，记载开通时间，且删除该账号的暂停时间。

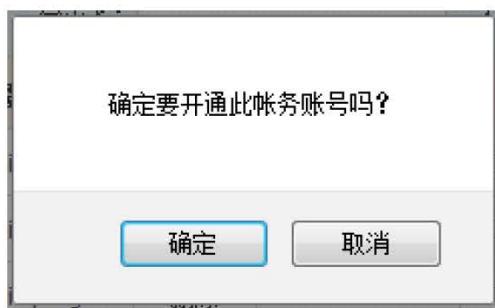


图 - 70

✓ 开通成功！其下属的业务账号仍为暂停状态。

图 - 71

⚠️ 开通失败！数据并发错误。

图 - 72

8.2.5. 删除帐务账号

用例名称	删除帐务账号
功能简述	具有“帐务账号管理”权限的管理员登录后，可以删除帐务账号
前置条件	具有“帐务账号管理”权限的管理员登录成功，且“帐务账号浏览”用例或“查询帐务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“帐务账号管理”权限的管理员登录后，执行“帐务账号浏览”或“查询帐务账号”用例得到帐务账号的列表数据； 单击界面上某行数据上的“删除”按钮，则显示提示信息，询问是否确定删除该帐务账号，界面如图 - 73 所示； 如果单击“取消”按钮，则取消删除操作，回到“帐务账号浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行删除帐务账号的操作，如果操作成功，则在界面提示删除成功，界面如图 - 74 所示； 单击图 - 74 所示界面上的“X”图标关闭提示信息框之后，将刷新“帐务账号浏览”界面上的数据显示； 如果删除帐务账号的操作失败，则在界面提示删除失败的信息，界面如图 - 75 所示； 单击图 - 75 所示界面上的“X”图标关闭提示信息框之后，将回到“帐务账号浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 帐务账号被删除后，数据依然保留，将状态记载为“删除”，并记载删除时间； 删除状态的帐务账号，不能再开通、暂停或者修改； 删除某帐务账号，同时删除其下属的所有业务账号。

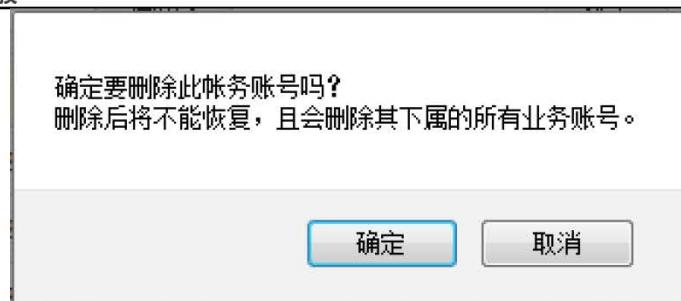


图 - 73



图 - 74

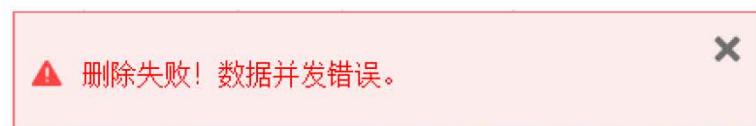


图 - 75

8.2.6. 帐务账号详细

用例名称	帐务账号详细
功能简述	具有“帐务账号管理”权限的管理员登录后，可以查看某帐务账号的详细信息
前置条件	具有“帐务账号管理”权限的管理员登录成功，且“帐务账号浏览”用例或“查询帐务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“帐务账号管理”权限的管理员登录后，执行“帐务账号浏览”或“查询帐务账号”用例得到帐务账号的列表数据； 单击界面上某行数据上的帐务账号的姓名链接，则进入“帐务账号详细”页面以查看该帐务账号的详细信息，界面如图 - 76 所示； 界面上将显示该帐务账号的所有详细信息，且所有数据项不能修改； 单击“返回”按钮将返回到“帐务账号浏览”界面。
分支与异常	<p>第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页；</p> <p>第2步：如果没有查询到相应的帐务账号数据，则直接返回到“帐务账号浏览”界面。</p>
备注	<ol style="list-style-type: none"> 根据帐务账号的状态，在页面上显示开通时间、暂停时间或删除时间。

帐务账号ID:	10
姓名:	张三
身份证:	230198765432123456
登录账号:	user1
电话:	13687865435
推荐人帐务账号ID:	5
推荐人身份证号码:	230198765432123456
状态:	开通 ▾
开通时间:	2013/01/21 12:00:00
上次登录时间:	2013/02/21 12:00:00
上次登录IP:	192.168.0.100
生日:	1980/01/21
Email:	aa@aa.com
职业:	干部 ▾
性别:	<input checked="" type="radio"/> 女 <input type="radio"/> 男
通信地址:	北京市海淀区北三环中路甲18号中鼎大厦
邮编:	100098
QQ:	12345678
返回	

图 - 76

8.2.7. 增加帐务账号

用例名称	增加帐务账号
功能简述	具有“帐务账号管理”权限的管理员登录后，可以增加新的帐务账号
前置条件	具有“帐务账号管理”权限的管理员登录成功，且“帐务账号浏览”用例或“查询帐务账号”用例执行成功
后置条件	无
基本流	1. 具有“帐务账号管理”权限的管理员登录后，单击“帐务账号”的功能链接，

	<p>则可以进入“帐务账号浏览”界面，然后单击界面上的“增加”按钮，则进入“增加帐务账号”界面，界面如图 - 77 所示；</p> <ol style="list-style-type: none"> 2. 图 - 77 所示的界面上，先显示所有必须录入的数据项（用红色 * 号表示）； 3. 单击界面上的可选项图标，则显示可选的录入数据项，界面如图 - 78 所示； 4. 在界面上录入各项数据，单击“保存”按钮； 5. 对录入的数据进行输入合法性验证，验证规则如表 - 7 所示； 6. 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 79 所示； 7. 如果录入的数据验证通过，则提交数据到服务器端，进行增加帐务账号的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 80 所示，该提示信息 3 秒后自动消失； 8. 数据提交到服务器端后，进行增加帐务账号的操作，如果操作失败，则在页面提示错误信息，如图 - 81 所示，该提示信息 3 秒后自动消失； 9. 如果单击页面的“取消”按钮，则回到“帐务账号浏览”界面。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 1. 新创建的帐务账号的状态为开通； 2. 帐务账号和身份证一一对应，即，每个客户，使用其身份证号只能开设一个帐务账号。



姓名: * 20长度以内的汉字、字母和数字的组合

身份证: * 正确的身份证号码格式

登录账号: * 30长度以内的字母、数字和下划线的组合

密码: * 30长度以内的字母、数字和下划线的组合

重复密码: * 两次密码必须相同

电话: * 正确的电话号码格式：手机或固话

可选项:

保存 **取消**

图 - 77

可选项： 

推荐人身份证号码:	<input type="text"/>	正确的身份证号码格式
生日:	由身份证号计算而来	
Email:	<input type="text"/>	50长度以内，合法的 Email 格式
职业:	<input type="text" value="干部"/>	
性别:	<input checked="" type="radio"/> 女 <input type="radio"/> 男	
通信地址:	<input type="text"/>	
邮编:	<input type="text"/>	6位数字
QQ:	<input type="text"/>	
<input type="button" value="保存"/> <input type="button" value="取消"/>		

图 - 78

表 - 7 “增加帐务账号”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
姓名	文本	非空，20 长度的汉字、字母和数字的组合
身份证	文本	非空，且符合身份证号码的格式要求 身份证号码唯一，即同一人只能开通一个帐务账号
登录账号	文本	非空，30 长度的字母、数字和下划线的组合
密码	文本	非空，30 长度的字母、数字和下划线的组合
重复密码	文本	非空，两次密码必须相同
电话	文本	非空，正确的电话号码格式：手机或者固话
推荐人身份证号码	文本	非空，且符合身份证号码的格式要求
生日	文本	不用录入，由身份证号码计算而来
Email	文本	50 长度以内，合法的 Email 格式
职业	文本	选择录入
性别	文本	选择录入
通信地址	文本	50 长度以内
邮编	文本	6 位数字，正确的邮编格式
QQ	文本	5 到 13 位数字的组合

图 - 79

✓ 保存成功！

图 - 80

✗ 保存失败，该身份证已经开通通过帐务账号！

图 - 81

8.2.8. 修改帐务账号

用例名称	修改帐务账号
功能简述	具有“帐务账号管理”权限的管理员登录后，可以修改已有帐务账号的数据
前置条件	具有“帐务账号管理”权限的管理员登录成功，且“帐务账号浏览”用例或者“查询帐务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“帐务账号管理”权限的管理员登录后，执行“帐务账号浏览”或“查询帐务账号”用例得到帐务账号的列表数据； 单击界面上某行数据上的“修改”按钮，则进入“修改帐务账号”界面，界面如图 - 82 所示； 界面上将显示所修改帐务账号的数据，其中，ID、登录账号、身份证号码以及生日不能修改； 单击“修改密码”复选框，则在界面上显示修改密码的选项，如图 - 83 所示； 在界面上修改帐务账号的数据，其中，红色 * 号表示必须录入的数据项； 单击“保存”按钮； 对录入的数据进行输入合法性验证，验证规则如表 - 7 所示； 如果录入的数据验证不通过，则在界面上显示录入错误的提示，如图 - 84 所示； 如果录入的数据验证通过，则提交数据到服务器端，进行修改帐务账号的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 85 所示，该提示信息 3 秒后自动消失；

	<p>10. 数据提交到服务器端后，进行修改帐务账号的操作，如果操作失败，则在页面提示错误信息，如图 - 86 所示，该提示信息 3 秒后自动消失；</p> <p>11. 如果单击页面的“取消”按钮，则回到“帐务账号浏览”界面。</p>
分支与异常	<p>第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“帐务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页；</p> <p>第2步：如果没有查询到相应的帐务账号数据，则直接返回到“帐务账号浏览”界面。</p>
备注	1. 录入正确的旧密码，才能修改密码。



帐务帐号ID:

姓名: * 20长度以内的汉字、字母和数字的组合

身份证号:

登录帐号: 修改密码

电话: * 正确的电话号码格式: 手机或固话

推荐人身份证号码: * 正确的身份证号码格式

生日:

Email: 50长度以内，合法的 Email 格式

职业:

性别: 女 男

通信地址: 50长度以内

邮编: 6位数字

QQ: 5到13位数字

图 - 82

The screenshot shows a password change form with the following fields and validation messages:

- 身份证号: 230198765432123456
- 登录帐号: user1 修改密码
- 旧密码: [redacted] * 30长度以内的字母、数字和下划线的组合
- 新密码: [redacted] * 30长度以内的字母、数字和下划线的组合
- 重输入新密码: [redacted] * 两次密码必须相同
- 电话: [redacted] * 正确的电话号码格式: 手机或固话

图 - 83

The screenshot shows a registration form with the following fields and validation messages:

- 帐务帐号ID: 10
- 姓名: 张三 * ✗ 20长度以内的汉字、字母和数字的组合
- 身份证号: 230198765432123456
- 登录帐号: user1 修改密码
- 电话: [redacted] * ✗ 正确的电话号码格式: 手机或固话
- 推荐人身份证号码: [redacted] * ✗ 正确的身份证号码格式
- 生日: 由身份证号计算而来

图 - 84



图 - 85



图 - 86

9. “业务账号管理”模块

9.1. 用例图

NetCTOSS 系统中“业务账号管理”模块的用例图如图 - 87 所示：

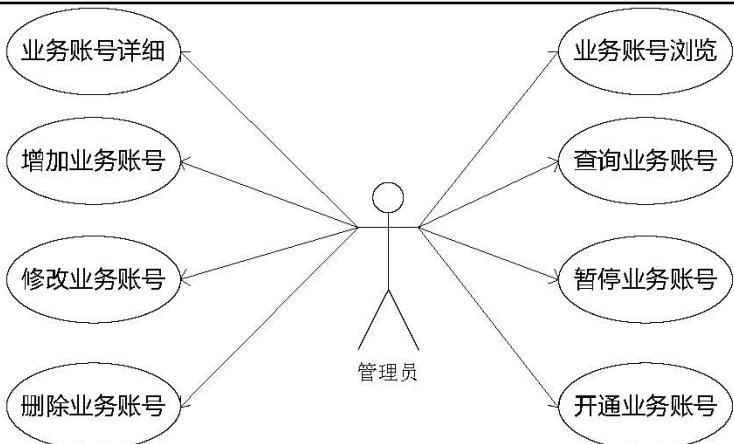


图 - 87

9.2. 用例描述

9.2.1. 业务账号浏览

用例名称	业务账号浏览
功能简述	具有“业务账号管理”权限的管理员登录后，可以查看所有业务账号的信息
前置条件	具有“业务账号管理”权限的管理员登录成功
后置条件	可以执行查询业务账号、查看业务账号详细、暂停业务账号、开通业务账号、增加业务账号、删除业务账号或者修改业务账号的用例
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，单击“业务账号”的功能链接，则可以进入“业务账号浏览”界面，以查看所有的业务账号信息，界面如图 - 88 所示； 界面上将分页列出所有的业务账号信息，每页显示 10 条数据； 如果数据小于等于 10 条，则不显示分页的页码； 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并显示与当前页面临近的 5 个页码的链接； 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示，并刷新临近 5 个页码的显示，如图 - 89 所示； 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化； 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化； 单击“首页”链接，将显示第一页的数据和页码，如果当前页已经是第一页，则页面数据无变化； 单击“末页”链接，将显示最后一页的数据和页码，如果当前页已经是最后一页，则页面数据无变化； 表格中的数据列中，“业务 ID”列为超级链接列，鼠标移入时会显示提示信息“查看明细”，单击则链接到“业务账号详细”页面，以执行“业务账号详细”用例； “资费”列显示当前业务账号所使用的资费名称，当鼠标移入当前行的“资费”列时，使用浮动框显示该资费的详细描述信息；

	<p>12. 单击“搜索”按钮，则执行“查询业务账号”用例；</p> <p>13. 单击“增加”按钮，则执行“增加业务账号”用例；</p> <p>14. 状态为“开通”的业务账号，可以暂停；单击“暂停”按钮，则执行“暂停业务账号”用例；</p> <p>15. 状态为“暂停”的业务账号，可以开通；单击“开通”按钮，则执行“开通业务账号”用例；</p> <p>16. 单击表格中的“删除”按钮，则执行“删除业务账号”用例；</p> <p>17. 单击表格中的“修改”按钮，则执行“修改业务账号”用例。</p>
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图-6所示的无权访问的错误提示页。
备注	<p>1. 业务账号有三种状态：开通、暂停和删除；</p> <p>2. 新创建的业务账号，状态为开通，并记载创建时间；</p> <p>3. 状态为“删除”的业务账号，只能查看详细信息，不能执行暂停、开通以及修改等操作。</p>



图 - 88



图 - 89

9.2.2. 查询业务账号

用例名称	查询业务账号
功能简述	具有“业务账号管理”权限的管理员登录后，可以查询业务账号数据
前置条件	具有“业务账号管理”权限的管理员登录成功，且“业务账号浏览”用例执行成功
后置条件	此用例执行成功后，可以执行业务账号浏览、业务账号详细、暂停业务账号、开通业

	务账号、增加业务账号、删除业务账号或者修改业务账号的用例
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，单击“业务账号”的功能链接，则可以进入如图 - 88 所示的“业务账号浏览”界面； 在界面上录入查询条件，如 OS 账号、服务器 IP 和身份证号码，并选择某种状态，如“全部”、“开通”、“暂停”或者“删除”，然后单击“搜索”按钮； 将查询条件数据提交到服务器端，查询业务账号数据； 如果查询成功，则将查询出的数据显示在图 - 88 所示的表格中； 如果没有查询到相应的数据，则只显示表格的表头。
分支与异常	第 1 步 :没有登录的用户访问此页面，则直接定位到登录页面 ;已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	无

9.2.3. 暂停业务账号

用例名称	暂停业务账号
功能简述	具有“业务账号管理”权限的管理员登录后，可以暂停“开通”状态下的业务账号
前置条件	具有“业务账号管理”权限的管理员登录成功，且“业务账号浏览”用例或“查询业务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，执行“业务账号浏览”或“查询业务账号”用例得到业务账号的列表数据； 如果某业务账号数据的状态为“开通”，则可以单击该行数据上的“暂停”按钮，并显示提示信息，询问是否确定暂停该业务账号，界面如图 - 90 所示； 如果单击“取消”按钮，则取消启用操作，回到“业务账号浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行暂停业务账号的操作，如果操作成功，则在界面提示操作成功，界面如图 - 91 所示； 单击图 - 91 所示界面上的“X”图标关闭提示信息框之后，将刷新“业务账号浏览”界面上的数据显示； 如果暂停业务账号的操作失败，则在界面提示暂停失败的信息，界面如图 - 92 所示； 单击图 - 92 所示界面上的“X”图标关闭提示信息框之后，将回到“业务账号浏览”界面。
分支与异常	第 1 步 :没有登录的用户访问此页面，则直接定位到登录页面 ;已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 只能暂停状态为“开通”的业务账号； 执行暂停操作时，需要记载该账号的暂停时间。

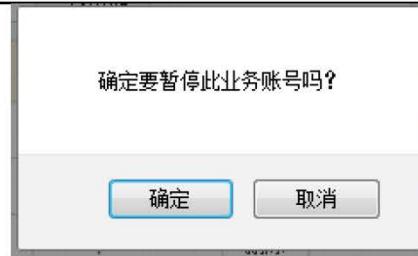


图 - 90



图 - 91

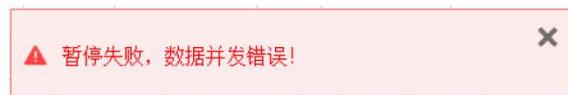


图 - 92

9.2.4. 开通业务账号

用例名称	开通业务账号
功能简述	具有“业务账号管理”权限的管理员登录后，可以开通“暂停”状态下的业务账号
前置条件	具有“业务账号管理”权限的管理员登录成功，且“业务账号浏览”用例或“查询业务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，执行“业务账号浏览”或“查询业务账号”用例得到业务账号的列表数据； 如果某业务账号数据的状态为“暂停”，则可以单击该行数据上的“开通”按钮，并显示提示信息，询问是否确定开通该业务账号，界面如图 - 93 所示； 如果单击“取消”按钮，则取消开通操作，回到“业务账号浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行开通业务账号的操作，如果操作成功，则在界面提示操作成功，界面如图 - 94 所示； 单击图 - 94 所示界面上的“X”图标关闭提示信息框之后，将刷新“业务账号浏览”界面上的数据显示； 如果开通业务账号的操作失败，则在界面提示开通失败的信息，界面如图 - 95 所示； 单击图 - 95 所示界面上的“X”图标关闭提示信息框之后，将回到“业务账号浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 只能开通状态为“暂停”的业务账号； 如果业务账号所属的帐务账号的状态为暂停或者删除，则不能开通此业务账号； 执行开通操作后，记载开通时间，且删除该账号的暂停时间。

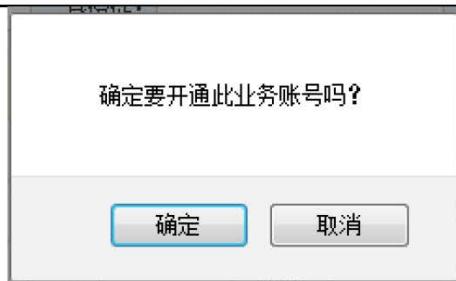


图 - 93



图 - 94

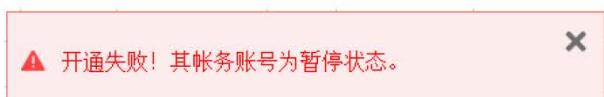


图 - 95

9.2.5. 删除业务账号

用例名称	删除业务账号
功能简述	具有“业务账号管理”权限的管理员登录后，可以删除业务账号
前置条件	具有“业务账号管理”权限的管理员登录成功，且“业务账号浏览”用例或“查询业务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，执行“业务账号浏览”或“查询业务账号”用例得到业务账号的列表数据； 单击界面上某行数据上的“删除”按钮，则显示提示信息，询问是否确定删除该业务账号，界面如图 - 96 所示； 如果单击“取消”按钮，则取消删除操作，回到“业务账号浏览”界面； 如果单击“确定”按钮，则提交数据到服务器端，进行删除业务账号的操作，如果操作成功，则在界面提示删除成功，界面如图 - 97 所示； 单击图 - 97 所示界面上的“X”图标关闭提示信息框之后，将刷新“业务账号浏览”界面上的数据显示； 如果删除业务账号的操作失败，则在界面提示删除失败的信息，界面如图 - 98 所示； 单击图 - 98 所示界面上的“X”图标关闭提示信息框之后，将回到“业务账号浏览”界面。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<ol style="list-style-type: none"> 业务账号被删除后，数据依然保留，将状态记载为“删除”，并记载删除时间； 删除状态的业务账号，不能再开通、暂停或者修改。

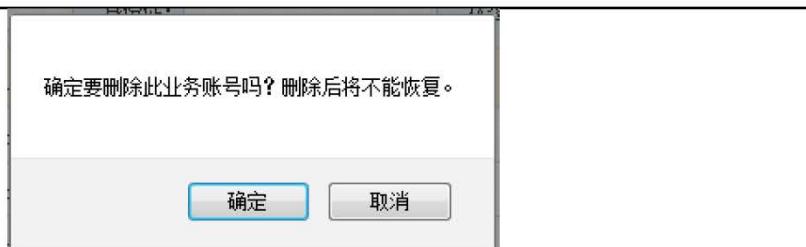


图 - 96



图 - 97



图 - 98

9.2.6. 业务账号详细

用例名称	业务账号详细
功能简述	具有“业务账号管理”权限的管理员登录后，可以查看某业务账号的详细信息
前置条件	具有“业务账号管理”权限的管理员登录成功，且“业务账号浏览”用例或“查询业务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，执行“业务账号浏览”或“查询业务账号”用例得到业务账号的列表数据； 单击界面上某行数据上的“业务 ID”的链接，则进入“业务账号详细”页面以查看该业务账号的详细信息，界面如图 - 99 所示； 界面上将显示该业务账号的所有详细信息，且所有数据项不能修改； 单击“返回”按钮将返回到“业务账号浏览”界面。
分支与异常	<p>第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页；</p> <p>第2步：如果没有查询到相应的业务账号数据，则直接返回到“业务账号浏览”界面。</p>
备注	<ol style="list-style-type: none"> 根据业务账号的状态，在页面上显示开通时间、暂停时间或删除时间。

业务帐号ID:	1
帐务帐号ID:	101
客户姓名:	张三
身份证件号码:	230111111111111111
服务器 IP:	192.168.0.23
OS 账号:	openlab1
状态:	开通
开通时间:	
资费 ID:	
资费名称:	
资费说明:	包 20 小时, 20 小时以内 24.5 元, 超出部分 0.03 元/分钟
返回	

图 - 99

9.2.7. 增加业务账号

用例名称	增加业务账号
功能简述	具有“业务账号管理”权限的管理员登录后，可以增加新的业务账号
前置条件	具有“业务账号管理”权限的管理员登录成功，且“业务账号浏览”用例或“查询业务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，单击“业务账号”的功能链接，则可以进入“业务账号浏览”界面，然后单击界面上的“增加”按钮，则进入“增加业务账号”界面，界面如图 - 100 所示； 在界面上录入各项数据，其中，红色 * 号表示必须录入的数据项； 单击“保存”按钮； 对录入的数据进行输入合法性验证，验证规则如表 - 8 所示； 如果录入的数据验证不通过，则在界面显示录入错误的提示，如图 - 101 所示； 单击“查询帐务账号”按钮，则根据身份证件号码查询该用户的帐务账号，并显示在界面上；如果没有查询到对应的帐务账号，则在界面提示错误信息，界面如图 - 102 所示； 身份证件号码和帐务账号录入完毕后，立即检查该信息是否正确，如果错误，则在界面上提示错误信息，界面如图 - 103 所示；

	<p>8. 如果录入的数据验证通过，则提交数据到服务器端，进行增加业务账号的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 104 所示，该提示信息 3 秒后自动消失；</p> <p>9. 数据提交到服务器端后，进行增加业务账号的操作，如果操作失败，则在页面提示错误信息，如图 - 105 所示，该提示信息 3 秒后自动消失；</p> <p>10. 如果单击页面的“取消”按钮，则回到“业务账号浏览”界面。</p>
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	<p>1. 新创建的业务账号的状态为开通；</p> <p>2. 状态为“暂停”或者“删除”的帐务账号，不能为其添加业务账号；</p> <p>3. 身份证号码和帐务账号必须匹配；</p> <p>4. 同一个服务器上的 OS 账号必须唯一。</p>

图 - 100

表 - 8 “增加业务账号”页面的输入列表及合法性

数据项名称	数据类型	输入合法性及验证要求
身份证	文本	非空，且符合身份证号码的格式要求 必须为已有某帐务账号的身份证号码
帐务账号	文本	非空，30 长度以内的字母、数字和下划线的组合 必须为已有的某帐务账号
资费类型	文本	非空，选择某种资费
服务器 IP	文本	非空，15 长度以内，符合 IP 地址的规范
登录 OS 账号	文本	非空，8 长度以内的字母、数字和下划线的组合 同一台服务器上，OS 账号必须唯一
密码	文本	非空，30 长度以内的字母、数字和下划线的组合

重复密码	文本	非空，两次密码必须相同
------	----	-------------

The screenshot shows a configuration form with the following fields and their current values:

- 身份证号: 230102129878654321
- 账务账号: zhangsan1
- 资费类型: 包 20 小时
- 服务器 IP: 192.168.0.23
- 登录 OS 账号: 创建即启用, 状态为开通
- 密码: (empty)
- 重复密码: (empty)

Validation errors are displayed in red boxes:

- * 没有此身份证号, 请重新录入。
- * 没有此账务账号, 请重新录入。
- * 15 长度, 符合IP的地址规范
- * 8长度以内的字母、数字和下划线的组合
- * 30长度以内的字母、数字和下划线的组合
- * 两次密码必须相同

Buttons at the bottom: 保存 (Save) and 取消 (Cancel).

图 - 101

The screenshot shows a configuration form with the following field:

- 身份证号: 230103199003135678

Validation error:

- * 没有对应的帐务账号, 请重新录入。

图 - 102

The screenshot shows a configuration form with the following field:

- 身份证号: 230103199003135678

Validation error:

- * 没有此身份证号, 请重新录入。

图 - 103



图 - 104

The screenshot shows a configuration form with the following fields and a failure message:

- 身份证号: 230102129878654321
- 账务账号: zhangsan1
- 资费类型: 包 20 小时
- 服务器 IP: 192.168.0.23
- 登录 OS 账号: mary

Failure message:

- 保存失败! 192.168.0.23服务器上已经开通过OS账号“mary”。

Validation errors:

- * 没有此账务账号, 请重新录入。
- * 15 长度, 符合IP的地址规范
- * 8长度以内的字母、数字和下划线的组合

图 - 105

9.2.8. 修改业务账号

用例名称	修改业务账号
功能简述	具有“业务账号管理”权限的管理员登录后，可以修改已有业务账号的数据
前置条件	具有“业务账号管理”权限的管理员登录成功，且“业务账号浏览”用例或者“查询业务账号”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“业务账号管理”权限的管理员登录后，执行“业务账号浏览”或“查询业务账号”用例得到业务账号的列表数据； 单击界面上某行数据上的“修改”按钮，则进入“修改业务账号”界面，界面如图 - 106 所示； 界面上将显示所修改业务账号的关键数据：业务账号 ID、OS 账号、服务器 IP 以及资费类型，其中只能修改资费类型； 修改资费类型后，单击“保存”按钮； 如果并未修改资费类型，则在界面上提示错误信息，如图 - 107 所示； 修改资费类型后，则提交数据到服务器端，进行修改业务账号资费类型的操作，如果操作成功，则在界面上显示操作成功的提示，如图 - 108 所示，该提示信息 3 秒后自动消失； 数据提交到服务器端后，进行修改业务账号的操作，如果操作失败，则在页面提示错误信息，如图 - 109 所示，该提示信息 3 秒后自动消失； 如果单击页面的“取消”按钮，则回到“业务账号浏览”界面。
分支与异常	<p>第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“业务账号管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页；</p> <p>第2步：如果没有查询到相应的业务账号数据，则直接返回到“业务账号浏览”界面。</p>
备注	<ol style="list-style-type: none"> 只能修改业务账号的资费类型； 修改资费类型，并保存成功后，新的资费标准从下月生效。

业务帐号ID: 1

OS 账号: openlab1

服务器 IP: 192.168.0.23

资费类型: 包 20 小时

保存 取消

图 - 106

资费类型: 包 20 小时

※ 请修改资费类型，或者取消修改操作。

图 - 107

✓ 资费修改成功！新资费标准将从下月生效。

图 - 108

✗ 资费修改失败！数据并发错误。

图 - 109

10. “账单管理”模块

10.1. 用例图

NetCTOSS 系统中“账单管理”模块的用例图如图 - 110 所示：

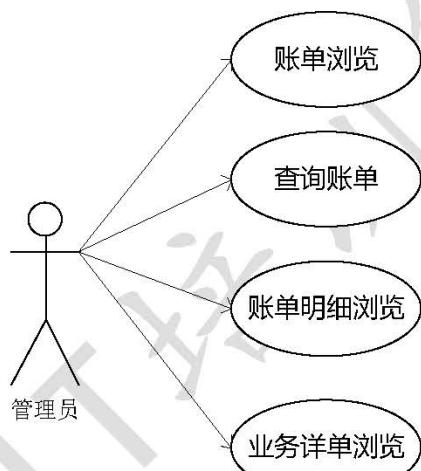


图 - 110

10.2. 用例描述

10.2.1. 账单浏览

用例名称	账单浏览
功能简述	具有“账单管理”权限的管理员登录后，可以查看所有账单的信息
前置条件	具有“账单管理”权限的管理员登录成功
后置条件	此用例执行成功后，可以执行查询账单、账单明细浏览或者业务详单浏览的用例
基本流	<ol style="list-style-type: none"> 具有“账单管理”权限的管理员登录后，单击“账单管理”的功能链接，则可以进入“账单浏览”界面，以查看所有的账单信息，界面如图 - 111 所示； 界面上将分页列出当年所有的账单信息，每页显示 10 条数据； 如果数据小于等于 10 条，则不显示分页的页码； 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并显示与当前页面临近的 5 个页码的链接； 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示，并刷新临

	近 5 个页码的显示，如图 - 112 所示； 6. 单击 “上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化； 7. 单击 “下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化； 8. 单击 “首页”链接，将显示第一页的数据和页码，如果当前页已经是第一页，则页面数据无变化； 9. 单击 “末页”链接，将显示最后一页的数据和页码，如果当前页已经是最后一页，则页面数据无变化； 10. 单击界面上的 “搜索”按钮，则执行“查询账单”用例； 11. 单击表格中某行数据上的 “明细”按钮，则执行“账单明细浏览”用例。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“账单管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	1. 当月账单不显示； 2. 账单的支付状态有两种：已支付和未支付； 3. 账单的支付方式有三种：现金、转账、卡付。



图 - 111



图 - 112

10.2.2. 查询账单

用例名称	查询账单
功能简述	具有“账单管理”权限的管理员登录后，可以查询账单数据
前置条件	具有“账单管理”权限的管理员登录成功，且“账单浏览”用例执行成功
后置条件	此用例执行成功后，可以执行账单浏览、账单明细浏览或者业务详单浏览的用例

基本流	<ol style="list-style-type: none"> 1. 具有“账单管理”权限的管理员登录后，单击“账单管理”的功能链接，则可以进入如图 - 111 所示的“账单浏览”界面； 2. 在界面上录入查询条件，如身份证号、帐务账号和姓名，并选择年份和月份，然后单击“搜索”按钮； 3. 将查询条件数据提交到服务器端，查询账单数据； 4. 如果查询成功，则将查询出的数据显示在图 - 111 所示的表格中； 5. 如果没有查询到相应的数据，则只显示表格的表头。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“账单管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页。
备注	1. 只查询近 3 年的账单，即只查询当前年及其前两年的账单数据。

10.2.3. 账单明细浏览

用例名称	账单明细浏览
功能简述	具有“账单管理”权限的管理员登录后，可以查看某账单的明细数据
前置条件	具有“账单管理”权限的管理员登录成功，且“账单浏览”用例或“查询账单”用例执行成功
后置条件	此用例执行成功，可以执行“业务详单浏览”用例
基本流	<ol style="list-style-type: none"> 1. 具有“账单管理”权限的管理员登录后，执行“账单浏览”或“查询账单”用例得到账单的列表数据； 2. 单击界面上某行数据上的“明细”的链接，则进入“账单明细浏览”页面以查看该账单的详细信息，界面上将列出当前账单的明细信息，并显示某计费周期内某帐务账号下所有业务账号的账单信息，界面如图 - 113 所示； 3. 每页显示 10 条数据，如果数据量小于等于 10 条，则不显示分页的页码； 4. 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并用超级链接的方式显示其他页码； 5. 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示； 6. 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化； 7. 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化； 8. 单击表格中的“详单”链接，链接到“业务详单浏览”页面，以执行“业务详单浏览”用例； 9. 单击页面上的“返回”按钮，返回到“账单浏览”页面。
分支与异常	第 1 步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“账单管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页； 第 2 步：如果没有查询到相应的账单明细数据，则直接返回到“账单浏览”界面。
备注	无



图 - 113

10.2.4. 业务详单浏览

用例名称	业务详单浏览
功能简述	具有“账单管理”权限的管理员登录后，可以查看某业务账号的业务详单
前置条件	具有“账单管理”权限的管理员登录成功，且“账单明细浏览”用例执行成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“账单管理”权限的管理员登录后，执行“账单明细浏览”用例得到账单明细的列表数据； 单击界面上某行数据上的“详单”的链接，则进入“业务详单浏览”页面以查看某业务账号账单的详细信息，界面上将列出当前业务账单的明细信息，并显示某计费周期内某帐务账号下某业务账号的所有详单，界面如图 - 114 所示； 每页显示 10 条数据，如果数据量小于等于 10 条，则不显示分页的页码； 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并用超级链接的方式显示其他页码； 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示； 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化； 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化； 单击页面上的“返回”按钮，则返回到“账单明细浏览”页面，并执行“账单明细浏览”用例。
分支与异常	第1步：没有登录的用户访问此页面，则直接定位到登录页面；已经登录但是没有“账单管理”权限的管理员访问此页面，则定位到图 - 6 所示的无权访问的错误提示页； 第2步：如果没有查询到相应的业务详单数据，则直接返回到“账单明细浏览”界面。
备注	无



图 - 114

11. “报表”模块

11.1. 用例图

NetCTOSS 系统中“报表”模块的用例图如图 - 115 所示：

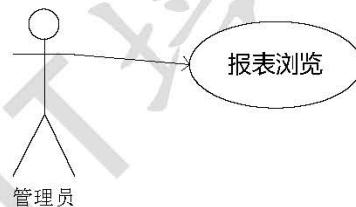


图 - 115

11.2. 用例描述

11.2.1. 报表浏览

用例名称	报表浏览
功能简述	具有“报表”权限的管理员登录后，可以查看所有报表的信息
前置条件	具有“报表”权限的管理员登录成功
后置条件	无
基本流	<ol style="list-style-type: none"> 具有“报表”权限的管理员登录后，单击“报表”的功能链接，则可以进入“报表浏览”界面，以查看所有的报表信息，界面如图 - 116 所示； 界面上首先显示“客户使用时长”报表，该报表用于显示每位客户每月的累计时长，数据以时间和客户 ID 升序排列； 另外两种报表以页签分别显示； 单击“时长排行榜”页签，则显示该报表数据，显示每台服务器上累计时长最高的前三名客户，界面如图 - 117 所示；

	<p>5. 单击“资费使用率”页签，则显示该报表数据，显示每台服务器上每种资费标准的使用次数，界面如图 - 118 所示；</p> <p>6. 显示数据时，每页显示 10 条数据；</p> <p>7. 如果数据小于等于 10 条，则不显示分页的页码；</p> <p>8. 如果数据大于 10 条，则显示头 10 条数据，当前页码为 1，并显示与当前页临近的 5 个页码的链接；</p> <p>9. 单击其他页码的链接，则显示其他页的数据，且当前页码高亮显示，并刷新临近 5 个页码的显示；</p> <p>10. 单击“上一页”链接，将显示上一页的数据，如果当前页已经是第一页，则页面数据无变化；</p> <p>11. 单击“下一页”链接，将显示下一页的数据，如果当前页已经是最后一页，则页面数据无变化；</p> <p>12. 单击“首页”链接，将显示第一页的数据和页码，如果当前页已经是第一页，则页面数据无变化；</p> <p>13. 单击“末页”链接，将显示最后一页的数据和页码，如果当前页已经是最后一页，则页面数据无变化。</p>
分支与异常	无
备注	无



账号 ID	账务帐号	客户名称	身份证号码	电话	月份	累积时长
1	mary	贾强	2202220202020202	13987654345	2013年1月	16小时32分
1	mary	贾强	2202220202020202	13987654345	2013年2月	16小时32分
1	mary	贾强	2202220202020202	13987654345	2013年3月	16小时32分
2	tony	真真	2202220202020202	13987654345	2013年1月	16小时32分

图 - 116

Unix 服务器IP	账务帐号	客户名称	身份证号码	累积时长
192.168.0.20	mary	贾强	220222020202020202	106小时32分
192.168.0.20	tony	米奇	220222020202020202	96小时32分
192.168.0.20	jecy	米妮	220222020202020202	16小时32分
192.168.0.23	mary	贾强	220222020202020202	16小时32分

首页 上一页 1 2 3 4 5 下一页 末页

[源自北美技术，最优秀的师资，最真实的企业环境，最适用的实战项目]
版权所有(C)加拿大达内IT培训集团公司

图 - 117

Unix 服务器IP	包月	套餐	计时
192.168.0.20	345	21	34
192.168.0.23	32	221	314

首页 上一页 1 2 3 4 5 下一页 末页

[源自北美技术，最优秀的师资，最真实的企业环境，最适用的实战项目]
版权所有(C)加拿大达内IT培训集团公司

图 - 118

数据库设计说明书

1. 概述

1.1. 数据库概述

数据库名称为 TARENA，适用于 Oracle 数据库，用于支持 NetCTOSS 项目。

1.2. 数据表清单

分析 NetCTOSS 项目的需求文档，该项目所需要的数据表如表 - 1 所示：

表 - 1 数据表清单

表名	描述
COST	资费信息表，针对 TELNET 服务的各种资费标准
HOST	UNIX 服务器信息表，记录提供远程登录服务的 UNIX 服务器（该表只用于 SQL 课程）
ACCOUNT	帐务信息表，从帐务角度看客户，即自然人，称为帐务账号
SERVICE	业务信息表，客户申请了远程登录业务，即获得一个业务账号，对应一台 UNIX 服务器上的 OS 账号，即 UNIX 服务器的 IP 地址 + OS 账号
SERVICE_DETAIL	业务详单表，记录每一次的 login 和 logout 服务器的信息
SERVICE_UPDATE_BAK	业务更新备份表，记载需要更新的业务账号的相关数据
BILL	账单信息表，客户（帐务账号）每月的总费用信息
BILL_ITEM	账单条目表，客户（帐务账号）每月的详细费用信息，使用每个服务器应付的费用
MODULE_INFO	模块表
FUNCTION_INFO	功能表
ROLE_INFO	角色表，记载系统的角色信息
ROLE_MODULE	角色模块表
ADMIN_INFO	管理员表，记载系统中所有管理员的信息
ADMIN_ROLE	管理员角色表，记载每个管理员的角色信息

1.3. E-R 模型图

数据库的 ER 图如图 - 1 和图 - 2 所示：

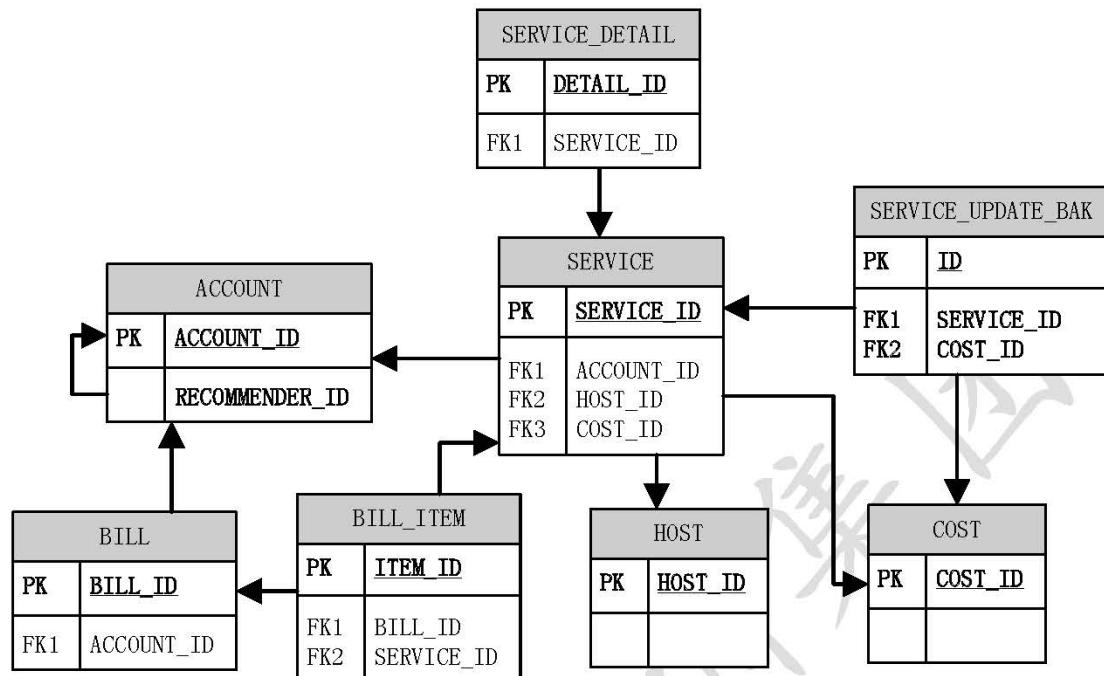


图 - 1

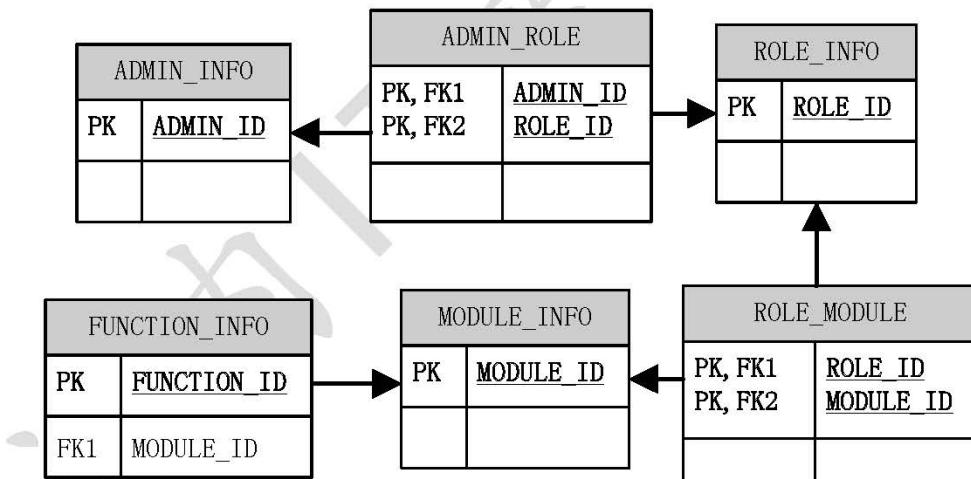


图 - 2

2. 数据表定义

2.1. 资费信息表 (COST)

● 业务说明

资费是运营商对于客户使用服务的收费标准。运营商会定义多种收费标准由客户选择。

比如，客户在 1 个月之内使用 20 小时，费用为 20.45 元；超出 20 小时的部分，以每分钟 0.01 元计费。

资费标准分为三类：

- 1) 包月：BASE_COST 记录包月费，BASE_DURATION 和 UNIT_COST 均为 NULL，即不限时长，没有额外费用；
- 2) 套餐： BASE_COST 记录月固定费，BASE_DURATION 记录套餐内包含的在线时长，UNIT_COST 记录超出部分的单位费用；
- 3) 计时：BASE_COST 和 BASE_DURATION 均为 NULL，即没有月固定费，不包在线时长，UNIT_COST 记录单位费用。

创建资费时，状态为暂停，需要记载资费的创建时间；资费在暂停状态下，可修改，可删除；资费开通后，记载开通时间，且开通后不能修改、不能再停用、也不能删除。

资费降价算作新建新资费，由客户提出申请，在业务账号管理处修改所关联的资费。

业务账号修改资费时，在下月底统一触发，修改业务账号所关联的资费 ID。

● 表结构

资费表的结构如表 - 2 所示：

表 - 2 资费表 (COST)

字段名称	类型	备注	字段描述
COST_ID	NUMBER(4)	PRIMARY KEY NOT NULL	主键，资费 ID
NAME	VARCHAR2(50)	NOT NULL	资费名称
BASE_DURATION	NUMBER(11)	NULL	包在线时长
BASE_COST	NUMBER(7,2)	NULL	月固定费，可能有小数
UNIT_COST	NUMBER(7,4)	NULL	单位费用 (元 / 小时)
STATUS	CHAR(1)	NOT NULL CHECK (0,1)	0 : 开通；1 : 暂停；
DESCR	VARCHAR2(100)	NULL	对资费信息的说明
CREATIME	DATE	DEFAULT SYSDATE	创建日期
STARTTIME	DATE	NULL	启用日期
COST_TYPE	CHAR(1)	NULL	1-包月,2-套餐,3-计时

2.2. UNIX 服务器信息表 (HOST)

● 业务说明

在电信的业务中，有一种 Unix 实验室出租业务。只要用户向电信运营商申请一个 Unix 账号 (OS 账号)，就可以远程登录某 Unix 服务器，以使用 Unix 系统。

需要创建 UNIX 服务器信息表 (HOST)，以记载服务器的相关信息。

● 表结构

UNIX 服务器信息表的结构如表 - 3 所示：

表 - 3 UNIX 服务器信息表 (HOST)

字段名称	类型	备注	字段描述
HOST_ID	VARCHAR2(15)	PRIMARY KEY	主键，记录 unix 服务器的 ip 地址
NAME	VARCHAR2(50)	NOT NULL	主机名
LOCATION	VARCHAR2(100)		主机所在位置

2.3. 帐务账号表 (ACCOUNT)

● 业务说明

帐务账号是运营商对于客户收费的账号，对应自然人。客户如果需要使用运营商提供的服务 (如使用某台 Unix 服务器上的某种服务)，则首先使用身份证件开设帐务账号。客户费用结算的信息就是通过帐务账号来管理。

帐务账号主要记录客户的基本信息、支付方式、所申请的各种业务账号等信息。每个客户只能开设一个帐务账号。关于帐务账号的相关业务，有如下说明：

- 1) 帐务账号 ID 和用户自服务系统的登录名是一一对应的；
- 2) 帐务账号 ID 和身份证件号 ID 是一一对应的关系，即一个帐务账号对应一个自然人，称之为客户；
- 3) 客户姓名是可以重复的，即不同的客户可以有同样的姓名；
- 4) 帐务账号一旦创建，则为开通状态，需要记载其创建时间；
- 5) 暂停帐务账号后，需要记载其暂停时间；
- 6) 重新开通帐务账号后，需要删除其暂停时间；
- 7) 帐务账号被删除后，记载其删除时间，标示为删除，不能再开通、修改、删除；
- 8) 暂停帐务账号时，需要同时暂停下属的所有业务账号；
- 9) 暂停后重新开通帐务账号，并不同时开启其下属的所有业务账号，需要在业务账号管理中单独开启。

● 表结构

帐务账号表的结构如表 - 4 所示：

表 - 4 帐务账号表 (ACCOUNT)

字段名称	类型	备注	字段描述
ACCOUNT_ID	NUMBER(9)	PRIMARY KEY NOT NULL	帐务账号 ID
RECOMMENDER_ID	NUMBER(9)	FORIEGN KEY	推荐人帐务账号 ID
LOGIN_NAME	VARCHAR2(30)	UNIQIUE NOT NULL	登录 NetCTOSS 系统的名称， 用于“用户自服务”系统
LOGIN_PASSWD	VARCHAR2(30)	NOT NULL	登录 NetCTOSS 的口令
STATUS	CHAR (1)	NOT NULL CHECK (0,1,2)	0 : 开通 ; 1 : 暂停 ; 2 : 删除
CREATE_DATE	DATE	DEFAULT SYSDATE	创建日期
PAUSE_DATE	DATE	NULL	暂停日期 (开通状态时为空)
CLOSE_DATE	DATE	NULL	删除日期
REAL_NAME	VARCHAR2(20)	NOT NULL	客户姓名
IDCARD_NO	CHAR(18)	NOT NULL UNIQUE	身份证号码
BIRTHDATE	DATE	NULL	出生日期
GENDER	CHAR(1)	NOT NULL CHECK (0,1)	性别 0:男 1:女
OCCUPATION	VARCHAR2(50)	NULL	职业
TELEPHONE	VARCHAR2(15)	NOT NULL	联系电话 (座机或手机)
EMAIL	VARCHAR2(50)	NULL	电子邮箱
MAILADDRESS	VARCHAR2(200)	NULL	通信地址
ZIPCODE	CHAR(6)	NULL	邮编
QQ	VARCHAR2(15)	NULL	QQ
LAST_LOGIN_TIME	DATE	NULL	最后一次登录时间
LAST_LOGIN_IP	VARCHAR2(15)	NULL	最后一次登录 IP 地址

2.4. 业务账号表 (SERVICE)

● 业务说明

客户开设了帐务账号之后，就可以申请开通其所需要的业务。

同一个客户可以使用多个 Unix 服务器上的服务，分别算作不同的业务，因此需要开设并使用不同的业务账号。比如某客户需要使用 20 服务器上的 telnet 远程登录服务，则需要开设一个业务账号；如果同时还需要使用 23 服务器上的 telnet 远程登录服务，则需要再开设一个业务账号。

客户在开设帐务账号之后，才可以开设业务账号，每个业务账号表示单独的业务。每个业务账号中包含资费类别等信息，用于统计每种业务的费用。

关于业务账号的相关业务，有如下说明：

- 1) 一个帐务账号可以对应多个业务账号；
- 2) 业务账号需要记载其对应的服务器信息，以及登录服务器的 OS 账号；
- 3) UNIX 服务器上的 OS 账号也是客户登录服务器的账号；
- 4) 在同一台服务器上，OS 账号是唯一的，但是在不同的 UNIX 服务器上的 OS 账号是可以相同的；
- 5) 业务账号一旦创建，即为开通状态，需要记载其创建时间；
- 6) 暂停业务账号时，需要记载其暂停时间；
- 7) 重新开通业务账号后，需要删除其暂停时间；
- 8) 如果业务账号所属的帐务账号的状态为暂停或者删除，则不能开通此业务账号；
- 9) 删除业务账号时，记载其删除时间，且标示为删除，不能再开通、修改、删除；
- 10) 业务账号不设计修改密码功能，由“用户自服务系统”中的功能实现；
- 11) 只能修改业务账号的资费类型；
- 12) 修改资费类型，并保存成功后，新的资费标准从下月生效。

● 表结构

业务账号表的结构如表 - 5 所示：

表 - 5 业务账号表 (SERVICE)

字段名称	类型	备注	字段描述
SERVICE_ID	NUMBER(10)	PRIMARY KEY NOT NULL	业务账号 ID
ACCOUNT_ID	NUMBER(9)	FOREIGN KEY NOT NULL	帐务账号 ID ,关联帐务账号表
UNIX_HOST	VARCHAR2(15)	和 OS_USERNAME 做联合唯一键 NOT NULL	UNIX 服务器 IP 地址
OS_USERNAME	VARCHAR2(8)	和 UNIX_HOST 做联合唯一键	登录 UNIX 服务器的

		NOT NULL	OS 账号
LOGIN_PASSWD	VARCHAR2(30)	NOT NULL	登录 UNIX 服务器的口令
STATUS	CHAR (1)	NOT NULL CHECK (0,1,2)	0 : 开通 ; 1 : 暂停 ; 2 : 删除
CREATE_DATE	DATE	DEFAULT SYSDATE	创建日期 , 创建即开通
PAUSE_DATE	DATE	NULL	暂停日期
CLOSE_DATE	DATE	NULL	删除日期
COST_ID	NUMBER(4)	FOREIGN KEY NOT NULL	资费编码 , 关联资费信息表

2.5. 业务详单表 (SERVICE_DETAIL)

● 业务说明

客户开设了业务账号以后，意味着客户可以使用某台 Unix 服务器上的某种服务，而使用服务是需要付费的，因此，需要记载客户的使用详情，用于计费。客户使用服务器的详细信息，即为业务详单。

业务详单需要记载客户每次使用某种服务的登入时刻、登出时刻、时长、登录地址、进程号等信息。这些信息标识：谁在什么时候使用了什么服务，共使用了多长时间。这些信息是计费的数据来源。

● 表结构

业务详单表的结构如表 - 6 所示：

表 - 6 业务详单表 (SERVICE_DETAIL)

字段名称	类型	备注	字段描述
DETAIL_ID	NUMBER(11)	PRIMARY KEY NOT NULL	主键，业务详单 ID
SERVICE_ID	NUMBER(10)	FOREIGN KEY NOT NULL	业务账号 ID , 关联业务账号表
CLIENT_HOST	VARCHAR2(15)	NULL	OS 账号从该 IP 地址登录 UNIX 服务器
OS_USERNAME	VARCHAR2(8)	NULL	UNIX 服务器上的 OS 账号
PID	NUMBER(11)	NULL	进程号
LOGIN_TIME	DATE	NULL	开始登录时间

LOGOUT_TIME	DATE	NULL	退出登录时间
DURATION	NUMBER(20,9)	NULL	时长(秒)
COST	NUMBER(20,6)	NULL	费用

2.6. 业务更新备份表 (SERVICE_UPDATE_BAK)

● 业务说明

用户可以修改某项业务的资费类型，比如由包 20 小时改为包月。用户修改某业务的资费类型，并保存成功后，新的资费标准将从下月生效。因此，即使成功更改了某业务所使用的资费类别，也不能立即修改业务账号表中的数据，而是需要先记载用户需要修改的相关数据，并在当月的月底进行统一修改。

业务更新备份表，就是用来记载用户需要修改的相关业务的数据，在当月的月底，进行统一修改。

关于更新业务账号相关联的资费操作，有如下说明：

- 1) 月底统一修改成功后，需要删除业务更新备份表中相应的数据；
- 2) 暂停或者删除某业务账号时，如果存在该账号所对应的业务更新备份数据，需要删除该数据。

● 表结构

业务更新备份表的结构如表 - 7 所示：

表 - 7 业务更新备份表 (SERVICE_UPDATE_BAK)

字段名称	类型	备注	字段描述
ID	NUMBER(10)	PRIMARY KEY NOT NULL	主键
SERVICE_ID	NUMBER(10)	FOREIGN KEY NOT NULL	业务账号 ID , 关联业务账号表
COST_ID	NUMBER(4)	FOREIGN KEY NOT NULL	资费编码 , 关联资费信息表

2.7. 账单信息表 (BILL)

● 业务说明

用户使用电信运营商提供的 Unix 实验室的服务需要按月缴纳一定的费用，每月的费用，称之为账单。

账单用于记载客户在某月的费用信息，如帐务账号信息、费用、支付方式等。其中，账单的支付方式和支付状态，用于为“用户自服务”系统中的支付功能预留；而账单的具体数

据，在每月的月底进行定时计算。

● 表结构

账单信息表的结构如表 - 8 所示：

表 - 8 账单信息表 (BILL)

字段名称	类型	备注	字段描述
BILL_ID	NUMBER(11)	PRIMARY KEY NOT NULL	主键,账单 ID
ACCOUNT_ID	NUMBER(9)	FOREIGN KEY NOT NULL	帐务账号 ID ,关联帐务账号表
BILL_MONTH	CHAR(6)	NOT NULL	账单月份 , 格式如 201301
COST	NUMBER(13,2)	NOT NULL	费用
PAYMENT_MODE	CHAR(1)	NULL CHECK (0,1,2,3)	0 : 现金 ; 1 : 银行转帐 ; 2 : 邮局汇款 ; 3 : 其他
PAY_STATE	CHAR(1)	DEFAULT 0 CHECK (0,1)	支付状态 0 : 未支付 ; 1 : 已支付 , 默认为 0

2.8. 账单条目表 (BILL_ITEM)

● 业务说明

对于每月的账单，用户可以查询账单的明细，即费用的明细。比如，某客户于 201404 账单期间内，共需要缴费 100 元，客户可以查询此账单的详细情况，即：分别使用何种服务，各花费了多少。因此，在账单条目表中，主要需要记载相关业务的花费，即账单明细。

● 表结构

账单条目表的结构如表 - 9 所示：

表 - 9 账单条目表 (BILL_ITEM)

字段名称	类型	备注	字段描述
ITEM_ID	NUMBER(11)	PRIMARY KEY	主键，账单条目 ID
BILL_ID	NUMBER(11)	FOREIGN KEY NOT NULL	账单 ID ,关联账单信息表
SERVICE_ID	NUMBER(10)	FOREIGN KEY NOT NULL	业务账号 ID ,关联业务信息表

COST	NUMBER(13 , 2)	NULL	所花费的费用
------	----------------	------	--------

2.9. 模块表 (MODULE_INFO)

- 业务说明

NetCTOSS 项目分为多个模块，如资费管理、帐务账号管理、报表查询等。具有不同权限的管理员登录进入 NetCTOSS 系统后，其可以操作的模块也不同。

因此，需要创建模块表 (MODULE_INFO)，以记载功能模块的相关信息。

- 表结构

模块表的结构如表 - 10 所示：

表 - 10 模块表 (MODULE_INFO)

字段名称	类型	备注	字段描述
MODULE_ID	NUMBER(4)	PRIMARY KEY NOT NULL	模块 ID
NAME	VARCHAR2(50)	NOT NULL	模块名称

2.10. 功能表 (FUNCTION_INFO)

- 业务说明

NetCTOSS 项目中的每个模块都包含多个功能。比如，资费管理模块下，包含资费列表、增加资费、修改资费等功能，每个功能对应一个 URL 地址，表示使用该功能的页面地址。

创建功能表 (FUNCTION_INFO)，以记载相关信息。

- 表结构

功能表的结构如表 - 11 所示：

表 - 11 功能表 (FUNCTION_INFO)

字段名称	类型	备注	字段描述
FUNCTION_ID	NUMBER(8)	PRIMARY KEY NOT NULL	功能 ID
FUNCTION_CODE	VARCHAR2(50)	NOT NULL UNIQUE	功能编码
MODULE_ID	NUMBER(4)	FOREIGN KEY NOT NULL	所属的模块 ID，关联模块表
NAME	VARCHAR2(50)	NOT NULL	功能名称

URL	VARCHAR2(100)	NOT NULL	功能对应的 URL 地址
-----	---------------	----------	--------------

2.11. 角色表 (ROLE_INFO)

- 业务说明

角色是多种权限的集合，可通过授予角色来简化权限的管理。比如，角色“账号管理员”同时具有“帐务账号”模块和“业务账号”模块的权限。如果设置某使用者的角色为“账号管理员”，则意味着该使用者可以同时操作“帐务账号”和“业务账号”模块。

创建角色表 (ROLE_INFO)，以记载相关信息。

- 表结构

角色表的结构如表 - 12 所示：

表 - 12 角色表 (ROLE_INFO)

字段名称	类型	备注	字段描述
ROLE_ID	NUMBER(4)	PRIMARY KEY NOT NULL	主键，角色 ID
NAME	VARCHAR2(50)	NOT NULL	角色名称

2.12. 角色模块表 (ROLE_MODULE)

- 业务说明

为了设置角色的权限，需要为角色分配其可以操作的模块。因此，需要创建角色模块表 (ROLE_MODULE)，以记载相关信息。

- 表结构

角色模块表的结构如表 - 13 所示：

表 - 13 角色模块表 (ROLE_MODULE)

字段名称	类型	备注	字段描述
ROLE_ID	NUMBER(4)	PRIMARY KEY NOT NULL FOREIGN KEY	角色 ID，关联角色表 联合主键
MODULE_ID	NUMBER(4)	PRIMARY KEY NOT NULL FOREIGN KEY	模块 ID，关联模块表 联合主键

2.13. 管理员表 (ADMIN_INFO)

- 业务说明

管理员指，可以登录 NetCTOSS 系统，并进行相关操作的用户。通过为管理员设置角色，使得管理员可以操作系统中相应的功能模块，以实现系统管理功能。

因此，需要创建管理员表 (ADMIN_INFO)，以记载其相关信息，如登录名、密码、联系方式等。

- 表结构

管理员表的结构如表 - 14 所示：

表 - 14 管理员表 (ADMIN_INFO)

字段名称	类型	备注	字段描述
ADMIN_ID	NUMBER(4)	PRIMARY KEY NOT NULL	主键，管理员 ID
ADMIN_CODE	VARCHAR2(30)	UNIQUE NOT NULL	管理员账号
PASSWORD	VARCHAR2(30)	NOT NULL	密码
NAME	VARCHAR2(30)	NOT NULL	姓名
TELEPHONE	VARCHAR2(15)	NULL	电话
EMAIL	VARCHAR2(50)	NULL	电子邮件
ENROLLODATE	DATE	NOT NULL DEFAULT SYSDATE	创建日期

2.14. 管理员角色表 (ADMIN_ROLE)

- 业务说明

每个管理员成功登录系统后，其可以操作的模块各不相同，取决于该管理员拥有何种权限。可以通过为管理员配置角色来设置该管理员的权限。

比如，管理员 mary 拥有角色“资费管理”，而“资费管理”角色可以操作模块“资费列表”、“增加资费”、“修改资费”。那么，管理员 mary 登录成功后，就可以操作上述三个模块。

创建管理员角色表 (ADMIN_ROLE)，以记载相关信息。

- 表结构

管理员角色表的结构如表 - 15 所示：

表 - 15 管理员角色表 (ADMIN_ROLE)

字段名称	类型	备注	字段描述
ADMIN_ID	NUMBER(8)	PRIMARY KEY NOT NULL FOREIGN KEY	管理员 ID , 关联管理员表 联合主键
ROLE_ID	NUMBER(4)	PRIMARY KEY NOT NULL FOREIGN KEY	角色 ID , 关联角色表 联合主键

静态结构

NETCTOSS 系统的包结构如图 - 1 所示。

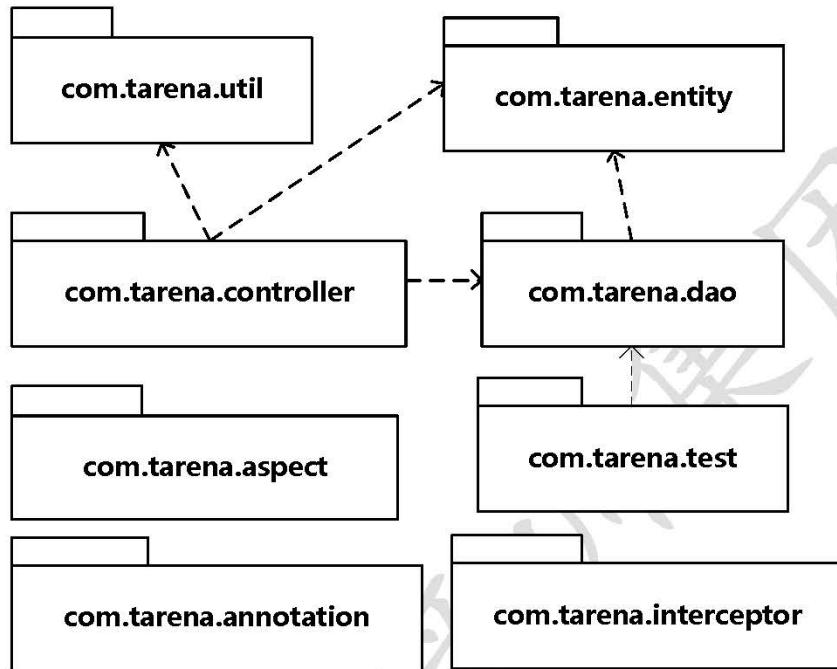


图 - 1

其中，com.tarena 是企业包，与企业域名保持一致，其下的各个包的含义如下：

1. 包 util 中包含了与工具相关的类；
2. 包 entity 中包含了数据库表与 Java 对象进行映射的实体类；
3. 包 dao 包含了访问数据库信息的类；
4. 包 test 包含了对 DAO 的测试类；
5. 包 controller 包含了数据和界面之间的控制类；
6. 包 aspect 包含了异常处理方面的类；
7. 包 annotation 包含了自定义的注解；
8. 包 interceptor 包含了程序中使用的拦截器类。

案例详解

1. 实现资费列表显示功能

- 问题

本案例要求为基于 Spring 和 MyBatis 实现电信计费系统项目的资费列表显示功能。

- 方案

采用 Spring 和 MyBatis 整合方式实现 CostDao 的 findAll 功能，然后将 CostDao 对象注入给 SpringMVC 的 CostController 完成查询列表显示操作。本案例的包结构如图 -1 所示。

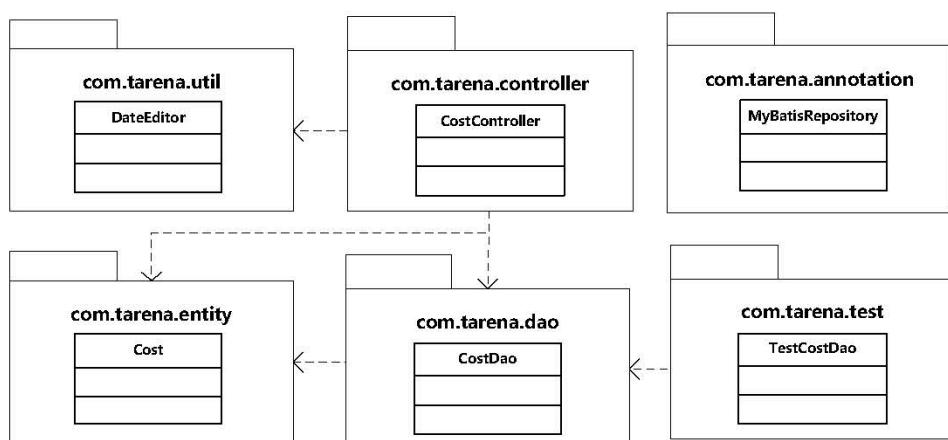


图 - 1

各包下的类含义如下：

1. 包 com.tarena.util 中的 DateEditor 类是处理日期格式的工具类；
2. 包 com.tarena.entity 中的 Cost 是数据库表 cost 与 Java 对象进行映射的实体类；
3. 包 com.tarena.dao 中的 CostDao 包含了操作数据库 cost 表中数据的方法；
4. 包 com.tarena.test 中的 TestCostDao 是对 CostDao 中的方法进行测试的类；
5. 包 com.tarena.controller 中的 CostController 是 CostDao 和页面 cost_list.jsp 之间的协调类；
6. 包 com.tarena.annotation 中的 MyBatisRepository 为自定义的注解；

- 步骤

实现此案例需要按照如下步骤进行。

步骤一：在 Oracle 数据库中创建表和添加测试数据

NetCTOSS 项目所涉及的表请参考 Oracle 课程的 Unit06 部分的“NetCTOSS 数据库设计说明书”。

步骤二：新建工程 NETCTOSS，导入 jar 包

新建名为 NETCTOSS 的 Web 工程，在该工程导入如图-2 所示的 jar 包。

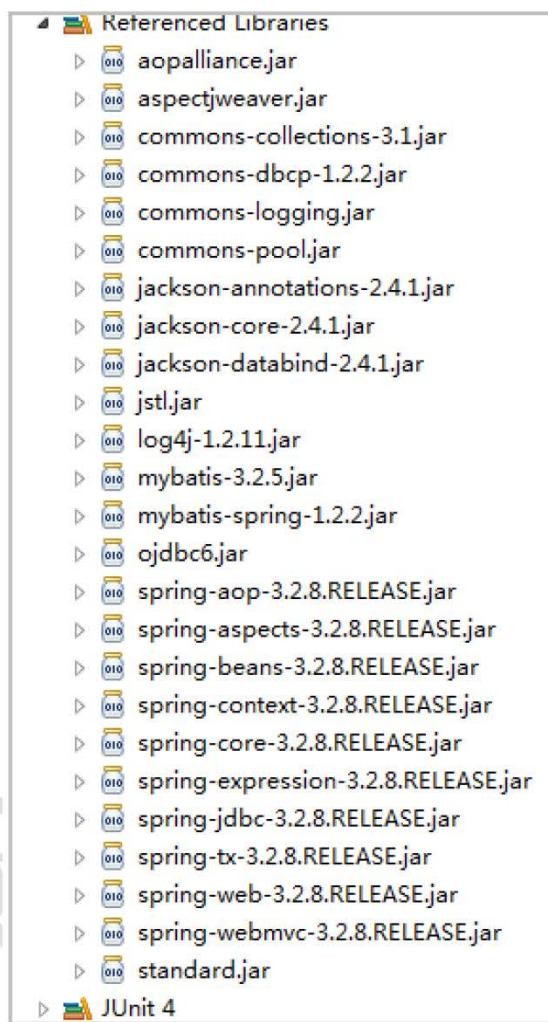


图 - 2

步骤三：导入 web 项目需要的 images、styles、jsp 文件

在 WebRoot 目录下导入开发需要的 images、styles、jsp 资源文件，如图-3 所示。

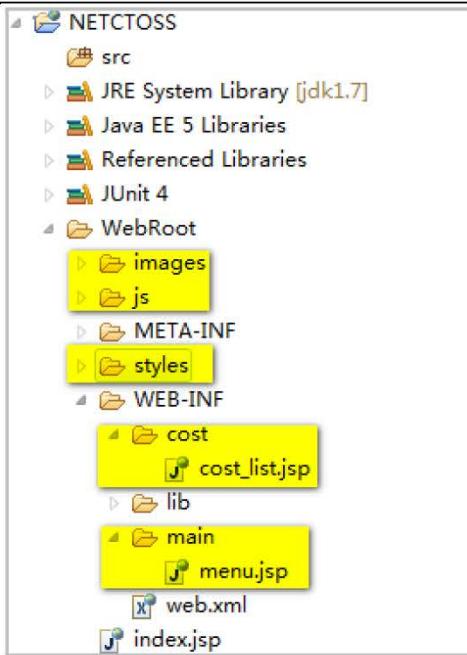


图 - 3

步骤四：修改 web.xml 配置文件

修改 web.xml 配置文件，修改后的 web.xml 的代码如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
    <display-name></display-name>
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
    <servlet>
        <servlet-name>SpringMVC</servlet-name>
        <servlet-class>
            org.springframework.web.servlet.DispatcherServlet
        </servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>
                classpath:applicationContext.xml
            </param-value>
        </init-param>
    </servlet>
    <servlet-mapping>
        <servlet-name>SpringMVC</servlet-name>
        <url-pattern>*.do</url-pattern>
    </servlet-mapping>
    <filter>
        <filter-name>encodingFilter</filter-name>
        <filter-class>
            org.springframework.web.filter.CharacterEncodingFilter
        </filter-class>
        <init-param>
            <param-name>encoding</param-name>
            <param-value>UTF-8</param-value>
        </init-param>
    </filter>
</?xml>
```

```

</filter>
<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>*.do</url-pattern>
</filter-mapping>
<listener>
    <listener-class>
        org.springframework.web.context.request.RequestContextListener
    </listener-class>
</listener>
</web-app>

```

步骤五：创建 Spring 配置文件 applicationContext.xml

在 src 下新建 Spring 配置文件 applicationContext.xml 该文件的位置如图-4 所示。

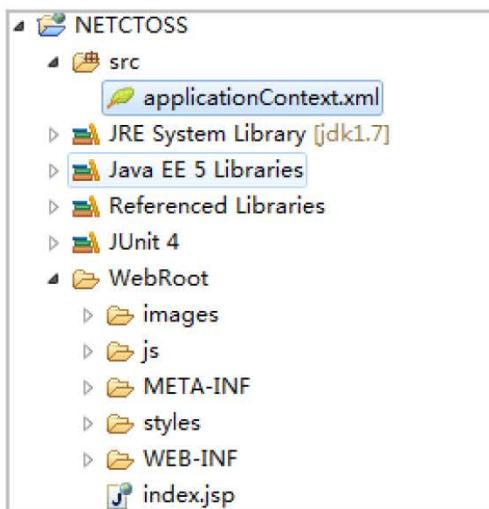


图 - 4

Spring 配置文件 applicationContext.xml 的代码如下所示：

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:jdbc="http://www.springframework.org/schema/jdbc"
    xmlns:jee="http://www.springframework.org/schema/jee"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xmlns:jpa="http://www.springframework.org/schema/data/jpa"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context-3.2.xsd
        http://www.springframework.org/schema/jdbc
        http://www.springframework.org/schema/jdbc/spring-jdbc-3.2.xsd
        http://www.springframework.org/schema/jee
        http://www.springframework.org/schema/jee/spring-jee-3.2.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx-3.2.xsd
        http://www.springframework.org/schema/data/jpa
        http://www.springframework.org/schema/data/jpa/spring-jpa-1.3.xsd
        http://www.springframework.org/schema/aop
        http://www.springframework.org/schema/aop/spring-aop-3.2.xsd
    "
/>

```

```

http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd">

<!-- 定义数据源 -->
<bean id="ds"
      class="org.apache.commons.dbcp.BasicDataSource"
      destroy-method="close">
    <property name="url" value="jdbc:oracle:thin:@localhost:1521:orcl"/>
    <property name="driverClassName" value="oracle.jdbc.OracleDriver"/>
    <property name="username" value="scott"/>
    <property name="password" value="tiger"/>
</bean>

<bean
  class="org.mybatis.spring.SqlSessionFactoryBean" id="sqlSessionFactory"
  >
  <property name="dataSource" ref="ds" />
  <property
    value="classpath:com/tarena/entity/*.xml" name="mapperLocations" />
</bean>

<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
  <property name="basePackage" value="com.tarena.dao" />
  <property name="annotationClass"
            value="com.tarena.annotation.MyBatisRepository"/>
</bean>

<context:component-scan base-package="com.tarena" />

<!-- 支持@RequestMapping 请求和 Controller 映射 -->
<mvc:annotation-driven />

<bean
  class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/WEB-INF/" />
  <property name="suffix" value=".jsp" />
</bean>

<!-- 声明式事务管理 -->
<bean id="txManager"
      class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="ds" />
</bean>
<tx:advice id="txAdvice" transaction-manager="txManager">
  <tx:attributes>
    <tx:method name="find*" read-only="true" />
    <tx:method name="to*" read-only="true" />
    <tx:method name="add*" propagation="REQUIRED" />
    <tx:method name="update*" propagation="REQUIRED" />
    <tx:method name="delete*" propagation="REQUIRED" />
    <tx:method name="start*" propagation="REQUIRED" />
    <tx:method name="pause*" propagation="REQUIRED" />
  </tx:attributes>
</tx:advice>
<aop:config proxy-target-class="true" advice-ref="txAdvice">
  <aop:advisor
    pointcut="within(com.tarena.controller..*)" />
</aop:config>
</beans>

```

步骤六：创建 Cost 实体类和映射文件 CostMapper.xml

首先 新建包 com.tarena.entity 然后新建实体类 Cost 和映射文件 CostMapper.xml ,

这两个文件的位置如图-5 所示。

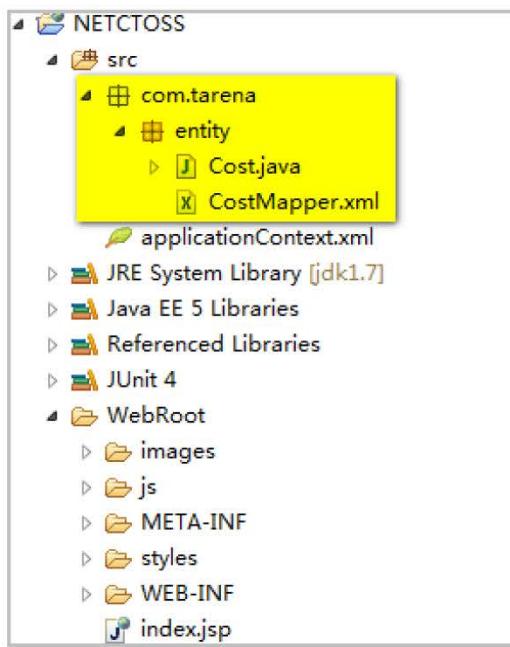


图 - 5

实体类 Cost 的代码如下所示：

```
package com.taren.entity;

import java.sql.Timestamp;

public class Cost {

    private Integer id;
    private String name;
    private Integer base_duration;
    private Double base_cost;
    private Double unit_cost;
    private String status;
    private String descr;
    private Timestamp creatime;
    private Timestamp starttime;
    private String cost_type;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public Integer getBase_duration() {
        return base_duration;
    }
    public void setBase_duration(Integer base_duration) {
        this.base_duration = base_duration;
    }
}
```

```

public Double getBase_cost() {
    return base_cost;
}
public void setBase_cost(Double base_cost) {
    this.base_cost = base_cost;
}
public Double getUnit_cost() {
    return unit_cost;
}
public void setUnit_cost(Double unit_cost) {
    this.unit_cost = unit_cost;
}
public String getStatus() {
    return status;
}
public void setStatus(String status) {
    this.status = status;
}
public String getDescr() {
    return descr;
}
public void setDescr(String descr) {
    this.descr = descr;
}
public Timestamp getCreatime() {
    return creatime;
}
public void setCreatime(Timestamp creatime) {
    this.creatime = creatime;
}
public Timestamp getStartime() {
    return startime;
}
public void setStartime(Timestamp startime) {
    this.startime = startime;
}
public String getCost_type() {
    return cost_type;
}
public void setCost_type(String cost_type) {
    this.cost_type = cost_type;
}
}

```

映射文件 CostMapper.xml 的代码如下所示：

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN"
"http://ibatis.apache.org/dtd/ibatis-3-mapper.dtd">

<mapper namespace="com.tarena.dao.CostDao">

<select id="findAll" resultType="com.tarena.entity.Cost">
    select * from cost order by id
</select>

</mapper>

```

步骤七：创建注解接口 MyBatisRepository

在包 com.tarena 下，新建 annotation 包，然后在该包下创建注解接口 MyBatisRepository，该注解在工程中位置如图-6 所示。

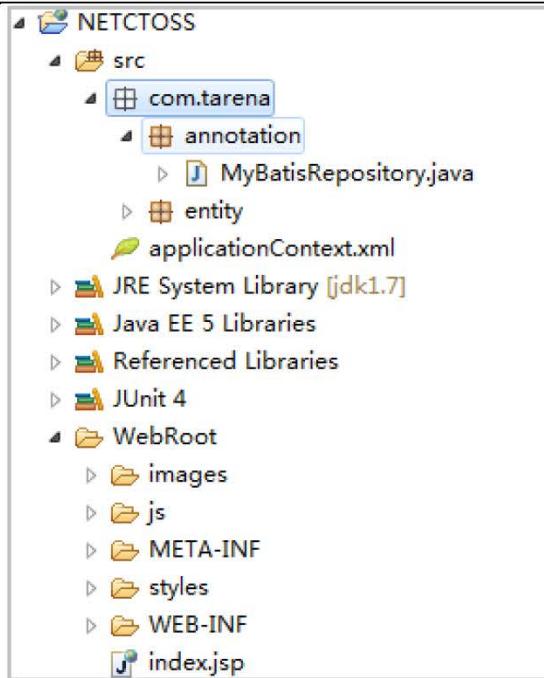


图 - 6

注解接口 MyBatisRepository 的代码如下所示：

```
package com.tarena.annotation;

public @interface MyBatisRepository { }
```

步骤八：创建 DAO 接口 CostMyBatisDao

首先在 com.tarena 下新建包 dao，然后，在该包下创建 DAO 接口 CostDao，该接口在工程中位置如图-7 所示。

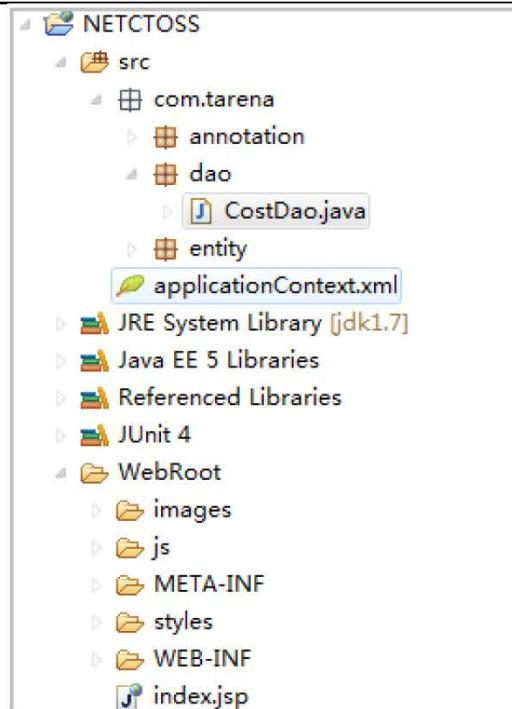


图 - 7

DAO 接口 CostDao 的代码如下所示：

```
package com.tarena.dao;

import java.util.List;

import com.tarena.annotation.MyBatisRepository;
import com.tarena.entity.Cost;

@MyBatisRepository
public interface CostDao {
    List<Cost> findAll();
}
```

步骤九：创建测试类 TestCostDao

首先在包 com.tarena 下新建包 test，然后，在该包下创建测试类 TestCostDao，该类在工程的位置如图-8 所示。

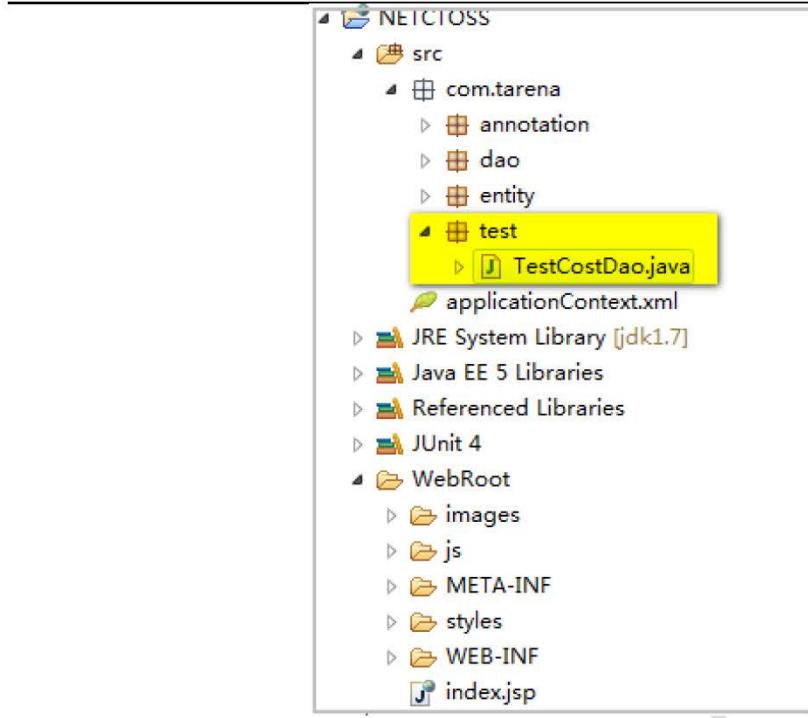


图 - 8

测试类 TestCostDao 的代码如下所示：

```
package com.tarena.test;

import java.util.List;

import org.junit.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.tarena.dao.CostDao;
import com.tarena.entity.Cost;

public class TestCostDao {

    private String conf = "applicationContext.xml";

    @Test
    public void testFindAll() {
        ApplicationContext ctx =
            new ClassPathXmlApplicationContext(conf);
        CostDao dao = ctx.getBean(CostDao.class);
        List<Cost> list = dao.findAll();
        for (Cost c : list) {
            System.out.println(
                c.getId() + " " +
                c.getName() + " " +
                c.getDescr());
        }
    }
}
```

运行测试类 testFindAll 方法，控制台会输出插入的测试数据。

步骤十：创建工具类 DateEditor

首先在包 com.tarena 下新建包 util，在该包下新建工具类 DateEditor，该类用于处理日期的现实格式，该类在工程中的位置如图-9 所示。

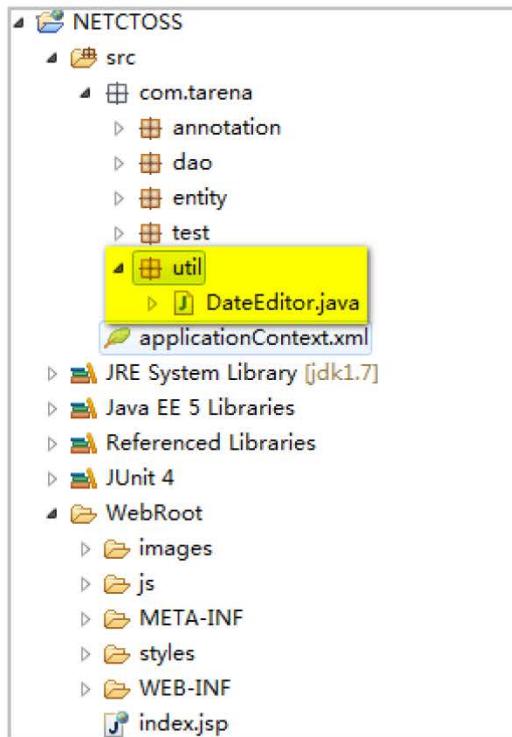


图 - 9

步骤十一：创建控制类的 BaseController 及 CostController

首先，在包 com.tarena 下新建包 controller，然后在该包下新建控制类 BaseController 及 CostController，这两个类在工程中的位置如图-10 所示。

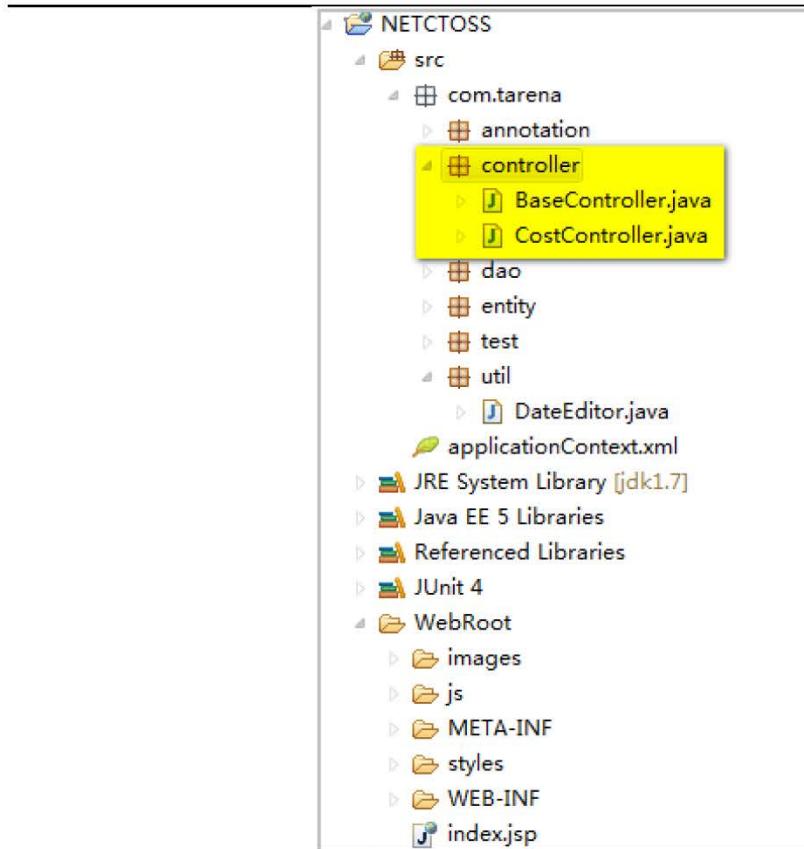


图 - 10

BaseController 类用于处理日期显示时，乱码的问题，**BaseController** 类的代码如下所示：

```
package com.tarena.controller;

import java.io.UnsupportedEncodingException;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import com.tarena.util.DateEditor;

public class BaseController {
    @InitBinder
    protected void initBinder(WebDataBinder binder) {
        binder.registerCustomEditor(
            java.sql.Date.class, new DateEditor());
    }

    protected String isoToUtf8(String s) {
        if (s == null)
            return null;
        try {
            return new String(s.getBytes("ISO8859-1"), "UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

控制类 **CostController** 的代码如下所示：

```

package com.tarena.controller;

import java.util.List;

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.SessionAttributes;

import com.tarena.dao.CostDao;
import com.tarena.entity.Cost;

@Controller
@RequestMapping("/cost")
@SessionAttributes("costPage")
public class CostController extends BaseController {

    @Resource
    private CostDao costDao;

    @RequestMapping("/findCost.do")
    public String find(Model model) {
        List<Cost> list = costDao.findAll();
        model.addAttribute("costs", list);
        return "cost/cost_list";
    }
}

```

步骤十二：修改 menu.jsp 和 cost_list.jsp 页面

修改 cost_list.jsp 页面，该页面的代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>达内 - NetCTOSS</title>
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
        <script language="javascript" type="text/javascript">
            //排序按钮的点击事件
            function sort(btnObj) {
                if (btnObj.className == "sort_desc") {
                    btnObj.className = "sort_asc";
                } else {
                    btnObj.className = "sort_desc";
                }
            }

            //启用
            function startFee() {
                var r = window.confirm("确定要启用此资费吗？资费启用后将不能修改和删除。");
                document.getElementById("operate_result_info").style.display
= "block";
            }

```

```

//删除
function deleteFee(id) {
    var r = window.confirm("确定要删除此资费吗？");
    if(r) {
        window.location.href = "deleteCost.do?id="+id;
    }
}
</script>
</head>
<body>
    <!--Logo 区域开始-->
    <div id="header">
        
        <a href="#">[退出]</a>
    </div>
    <!--Logo 区域结束-->
    <!--导航区域开始-->
    <div id="navi">
        <jsp:include page="/WEB-INF/main/menu.jsp" />
        <!-- <ul id="menu">
            <li><a href="../index.html" class="index_off"></a></li>
            <li><a href="../role/role_list.html" class="role_off"></a></li>
            <li><a href="../admin/admin_list.html" class="admin_off"></a></li>
            <li><a href="../fee/fee_list.html" class="fee_on"></a></li>
            <li><a href="../account/account_list.html" class="account_off"></a></li>
            <li><a href="../service/service_list.html" class="service_off"></a></li>
            <li><a href="../bill/bill_list.html" class="bill_off"></a></li>
            <li><a href="../report/report_list.html" class="report_off"></a></li>
            <li><a href="../user/user_info.html" class="information_off"></a></li>
            <li><a href="../user/user_modi_pwd.html" class="password_off"></a></li>
        </ul> -->
    </div>
    <!--导航区域结束-->
    <!--主要区域开始-->
    <div id="main">
        <form action="" method="">
            <!--排序-->
            <div class="search_add">
                <div>
                    <!--<input type="button" value="月租" class="sort_asc" onclick="sort(this);"/>-->
                    <input type="button" value=" 基 费 " class="sort_asc" onclick="sort(this);"/>
                    <input type="button" value=" 时 长 " class="sort_asc" onclick="sort(this);"/>
                </div>
                <input type="button" value=" 增 加 " class="btn_add" onclick="location.href='toAddCost.do';"/>
            </div>
            <!--启用操作的操作提示-->
            <div id="operate_result_info" class="operate_success">
                
            </div>
        </form>
    </div>

```

```

    删除成功！

</div>
<!--数据区域：用表格展示数据-->
<div id="data">
    <table id="datalist">
        <tr>
            <th>资费 ID</th>
            <th class="width100">资费名称</th>
            <th>基本时长</th>
            <th>基本费用</th>
            <th>单位费用</th>
            <th>创建时间</th>
            <th>开通时间</th>
            <th class="width50">状态</th>
            <th class="width200"></th>
        </tr>
        <c:forEach items="${costs }" var="cost">
            <tr>
                <td>${cost.id }</td>
                <td><a href="#fee_detail.html">${cost.name }</a></td>
                <td>${cost.base_duration }</td>
                <td>${cost.base_cost }</td>
                <td>${cost.unit_cost }</td>
                <td><fmt:formatDate value="${cost.creatime }" pattern="yyyy-MM-dd HH:mm:ss"/></td>
                <td><fmt:formatDate value="${cost.startime }" pattern="yyyy-MM-dd HH:mm:ss"/></td>
                <td>
                    <c:choose>
                        <c:when test="${cost.status==0 }"> 开通
                        <c:otherwise>暂停</c:otherwise>
                    </c:choose>
                </td>
                <td>
                    <input type="button" value="启用" class="btn start" onclick="startFee();"/>
                    <input type="button" value="修改" class="btn_modify" onclick="location.href='toUpdateCost.do?id=${cost.id}';"/>
                    <input type="button" value="删除" class="btn delete" onclick="deleteFee(${cost.id});"/>
                </td>
            </tr>
        </c:forEach>
    </table>
    <p>业务说明：<br />
        1、创建资费时，状态为暂停，记载创建时间；<br />
        2、暂停状态下，可修改，可删除；<br />
        3、开通后，记载开通时间，且开通后不能修改、不能再停用、也不能删除；<br />
        4、业务账号修改资费时，在下月底统一触发，修改其关联的资费 ID（此触发动作由程序处理）
    </p>
</div>
<!--分页-->

```

```

<div id="pages">
    <c:choose>
        <c:when test="${costPage.currentPage==1 }">
            <a href="#">上一页</a>
        </c:when>
        <c:otherwise>
            <a
                href="findCost.do?currentPage=${costPage.currentPage-1 }">上一页</a>
        </c:otherwise>
    </c:choose>

    <c:forEach begin="1" end="${costPage.totalPage }" var="p">
        <c:choose>
            <c:when test="${p==costPage.currentPage }">
                <a href="findCost.do?currentPage=${p }" class="current_page">${p }</a>
            </c:when>
            <c:otherwise>
                <a href="findCost.do?currentPage=${p }">${p }</a>
            </c:otherwise>
        </c:choose>
    </c:forEach>

    <c:choose>
        <c:when
            test="${costPage.currentPage==costPage.totalPage }">
            <a href="#">下一页</a>
        </c:when>
        <c:otherwise>
            <a
                href="findCost.do?currentPage=${costPage.currentPage+1 }">下一页</a>
        </c:otherwise>
    </c:choose>
</div>
<!--主要区域结束-->
<div id="footer">
    <p>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目]</p>
    <p>版权所有(C)加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>

```

menu.jsp 页面的代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort()
+ path + "/";
%>
<ul id="menu">
    <c:choose>
        <c:when test="${currentModule==0 }">
            <li><a href="<%=path%>/login/toIndex.do"
class="index on"></a></li>
        </c:when>
        <c:otherwise>
            <li><a href="<%=path%>/login/toIndex.do"

```

```

class="index off">></a></li>
    </c:otherwise>
</c:choose>

<c:forEach items="${allModules }" var="module">
    <c:if test="${module.module_id==1 }">
        <c:choose>
            <c:when test="${currentModule==1 }">
                <li><a href="<%=path%>/role/findRole.do"
class="role_on"></a></li>
            </c:when>
            <c:otherwise>
                <li><a href="<%=path%>/role/findRole.do"
class="role_off"></a></li>
            </c:otherwise>
        </c:choose>
    </c:if>
    <c:if test="${module.module_id==2 }">
        <c:choose>
            <c:when test="${currentModule==2 }">
                <li><a href="<%=path%>/admin/findAdmin.do"
class="admin_on"></a></li>
            </c:when>
            <c:otherwise>
                <li><a href="<%=path%>/admin/findAdmin.do"
class="admin_off"></a></li>
            </c:otherwise>
        </c:choose>
    </c:if>
    <c:if test="${module.module_id==3 }">
        <c:choose>
            <c:when test="${currentModule==3 }">
                <li><a href="<%=path%>/cost/findCost.do"
class="fee_on"></a></li>
            </c:when>
            <c:otherwise>
                <li><a href="<%=path%>/cost/findCost.do"
class="fee_off"></a></li>
            </c:otherwise>
        </c:choose>
    </c:if>
    <c:if test="${module.module_id==4 }">
        <c:choose>
            <c:when test="${currentModule==4 }">
                <li><a href="<%=path%>/account/findAccount.do"
class="account_on"></a></li>
            </c:when>
            <c:otherwise>
                <li><a href="<%=path%>/account/findAccount.do"
class="account_off"></a></li>
            </c:otherwise>
        </c:choose>
    </c:if>
    <c:if test="${module.module_id==5 }">
        <c:choose>
            <c:when test="${currentModule==5 }">
                <li><a href="<%=path%>/service/findService.do"
class="service_on"></a></li>
            </c:when>
            <c:otherwise>
                <li><a href="<%=path%>/service/findService.do"
class="service_off"></a></li>
            </c:otherwise>
        </c:choose>
    </c:if>
    <%-->
    <c:if test="${module.module_id==6 }">
        <li><a href="bill/bill_list.html" class="bill_off"></a></li>
    </c:if>

```

```

</c:if>
<c:if test="${module.module_id==7 }">
    <li><a href="report/report_list.html" class="report_off"></a></li>
</c:if>
--%>
</c:forEach>
<li><a href="user/user_info.html" class="information off"></a></li>
<li><a href="user/user_modi_pwd.html" class="password off"></a></li>
</ul>

```

步骤十三：部署、运行

将 NETCTOSS 工程部署到 tomcat 下，并启动服务器。打开浏览器访问 URL 地址：

<http://localhost:8080/NETCTOSS/cost/findCost.do>

出现如图-11 所示页面则表示成功：



资费ID	资费名称	基本时长	基本费用	单位费用	创建时间	开通时间	状态	操作
1	5.9元套餐	20	5.9	0.4	2014-06-24 16:30:40		开通	<input type="button" value="启用"/> <input type="button" value="修改"/> <input type="button" value="删除"/>
2	6.9元套餐哈哈	40	6.9	0.3	2014-06-24 16:30:40		暂停	<input type="button" value="启用"/> <input type="button" value="修改"/> <input type="button" value="删除"/>
3	8.5元套餐	100	8.5	0.2	2014-06-24 16:30:40		开通	<input type="button" value="启用"/> <input type="button" value="修改"/> <input type="button" value="删除"/>
4	10.5元套餐	200	10.5	0.1	2014-06-24 16:30:40		暂停	<input type="button" value="启用"/> <input type="button" value="修改"/> <input type="button" value="删除"/>
5	计时收费			0.5	2014-06-24 16:30:40		开通	<input type="button" value="启用"/> <input type="button" value="修改"/> <input type="button" value="删除"/>
6	包月		20.0		2014-06-24 16:30:40		开通	<input type="button" value="启用"/> <input type="button" value="修改"/> <input type="button" value="删除"/>

业务说明：
1、创建资费时，状态为暂停，记载创建时间；
2、暂停状态下，可修改、可删除；
3、开通后，记载开通时间，且开通后不能修改，不能再停用、也不能删除；
4、业务账号修改资费时，在下月底统一触发，修改其关联的资费ID（此触发动作由程序处理）

上一页 下一页

【源自北美技术，最优秀的师资，最真实的企业环境，最适合的实战项目】

图 - 11

• 完整代码

web.xml 文件的完整代码如下所示：

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
    <display-name></display-name>
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
    <servlet>

```

```

<servlet-name>SpringMVC</servlet-name>
<servlet-class>
    org.springframework.web.servlet.DispatcherServlet
</servlet-class>
<init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        classpath:applicationContext.xml
    </param-value>
</init-param>
</servlet>
<servlet-mapping>
    <servlet-name>SpringMVC</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>
<filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>
        org.springframework.web.filter.CharacterEncodingFilter
    </filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>UTF-8</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>*.do</url-pattern>
</filter-mapping>
<listener>
    <listener-class>
        org.springframework.web.context.request.RequestContextListener
    </listener-class>
</listener>
</web-app>

```

实体类 Cost 的完整代码如下所示：

```

package com.tarena.entity;

import java.sql.Timestamp;

public class Cost {

    private Integer id;
    private String name;
    private Integer base_duration;
    private Double base_cost;
    private Double unit_cost;
    private String status;
    private String descr;
    private Timestamp creatime;
    private Timestamp starttime;
    private String cost_type;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}

```

```
    }
    public Integer getBase_duration() {
        return base_duration;
    }
    public void setBase_duration(Integer base_duration) {
        this.base_duration = base_duration;
    }
    public Double getBase_cost() {
        return base_cost;
    }
    public void setBase_cost(Double base_cost) {
        this.base_cost = base_cost;
    }
    public Double getUnit_cost() {
        return unit_cost;
    }
    public void setUnit_cost(Double unit_cost) {
        this.unit_cost = unit_cost;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public String getDescr() {
        return descr;
    }
    public void setDescr(String descr) {
        this.descr = descr;
    }
    public Timestamp getCreatime() {
        return creatime;
    }
    public void setCreatime(Timestamp creatime) {
        this.creatime = creatime;
    }
    public Timestamp getStartime() {
        return startime;
    }
    public void setStartime(Timestamp startime) {
        this.startime = startime;
    }
    public String getCost_type() {
        return cost_type;
    }
    public void setCost_type(String cost_type) {
        this.cost_type = cost_type;
    }
}
```

映射文件 CostMapper.xml 的完整代码如下所示：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN"
"http://ibatis.apache.org/dtd/ibatis-3-mapper.dtd">

<mapper namespace="com.tarena.dao.CostDao">
    <select id="findAll" resultType="com.tarena.entity.Cost">
        select * from cost order by id
    </select>
</mapper>
```

注解接口 MyBatisRepository 的完整代码如下所示：

```
package com.tarena.annotation;

public @interface MyBatisRepository {
```

```
}
```

DAO 接口 CostDao 的完整代码如下所示：

```
package com.tarena.dao;

import java.util.List;

import com.tarena.annotation.MyBatisRepository;
import com.tarena.entity.Cost;

@MyBatisRepository
public interface CostDao {
    List<Cost> findAll();
}
```

测试类 TestCostDao 类的完整代码如下所示：

```
package com.tarena.test;

import java.util.List;

import org.junit.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.tarena.dao.CostDao;
import com.tarena.entity.Cost;

public class TestCostDao {

    private String conf = "applicationContext.xml";

    @Test
    public void testfindAll() {
        ApplicationContext ctx =
            new ClassPathXmlApplicationContext(conf);
        CostDao dao = ctx.getBean(CostDao.class);
        List<Cost> list = dao.findAll();
        for (Cost c : list) {
            System.out.println(
                c.getId() + " " +
                c.getName() + " " +
                c.getDescr()
            );
        }
    }
}
```

类 DateEditor 的完整代码如下所示：

```
package com.tarena.util;
```

```
import java.beans.PropertyEditorSupport;
import java.sql.Date;
import java.text.SimpleDateFormat;

public class DateEditor extends PropertyEditorSupport {

    private String pattern = "yyyy-MM-dd";

    public DateEditor() {

    }

    public DateEditor(String pattern) {
        this.pattern = pattern;
    }

    @Override
    public void setAsText(String text)
            throws IllegalArgumentException {
        if(text == null || text.length() == 0) {
            setValue(null);
        } else {
            SimpleDateFormat sf = new SimpleDateFormat(this.pattern);
            String dateStr = sf.format(Date.valueOf(text));
            setValue(Date.valueOf(dateStr));
        }
    }
}
```

控制类 BaseController 的完整代码如下所示：

```
package com.tarena.controller;

import java.io.UnsupportedEncodingException;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import com.tarena.util.DateEditor;

public class BaseController {
    @InitBinder
    protected void initBinder(WebDataBinder binder) {
        binder.registerCustomEditor(
                java.sql.Date.class, new DateEditor());
    }

    protected String isoToUtf8(String s) {
        if (s == null)
            return null;
        try {
            return new String(s.getBytes("ISO8859-1"), "UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

控制类 CostController 的完整代码如下所示：

```
package com.tarena.controller;

import java.util.List;
```

```

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.SessionAttributes;

import com.tarena.dao.CostDao;
import com.tarena.entity.Cost;

@Controller
@RequestMapping("/cost")
@SessionAttributes("costPage")
public class CostController extends BaseController {

    @Resource
    private CostDao costDao;

    @RequestMapping("/findCost.do")
    public String find(Model model) {
        List<Cost> list = costDao.findAll();
        model.addAttribute("costs", list);
        return "cost/cost_list";
    }
}

```

页面 menu.jsp 的完整代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%
String path = request.getContextPath();
String basePath = request.getScheme() + "://" + request.getServerName() + ":" + request.getServerPort()
+ path + "/";
%>
<ul id="menu">
<c:choose>
    <c:when test="${currentModule==0 }">
        <li><a href="<%=path%>/login/toIndex.do"
class="index_on"></a></li>
    </c:when>
    <c:otherwise>
        <li><a href="<%=path%>/login/toIndex.do"
class="index_off"></a></li>
    </c:otherwise>
    </c:choose>

    <c:forEach items="${allModules }" var="module">
        <c:if test="${module.module_id==1 }">
            <c:choose>
                <c:when test="${currentModule==1 }">
                    <li><a href="<%=path%>/role/findRole.do"
class="role_on"></a></li>
                </c:when>
                <c:otherwise>
                    <li><a href="<%=path%>/role/findRole.do"
class="role_off"></a></li>
                </c:otherwise>
            </c:choose>
        </c:if>
        <c:if test="${module.module_id==2 }">
            <c:choose>
                <c:when test="${currentModule==2 }">
                    <li><a href="<%=path%>/admin/findAdmin.do"
class="admin_on"></a></li>
                </c:when>
            </c:choose>
        </c:if>
    </c:forEach>
</c:choose>
</ul>

```

```

        </c:when>
        <c:otherwise>
            <li><a href="<%path%>/admin/findAdmin.do"
class="admin_off"></a></li>
        </c:otherwise>
    </c:choose>
</c:if>
<c:if test="${module.module_id==3 }">
    <c:choose>
        <c:when test="${currentModule==3 }">
            <li><a href="<%path%>/cost/findCost.do"
class="fee_on"></a></li>
        </c:when>
        <c:otherwise>
            <li><a href="<%path%>/cost/findCost.do"
class="fee_off"></a></li>
        </c:otherwise>
    </c:choose>
</c:if>
<c:if test="${module.module_id==4 }">
    <c:choose>
        <c:when test="${currentModule==4 }">
            <li><a href="<%path%>/account/findAccount.do"
class="account_on"></a></li>
        </c:when>
        <c:otherwise>
            <li><a href="<%path%>/account/findAccount.do"
class="account_off"></a></li>
        </c:otherwise>
    </c:choose>
</c:if>
<c:if test="${module.module_id==5 }">
    <c:choose>
        <c:when test="${currentModule==5 }">
            <li><a href="<%path%>/service/findService.do"
class="service_on"></a></li>
        </c:when>
        <c:otherwise>
            <li><a href="<%path%>/service/findService.do"
class="service_off"></a></li>
        </c:otherwise>
    </c:choose>
</c:if>
<%-->
<c:if test="${module.module_id==6 }">
    <li><a href="bill/bill_list.html" class="bill_off"></a></li>
</c:if>
<c:if test="${module.module_id==7 }">
    <li><a href="report/report_list.html" class="report_off"></a></li>
</c:if>
<%-->
</c:forEach>
<li><a href="user/user_info.html" class="information_off"></a></li>
<li><a href="user/user_modi_pwd.html" class="password_off"></a></li>
</ul>

```

页面 cost_list.jsp 页面的完整代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt "%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

```

<title>达内-NetCTOSS</title>
<link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
<link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
<script language="javascript" type="text/javascript">
    //排序按钮的点击事件
    function sort(btnObj) {
        if (btnObj.className == "sort_desc")
            btnObj.className = "sort asc";
        else
            btnObj.className = "sort_desc";
    }

    //启用
    function startFee() {
        var r = window.confirm("确定要启用此资费吗？资费启用后将不能修改和删除。");
        document.getElementById("operate result info").style.display
= "block";
    }
    //删除
    function deleteFee(id) {
        var r = window.confirm("确定要删除此资费吗？");
        if(r)
            window.location.href = "deleteCost.do?id="+id;
    }
</script>
</head>
<body>
    <!--Logo 区域开始-->
    <div id="header">
        
        <a href="#">[退出]</a>
    </div>
    <!--Logo 区域结束-->
    <!--导航区域开始-->
    <div id="navi">
        <jsp:include page="/WEB-INF/main/menu.jsp" />
        <!-- <ul id="menu">
            <li><a href="../index.html" class="index_off"></a></li>
            <li><a href="../role/role_list.html"
class="role off"></a></li>
            <li><a href="..//admin/admin_list.html"
class="admin_off"></a></li>
            <li><a href="..//fee/fee_list.html" class="fee_on"></a></li>
            <li><a href="..//account/account_list.html"
class="account off"></a></li>
            <li><a href="..//service/service_list.html"
class="service_off"></a></li>
            <li><a href="..//bill/bill_list.html"
class="bill_off"></a></li>
            <li><a href="..//report/report_list.html"
class="report off"></a></li>
            <li><a href="..//user/user_info.html"
class="information off"></a></li>
            <li><a href="..//user/user_modi_pwd.html"
class="password off"></a></li>
        </ul> -->
    </div>
    <!--导航区域结束-->
    <!--主要区域开始-->
    <div id="main">
        <form action="" method="">
            <!--排序-->

```

```

<div class="search add">
    <div>
        <!--<input type="button" value="月租" class="sort asc"
        onclick="sort(this);"/><!--&gt;
        &lt;input type="button" value=" 基 费 " class="sort_asc"
        onclick="sort(this);"/&gt;
        &lt;input type="button" value=" 时 长 " class="sort_asc"
        onclick="sort(this);"/&gt;
    &lt;/div&gt;
    &lt;input type="button" value=" 增 加 " class="btn_add"
    onclick="location.href='toAddCost.do';"/&gt;
&lt;/div&gt;
&lt;!--启用操作的操作提示--&gt;
&lt;div id="operate result info" class="operate success"&gt;
    &lt;img src="..../images/close.png"
    onclick="this.parentNode.style.display='none';" /&gt;
    删除成功!
&lt;/div&gt;
&lt;!--数据区域: 用表格展示数据--&gt;
&lt;div id="data"&gt;
    &lt;table id="datalist"&gt;
        &lt;tr&gt;
            &lt;th&gt;资费 ID&lt;/th&gt;
            &lt;th class="width100"&gt;资费名称&lt;/th&gt;
            &lt;th&gt;基本时长&lt;/th&gt;
            &lt;th&gt;基本费用&lt;/th&gt;
            &lt;th&gt;单位费用&lt;/th&gt;
            &lt;th&gt;创建时间&lt;/th&gt;
            &lt;th&gt;开通时间&lt;/th&gt;
            &lt;th class="width50"&gt;状态&lt;/th&gt;
            &lt;th class="width200"&gt;&lt;/th&gt;
        &lt;/tr&gt;
        &lt;c:forEach items="${costs }" var="cost"&gt;
        &lt;tr&gt;
            &lt;td&gt;${cost.id }&lt;/td&gt;
            &lt;td&gt;&lt;a href="fee_detail.html"&gt;${cost.name }&lt;/a&gt;&lt;/td&gt;
            &lt;td&gt;${cost.base_duration }&lt;/td&gt;
            &lt;td&gt;${cost.base_cost }&lt;/td&gt;
            &lt;td&gt;${cost.unit_cost }&lt;/td&gt;
            &lt;td&gt;&lt;fmt:formatDate pattern="yyyy-MM-dd HH:mm:ss" value="${cost.creatime }" /&gt;&lt;/td&gt;
            &lt;td&gt;&lt;fmt:formatDate pattern="yyyy-MM-dd HH:mm:ss" value="${cost.startime }" /&gt;&lt;/td&gt;
            &lt;td&gt;
                &lt;c:choose&gt;
                    &lt;c:when test="${cost.status==0 }"&gt; 开 通
                &lt;/c:when&gt;
                    &lt;c:otherwise&gt;暂停&lt;/c:otherwise&gt;
                &lt;/c:choose&gt;
            &lt;/td&gt;
            &lt;td&gt;
                &lt;input type="button" value=" 启 用 "
                class="btn_start" onclick="startFee();"/&gt;
                &lt;input type="button" value=" 修 改 "
                class="btn_modify" onclick="location.href='toUpdateCost.do?id=${cost.id}';"/&gt;
                &lt;input type="button" value=" 删 除 "
                class="btn_delete" onclick="deleteFee(${cost.id});"/&gt;
            &lt;/td&gt;
        &lt;/tr&gt;
    &lt;/c:forEach&gt;
&lt;/table&gt;
&lt;p&gt;业务说明:&lt;br /&gt;
1、创建资费时，状态为暂停，记载创建时间;&lt;br /&gt;
</pre>

```

```

2、暂停状态下，可修改，可删除；<br />
3、开通后，记载开通时间，且开通后不能修改、不能再停用、也不能删除；
<br />
4、业务账号修改资费时，在下月底统一触发，修改其关联的资费 ID（此触发
动作由程序处理）
        </p>
    </div>
    <!--分页-->
    <div id="pages">
        <c:choose>
            <c:when test="${costPage.currentPage==1 }">
                <a href="#">上一页</a>
            </c:when>
            <c:otherwise>
                <a
href="findCost.do?currentPage=${costPage.currentPage-1 }">上一页</a>
                </c:otherwise>
            </c:choose>

            <c:forEach begin="1" end="${costPage.totalPage }" var="p">
                <c:choose>
                    <c:when test="${p==costPage.currentPage }">
                        <a href="findCost.do?currentPage=${p }" class="current page">${p }</a>
                    </c:when>
                    <c:otherwise>
                        <a href="findCost.do?currentPage=${p }">${p }</a>
                    </c:otherwise>
                </c:choose>
            </c:forEach>

            <c:choose>
                <c:when
test="${costPage.currentPage==costPage.totalPage }">
                    <a href="#">下一页</a>
                </c:when>
                <c:otherwise>
                    <a
href="findCost.do?currentPage=${costPage.currentPage+1 }">下一页</a>
                </c:otherwise>
            </c:choose>
        </div>
        </form>
    </div>
    <!--主要区域结束-->
    <div id="footer">
        <p>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目]</p>
        <p>版权所有(C)加拿大达内 IT 培训集团公司 </p>
    </div>
</body>
</html>

```

2. 实现帐务账号的增改查以及暂停

- 问题

本案例要求完成 NETCTOSS 项目需求中的帐务账号的增删改查功能。

- 方案

首先采用 Spring 和 MyBatis 整合方式实现 AccountDao 中的增改查以及暂停方法，然后将 AccountsDao 对象注入给 SpringMVC 的 AccountController 完成增改查以及暂停操作。本案例的工程结构如图-12 所示。

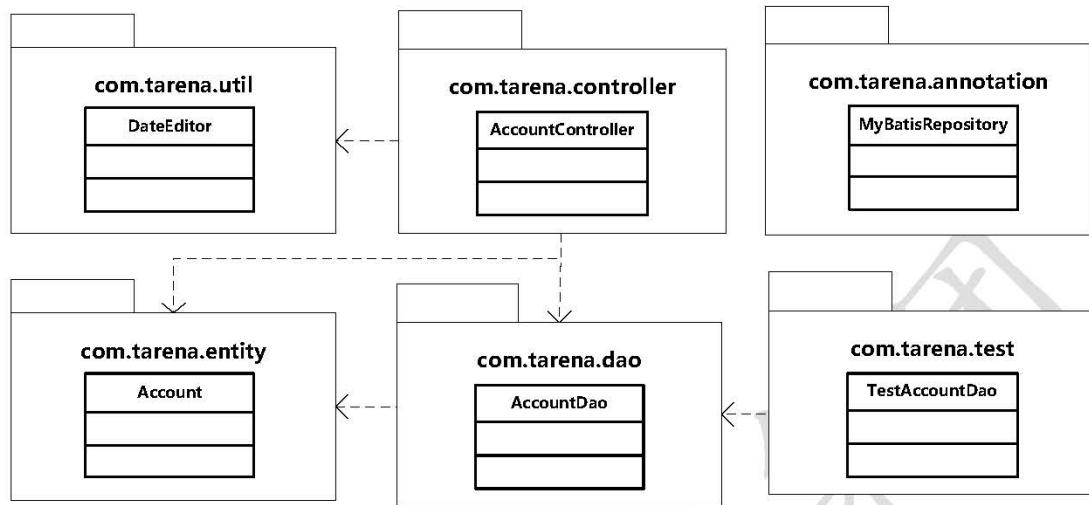


图 - 12

各包下的类含义如下：

1. 包 com.tarena.util 中的 DateEditor 类是处理日期格式和的工具类；
2. 包 com.tarena.entity 中的 Account 是数据库表 account 与 Java 对象进行映射的实体类；
3. 包 com.tarena.dao 中的 AccountDao 包含了操作数据库 account 表中的数据的方法；
4. 包 com.tarena.test 中的 TestAccountDao 是对 AccountDao 中的方法进行测试的类；
5. 包 com.tarena.controller 中的 AccountController 是 AccountDao 和页面之间的协调类；
6. 包 com.tarena.annotation 中的 MyBatisRepository 为自定义的注解；

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：帐务账号的列表显示功能

1. 在 com.tarena.entity 包下创建实体类 Account 和映射文件 AccountMapper.xml , Account 类的代码如下所示：

```

package com.tarena.entity;

import java.sql.Date;
import java.sql.Timestamp;
  
```

```
public class Account {

    private Integer id;
    private Integer recommender_id;
    private String login_name;
    private String login_passwd;
    private String status;
    private Timestamp create_date;
    private Timestamp pause_date;
    private Timestamp close_date;
    private String real_name;
    private String idcard_no;
    private Date birthdate;
    private String gender;
    private String occupation;
    private String telephone;
    private String email;
    private String mailaddress;
    private String zipcode;
    private String qq;
    private Timestamp last_login_time;
    private String last_login_ip;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public Integer getRecommender_id() {
        return recommender_id;
    }
    public void setRecommender_id(Integer recommender_id) {
        this.recommender_id = recommender_id;
    }
    public String getLogin_name() {
        return login_name;
    }
    public void setLogin_name(String login_name) {
        this.login_name = login_name;
    }
    public String getLogin_passwd() {
        return login_passwd;
    }
    public void setLogin_passwd(String login_passwd) {
        this.login_passwd = login_passwd;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public Timestamp getCreate_date() {
        return create_date;
    }
    public void setCreate_date(Timestamp create_date) {
        this.create_date = create_date;
    }
    public Timestamp getPause_date() {
        return pause_date;
    }
    public void setPause_date(Timestamp pause_date) {
        this.pause_date = pause_date;
    }
    public Timestamp getClose_date() {
        return close_date;
    }
}
```

```
public void setClose_date(Timestamp close_date) {
    this.close_date = close_date;
}
public String getReal_name() {
    return real_name;
}
public void setReal_name(String real_name) {
    this.real_name = real_name;
}
public String getIdcard_no() {
    return idcard_no;
}
public void setIdcard_no(String idcard_no) {
    this.idcard_no = idcard_no;
}
public Date getBirthdate() {
    return birthdate;
}
public void setBirthdate(Date birthdate) {
    this.birthdate = birthdate;
}
public String getGender() {
    return gender;
}
public void setGender(String gender) {
    this.gender = gender;
}
public String getOccupation() {
    return occupation;
}
public void setOccupation(String occupation) {
    this.occupation = occupation;
}
public String getTelephone() {
    return telephone;
}
public void setTelephone(String telephone) {
    this.telephone = telephone;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getMailaddress() {
    return mailaddress;
}
public void setMailaddress(String mailaddress) {
    this.mailaddress = mailaddress;
}
public String getZipcode() {
    return zipcode;
}
public void setZipcode(String zipcode) {
    this.zipcode = zipcode;
}
public String getQq() {
    return qq;
}
public void setQq(String qq) {
    this.qq = qq;
}
public Timestamp getLast_login_time() {
    return last_login_time;
}
public void setLast_login_time(Timestamp last_login_time) {
    this.last_login_time = last_login_time;
}
```

```

    }
    public String getLast_login_ip() {
        return last_login_ip;
    }
    public void setLast_login_ip(String last_login_ip) {
        this.last_login_ip = last_login_ip;
    }
}

```

新建映射文件 AccountMapper.xml , 代码如下所示 :

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN"
"http://ibatis.apache.org/dtd/ibatis-3-mapper.dtd">

<mapper namespace="com.tarena.dao.AccountDao">

<select id="findByPage"
parameterType="com.tarena.entity.page.Page"
resultType="com.tarena.entity.Account">
select * from (
    select a.* , ROWNUM r from (
        select * from account
        <where>
            <if test="idcard_no!=null && idcard_no.length()>0">
                and idcard_no=#{idcard_no}
            </if>
            <if test="real_name!=null && real_name.length()>0">
                and real_name=#{real_name}
            </if>
            <if test="login_name!=null && login_name.length()>0">
                and login_name=#{login_name}
            </if>
            <if test='status!=null && !status.equals("-1")'>
                and status=#{status}
            </if>
        </where>
        order by id
    ) a
) where r<#{end} and r>#{begin}
</select>

<select id="findRows"
parameterType="com.tarena.entity.page.Page"
resultType="java.lang.Integer">
select count(*) from account
<where>
    <if test="idcard_no!=null && idcard_no.length()>0">
        and idcard_no=#{idcard_no}
    </if>
    <if test="real_name!=null && real_name.length()>0">
        and real_name=#{real_name}
    </if>
    <if test="login_name!=null && login_name.length()>0">
        and login_name=#{login_name}
    </if>
    <if test='status!=null && !status.equals("-1")'>
        and status=#{status}
    </if>
</where>
</select>
</mapper>

```

2. 在 com.tarena.entity 包下新建包 page 然后在该包下创建分页所需的 Page 类，代码如下所示：

```
package com.tarena.entity.page;

public abstract class Page {

    // 分页时输入的参数
    private int currentPage = 1; // 当前页
    private int pageSize = 5; // 每页显示最大行数

    // 写分页 SQL 时需要的条件，由输入参数计算而来
    private int begin; // 当前页的起始行
    private int end; // 当前页的终止行

    // 用于计算总页数
    private int rows; // 总行数

    // 总页数，由 rows 和 pageSize 计算而来
    private int totalPage;

    public int getBegin() {
        begin = (currentPage - 1) * pageSize;
        return begin;
    }

    public int getEnd() {
        end = currentPage * pageSize + 1;
        return end;
    }

    public int getTotalPage() {
        if (rows % pageSize == 0) {
            totalPage = rows / pageSize;
        } else {
            totalPage = rows / pageSize + 1;
        }
        return totalPage;
    }

    public int getCurrentPage() {
        return currentPage;
    }

    public void setCurrentPage(int currentPage) {
        this.currentPage = currentPage;
    }

    public int getPageSize() {
        return pageSize;
    }

    public void setPageSize(int pageSize) {
        this.pageSize = pageSize;
    }

    public int getRows() {
        return rows;
    }

    public void setRows(int rows) {
```

```
        this.rows = rows;
    }
}
```

3. 在 com.tarena.entity.page 包下创建 AccountPage 类，该类继承自 Page，包含了分页所需信息和查询条件所需的信息，代码如下所示：

```
package com.tarena.entity.page;

public class AccountPage extends Page {

    private String idcard_no;
    private String real_name;
    private String login_name;
    private String status;

    public String getIdcard_no() {
        return idcard_no;
    }

    public void setIdcard_no(String idcard_no) {
        this.idcard_no = idcard_no;
    }

    public String getReal_name() {
        return real_name;
    }

    public void setReal_name(String real_name) {
        this.real_name = real_name;
    }

    public String getLogin_name() {
        return login_name;
    }

    public void setLogin_name(String login_name) {
        this.login_name = login_name;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }
}
```

4. 在 com.tarena.dao 包下 创建 DAO 接口 AccountDao，代码如下所示：

```
package com.tarena.dao;

import java.util.List;

import com.tarena.annotation.MyBatisRepository;
import com.tarena.entity.Account;
import com.tarena.entity.page.Page;

@MyBatisRepository
public interface AccountDao {
```

```

List<Account> findByPage(Page page);

int findRows(Page page);
}

```

5. 创建控制类 AccountController , 代码如下所示 :

```

package com.tarena.controller;

import java.util.List;
import javax.annotation.Resource;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.SessionAttributes;
import com.tarena.dao.AccountDao;
import com.tarena.entity.Account;
import com.tarena.entity.page.AccountPage;

@Controller
@RequestMapping("/account")
@SessionAttributes("accountPage")
public class AccountController extends BaseController {

    @Resource
    private AccountDao accountDao;

    @RequestMapping("/findAccount.do")
    public String find(AccountPage page, Model model) {
        List<Account> list = accountDao.findByPage(page);
        model.addAttribute("accounts", list);

        page.setRows(accountDao.findRows(page));
        model.addAttribute("accountPage", page);

        return "account/account_list";
    }
}

```

6. 创建 account_list.jsp , 代码如下所示 :

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>达内 - NetCTOSS</title>
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
        <script language="javascript" type="text/javascript">
            //删除
            function deleteAccount() {
                var r = window.confirm("确定要删除此帐务账号吗 ? \r\n 删除后将不能恢
复, 且会删除其下属的所有业务账号。");
                document.getElementById("operate_result_info").style.display
= "block";
            }

```

```

//开通或暂停
function setState() {
    var r = window.confirm("确定要开通此帐务账号吗？");
    document.getElementById("operate_result_info").style.display
= "block";
}
//翻页
function to_page(page) {
    document.getElementById("current_page").value = page;
    document.forms[0].submit();
}
//暂停
function pause account(id) {
    var r = window.confirm("确定要暂停此帐务账号吗？");
    if(r) {
        window.location.href = "pauseAccount.do?id="+id;
    }
}
</script>
</head>
<body>

<!--Logo 区域开始--&gt;
&lt;div id="header"&gt;
    &lt;img src="../images/logo.png" alt="logo" class="left"/&gt;
    &lt;a href="#"&gt;[退出]&lt;/a&gt;
&lt;/div&gt;
<!--Logo 区域结束--&gt;
<!--导航区域开始--&gt;
&lt;div id="navi"&gt;
    &lt;jsp:include page="/WEB-INF/main/menu.jsp" /&gt;
    &lt;!-- &lt;ul id="menu"&gt;
        &lt;li&gt;&lt;a href="../index.html" class="index_off"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="role_off"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="admin off"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="account_on"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="service off"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="bill off"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="report off"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="information off"&gt;&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="..&gt;</li&gt;
class="password off"&gt;&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt; --&gt;
&lt;/div&gt;
<!--导航区域结束--&gt;
<!--主要区域开始--&gt;
&lt;div id="main"&gt;
    &lt;form action="findAccount.do" method="post"&gt;
        &lt;input type="hidden" name="currentPage" id="current_page"
value="${accountPage.currentPage }"/&gt;
        &lt;!--查询--&gt;
        &lt;div class="search_add"&gt;
            &lt;div&gt; 身份证 : &lt;input type="text" class="text_search"
name="idcard_no" value="${accountPage.idcard_no }" /&gt;&lt;/div&gt;
</pre>

```

```

        <div>姓名 : <input type="text" class="width70 text_search"
name="real_name" value="${accountPage.real_name }" /></div>
        <div> 登录名 : <input type="text" class="text_search"
name="login_name" value="${accountPage.login_name }" /></div>
        <div>

            状态 :

            <select class="select search" name="status">
                <option value="-1">全部</option>
                <option value="0" >${accountPage.status}</option>
                <c:if test="${accountPage.status==0 }">selected</c:if>>开通</option>
                <option value="1" >${accountPage.status}</option>
                <c:if test="${accountPage.status==1 }">selected</c:if>>暂停</option>
                <option value="2" >${accountPage.status}</option>
                <c:if test="${accountPage.status==2 }">selected</c:if>>删除</option>
            </select>
        </div>
        <div><input type="button" value="搜索" class="btn_search"
onclick="to page(1);"/></div>
        <input type="button" value=" 增 加 " class="btn_add"
onclick="location.href='toAddAccount.do';" />
<!--删除等的操作提示-->
<div id="operate_result_info" class="operate_success">
    
    删除成功 , 且已删除其下属的业务账号 !
</div>
<!--数据区域 : 用表格展示数据-->
<div id="data">
    <table id="datalist">
        <tr>
            <th>账号 ID</th>
            <th>姓名</th>
            <th class="width150">身份证</th>
            <th>登录名</th>
            <th>状态</th>
            <th class="width100">创建日期</th>
            <th class="width150">上次登录时间</th>
            <th class="width200"></th>
        </tr>
        <c:forEach items="${accounts }" var="acc">
            <tr>
                <td>${acc.id }</td>
                <td><a href="account_detail.html">${acc.real_name }</a></td>
                <td>${acc.idcard_no }</td>
                <td>${acc.login_name }</td>
                <td>
                    <c:choose>
                        <c:when test="${acc.status==0 }">开通</c:when>
                        <c:when test="${acc.status==1 }">暂停</c:when>
                        <c:otherwise>删除</c:otherwise>
                    </c:choose>
                </td>
                <td><fmt:formatDate value="${acc.create_date }"
pattern="yyyy-MM-dd"/></td>
            </tr>
        </c:forEach>
    </table>
</div>

```

```

        <td><fmt:formatDate value="${acc.last_login_time}" pattern="yyyy-MM-dd"/></td>
        <td class="td_modi">
            <c:choose>
                <c:when test="${acc.status==0 }">
                    <input type="button" value=" 暂停 " class="btn pause" onclick="pause_account(${acc.id});" />
                    <input type="button" value=" 修改 " class="btn_modify" onclick="location.href='toUpdateAccount.do?id=${acc.id}';" />
                    <input type="button" value=" 删 除 " class="btn_delete" onclick="deleteAccount();;" />
                </c:when>
                <c:when test="${acc.status==1 }">
                    <input type="button" value=" 开通 " class="btn_start" onclick="setState();;" />
                    <input type="button" value=" 修改 " class="btn_modify" onclick="location.href='toUpdateAccount.do?id=${acc.id}';" />
                    <input type="button" value=" 删 除 " class="btn_delete" onclick="deleteAccount();;" />
                </c:when>
                <c:otherwise>
                    </c:otherwise>
                </c:choose>
            </td>
        </tr>
    </c:forEach>
</table>
<p>业务说明 :<br />
1、创建则开通，记载创建时间；<br />
2、暂停后，记载暂停时间；<br />
3、重新开通后，删除暂停时间；<br />
4、删除后，记载删除时间，标示为删除，不能再开通、修改、删除；<br />
5、暂停帐务账号，同时暂停下属的所有业务账号；<br />
6、暂停后重新开通帐务账号，并不同时开启下属的所有业务账号，需要在业务账号管理中单独开启；<br />
7、删除帐务账号，同时删除下属的所有业务账号。</p>
</div>
<!--分页-->
<div id="pages">
    <a href="javascript:to_page(1);">首页</a>
    <c:choose>
        <c:when test="${accountPage.currentPage==1 }">
            <a href="#">上一页</a>
        </c:when>
        <c:otherwise>
            <a href="javascript:to_page(${accountPage.currentPage-1});">上一页</a>
        </c:otherwise>
    </c:choose>
    <c:forEach begin="1" end="${accountPage.totalPage }" var="p">
        <c:choose>
            <c:when test="${p==accountPage.currentPage }">
                <a href="javascript:to_page(${p});">

```

```
class="current_page">>${p }</a>
    </c:when>
    <c:otherwise>
        <a href="javascript:to_page(${p });">${p }</a>
    </c:otherwise>
</c:choose>
</c:forEach>

<c:choose>
    <c:when
test="${accountPage.currentPage==accountPage.totalPage }">
        <a href="#">下一页</a>
    </c:when>
    <c:otherwise>
        <a
href="javascript:to_page(${accountPage.currentPage+1 });">下一页</a>
    </c:otherwise>
</c:choose>
<a href="javascript:to_page(${accountPage.totalPage });">
末页</a>
</div>
</form>
</div>
<!--主要区域结束-->
<div id="footer">
    <p>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目]</p>
    <p>版权所有(C)加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>
```

7. 部署项目，打开浏览器访问 URL 地址：

http://localhost:8080/NETCTOSS/account/findAccount.do

出现如图-13 所示页面。



图 - 13

点击第“2”页，进入第二页的数据，界面效果如图-14所示。

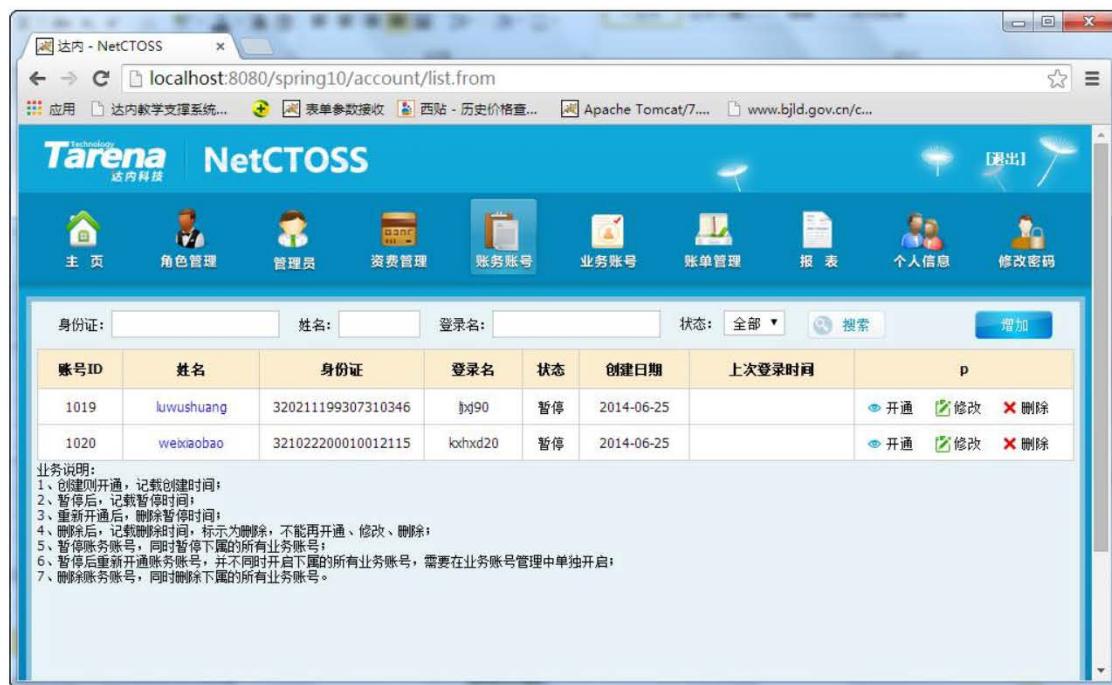


图 - 14

可以根据条件进行查询，例如输入身份证“320211199307310346”点击“搜索”按钮，界面如图-15所示。

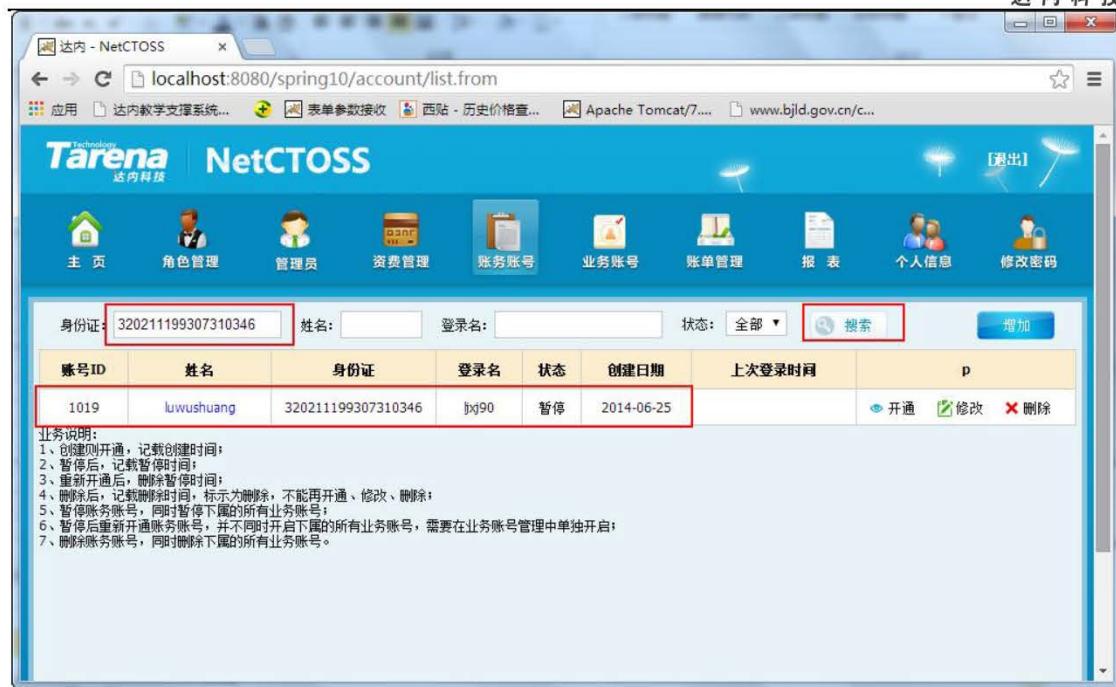


图 - 15

步骤二：更新帐务账号

修改映射文件 AccountMapper.xml，加入如图-16 所示的代码。

```

<select id="findById"
    parameterType="java.lang.Integer"
    resultType="com.tarena.entity.Account">
    select * from account where id=#{id}
</select>

<update id="update" parameterType="com.tarena.entity.Account">
<![CDATA[
    update account set
        real_name=#{real_name,jdbcType=VARCHAR},
        telephone=#{telephone,jdbcType=VARCHAR},
        email=#{email,jdbcType=VARCHAR},
        occupation=#{occupation,jdbcType=VARCHAR},
        gender=#{gender,jdbcType=CHAR},
        mailaddress=#{mailaddress,jdbcType=VARCHAR},
        zipcode=#{zipcode,jdbcType=CHAR},
        qq=#{qq,jdbcType=VARCHAR}
    where id=#{id}
]]>
</update>

```

图 - 16

2. 修改 DAO 接口 AccountDao，加入如下图-17 所示的代码。

```

Account findById(int id);

void update(Account account);

```

图 - 17

3. 修改控制类 AccountController , 加入图-18 所示的代码。

```

@RequestMapping("/toUpdateAccount.do")
public String toUpdate(@RequestParam("id") int id, Model model) {
    Account a = accountDao.findById(id);
    model.addAttribute("account", a);
    return "account/update_account";
}

@RequestMapping("/updateAccount.do")
public String update(Account account) {
    accountDao.update(account);
    return "redirect:findAccount.do";
}

```

图 - 18

4. 创建 update_account.jsp 页面 , 代码如下所示 :

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>达内 - NetCTOSS</title>
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
        <script language="javascript" type="text/javascript">
            //保存成功的提示信息
            function showResult() {
                showResultDiv(true);
                window.setTimeout("showResultDiv(false);", 3000);
            }
            function showResultDiv(flag) {
                var divResult = document.getElementById("save_result_info");
                if (flag)
                    divResult.style.display = "block";
                else
                    divResult.style.display = "none";
            }

            //显示修改密码的信息项
            function showPwd(chkObj) {
                if (chkObj.checked)
                    document.getElementById("divPwds").style.display =
"block";
                else
                    document.getElementById("divPwds").style.display = "none";
            }
        </script>
    </head>
    <body>
        <!--Logo 区域开始-->
        <div id="header">
            

```

```
<a href="#">[退出]</a>
</div>
<!--Logo 区域结束-->
<!--导航区域开始-->
<div id="navi">
    <jsp:include page="/WEB-INF/main/menu.jsp" />
    <!-- <ul id="menu">
        <li><a href="../index.html" class="index_off"></a></li>
        <li><a href="..>
```

```

        <input id="chkModiPwd" type="checkbox"
onclick="showPwd(this);"/>
        <label for="chkModiPwd">修改密码</label>
    </div>
-->
</div>
<!--修改密码部分-->
<!--
<div id="divPwds">
    <div class="text_info clearfix"><span>旧密码 :</span></div>
</div>
    <div class="input_info">
        <input type="password" />
        <span class="required">*</span>
        <div class="validate_msg_long">30 长度以内的字母、数字和下划线的组合</div>
    </div>
    <div class="text_info clearfix"><span>新密码 :</span></div>
    <div class="input_info">
        <input type="password" />
        <span class="required">*</span>
        <div class="validate_msg_long">30 长度以内的字母、数字和下划线的组合</div>
    </div>
    <div class="text_info clearfix"><span>重复新密码 :</span></div>
    <div class="input_info">
        <input type="password" />
        <span class="required">*</span>
        <div class="validate_msg_long">两次密码必须相同</div>
    </div>
-->
<div class="text_info clearfix"><span>电话 :</span></div>
<div class="input_info">
    <input type="text" name="telephone" class="width200" value="${account.telephone}"/>
    <span class="required">*</span>
    <div class="validate_msg_medium error_msg">正确的电话号码格式：手机或固话</div>
</div>
<div class="text_info clearfix"><span>推荐人身份证号码 :</span></div>
<div class="input_info">
    <input type="text" />
    <div class="validate_msg_long error_msgs">正确的身份证号码格式</div>
</div>
<div class="text_info clearfix"><span>生日 :</span></div>
<div class="input_info">
    <input type="text" name="birthdate" value="${account.birthdate}" readonly class="readonly" />
</div>
<div class="text_info clearfix"><span>Email :</span></div>
<div class="input_info">
    <input type="text" name="email" class="width200" value="${account.email}"/>

```

```

        <div class="validate_msg_medium">50 长度以内，合法的 Email
格式</div>
        </div>
        <div class="text_info clearfix"><span>职业 :</span></div>
        <div class="input_info">
            <select name="occupation">
                <option value="">全部</option>
                <option value="1" >干部</option>
                <c:if test="${account.occupation==1 }">selected</c:if>
                <option value="2" >学生</option>
                <c:if test="${account.occupation==2 }">selected</c:if>
                <option value="3" >技术人员</option>
                <c:if test="${account.occupation==3 }">selected</c:if>
                <option value="4" >其他</option>
                <c:if test="${account.occupation==4 }">selected</c:if>
            </select>
        </div>
        <div class="text_info clearfix"><span>性别 :</span></div>
        <div class="input_info fee_type">
            <input type="radio" name="gender" value="1" >
            <c:if test="${account.gender==1 }">checked</c:if>
            <input type="radio" name="gender" value="0" >
            <c:if test="${account.gender==0 }">checked</c:if>
            <label for="female">女</label>
            <label for="male">男</label>
        </div>
        <div class="text_info clearfix"><span>通 信 地 址 :</span></div>
        <div class="input_info">
            <input type="text" name="mailaddress" class="width350" value="${account.mailaddress }"/>
            <div class="validate_msg_tiny">50 长度以内</div>
        </div>
        <div class="text_info clearfix"><span>邮编 :</span></div>
        <div class="input_info">
            <input type="text" name="zipcode" value="${account.zipcode }"/>
            <div class="validate_msg_long">6 位数字</div>
        </div>
        <div class="text_info clearfix"><span>QQ :</span></div>
        <div class="input_info">
            <input type="text" name="qq" value="${account.qq }"/>
            <div class="validate_msg_long">5 到 13 位数字</div>
        </div>
        <!--操作按钮-->
        <div class="button_info clearfix">
            <input type="submit" value="保存" class="btn_save" />
            <input type="button" value="取消" class="btn_save" onclick="history.back();"/>
        </div>
    </form>
</div>
<!--主要区域结束-->
<div id="footer">
    <span>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项

```

```
目]</span>
<br />
<span>版权所有 (C) 加拿大达内 IT 培训集团公司 </span>
</div>
</body>
</html>
```

5. 将 NETCTOSS 工程部署到 tomcat 下，并启动服务器。打开浏览器访问 URL 地址：

<http://localhost:8080/NETCTOSS/account/findAccount.do>

出现如图-19 所示页面。

账号ID	姓名	身份证	登录名	状态	创建日期	上次登录时间
1005	zhangsanfeng	410381194302256528	taij001	开通	2014-06-25	
1010	guojing	330682196903190613	xl18z60	开通	2014-06-25	
1011	huangrong	330902197108270429	dgbf70	暂停	2014-06-25	
1015	zhangwuli	610121198906041115	mjjzh64	暂停	2014-06-25	
1018	guofurong	35058120020101322	jmdxg00	暂停	2014-06-25	

业务说明：

- 1、创建则开通，记录创建时间；
- 2、暂停后，记录暂停时间；
- 3、重新开通后，删除暂停时间；
- 4、删除后，记录删除时间，标识为删除，不能再开通、修改、删除；
- 5、暂停账号销号，同时暂停下属的所有业务账号；
- 6、暂停后重新开通账号，并不能开启下属的所有业务账号，需要在业务账号管理中单独开启；
- 7、删除账号销号，同时删除下属的所有业务账号。

图 - 19

点击图-19 中的“修改”按钮 进入如图-20 所示的修改页面。修改姓名为“shiyuanli”。



The screenshot shows a web page titled "NetCTOSS" with a blue header. The main content is a form for updating an account. The fields include:

- 账务账号ID: 1005
- 姓名: shiyuanli (highlighted in yellow) - Error message: 20长度以内的汉字、字母和数字的组合
- 身份证: 410381194302256528
- 登录账号: taiji001
- 电话: 13669351234 (highlighted in yellow) - Error message: 正确的电话号码格式: 手机或固话
- 推荐人身份证号码: (empty)
- 生日: (empty)
- Email: (empty) - Error message: 50长度以内, 合法的 Email 格式
- 职业: 全部

图 - 20

点击“保存”按钮，返回到列表页面，如图-21所示。



The screenshot shows a table of accounts. The columns are:

账号ID	姓名	身份证	登录名	状态	创建日期	上次登录时间	操作
1005	shiyuanli	410381194302256528	taiji001	暂停	2014-06-25		开通 修改 删除
1010	guojing	330682196903190613	xl18z60	开通	2014-06-25		暂停 修改 删除
1011	huangrong	330902197108270429	dgbf70	暂停	2014-06-25		开通 修改 删除
1015	zhangwuji	610121198906041115	mjzh64	暂停	2014-06-25		开通 修改 删除
1018	guofurong	350581200201010322	jmdxj00	暂停	2014-06-25		开通 修改 删除

业务说明:

- 1、创建则开通，记录创建时间；
- 2、暂停后，记录暂停时间；
- 3、开通后，记录开通时间；
- 4、删除后，记录删除时间，表示为删除，不能再开通、修改、删除；
- 5、暂停账号密码，同时暂停下属的所有业务账号；
- 6、暂停后重新开通账号后，并不能打开下属的所有业务账号，需要在业务账号管理中单独开启；
- 7、删除账号密码，同时删除下属的所有业务账号。

图 - 21

由图-21可以看出修改帐务账号的姓名成功。

步骤三：实现帐务账号的添加

1. 修改映射文件 AccountMapper.xml，加入如图-22所示的代码。

```

<insert id="save" parameterType="com.tarena.entity.Account">
<! [CDATA[
    insert into account values(
        account_seq.nextval,
        #{recommender_id,jdbcType=NUMERIC},
        #{login_name,jdbcType=VARCHAR},
        #{login_passwd,jdbcType=VARCHAR},
        #{status,jdbcType=CHAR},
        #{create_date,jdbcType=TIMESTAMP},
        #{pause_date,jdbcType=TIMESTAMP},
        #{close_date,jdbcType=TIMESTAMP},
        #{real_name,jdbcType=VARCHAR},
        #{idcard_no,jdbcType=VARCHAR},
        #{birthdate,jdbcType=DATE},
        #{gender,jdbcType=CHAR},
        #{occupation,jdbcType=VARCHAR},
        #{telephone,jdbcType=VARCHAR},
        #{email,jdbcType=VARCHAR},
        #{mailaddress,jdbcType=VARCHAR},
        #{zipcode,jdbcType=CHAR},
        #{qq,jdbcType=VARCHAR},
        #{last_login_time,jdbcType=TIMESTAMP},
        #{last_login_ip,jdbcType=VARCHAR}
    )
]]>
</insert>

```

图 - 22

2. 修改 DAO 接口 AccountDao , 加入如图-23 的代码 :

```
void save(Account account);
```

图 - 23

3. 修改控制类 AccountController , 加入图-24 所示的代码 :

```

@RequestMapping("/addAccount.do")
public String add(Account account) {
    account.setStatus("0");
    account.setCreate_date(
        new Timestamp(System.currentTimeMillis()));
    accountDao.save(account);
    return "redirect:findAccount.do";
}

```

图 - 24

4. 创建 add_account.jsp 页面 , 代码如下所示 :

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

```

<title>达内 - NetCTOSS</title>
<link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
<link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
<script language="javascript" type="text/javascript"
src="../js/jquery-1.11.1.js"></script>
<script language="javascript" type="text/javascript">

//校验身份证证
function check_idcardno(idcardno) {
    //重置生日
    $("#birthdate").val("");

    //如果身份证为空，则给出提示
    if(idcardno == "") {
        $("#idcardno_msg").text(" 身份证号不能为空！");
        .addClass("error_msg");
        return;
    }

    //如果身份证格式不对，则给出提示
    var reg = /^[1-9]\d{5}[1-9]\d{3}((0\d)|(1[0-2]))(([0|1|2]\d)|3[0-1])\d{4}$/;
    if(!reg.test(idcardno)){
        $("#idcardno_msg").text(" 身份证号格式不正确！");
        .addClass("error_msg");
        return;
    }

    //校验通过，给予提示，并移除错误样式
    $("#idcardno_msg").text(" 有效的身份证号");
    .removeClass("error_msg");

    //根据身份证提取出生日，赋值给生日字段
    var birthdate = idcardno.substring(6,10) + "—" +
idcardno.substring(10,12) + "-" + idcardno.substring(12,14);
    $("#birthdate").val(birthdate);
}

//保存成功的提示信息
function showResult() {
    showResultDiv(true);
    window.setTimeout("showResultDiv(false);", 3000);
}
function showResultDiv(flag) {
    var divResult = document.getElementById("save_result_info");
    if (flag)
        divResult.style.display = "block";
    else
        divResult.style.display = "none";
}

//显示选填的信息项
function showOptionalInfo(imgObj) {
    var div = document.getElementById("optionalInfo");
    if (div.className == "hide") {
        div.className = "show";
        imgObj.src = "../images/hide.png";
    }
    else {
        div.className = "hide";
        imgObj.src = "../images/show.png";
    }
}

```

```

        }
    }
</script>
</head>
<body>

<!--Logo 区域开始-->
<div id="header">
    
    <a href="#">[退出]</a>
</div>
<!--Logo 区域结束-->
<!--导航区域开始-->
<div id="navi">
    <jsp:include page="/WEB-INF/main/menu.jsp" />
    <!-- <ul id="menu">
        <li><a href="../index.html" class="index_off"></a></li>
        <li><a href="../role/role_list.html" class="role off"></a></li>
        <li><a href="../admin/admin_list.html" class="admin off"></a></li>
        <li><a href="../fee/fee_list.html" class="fee off"></a></li>
        <li><a href="../account/account_list.html" class="account_on"></a></li>
        <li><a href="../service/service_list.html" class="service off"></a></li>
        <li><a href="../bill/bill_list.html" class="bill off"></a></li>
        <li><a href="../report/report_list.html" class="report off"></a></li>
        <li><a href="../user/user_info.html" class="information off"></a></li>
        <li><a href="../user/user_modi_pwd.html" class="password off"></a></li>
    </ul> -->
</div>
<!--导航区域结束-->
<!--主要区域开始-->
<div id="main">
    <!--保存成功或者失败的提示消息-->
    <div id="save_result_info" class="save_fail">保存失败，该身份证已经开通通过账务账号！</div>
    <form action="addAccount.do" method="post" class="main_form">
        <!--必填项-->
        <div class="text_info clearfix"><span>姓名 :</span></div>
        <div class="input_info">
            <input type="text" name="real_name" />
            <span class="required">*</span>
            <div class="validate_msg_long" id="real_name_msg">20 长度以内的汉字、字母和数字的组合</div>
        </div>
        <div class="text_info clearfix"><span>身份证 :</span></div>
        <div class="input_info">
            <input type="text" name="idcard_no" onblur="check_idcardno(this.value);"/>
            <span class="required">*</span>
            <div class="validate_msg_long" id="idcardno_msg">正确的身份证号码格式</div>
        </div>
        <div class="text_info clearfix"><span>登录账号 :</span></div>
    </form>
</div>

```

```

<div class="input_info">
    <input type="text" name="login_name" />
    <span class="required">*</span>
    <div class="validate_msg_long">30 长度以内的字母、数字和下划线
的组合</div>
</div>

<div class="text_info clearfix"><span>密码 :</span></div>
<div class="input_info">
    <input type="password" name="login_passwd" />
    <span class="required">*</span>
    <div class="validate_msg_long">30 长度以内的字母、数字和下划线
的组合</div>
</div>

<div class="text_info clearfix"><span>重复密码 :</span></div>
<div class="input_info">
    <input type="password" />
    <span class="required">*</span>
    <div class="validate_msg_long">两次密码必须相同</div>
</div>

<div class="text_info clearfix"><span>电话 :</span></div>
<div class="input_info">
    <input type="text" class="width200" name="telephone"/>
    <span class="required">*</span>
    <div class="validate_msg_medium">正确的电话号码格式 : 手机或固
话</div>
</div>

<!--可选项-->

<div class="text_info clearfix"><span>可选项 :</span></div>
<div class="input_info">
    
    <div id="optionalInfo" class="hide">
        <div class="text_info clearfix"><span>推荐人身份证号码 :</span></div>
        <div class="input_info">
            <input type="text"/>
            <div class="validate_msg_long">正确的身份证号码格式</div>
        </div>
        <div class="text_info clearfix"><span>生日 :</span></div>
        <div class="input_info">
            <input type="text" name="birthdate" readonly
class="readonly" id="birthdate"/>
        </div>
        <div class="text_info clearfix"><span>Email :</span></div>
        <div class="input_info">
            <input type="text" class="width350" name="email"/>
            <div class="validate_msg_tiny">50 长度以内,合法的 Email 格
式</div>
        </div>
        <div class="text_info clearfix"><span>职业 :</span></div>
        <div class="input_info">
            <select name="occupation">
                <option value="">全部</option>
                <option value="1">干部</option>
                <option value="2">学生</option>
            </select>
        </div>
    </div>
</div>

```

```

        <option value="3">技术人员</option>
        <option value="4">其他</option>
    </select>
</div>
<div class="text_info clearfix"><span>性别:</span></div>
<div class="input_info fee type">
    <input type="radio" name="gender" value="1"
checked="checked" id="female" />
    <label for="female">女</label>
    <input type="radio" name="gender" value="0" id="male" />
    <label for="male">男</label>
</div>
<div class="text_info clearfix"><span>通信地址:</span></div>
<div class="input_info">
    <input type="text" class="width350"
name="mailaddress"/>
        <div class="validate_msg_tiny">50 长度以内</div>
</div>
<div class="text_info clearfix"><span>邮编:</span></div>
<div class="input_info">
    <input type="text" name="zipcode"/>
        <div class="validate_msg_long">6 位数字</div>
</div>
<div class="text_info clearfix"><span>QQ:</span></div>
<div class="input_info">
    <input type="text" name="qq"/>
        <div class="validate_msg_long">5 到 13 位数字</div>
</div>
</div>
<!--操作按钮-->
<div class="button_info clearfix">
    <input type="submit" value="保存" class="btn_save" />
    <input type="button" value="取消" class="btn_save"
onclick="history.back();"/>
</div>
</form>
</div>
<!--主要区域结束-->
<div id="footer">
    <span>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项
目]</span>
    <br />
    <span>版权所有(C)加拿大达内 IT 培训集团公司 </span>
</div>
</body>
</html>

```

5. 将 NETCTOSS 工程部署到 tomcat 下，并启动服务器。打开浏览器访问 URL 地址：

<http://localhost:8080/NETCTOSS/account/findAccount.do>

出现如图-25 所示页面。



业务说明:

- 创建则开通，记载创建时间；
- 暂停后，记载暂停时间；
- 开通后，记载开通时间；
- 删除后，记载删除时间；
- 剩余分，记载剩余时间，标示为删除，不能再开通、修改、删除；
- 暂停账号号，同时暂停下属的所有业务账号；
- 暂停后重新开通账号，并不能开启下属的所有业务账号，需要在业务账号管理中单独开启；
- 删除账号号，同时删除下属的所有业务账号。

图 - 25

点击图-25 页面上“增加”按钮，跳转进入添加页面，填写帐务账号的各种信息，如图-26 所示。



图 - 26

点击图-26 的“保存”按钮，页面跳转到如图-27 所示的列表页面，在列表页面可以发现多了与上述添加页面填入的信息相同的数据，则表示添加资费操作成功。



图 - 27

步骤四：实现帐务账号的暂停

1. 修改映射文件 AccountMapper.xml , 加入如图-28 所示的代码。

```
<update id="updateStatus" parameterType="com.tarena.entity.Account">
    update account set status=#{status},
    <choose>
        <when test="status==0">
            pause_date=null
        </when>
        <when test="status==1">
            pause_date=SYSDATE
        </when>
        <otherwise>
            close_date=SYSDATE
        </otherwise>
    </choose>
    where id=#{id}
</update>
```

图 - 28

2. 修改 DAO 接口 AccountDao , 加入如图-29 的代码 :

```
void updateStatus(Account account);
```

图 - 29

3. com.tarena.entity 包下添加 ServiceMapper.xml 文件 , 代码如下所示 :

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN"
```

```

"http://ibatis.apache.org/dtd/ibatis-3-mapper.dtd">

<mapper namespace="com.tarena.dao.ServiceDao">
    <update id="pauseByAccount" parameterType="java.lang.Integer">
        update service set status='1', pause_date=SYSDATE
        where account_id=#{accountId}
    </update>
</mapper>

```

4. 在 com.tarena.dao 包下添加 ServiceDao，在该 DAO 中添加方法 pauseByAccount 方法，该方法用于暂停帐务账号时，暂停该帐务账号下的所有业务，代码如下所示：

```

package com.tarena.dao;

@MyBatisRepository
public interface ServiceDao {

    void pauseByAccount(int accountId);
}

```

5. 修改控制类 AccountController，加入如下突出显示的代码：

```

package com.tarena.controller;

import java.sql.Timestamp;
import java.util.List;
import javax.annotation.Resource;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.SessionAttributes;
import com.tarena.dao.AccountDao;

import com.tarena.dao.ServiceDao;

import com.tarena.entity.Account;
import com.tarena.entity.AccountPage;

@Controller
@RequestMapping("/account")
@SessionAttributes("accountPage")
public class AccountController extends BaseController {

    @Resource
    private AccountDao accountDao;

    @Resource
    private ServiceDao serviceDao;

    @RequestMapping("/findAccount.do")
    public String find(AccountPage page, Model model) {
        List<Account> list = accountDao.findByPage(page);
        model.addAttribute("accounts", list);

        page.setRows(accountDao.findRows(page));
        model.addAttribute("accountPage", page);
    }
}

```

```
        return "account/account list";
    }

    @RequestMapping("/pauseAccount.do")
    public String pause(@RequestParam("id") int id) {
        //暂停帐务账号
        Account a = new Account();
        a.setId(id);
        a.setStatus("1");
        accountDao.updateStatus(a);

        //暂停其下属的业务账号
        serviceDao.pauseByAccount(id);
        return "redirect:findAccount.do";
    }

    @RequestMapping("/toUpdateAccount.do")
    public String toUpdate(@RequestParam("id") int id, Model model) {
        ...
    }

    @RequestMapping("/updateAccount.do")
    public String update(Account account) {
        ...
    }

    @RequestMapping("/toAddAccount.do")
    public String toAdd() {
        ...
    }

    @RequestMapping("/addAccount.do")
    public String add(Account account) {
        ...
    }
}
```

6. 将 NETCTOSS 工程部署到 tomcat 下，并启动服务器。打开浏览器访问 URL 地址：

<http://localhost:8080/NETCTOSS/account/findAccount.do>

出现如图-30 所示页面。



图 - 30

将上一步添加的账号 ID 为“21”的帐务账号进行暂停。点击图-30 页面上账号 ID 为“21”行的“暂停”按钮后，重新显示该页面，如图-32 所示。



图 - 31

由图-31 可以看出，账号 ID 为“21”的帐务账号状态发生了变化。

- 完整代码

Account 类的完整代码如下所示：

```
package com.tarena.entity;

import java.sql.Date;
import java.sql.Timestamp;

public class Account {

    private Integer id;
    private Integer recommender_id;
    private String login_name;
    private String login_passwd;
    private String status;
    private Timestamp create_date;
    private Timestamp pause_date;
    private Timestamp close_date;
    private String real_name;
    private String idcard_no;
    private Date birthdate;
    private String gender;
    private String occupation;
    private String telephone;
    private String email;
    private String mailaddress;
    private String zipcode;
    private String qq;
    private Timestamp last_login_time;
    private String last_login_ip;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public Integer getRecommender_id() {
        return recommender_id;
    }
    public void setRecommender_id(Integer recommender_id) {
        this.recommender_id = recommender_id;
    }
    public String getLogin_name() {
        return login_name;
    }
    public void setLogin_name(String login_name) {
        this.login_name = login_name;
    }
    public String getLogin_passwd() {
        return login_passwd;
    }
    public void setLogin_passwd(String login_passwd) {
        this.login_passwd = login_passwd;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public Timestamp getCreate_date() {
        return create_date;
    }
    public void setCreate_date(Timestamp create_date) {
        this.create_date = create_date;
    }
}
```

```
public Timestamp getPause_date() {
    return pause_date;
}
public void setPause_date(Timestamp pause_date) {
    this.pause_date = pause_date;
}
public Timestamp getClose_date() {
    return close_date;
}
public void setClose_date(Timestamp close_date) {
    this.close_date = close_date;
}
public String getReal_name() {
    return real_name;
}
public void setReal_name(String real_name) {
    this.real_name = real_name;
}
public String getIdcard_no() {
    return idcard_no;
}
public void setIdcard_no(String idcard_no) {
    this.idcard_no = idcard_no;
}
public Date getBirthdate() {
    return birthdate;
}
public void setBirthdate(Date birthdate) {
    this.birthdate = birthdate;
}
public String getGender() {
    return gender;
}
public void setGender(String gender) {
    this.gender = gender;
}
public String getOccupation() {
    return occupation;
}
public void setOccupation(String occupation) {
    this.occupation = occupation;
}
public String getTelephone() {
    return telephone;
}
public void setTelephone(String telephone) {
    this.telephone = telephone;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getMailaddress() {
    return mailaddress;
}
public void setMailaddress(String mailaddress) {
    this.mailaddress = mailaddress;
}
public String getZipcode() {
    return zipcode;
}
public void setZipcode(String zipcode) {
    this.zipcode = zipcode;
}
public String getQq() {
    return qq;
}
```

```

        }
        public void setQq(String qq) {
            this.qq = qq;
        }
        public Timestamp getLast_login_time() {
            return last_login_time;
        }
        public void setLast_login_time(Timestamp last_login_time) {
            this.last_login_time = last_login_time;
        }
        public String getLast_login_ip() {
            return last_login_ip;
        }
        public void setLast_login_ip(String last_login_ip) {
            this.last_login_ip = last_login_ip;
        }
    }
}

```

映射文件 AccountMapper.xml 的完整代码如下所示：

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN"
"http://ibatis.apache.org/dtd/ibatis-3-mapper.dtd">

<mapper namespace="com.tarena.dao.AccountDao">

    <select id="findByPage"
        parameterType="com.tarena.entity.page.Page"
        resultType="com.tarena.entity.Account">
        select * from (
            select a.* , ROWNUM r from (
                select * from account
                <where>
                    <if test="idcard_no!=null && idcard_no.length()>0">
                        and idcard_no=#{idcard_no}
                    </if>
                    <if test="real_name!=null && real_name.length()>0">
                        and real_name=#{real_name}
                    </if>
                    <if test="login_name!=null & login_name.length()>0">
                        and login_name=#{login_name}
                    </if>
                    <if test='status!=null & status!= -1'>
                        and status=#{status}
                    </if>
                </where>
                order by id
            ) a
        ) where r<#{end} and r>#{begin}
    </select>

    <select id="findRows"
        parameterType="com.tarena.entity.page.Page"
        resultType="java.lang.Integer">
        select count(*) from account
        <where>
            <if test="idcard_no!=null && idcard_no.length()>0">
                and idcard_no=#{idcard_no}
            </if>
            <if test="real_name!=null && real_name.length()>0">
                and real_name=#{real_name}
            </if>
            <if test="login_name!=null && login_name.length()>0">
                and login_name=#{login_name}
            </if>
        </where>
    </select>

```

```

        </if>
        <if test='status!=null && !status.equals("-1")'>
            and status=#{status}
        </if>
    </where>
</select>

<update id="updateStatus" parameterType="com.tarena.entity.Account">
    update account set status=#{status},
    <choose>
        <when test="status==0">
            pause date=null
        </when>
        <when test="status==1">
            pause date=SYSDATE
        </when>
        <otherwise>
            close_date=SYSDATE
        </otherwise>
    </choose>
    where id=#{id}
</update>

<select id="findById"
    parameterType="java.lang.Integer"
    resultType="com.tarena.entity.Account">
    select * from account where id=#{id}
</select>

<update id="update" parameterType="com.tarena.entity.Account">
<! [CDATA[
    update account set
        real_name=#{real_name,jdbcType=VARCHAR},
        telephone=#{telephone,jdbcType=VARCHAR},
        email=#{email,jdbcType=VARCHAR},
        occupation=#{occupation,jdbcType=VARCHAR},
        gender=#{gender,jdbcType=CHAR},
        mailaddress=#{mailaddress,jdbcType=VARCHAR},
        zipcode=#{zipcode,jdbcType=CHAR},
        qq=#{qq,jdbcType=VARCHAR}
    where id=#{id}
]]>
</update>

<insert id="save" parameterType="com.tarena.entity.Account">
<! [CDATA[
    insert into account values(
        account_seq.nextval,
        #{recommender_id,jdbcType=NUMERIC},
        #{login_name,jdbcType=VARCHAR},
        #{login_passwd,jdbcType=VARCHAR},
        #{status,jdbcType=CHAR},
        #{create_date,jdbcType=TIMESTAMP},
        #{pause_date,jdbcType=TIMESTAMP},
        #{close_date,jdbcType=TIMESTAMP},
        #{real_name,jdbcType=VARCHAR},
        #{idcard_no,jdbcType=VARCHAR},
        #{birthdate,jdbcType=DATE},
        #{gender,jdbcType=CHAR},
        #{occupation,jdbcType=VARCHAR},
        #{telephone,jdbcType=VARCHAR},
        #{email,jdbcType=VARCHAR},
        #{mailaddress,jdbcType=VARCHAR},
        #{zipcode,jdbcType=CHAR},
        #{qq,jdbcType=VARCHAR},
        #{last_login_time,jdbcType=TIMESTAMP},
        #{last_login_ip,jdbcType=VARCHAR}
    )
]]>

```

```
]]>
</insert>

</mapper>
```

Page 类的完整代码如下所示：

```
package com.tarena.entity.page;

public abstract class Page {

    // 分页时输入的参数
    private int currentPage = 1; // 当前页
    private int pageSize = 5; // 每页显示最大行数

    // 写分页 SQL 时需要的条件，由输入参数计算而来
    private int begin; // 当前页的起始行
    private int end; // 当前页的终止行

    // 用于计算总页数
    private int rows; // 总行数

    // 总页数，由 rows 和 pageSize 计算而来
    private int totalPage;

    public int getBegin() {
        begin = (currentPage - 1) * pageSize;
        return begin;
    }

    public int getEnd() {
        end = currentPage * pageSize + 1;
        return end;
    }

    public int getTotalPage() {
        if (rows % pageSize == 0) {
            totalPage = rows / pageSize;
        } else {
            totalPage = rows / pageSize + 1;
        }
        return totalPage;
    }

    public int getCurrentPage() {
        return currentPage;
    }

    public void setCurrentPage(int currentPage) {
        this.currentPage = currentPage;
    }

    public int getPageSize() {
        return pageSize;
    }

    public void setPageSize(int pageSize) {
        this.pageSize = pageSize;
    }

    public int getRows() {
        return rows;
    }
}
```

```
public void setRows(int rows) {
    this.rows = rows;
}
```

AccountPage 类的完整代码如下所示：

```
package com.tarena.entity.page;

public class AccountPage extends Page {

    private String idcard_no;
    private String real_name;
    private String login_name;
    private String status;

    public String getIdcard_no() {
        return idcard_no;
    }

    public void setIdcard_no(String idcard_no) {
        this.idcard_no = idcard_no;
    }

    public String getReal_name() {
        return real_name;
    }

    public void setReal_name(String real_name) {
        this.real_name = real_name;
    }

    public String getLogin_name() {
        return login_name;
    }

    public void setLogin_name(String login_name) {
        this.login_name = login_name;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }
}
```

接口 AccountDao 的完整代码如下所示：

```
package com.tarena.dao;

import java.util.List;

import com.tarena.annotation.MyBatisRepository;
import com.tarena.entity.Account;
import com.tarena.entity.page.Page;

@MyBatisRepository
public interface AccountDao {

    List<Account> findByPage(Page page);
}
```

```
int findRows(Page page);

void updateStatus(Account account);

Account findById(int id);

void update(Account account);

void save(Account account);

}
```

控制类 AccountController 的完整代码如下所示：

```
package com.tarena.controller;

import java.sql.Timestamp;
import java.util.List;
import javax.annotation.Resource;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.SessionAttributes;
import com.tarena.dao.AccountDao;
import com.tarena.dao.ServiceDao;
import com.tarena.entity.Account;
import com.tarena.entity.page.AccountPage;

@Controller
@RequestMapping("/account")
@SessionAttributes("accountPage")
public class AccountController extends BaseController {

    @Resource
    private AccountDao accountDao;

    @Resource
    private ServiceDao serviceDao;

    @RequestMapping("/findAccount.do")
    public String find(AccountPage page, Model model) {
        List<Account> list = accountDao.findByPage(page);
        model.addAttribute("accounts", list);

        page.setRows(accountDao.findRows(page));
        model.addAttribute("accountPage", page);

        return "account/account list";
    }

    @RequestMapping("/pauseAccount.do")
    public String pause(@RequestParam("id") int id) {
        //暂停帐务账号
        Account a = new Account();
        a.setId(id);
        a.setStatus("1");
        accountDao.updateStatus(a);

        //暂停其下属的业务账号
        serviceDao.pauseByAccount(id);
        return "redirect:findAccount.do";
    }

    @RequestMapping("/toUpdateAccount.do")
```

```

public String toUpdate(@RequestParam("id") int id, Model model) {
    Account a = accountDao.findById(id);
    model.addAttribute("account", a);
    return "account/update_account";
}

@RequestMapping("/updateAccount.do")
public String update(Account account) {
    accountDao.update(account);
    return "redirect:findAccount.do";
}

@RequestMapping("/toAddAccount.do")
public String toAdd() {
    return "account/add account";
}

@RequestMapping("/addAccount.do")
public String add(Account account) {
    account.setStatus("0");
    account.setCreate_date(
        new Timestamp(System.currentTimeMillis()));
    accountDao.save(account);
    return "redirect:findAccount.do";
}
}

```

映射文件 ServiceMapper.xml 完整代码如下所示：

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN"
"http://ibatis.apache.org/dtd/ibatis-3-mapper.dtd">

<mapper namespace="com.tarena.dao.ServiceDao">
    <update id="pauseByAccount" parameterType="java.lang.Integer">
        update service set status='1', pause date=SYSDATE
        where account id=#{accountId}
    </update>
</mapper>

```

ServiceDao 类的完整代码如下所示：

```

package com.tarena.dao;

import java.util.List;
import java.util.Map;

import com.tarena.annotation.MyBatisRepository;
import com.tarena.entity.page.Page;

@MyBatisRepository
public interface ServiceDao {

    void pauseByAccount(int accountId);

}

```

页面 account_list.jsp 的完整代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

```

```

<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>达内—NetCTOSS</title>
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
        <script language="javascript" type="text/javascript">
            //删除
            function deleteAccount() {
                var r = window.confirm("确定要删除此帐务账号吗? \r\n 删除后将不能恢
复, 且会删除其下属的所有业务账号。");
                document.getElementById("operate result info").style.display
= "block";
            }
            //开通或暂停
            function setState() {
                var r = window.confirm("确定要开通此帐务账号吗? ");
                document.getElementById("operate_result_info").style.display
= "block";
            }
            //翻页
            function toPage(page) {
                document.getElementById("current page").value = page;
                document.forms[0].submit();
            }
            //暂停
            function pause_account(id) {
                var r = window.confirm("确定要暂停此帐务账号吗? ");
                if(r) {
                    window.location.href = "pauseAccount.do?id="+id;
                }
            }
        </script>
    </head>
    <body>
        <!--Logo 区域开始-->
        <div id="header">
            
            <a href="#">[退出]</a>
        </div>
        <!--Logo 区域结束-->
        <!--导航区域开始-->
        <div id="navi">
            <jsp:include page="/WEB-INF/main/menu.jsp" />
            <!-- <ul id="menu">
                <li><a href="../index.html" class="index off"></a></li>
                <li><a href="../role/role_list.html"
class="role off"></a></li>
                <li><a href="../admin/admin_list.html"
class="admin_off"></a></li>
                <li><a href="../fee/fee_list.html" class="fee_off"></a></li>
                <li><a href="../account/account_list.html"
class="account on"></a></li>
                <li><a href="../service/service_list.html"
class="service_off"></a></li>
                <li><a href="../bill/bill_list.html"
class="bill_off"></a></li>
                <li><a href="../report/report_list.html"
class="report_off"></a></li>
                <li><a href="../user/user_info.html"
class="information_off"></a></li>
            </ul>
        </div>
    </body>

```

```

<li><a href=".../user/user_modi_pwd.html"
class="password off"></a></li>
</ul> -->
</div>
<!--导航区域结束-->
<!--主要区域开始-->
<div id="main">
    <form action="findAccount.do" method="post">
        <input type="hidden" name="currentPage" id="current_page"
value="${accountPage.currentPage }"/>
        <!--查询-->
        <div class="search add">
            <div>身份证 : <input type="text" class="text search"
name="idcard_no" value="${accountPage.idcard_no }" /></div>
            <div>姓名: <input type="text" class="width70 text_search"
name="real_name" value="${accountPage.real_name }" /></div>
            <div>登录名 : <input type="text" class="text_search"
name="login_name" value="${accountPage.login_name }" /></div>
            <div>
                状态:
                <select class="select_search" name="status">
                    <option value="-1">全部</option>
                    <option value="0" <c:if
test="${accountPage.status==0 }">selected</c:if>>开通</option>
                    <option value="1" <c:if
test="${accountPage.status==1 }">selected</c:if>>暂停</option>
                    <option value="2" <c:if
test="${accountPage.status==2 }">selected</c:if>>删除</option>
                </select>
            </div>
            <div><input type="button" value="搜索" class="btn_search"
onclick="to_page(1);"/></div>
            <input type="button" value="增加" class="btn_add"
onclick="location.href='toAddAccount.do';" />
        <!--删除等的操作提示-->
        <div id="operate_result_info" class="operate_success">
            
            删除成功，且已删除其下属的业务账号！
        </div>
        <!--数据区域: 用表格展示数据-->
        <div id="data">
            <table id="datalist">
                <tr>
                    <th>账号 ID</th>
                    <th>姓名</th>
                    <th class="width150">身份证</th>
                    <th>登录名</th>
                    <th>状态</th>
                    <th class="width100">创建日期</th>
                    <th class="width150">上次登录时间</th>
                    <th class="width200"></th>
                </tr>
                <c:forEach items="${accounts }" var="acc">
                    <tr>
                        <td>${acc.id }</td>
                        <td><a href="${account_detail.html }">${acc.real_name }</a></td>
                        <td>${acc.idcard_no }</td>
                        <td>${acc.login_name }</td>
                        <td>
                            <c:choose>
                                <c:when test="${acc.status==0 }">开通</c:when>

```

```

        <c:when test="${acc.status==1 }">暂停</c:when>
        <c:otherwise>删除</c:otherwise>
    </c:choose>
</td>
<td><fmt:formatDate value="${acc.create_date }" pattern="yyyy-MM-dd"/></td>
<td><fmt:formatDate value="${acc.last_login_time }" pattern="yyyy-MM-dd"/></td>
<td class="td modi">
    <c:choose>
        <c:when test="${acc.status==0 }">
            <input type="button" value="暂停" class="btn_pause" onclick="pause_account(${acc.id});" />
            <input type="button" value="修改" class="btn_modify" onclick="location.href='toUpdateAccount.do?id=${acc.id}';" />
            <input type="button" value="删除" class="btn_delete" onclick="deleteAccount();" />
        </c:when>
        <c:when test="${acc.status==1 }">
            <input type="button" value="开通" class="btn_start" onclick="setState();" />
            <input type="button" value="修改" class="btn_modify" onclick="location.href='toUpdateAccount.do?id=${acc.id}';" />
            <input type="button" value="删除" class="btn_delete" onclick="deleteAccount();" />
        </c:when>
        <c:otherwise>
    </c:otherwise>
</c:choose>
</td>
</tr>
</c:forEach>
</table>
<p>业务说明：<br />
1、创建则开通，记载创建时间；<br />
2、暂停后，记载暂停时间；<br />
3、重新开通后，删除暂停时间；<br />
4、删除后，记载删除时间，标示为删除，不能再开通、修改、删除；<br />
5、暂停帐务账号，同时暂停下属的所有业务账号；<br />
6、暂停后重新开通帐务账号，并不同时开启下属的所有业务账号，需要在业务账号管理中单独开启；<br />
7、删除帐务账号，同时删除下属的所有业务账号。</p>
</div>
<!--分页-->
<div id="pages">
    <a href="javascript:to_page(1);">首页</a>
    <c:choose>
        <c:when test="${accountPage.currentPage==1 }">
            <a href="#">上一页</a>
        </c:when>
        <c:otherwise>
            <a href="javascript:to_page(${accountPage.currentPage-1});">上一页</a>
        </c:otherwise>
    </c:choose>
    <c:forEach begin="1" end="${accountPage.totalPage }" var="p">
        <c:choose>
            <c:when test="${p==accountPage.currentPage }">
                <a href="javascript:to_page(${p});" class="current_page">${p}</a>
            </c:when>
            <c:otherwise>
                <a href="javascript:to_page(${p});" class="normal_page">${p}</a>
            </c:otherwise>
        </c:choose>
    </c:forEach>
</div>

```

```

        </c:when>
        <c:otherwise>
            <a href="javascript:to_page(${p});">${p}</a>
        </c:otherwise>
    </c:choose>
    </c:forEach>

    <c:choose>
        <c:when
test="${accountPage.currentPage==accountPage.totalPage }">
            <a href="#">下一页</a>
        </c:when>
        <c:otherwise>
            <a href="javascript:to_page(${accountPage.currentPage+1});">下一页</a>
        </c:otherwise>
    </c:choose>
    <a href="javascript:to_page(${accountPage.totalPage});">
末页</a>
        </div>
    </form>
</div>
<!--主要区域结束-->
<div id="footer">
    <p>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目]</p>
    <p>版权所有(C)加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>

```

页面 add_account.jsp 的完整代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>达内-NetCTOSS</title>
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
        <script language="javascript" type="text/javascript"
src="../js/jquery-1.11.1.js"></script>
        <script language="javascript" type="text/javascript">
            //校验身份证
            function check_idcardno(idcardno) {
                //重置生日
                $("#birthdate").val("");
                //如果身份证为空，则给出提示
                if(idcardno == ""){
                    $("#idcardno msg").text(" 身份证号不能为空！");
                    .addClass("error msg");
                    return;
                }
                //如果身份证格式不对，则给出提示
                var reg = /^[1-9]\d{5}[1-9]\d{3}((0\d)|(1[0-2]))(([0|1|2]\d)|3[0-1])\d{4}$/;
                if(!reg.test(idcardno)){
                    $("#idcardno_msg").text(" 身份证号格式不正确！");
                }
            }
        </script>
    </head>
    <body>
        <div style="text-align: center; margin-top: 100px;">
            <h1>添加学员信息</h1>
            <form action="add_account" method="post">
                <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;">
                    <tr>
                        <td>学员姓名:</td>
                        <td><input type="text" name="name" value="" /></td>
                    </tr>
                    <tr>
                        <td>学员性别:</td>
                        <td><input checked="" type="radio" name="sex" value="男" />男 <input type="radio" name="sex" value="女" />女</td>
                    </tr>
                    <tr>
                        <td>出生日期:</td>
                        <td><input type="text" name="birthdate" value="" /></td>
                    </tr>
                    <tr>
                        <td>身份证号:</td>
                        <td><input type="text" name="idcardno" value="" /><br/>
                            <small>身份证号格式为：18位数字</small>
                        </td>
                    </tr>
                    <tr>
                        <td>学号:</td>
                        <td><input type="text" name="学号" value="" /></td>
                    </tr>
                    <tr>
                        <td>年龄:</td>
                        <td><input type="text" name="age" value="" /></td>
                    </tr>
                    <tr>
                        <td>性别:</td>
                        <td><input checked="" type="radio" name="gender" value="男" />男 <input type="radio" name="gender" value="女" />女</td>
                    </tr>
                    <tr>
                        <td>入学时间:</td>
                        <td><input type="text" name="enroll_time" value="" /></td>
                    </tr>
                    <tr>
                        <td>专业:</td>
                        <td><input type="text" name="major" value="" /></td>
                    </tr>
                    <tr>
                        <td>状态:</td>
                        <td><input checked="" type="radio" name="status" value="正常" />正常 <input type="radio" name="status" value="已退学" />已退学</td>
                    </tr>
                    <tr>
                        <td colspan="2" style="text-align: center; padding-top: 10px;">
                            <input type="submit" value="提交" /> <input type="button" value="重置" />
                        </td>
                    </tr>
                </table>
            </form>
        </div>
    </body>
</html>

```

```

        ").addClass("error msg");
            return;
        }

        //校验通过，给予提示，并移除错误样式
        $("#idcardno_msg").text(" 有 效 的 身 份 证 号"
").removeClass("error_msg");

        //根据身份证提取出生日，赋值给生日字段
        var birthdate = idcardno.substring(6,10) + "—" +
idcardno.substring(10,12) + "—" + idcardno.substring(12,14);
        $("#birthdate").val(birthdate);
    }

    //保存成功的提示信息
    function showResult() {
        showResultDiv(true);
        window.setTimeout("showResultDiv(false);", 3000);
    }
    function showResultDiv(flag) {
        var divResult = document.getElementById("save_result_info");
        if (flag)
            divResult.style.display = "block";
        else
            divResult.style.display = "none";
    }

    //显示选填的信息项
    function showOptionalInfo(imgObj) {
        var div = document.getElementById("optionalInfo");
        if (div.className == "hide") {
            div.className = "show";
            imgObj.src = "../images/hide.png";
        }
        else {
            div.className = "hide";
            imgObj.src = "../images/show.png";
        }
    }
    </script>
</head>
<body>

<!--Logo 区域开始-->
<div id="header">
    
    <a href="#">[退出]</a>
</div>
<!--Logo 区域结束-->
<!--导航区域开始-->
<div id="navi">
    <jsp:include page="/WEB-INF/main/menu.jsp" />
    <!-- <ul id="menu">
        <li><a href="../index.html" class="index_off"></a></li>
        <li><a href="../role/role_list.html" class="role_off"></a></li>
        <li><a href="../admin/admin_list.html" class="admin_off"></a></li>
        <li><a href="../fee/fee_list.html" class="fee_off"></a></li>
        <li><a href="../account/account_list.html" class="account_on"></a></li>
        <li><a href="../service/service_list.html" class="service_off"></a></li>
        <li><a href="../bill/bill_list.html" class="bill_off"></a></li>
        <li><a href="../report/report_list.html" class="report_off"></a></li>
    </ul>
</div>

```

```

<li><a href=".../user/user_info.html" class="information off"></a></li>
<li><a href=".../user/user_modi_pwd.html" class="password_off"></a></li>
</ul> -->
</div>
<!--导航区域结束-->
<!--主要区域开始-->
<div id="main">
    <!--保存成功或者失败的提示消息-->
    <div id="save_result_info" class="save_fail">保存失败，该身份证已经  
开通过账务账号！</div>
    <form action="addAccount.do" method="post" class="main_form">
        <!--必填项-->
        <div class="text_info clearfix"><span>姓名: </span></div>
        <div class="input_info">
            <input type="text" name="real_name" />
            <span class="required">*</span>
            <div class="validate_msg_long">20 长度以内的汉字、字母和数字的  
组合</div>
        </div>
        <div class="text_info clearfix"><span>身份证号: </span></div>
        <div class="input_info" onblur="check_idcardno(this.value);">
            <input type="text" name="idcard_no" />
            <span class="required">*</span>
            <div class="validate_msg_long" id="idcardno_msg">正确的身份  
证号码格式</div>
        </div>
        <div class="text_info clearfix"><span>登录账号: </span></div>
        <div class="input_info">
            <input type="text" name="login_name" />
            <span class="required">*</span>
            <div class="validate_msg_long">30 长度以内的字母、数字和下划线  
的组合</div>
        </div>
        <div class="text_info clearfix"><span>密码: </span></div>
        <div class="input_info">
            <input type="password" name="login_passwd" />
            <span class="required">*</span>
            <div class="validate_msg_long">30 长度以内的字母、数字和下划线  
的组合</div>
        </div>
        <div class="text_info clearfix"><span>重复密码: </span></div>
        <div class="input_info">
            <input type="password" />
            <span class="required">*</span>
            <div class="validate_msg_long">两次密码必须相同</div>
        </div>
        <div class="text_info clearfix"><span>电话: </span></div>
        <div class="input_info">
            <input type="text" class="width200" name="telephone"/>
            <span class="required">*</span>
            <div class="validate_msg_medium">正确的电话号码格式：手机或固  
话</div>
        </div>
        <!--可选项-->
        <div class="text_info clearfix"><span>可选项: </span></div>
        <div class="input_info" onclick="showOptionalInfo(this);">
            
        </div>
        <div id="optionalInfo" class="hide">
            <div class="text_info clearfix"><span>推荐人身份证号码: </span></div>
        </div>
    </form>
</div>

```

```

</span></div>
    <div class="input_info">
        <input type="text"/>
        <div class="validate_msg_long">正确的身份证号码格式</div>
    </div>
    <div class="text_info clearfix"><span>生日: </span></div>
    <div class="input_info">
        <input type="text" name="birthdate" readonly
class="readonly" id="birthdate"/>
    </div>
    <div class="text_info clearfix"><span>Email: </span></div>
    <div class="input_info">
        <input type="text" class="width350" name="email"/>
        <div class="validate_msg_tiny">50 长度以内,合法的 Email 格
式</div>
    </div>
    <div class="text_info clearfix"><span>职业: </span></div>
    <div class="input_info">
        <select name="occupation">
            <option value="">全部</option>
            <option value="1">干部</option>
            <option value="2">学生</option>
            <option value="3">技术人员</option>
            <option value="4">其他</option>
        </select>
    </div>
    <div class="text_info clearfix"><span>性别: </span></div>
    <div class="input_info fee_type">
        <input type="radio" name="gender" value="1"
checked="checked" id="female" />
        <label for="female">女</label>
        <input type="radio" name="gender" value="0" id="male" />
        <label for="male">男</label>
    </div>
    <div class="text_info clearfix"><span>通 信 地 址 :</span></div>
    <div class="input_info">
        <input type="text" class="width350"
name="mailaddress"/>
        <div class="validate_msg_tiny">50 长度以内</div>
    </div>
    <div class="text_info clearfix"><span>邮编: </span></div>
    <div class="input_info">
        <input type="text" name="zipcode"/>
        <div class="validate_msg_long">6 位数字</div>
    </div>
    <div class="text_info clearfix"><span>QQ: </span></div>
    <div class="input_info">
        <input type="text" name="qq"/>
        <div class="validate_msg_long">5 到 13 位数字</div>
    </div>
</div>
<!--操作按钮-->
<div class="button_info clearfix">
    <input type="submit" value="保存" class="btn_save" />
    <input type="button" value=" 取 消 " class="btn_save"
onclick="history.back();"/>
</div>
</form>
</div>
<!--主要区域结束-->
<div id="footer">
    <span>[源自北美的技术, 最优秀的师资, 最真实的企业环境, 最适用的实战项
目]</span>

```

```

<br />
<span>版权所有(C)加拿大达内IT培训集团公司 </span>
</div>
</body>
</html>

```

页面 update_account.jsp 的完整代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>达内-NetCTOSS</title>
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
        <script language="javascript" type="text/javascript">
            //保存成功的提示信息
            function showResult() {
                showResultDiv(true);
                window.setTimeout("showResultDiv(false);", 3000);
            }
            function showResultDiv(flag) {
                var divResult = document.getElementById("save result info");
                if (flag)
                    divResult.style.display = "block";
                else
                    divResult.style.display = "none";
            }
        //显示修改密码的信息项
        function showPwd(chkObj) {
            if (chkObj.checked)
                document.getElementById("divPwds").style.display =
"block";
            else
                document.getElementById("divPwds").style.display = "none";
        }
    </script>
</head>
<body>
    <!--Logo 区域开始-->
    <div id="header">
        
        <a href="#">[退出]</a>
    </div>
    <!--Logo 区域结束-->
    <!--导航区域开始-->
    <div id="navi">
        <jsp:include page="/WEB-INF/main/menu.jsp" />
        <!-- <ul id="menu">
            <li><a href="../index.html" class="index off"></a></li>
            <li><a href="../role/role_list.html"
class="role_off"></a></li>
            <li><a href="../admin/admin_list.html"
class="admin_off"></a></li>
            <li><a href="../fee/fee_list.html" class="fee off"></a></li>
            <li><a href="../account/account_list.html"
class="account on"></a></li>
            <li><a href="../service/service_list.html"

```

```

class="service off"></a></li>
    <li><a href="..../bill/bill_list.html"
class="bill_off"></a></li>
    <li><a href="..../report/report_list.html"
class="report_off"></a></li>
    <li><a href="..../user/user_info.html"
class="information_off"></a></li>
    <li><a href="..../user/user_modi_pwd.html"
class="password_off"></a></li>
        </ul> -->
    </div>
<!--导航区域结束-->
<!--主要区域开始-->
<div id="main">
    <!--保存成功或者失败的提示消息-->
    <div id="save_result_info" class="save_fail">保存失败，旧密码错误！
</div>
    <form action="updateAccount.do" method="post" class="main_form">
        <!--必填项-->
        <div class="text_info clearfix"><span>帐务账号 ID :</span></div>
        <div class="input_info">
            <input type="text" name="id" value="${account.id}" readonly class="readonly" />
        </div>
        <div class="text_info clearfix"><span>姓名:</span></div>
        <div class="input_info">
            <input type="text" name="real_name" value="${account.real_name}" />
            <span class="required">*</span>
            <div class="validate_msg_long error_msg">20 长度以内的汉字、字母和数字的组合</div>
        </div>
        <div class="text_info clearfix"><span>身份证:</span></div>
        <div class="input_info">
            <input type="text" name="idcard_no" value="${account.idcard_no}" readonly class="readonly" />
        </div>
        <div class="text_info clearfix"><span>登录账号:</span></div>
        <div class="input_info">
            <input type="text" name="login_name" value="${account.login_name}" readonly class="readonly" />
            <!--
            <div class="change_pwd">
                <input id="chkModiPwd" type="checkbox" onclick="showPwd(this);"/>
                <label for="chkModiPwd">修改密码</label>
            </div>
            -->
        </div>
        <!--修改密码部分-->
        <!--
        <div id="divPwds">
            <div class="text_info clearfix"><span>旧密码:</span></div>
            <div class="input_info">
                <input type="password" />
                <span class="required">*</span>
                <div class="validate msg_long">30 长度以内的字母、数字和下划线的组合</div>
            </div>
            <div class="text_info clearfix"><span>新密码:</span></div>
            <div class="input_info">

```

```

        <input type="password" />
        <span class="required">*</span>
        <div class="validate msg long">30 长度以内的字母、数字
和下划线的组合</div>
    </div>
    <div class="text info clearfix"><span>重 复 新 密 码 :</span>
</div></div>
    <div class="input_info">
        <input type="password" />
        <span class="required">*</span>
        <div class="validate msg long">两次密码必须相同</div>
    </div>
    </div>
    -->
    <div class="text_info clearfix"><span>电话:</span></div>
    <div class="input_info">
        <input type="text" name="telephone" class="width200" value="${account.telephone }"/>
        <span class="required">*</span>
        <div class="validate msg medium error msg">正确的电话号码格式: 手机或固话</div>
    </div>
    <div class="text_info clearfix"><span>推荐人身份证号码:</span></div>
    <div class="input_info">
        <input type="text" />
        <div class="validate_msg_long error_msgs">正确的身份证号码格式</div>
    </div>
    <div class="text_info clearfix"><span>生日:</span></div>
    <div class="input_info">
        <input type="text" name="birthdate" value="${account.birthdate }" readonly class="readonly" />
    </div>
    <div class="text_info clearfix"><span>Email:</span></div>
    <div class="input_info">
        <input type="text" name="email" class="width200" value="${account.email }"/>
        <div class="validate_msg_medium">50 长度以内, 合法的 Email 格式</div>
    </div>
    <div class="text_info clearfix"><span>职业:</span></div>
    <div class="input_info">
        <select name="occupation">
            <option value="">全部</option>
            <option value="1" ${test=="${account.occupation}==1 ? 'selected' : ''}">干部</option>
            <option value="2" ${test=="${account.occupation}==2 ? 'selected' : ''}">学生</option>
            <option value="3" ${test=="${account.occupation}==3 ? 'selected' : ''}">技术人员</option>
            <option value="4" ${test=="${account.occupation}==4 ? 'selected' : ''}">其他</option>
        </select>
    </div>
    <div class="text_info clearfix"><span>性别:</span></div>
    <div class="input_info fee_type">
        <input type="radio" name="gender" value="1" ${test=="${account.gender}==1 ? 'checked' : ''} checked="" id="female" onclick="feeTypeChange(1);"/>
        <label for="female">女</label>
        <input type="radio" name="gender" value="0" ${test=="${account.gender}==0 ? 'checked' : ''} checked="" id="male" onclick="feeTypeChange(2);"/>
        <label for="male">男</label>
    </div>

```

```

        <label for="male">男</label>
    </div>
    <div class="text_info clearfix"><span>通 信 地 址 :</span></div>
    <div class="input_info">
        <input type="text" name="mailaddress" class="width350" value="${account.mailaddress }"/>
        <div class="validate_msg_tiny">50 长度以内</div>
    </div>
    <div class="text_info clearfix"><span>邮编:</span></div>
    <div class="input_info">
        <input type="text" name="zipcode" value="${account.zipcode }"/>
        <div class="validate_msg_long">6 位数字</div>
    </div>
    <div class="text_info clearfix"><span>QQ:</span></div>
    <div class="input_info">
        <input type="text" name="qq" value="${account.qq }"/>
        <div class="validate_msg_long">5 到 13 位数字</div>
    </div>
    <!--操作按钮-->
    <div class="button_info clearfix">
        <input type="submit" value="保存" class="btn_save" />
        <input type="button" value="取消" class="btn_save" onclick="history.back();"/>
    </div>
    </form>
</div>
<!--主要区域结束-->
<div id="footer">
    <span>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目]</span>
    <br />
    <span>版权所有 (C) 加拿大达内 IT 培训集团公司 </span>
</div>
</body>
</html>

```

3. 实现业务账号的列表显示以及开通

- **问题**

本案例要求完成 NETCTOSS 项目需求中的业务账号的列表显示以及开通功能。

- **方案**

采用 Spring 和 MyBatis 整合方式实现 ServiceDao 的列表实现功能和开通功能，然后将 ServiceDao 对象注入给 SpringMVC 的 ServiceController 完成查询列表显示操作以及开通操作。本案例的包结构如图-32 所示。

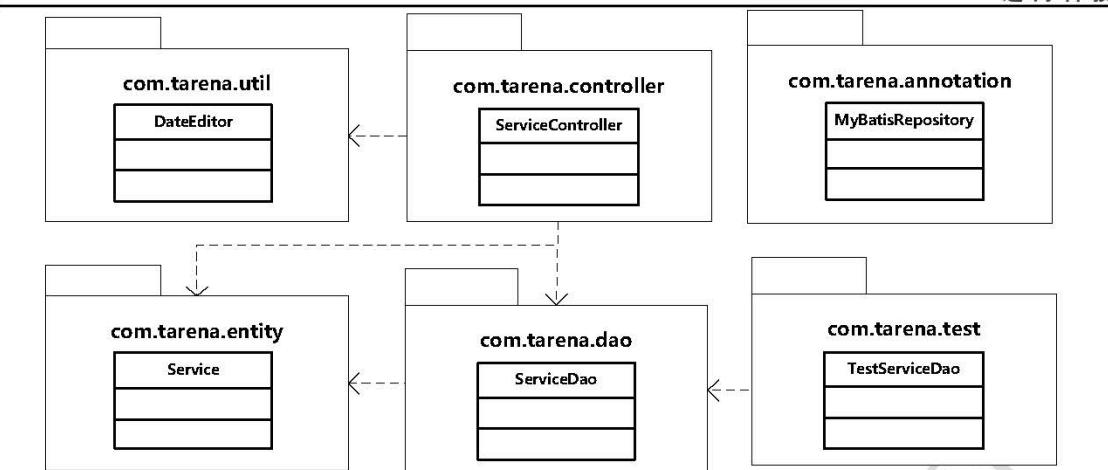


图 - 32

各包下的类含义如下：

1. 包 com.tarena.util 中的 DateEditor 类是处理日期格式的工具类；
2. 包 com.tarena.entity 中的 Service 是数据库表 service 与 Java 对象进行映射的实体类；
3. 包 com.tarena.dao 中的 ServiceDao 包含了操作数据库 service 表中的数据的方法；
4. 包 com.tarena.test 中的 TestServiceDao 是对 ServiceDao 中的方法进行测试的类；
5. 包 com.tarena.controller 中的 ServiceController 是 ServiceDao 和页面之间的协调类；
6. 包 com.tarena.annotation 中的 MyBatisRepository 为自定义的注解；

• 步骤

实现此案例需要按照如下步骤进行。

步骤一：业务账号的列表显示功能

1. 在 com.tarena.entity 包下创建实体类 Service 以及修改映射文件 ServiceMapper.xml , Service 类的代码如下所示：

```

package com.tarena.entity;

import java.sql.Timestamp;

public class Service {

    private Integer id;
    private Integer account_id;
    private String unix_host;
    private String os_username;
    private String login_passwd;
    private String status;
    private Timestamp create_date;
}

```

```
private Timestamp pause_date;
private Timestamp close_date;
private Integer cost_id;

public Integer getId() {
    return id;
}
public void setId(Integer id) {
    this.id = id;
}
public Integer getAccount_id() {
    return account_id;
}
public void setAccount_id(Integer account_id) {
    this.account_id = account_id;
}
public String getUnix_host() {
    return unix_host;
}
public void setUnix_host(String unix_host) {
    this.unix_host = unix_host;
}
public String getOs_username() {
    return os_username;
}
public void setOs_username(String os_username) {
    this.os_username = os_username;
}
public String getLogin_passwd() {
    return login_passwd;
}
public void setLogin_passwd(String login_passwd) {
    this.login_passwd = login_passwd;
}
public String getStatus() {
    return status;
}
public void setStatus(String status) {
    this.status = status;
}
public Timestamp getCreate_date() {
    return create_date;
}
public void setCreate_date(Timestamp create_date) {
    this.create_date = create_date;
}
public Timestamp getPause_date() {
    return pause_date;
}
public void setPause_date(Timestamp pause_date) {
    this.pause_date = pause_date;
}
public Timestamp getClose_date() {
    return close_date;
}
public void setClose_date(Timestamp close_date) {
    this.close_date = close_date;
}
public Integer getCost_id() {
    return cost_id;
}
public void setCost_id(Integer cost_id) {
    this.cost_id = cost_id;
}
```

修改映射文件 ServiceMapper.xml , 添加如图-33、图-34 所示的代码 :

```

<select id="findByPage"
    parameterType="com.tarena.entity.page.Page"
    resultType="java.util.HashMap">
    select * from (
        select a.* , ROWNUM r from (
            select
                s.*,
                a.idcard_no,
                a.real_name,
                c.name as cost_name,
                c.descr
            from service s
            inner join account a on a.id=s.account_id
            inner join cost c on c.id=s.cost_id
            <where>
                <if test="os_username!=null && os_username.length()>0">
                    and s.os_username=#{os_username}
                </if>
                <if test="unix_host!=null && unix_host.length()>0">
                    and s.unix_host=#{unix_host}
                </if>
                <if test="idcard_no!=null && idcard_no.length()>0">
                    and a.idcard_no=#{idcard_no}
                </if>
                <if test='status!=null && !status.equals("-1")'>
                    and s.status=#{status}
                </if>
            </where>
            order by s.id
        ) a
    ) where r<#{end} and r>#{begin}
</select>

```

图 - 33

```

<select id="findRows"
    parameterType="com.tarena.entity.page.Page"
    resultType="java.lang.Integer">
    select
        count(*)
    from service s
    inner join account a on a.id=s.account_id
    inner join cost c on c.id=s.cost_id
    <where>
        <if test="os_username!=null && os_username.length()>0">
            and s.os_username=#{os_username}
        </if>
        <if test="unix_host!=null && unix_host.length()>0">
            and s.unix_host=#{unix_host}
        </if>
        <if test="idcard_no!=null && idcard_no.length()>0">
            and a.idcard_no=#{idcard_no}
        </if>
        <if test='status!=null && !status.equals("-1")'>
            and s.status=#{status}
        </if>
    </where>
</select>

```

图 - 34

2. 在 com.tarena.entity.page 包下创建 ServicePage 类，该类继承自 Page，包含了分页所需信息和查询条件所需的信息，代码如下所示：

```
package com.tarena.entity.page;

public class ServicePage extends Page {

    private String os_username;
    private String unix_host;
    private String idcard_no;
    private String status;

    public String getOs_username() {
        return os_username;
    }

    public void setOs_username(String os_username) {
        this.os_username = os_username;
    }

    public String getUnix_host() {
        return unix_host;
    }

    public void setUnix_host(String unix_host) {
        this.unix_host = unix_host;
    }

    public String getIdcard_no() {
        return idcard_no;
    }

    public void setIdcard_no(String idcard_no) {
        this.idcard_no = idcard_no;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }
}
```

3. 修改 com.tarena.dao 包下的 ServiceDao 接口，添加列表实现的代码，代码如图-35 所示：

```
List<Map<String, Object>> findByPage(Page page);

int findRows(Page page);
```

图 - 35

4. 创建控制类 ServiceController，代码如下所示：

```
package com.tarena.controller;

import java.util.HashMap;
```

```

import java.util.List;
import java.util.Map;

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.SessionAttributes;

import com.tarena.dao.AccountDao;
import com.tarena.dao.ServiceDao;
import com.tarena.entity.Account;
import com.tarena.entity.Service;
import com.tarena.entity.page.ServicePage;

@Controller
@RequestMapping("/service")
@SessionAttributes("servicePage")
public class ServiceController extends BaseController {

    @Resource
    private ServiceDao serviceDao;

    @RequestMapping("/findService.do")
    public String find(ServicePage page, Model model) {
        List<Map<String, Object>> list =
            serviceDao.findByPage(page);
        model.addAttribute("services", list);

        page.setRows(serviceDao.findRows(page));
        model.addAttribute("servicePage", page);

        return "service/service_list";
    }
}

```

6. 创建 service_list.jsp , 代码如下所示 :

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>达内 - NetCTOSS</title>
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
        <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
        <script language="javascript" type="text/javascript"
src="../js/jquery-1.11.1.js"></script>
        <script language="javascript" type="text/javascript">
            //显示角色详细信息
            function showDetail(flag, a) {
                var detailDiv = a.parentNode.getElementsByTagName("div") [0];
                if (flag) {
                    detailDiv.style.display = "block";
                }
                else
                    detailDiv.style.display = "none";
            }
        </script>
    </head>
    <body>
        <div>
            <table border="1">
                <tr>
                    <td>序号</td>
                    <td>服务名</td>
                    <td>状态</td>
                    <td>操作</td>
                </tr>
                <c:forEach items="${services}" var="service" varStatus="status">
                    <tr>
                        <td>${status.index+1}</td>
                        <td>${service.name}</td>
                        <td>${service.state}</td>
                        <td>
                            <a href="#" onclick="showDetail(true, this);">查看</a>
                            <a href="#" onclick="showDetail(false, this);">隐藏</a>
                            <a href="#">修改</a>
                            <a href="#">删除</a>
                        </td>
                    </tr>
                </c:forEach>
            </table>
        </div>
    </body>
</html>

```

```

        }
        //删除
        function deleteAccount() {
            var r = window.confirm("确定要删除此业务账号吗？删除后将不能恢复。");
            document.getElementById("operate_result_info").style.display
= "block";
        }
        //开通或暂停
        function setState() {
            var r = window.confirm("确定要开通此业务账号吗？");
            document.getElementById("operate_result_info").style.display
= "block";
        }
        //翻页
        function to_page(page) {
            document.getElementById("current_page").value = page;
            document.forms[0].submit();
        }
        //暂停
        function pauseService(id) {
            var r = window.confirm("确定要暂停此业务账号吗？");
            if(!r) return;
            //采用异步的方式，暂停业务账号
            $.post(
                "pauseService.do",
                {"id":id},
                function(data) {
                    //设置提示信息
                    $("#operate_msg").text(data.message);
                    //显示出提示信息
                    $("#operate_result_info").show();
                    //延迟2s刷新
                    setTimeout(function(){
                        //如果暂停成功，刷新页面
                        if(data.success) {
                            //调用分页函数，刷新当前页
                            to_page($("#current_page").val());
                        } else {
                            $("#operate_result_info").hide();
                        }
                    }, 2000);
                }
            );
        }
        //开通
        function startService(id) {
            var r = window.confirm("确定要开通此业务账号吗？");
            if(!r) return;
            //采用异步的方式，开通业务账号
            $.post(
                "startService.do",
                {"id":id},
                function(data) {
                    //设置提示信息
                    $("#operate_msg").text(data.message);
                    //显示出提示信息
                    $("#operate_result_info").show();
                    //延迟2s刷新
                }
            );
        }
    }
}

```

```

        setTimeout(function(){
            //如果开通成功，刷新页面
            if(data.success) {
                //调用分页函数，刷新当前页
                to_page($("#current_page").val());
            } else {
                $("#operate result info").hide();
            }
            }, 2000);
        );
    }
</script>
</head>
<body>
    <!--Logo 区域开始-->
    <div id="header">
        
        <a href="#">[退出]</a>
    </div>
    <!--Logo 区域结束-->
    <!--导航区域开始-->
    <div id="navi">
        <jsp:include page="/WEB-INF/main/menu.jsp" />
        <!-- <ul id="menu">
            <li><a href="../index.html" class="index off"></a></li>
            <li><a href="../role/role_list.html" class="role_off"></a></li>
            <li><a href="../admin/admin_list.html" class="admin off"></a></li>
            <li><a href="../fee/fee_list.html" class="fee off"></a></li>
            <li><a href="../account/account_list.html" class="account_off"></a></li>
            <li><a href="../service/service_list.html" class="service_on"></a></li>
            <li><a href="../bill/bill_list.html" class="bill off"></a></li>
            <li><a href="../report/report_list.html" class="report_off"></a></li>
            <li><a href="../user/user_info.html" class="information_off"></a></li>
            <li><a href="../user/user_modi_pwd.html" class="password_off"></a></li>
        </ul> -->
    </div>
    <!--导航区域结束-->
    <!--主要区域开始-->
    <div id="main">
        <form action="findService.do" method="post">
            <input type="hidden" name="currentPage" id="current_page" value="${servicePage.currentPage }"/>
            <!--查询-->
            <div class="search_add">
                <div>OS 账号 : <input type="text" name="os_username" value="${servicePage.os username }" class="width100 text search" /></div>
                <div>服务器 IP : <input type="text" name="unix_host" value="${servicePage.unix host }" class="width100 text search" /></div>
                <div>身份证 : <input type="text" name="idcard_no" value="${servicePage.idcard_no }" class="text_search" /></div>
                <div>状态 :
                    <select class="select_search" name="status">

```

```

        <option value="-1">全部</option>
        <option value="0" ><c:if
test="${servicePage.status==0 }">selected</c:if>>开通</option>
        <option value="1" ><c:if
test="${servicePage.status==1 }">selected</c:if>>暂停</option>
        <option value="2" ><c:if
test="${servicePage.status==2 }">selected</c:if>>删除</option>
            </select>
        </div>
        <div><input type="button" value="搜索" class="btn_search"
onclick="to_page(1);"/></div>
        <input type="button" value="增加" class="btn_add"
onclick="location.href='toAddService.do';" />
        </div>
        <!--删除的操作提示-->
        <div id="operate_result_info" class="operate_success">
            
            <span id="operate_msg">删除成功！</span>
        </div>
        <!--数据区域：用表格展示数据-->
        <div id="data">
            <table id="datalist">
                <tr>
                    <th class="width50">业务 ID</th>
                    <th class="width70">账务账号 ID</th>
                    <th class="width150">身份证证</th>
                    <th class="width70">姓名</th>
                    <th>OS 账号</th>
                    <th class="width50">状态</th>
                    <th class="width100">服务器 IP</th>
                    <th class="width100">资费</th>
                    <th class="width200"></th>
                </tr>
                <c:forEach items="${services }" var="service">
                    <tr>
                        <td><a href="service_detail.html" title="查看明细
">${service.ID }</a></td>
                        <td>${service.ACCOUNT_ID }</td>
                        <td>${service.IDCARD_NO }</td>
                        <td>${service.REAL_NAME }</td>
                        <td>${service.OS_USERNAME }</td>
                        <td>
                            <c:choose>
                                <c:when test="${service.STATUS==0 }">
                                    开通
                                </c:when>
                                <c:when test="${service.STATUS==1 }">
                                    暂停
                                </c:when>
                                <c:otherwise>
                                    删除
                                </c:otherwise>
                            </c:choose>
                        </td>
                        <td>${service.UNIX_HOST }</td>
                        <td>

```

```

        <a class="summary"
onmouseover="showDetail(true,this);"
onmouseout="showDetail(false,this);">${service.COST_NAME}</a>
        <!--浮动的详细信息-->
        <div class="detail info">
            ${service.DESCR}
        </div>
    </td>
    <td class="td modi">
        <c:choose>
            <c:when test="${service.STATUS==0 }">
                <input type="button" value=" 暂 停 "
class="btn pause" onclick="pauseService(${service.ID});" />
                <input type="button" value=" 修 改 "
class="btn modify"
onclick="location.href='toUpdateService.do?id=${service.ID}';" />
                <input type="button" value=" 删 除 "
class="btn delete" onclick="deleteAccount();" />
            </c:when>
            <c:when test="${service.STATUS==1 }">
                <input type="button" value=" 开 通 "
class="btn_start" onclick="startService(${service.ID});" />
                <input type="button" value=" 修 改 "
class="btn_modify"
onclick="location.href='toUpdateService.do?id=${service.ID}';" />
                <input type="button" value=" 删 除 "
class="btn_delete" onclick="deleteAccount();" />
            </c:when>
            <c:otherwise>
                </c:otherwise>
            </c:choose>
        </td>
    </tr>
</c:forEach>
</table>
<p>业务说明 :<br />
1、创建即开通，记载创建时间；<br />
2、暂停后，记载暂停时间；<br />
3、重新开通后，删除暂停时间；<br />
4、删除后，记载删除时间，标示为删除，不能再开通、修改、删除；<br />
5、业务账号不设计修改密码功能，由用户自服务功能实现；<br />
6、暂停和删除状态的账务账号下属的业务账号不能被开通。</p>
</div>
<!--分页-->
<div id="pages">
    <a href="javascript:to_page(1);">首页</a>
    <c:choose>
        <c:when test="${servicePage.currentPage==1 }">
            <a href="#">上一页</a>
        </c:when>
        <c:otherwise>
            <a href="javascript:to_page(${servicePage.currentPage-1});">上一页</a>
        </c:otherwise>
    </c:choose>
    <c:forEach begin="1" end="${servicePage.totalPage}" %>
        <var="p">

```

```
<c:choose>
    <c:when test="${p==servicePage.currentPage }">
        <a href="javascript:to_page(${p});" class="current_page">${p}</a>
    </c:when>
    <c:otherwise>
        <a href="javascript:to_page(${p});">${p}</a>
    </c:otherwise>
</c:choose>
</c:forEach>

<c:choose>
    <c:when
test="${servicePage.currentPage==servicePage.totalPage }">
        <a href="#">下一页</a>
    </c:when>
    <c:otherwise>
        <a href="javascript:to_page(${servicePage.currentPage+1});">下一页</a>
    </c:otherwise>
</c:choose>
<a href="javascript:to_page(${servicePage.totalPage});">
末页</a>
</div>
</form>
</div>
<!--主要区域结束-->
<div id="footer">
    <p>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目]</p>
    <p>版权所有(C)加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>
```

6. 部署项目，打开浏览器访问 URL 地址：

<http://localhost:8080/NETCTOSS/service/findService.do>

出现如图-36 所示页面。



图 - 36

点击第“2”页，进入第二页的数据，界面效果如图-37 所示。



图 - 37

可以根据条件进行查询，例如输入身份证 “320211199307310346” 点击 “搜索” 按钮，界面如图-38 所示。



图 - 38

步骤二：业务账号的开通功能

1. 修改映射文件 ServiceMapper.xml，加入如图-39 所示的代码。

```

<update id="updateStatus" parameterType="com.tarena.entity.Service">
    update service set status=#{status},
    <choose>
        <when test="status==0">
            pause_date=null
        </when>
        <when test="status==1">
            pause_date=SYSDATE
        </when>
        <otherwise>
            close_date=SYSDATE
        </otherwise>
    </choose>
    where id=#{id}
</update>

<select id="findById"
    parameterType="java.lang.Integer"
    resultType="com.tarena.entity.Service">
    select * from service where id=#{id}
</select>

```

图 - 39

2. 修改 DAO 接口 ServiceDao，加入如图-40 的代码：

```
void updateStatus(Service service);  
  
Service findById(int id);
```

图 - 40

3. 修改控制类 ServiceController , 加入如下突出显示的代码 :

```
package com.tarena.controller;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
import javax.annotation.Resource;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.ResponseBody;  
import org.springframework.web.bind.annotation.SessionAttributes;  
  
import com.tarena.dao.AccountDao;  
import com.tarena.dao.ServiceDao;  
import com.tarena.entity.Account;  
import com.tarena.entity.Service;  
import com.tarena.entity.page.ServicePage;  
  
@Controller  
@RequestMapping("/service")  
@SessionAttributes("servicePage")  
public class ServiceController extends BaseController {  
  
    @Resource  
    private ServiceDao serviceDao;  
  
    @Resource  
    private AccountDao accountDao;  
  
    @RequestMapping("/findService.do")  
    public String find(ServicePage page, Model model) {  
        ...  
    }  
  
    @RequestMapping("/startService.do")  
    @ResponseBody  
    public Map<String, Object> start(@RequestParam("id") int id) {  
        Map<String, Object> info =  
            new HashMap<String, Object>();  
  
        //判断对应的账务账号是否处于开通态  
        Service service = serviceDao.findById(id);  
        Account account = accountDao.findById(service.getAccount_id());  
        if(!account.getStatus().equals("0")) {  
            info.put("success", false);  
            info.put("message", "账务账号没有开通，不允许开通当前业务账号!");  
        }  
    }
```

```

        return info;
    }

    Service s = new Service();
    s.setId(id);
    s.setStatus("0");
    try {
        serviceDao.updateStatus(s);
        info.put("success", true);
        info.put("message", "开通成功!");
    } catch (Exception e) {
        e.printStackTrace();
        info.put("success", false);
        info.put("message", "开通失败，系统发生异常!");
    }
    return info;
}

}

```

4. 将 NETCTOSS 工程部署到 tomcat 下，并启动服务器。打开浏览器访问 URL 地址：

<http://localhost:8080/NETCTOSS/service/findService.do>

出现如图-41 所示页面。



图 - 41

将的业务 ID 为 “2002” 的业务账号进行暂停。点击图-41 页面上业务 ID 为 “2002” 行的 “开通” 按钮后，重新显示该页面，如图-42 所示。



图 - 42

由图-42可以看出，业务 ID 为“2002”的业务账号没有成功开通，由于申请该业务的帐务账号没有开通。

将业务 ID 为“2005”的业务账号进行开通。点击图-41 页面上业务 ID 为“2005”行的“开通”按钮后，重新显示该页面，如图-43 所示。



图 - 43

由图-43可以看出，业务 ID 为“2005”的业务账号成功开通。

- 完整代码

类 Service 的完整代码如下所示：

```
package com.tarena.entity;

import java.sql.Timestamp;

public class Service {

    private Integer id;
    private Integer account_id;
    private String unix_host;
    private String os_username;
    private String login_passwd;
    private String status;
    private Timestamp create_date;
    private Timestamp pause_date;
    private Timestamp close_date;
    private Integer cost_id;

    public Integer getId() {
        return id;
    }
    public void setId(Integer id) {
        this.id = id;
    }
    public Integer getAccount_id() {
        return account_id;
    }
    public void setAccount_id(Integer account_id) {
        this.account_id = account_id;
    }
    public String getUnix_host() {
        return unix_host;
    }
    public void setUnix_host(String unix_host) {
        this.unix_host = unix_host;
    }
    public String getOs_username() {
        return os_username;
    }
    public void setOs_username(String os_username) {
        this.os_username = os_username;
    }
    public String getLogin_passwd() {
        return login_passwd;
    }
    public void setLogin_passwd(String login_passwd) {
        this.login_passwd = login_passwd;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
    public Timestamp getCreate_date() {
        return create_date;
    }
    public void setCreate_date(Timestamp create_date) {
        this.create_date = create_date;
    }
    public Timestamp getPause_date() {
        return pause_date;
    }
}
```

```
public void setPause date(Timestamp pause date) {
    this.pause date = pause date;
}
public Timestamp getClose_date() {
    return close_date;
}
public void setClose date(Timestamp close date) {
    this.close date = close date;
}
public Integer getCost_id() {
    return cost_id;
}
public void setCost id(Integer cost id) {
    this.cost id = cost id;
}

}
```

类 ServicePage 的完整代码如下所示：

```
package com.tarena.entity.page;

public class ServicePage extends Page {

    private String os_username;
    private String unix_host;
    private String idcard_no;
    private String status;

    public String getOs_username() {
        return os_username;
    }

    public void setOs_username(String os_username) {
        this.os_username = os_username;
    }

    public String getUnix_host() {
        return unix_host;
    }

    public void setUnix_host(String unix_host) {
        this.unix_host = unix_host;
    }

    public String getIdcard_no() {
        return idcard_no;
    }

    public void setIdcard_no(String idcard_no) {
        this.idcard_no = idcard_no;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

}
```

映射文件 ServiceMapper.xml 的完整代码如下所示：

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//ibatis.apache.org//DTD Mapper 3.0//EN"
"http://ibatis.apache.org/dtd/ibatis-3-mapper.dtd">

<mapper namespace="com.tarena.dao.ServiceDao">
    <update id="pauseByAccount" parameterType="java.lang.Integer">
        update service set status='1', pause date=SYSDATE
        where account id=#{accountId}
    </update>

    <select id="findByPage"
        parameterType="com.tarena.entity.page.Page"
        resultType="java.util.HashMap">
        select * from (
            select a.* , ROWNUM r from (
                select
                    s.* ,
                    a.idcard_no,
                    a.real name,
                    c.name as cost name,
                    c.descr
                from service s
                inner join account a on a.id=s.account_id
                inner join cost c on c.id=s.cost_id
                <where>
                    <if test="os username!=null && os.username.length()>0">
                        and s.os_username=#{os_username}
                    </if>
                    <if test="unix_host!=null && unix_host.length()>0">
                        and s.unix_host=#{unix_host}
                    </if>
                    <if test="idcard_no!=null && idcard_no.length()>0">
                        and a.idcard_no=#{idcard_no}
                    </if>
                    <if test='status!=null && !status.equals("-1")'>
                        and s.status=#{status}
                    </if>
                </where>
                order by s.id
            ) a
        ) where r<#{end} and r>#{begin}
    </select>

    <select id="findRows"
        parameterType="com.tarena.entity.page.Page"
        resultType="java.lang.Integer">
        select
            count(*)
        from service s
        inner join account a on a.id=s.account_id
        inner join cost c on c.id=s.cost_id
        <where>
            <if test="os_username!=null && os_username.length()>0">
                and s.os_username=#{os_username}
            </if>
            <if test="unix_host!=null && unix_host.length()>0">
                and s.unix_host=#{unix_host}
            </if>
            <if test="idcard_no!=null && idcard_no.length()>0">
                and a.idcard_no=#{idcard_no}
            </if>
            <if test='status!=null && !status.equals("-1")'>
                and s.status=#{status}
            </if>
        </where>
    </select>

```

```
<update id="updateStatus" parameterType="com.tarena.entity.Service">
    update service set status=#{status},
    <choose>
        <when test="status==0">
            pause_date=null
        </when>
        <when test="status==1">
            pause date=SYSDATE
        </when>
        <otherwise>
            close_date=SYSDATE
        </otherwise>
    </choose>
    where id=#{id}
</update>

<select id="findById"
    parameterType="java.lang.Integer"
    resultType="com.tarena.entity.Service">
    select * from service where id=#{id}
</select>

</mapper>
```

类 ServiceDao 的完整代码如下：

```
package com.tarena.dao;

import java.util.List;
import java.util.Map;

import com.tarena.annotation.MyBatisRepository;
import com.tarena.entity.Service;
import com.tarena.entity.page;

@MyBatisRepository
public interface ServiceDao {
    void pauseByAccount(int accountId);

    List<Map<String, Object>> findByPage(Page page);

    int findRows(Page page);

    void updateStatus(Service service);

    Service findById(int id);
}
```

类 ServiceController 的完整代码如下所示：

```
package com.tarena.controller;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
```

```

import org.springframework.web.bind.annotation.SessionAttributes;

import com.tarena.dao.AccountDao;
import com.tarena.dao.ServiceDao;
import com.tarena.entity.Account;
import com.tarena.entity.Service;
import com.tarena.entity.page.ServicePage;

@Controller
@RequestMapping("/service")
@SessionAttributes("servicePage")
public class ServiceController extends BaseController {

    @Resource
    private ServiceDao serviceDao;

    @Resource
    private AccountDao accountDao;

    @RequestMapping("/findService.do")
    public String find(ServicePage page, Model model) {
        List<Map<String, Object>> list =
            serviceDao.findByPage(page);
        model.addAttribute("services", list);

        page.setRows(serviceDao.findRows(page));
        model.addAttribute("servicePage", page);

        return "service/service_list";
    }

    @RequestMapping("/startService.do")
    @ResponseBody
    public Map<String, Object> start(@RequestParam("id") int id) {
        Map<String, Object> info =
            new HashMap<String, Object>();

        //判断对应的账务账号是否处于开通态
        Service service = serviceDao.findById(id);
        Account account = accountDao.findById(service.getAccount_id());
        if(!account.getStatus().equals("0")) {
            info.put("success", false);
            info.put("message", "帐务账号没有开通，不允许开通当前业务账号！");
            return info;
        }

        Service s = new Service();
        s.setId(id);
        s.setStatus("0");
        try {
            serviceDao.updateStatus(s);
            info.put("success", true);
            info.put("message", "开通成功！");
        } catch (Exception e) {
            e.printStackTrace();
            info.put("success", false);
            info.put("message", "开通失败，系统发生异常！");
        }
        return info;
    }
}

```

页面 service_list.jsp 完整代码如下所示：

```

<%@page pageEncoding="utf-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>达内—NetCTOSS</title>
    <link type="text/css" rel="stylesheet" media="all"
href="../styles/global.css" />
    <link type="text/css" rel="stylesheet" media="all"
href="../styles/global_color.css" />
    <script language="javascript" type="text/javascript"
src="../js/jquery-1.11.1.js"></script>
    <script language="javascript" type="text/javascript">
      //显示角色详细信息
      function showDetail(flag, a) {
        var detailDiv = a.parentNode.getElementsByTagName("div")[0];
        if (flag) {
          detailDiv.style.display = "block";
        }
        else
          detailDiv.style.display = "none";
      }
      //删除
      function deleteAccount() {
        var r = window.confirm("确定要删除此业务账号吗？删除后将不能恢复。");
        document.getElementById("operate_result_info").style.display
= "block";
      }
      //开通或暂停
      function setState() {
        var r = window.confirm("确定要开通此业务账号吗？");
        document.getElementById("operate_result_info").style.display
= "block";
      }
      //翻页
      function toPage(page) {
        document.getElementById("current_page").value = page;
        document.forms[0].submit();
      }
      //暂停
      function pauseService(id) {
        var r = window.confirm("确定要暂停此业务账号吗？");
        if(!r) return;
        //采用异步的方式，暂停业务账号
        $.post(
          "pauseService.do",
          {"id":id},
          function(data) {
            //设置提示信息
            $("#operate_msg").text(data.message);
            //显示出提示信息
            $("#operate_result_info").show();
            //延迟2s刷新
            setTimeout(function(){
              //如果暂停成功，刷新页面
              if(data.success) {
                //调用分页函数，刷新当前页
                toPage($("#current_page").val());
              } else {
                $("#operate_result_info").hide();
              }
            }, 2000);
        });
      }
    
```

```

        }
    );
    //开通
    function start_service(id) {
        var r = window.confirm("确定要开通此业务账号吗？");
        if(!r) return;
        //采用异步的方式，开通业务账号
        $.post(
            "startService.do",
            {"id":id},
            function(data) {
                //设置提示信息
                $("#operate_msg").text(data.message);
                //显示出提示信息
                $("#operate_result_info").show();
                //延迟2s刷新
                setTimeout(function(){
                    //如果开通成功，刷新页面
                    if(data.success) {
                        //调用分页函数，刷新当前页
                        toPage($("#current_page").val());
                    } else {
                        $("#operate_result_info").hide();
                    }
                }, 2000);
            }
        );
    }

```

</script>

</head>

<body>

<!--Logo 区域开始-->

<div id="header">

[退出]

</div>

<!--Logo 区域结束-->

<!--导航区域开始-->

<div id="navi">

<jsp:include page="/WEB-INF/main/menu.jsp" />

<!-- <ul id="menu">

 -->

</div>

<!--导航区域结束-->

<!--主要区域开始-->

<div id="main">

<form action="findService.do" method="post">

```

<input type="hidden" name="currentPage" id="current page"
value="${servicePage.currentPage }"/>
    <!--查询-->
    <div class="search_add">
        <div>OS 账 号 : <input type="text" name="os_username"
value="${servicePage.os_username }" class="width100 text_search" /></div>
        <div>服 务 器 IP : <input type="text" name="unix_host"
value="${servicePage.unix_host }" class="width100 text_search" /></div>
        <div>身 份 证 : <input type="text" name="idcard_no"
value="${servicePage.idcard_no }" class="text_search" /></div>
        <div>状 态:
            <select class="select_search" name="status">
                <option value="-1">全部</option>
                <option value="0" >开通</option>
                <c:if test="${servicePage.status==0 }">selected</c:if></option>
                <option value="1" >暂停</option>
                <c:if test="${servicePage.status==1 }">selected</c:if></option>
                <option value="2" >删除</option>
                <c:if test="${servicePage.status==2 }">selected</c:if></option>
            </select>
        </div>
        <div><input type="button" value="搜索" class="btn_search"
onclick="to page(1);"/></div>
        <input type="button" value="增 加 " class="btn add"
onclick="location.href='toAddService.do';" />
    <!--删除的操作提示-->
    <div id="operate_result_info" class="operate_success">
        
        <span id="operate_msg">删除成功! </span>
    </div>
    <!--数据区域: 用表格展示数据-->
    <div id="data">
        <table id="datalist">
            <tr>
                <th class="width50">业务 ID</th>
                <th class="width70">账务账号 ID</th>
                <th class="width150">身份证号</th>
                <th class="width70">姓名</th>
                <th>OS 账号</th>
                <th class="width50">状态</th>
                <th class="width100">服务器 IP</th>
                <th class="width100">资费</th>
                <th class="width200"></th>
            </tr>
            <c:forEach items="${services }" var="service">
                <tr>
                    <td><a href="service_detail.html" title="查看明细"
"${service.ID }</a></td>
                    <td>${service.ACCOUNT_ID }</td>
                    <td>${service.IDCARD_NO }</td>
                    <td>${service.REAL_NAME }</td>
                    <td>${service.OS_USERNAME }</td>
                    <td>
                        <c:choose>
                            <c:when test="${service.STATUS==0 }">
                                开通
                            </c:when>
                            <c:when test="${service.STATUS==1 }">
                                暂停
                            </c:when>
                            <c:otherwise>
                                删除
                            </c:otherwise>
                        </c:choose>
                    </td>
                </tr>
            </c:forEach>
        </table>
    </div>

```

```

                </c:otherwise>
            </c:choose>
        </td>
        <td>${service.UNIX_HOST }</td>
        <td>
            <a class="summary"
onmouseover="showDetail(true,this);"
onmouseout="showDetail(false,this);">${service.COST_NAME }</a>
            <!--浮动的详细信息-->
            <div class="detail_info">
                ${service.DESCR }
            </div>
        </td>
        <td class="td_modi">
            <c:choose>
                <c:when test="${service.STATUS==0 }">
                    <input type="button" value="暂停" class="btn_pause" onclick="pause_service(${service.ID});" />
                    <input type="button" value="修改" class="btn_modify"
onclick="location.href='toUpdateService.do?id=${service.ID}';" />
                    <input type="button" value="删除" class="btn_delete" onclick="deleteAccount();" />
                </c:when>
                <c:when test="${service.STATUS==1 }">
                    <input type="button" value="开通" class="btn_start" onclick="start_service(${service.ID});" />
                    <input type="button" value="修改" class="btn_modify"
onclick="location.href='toUpdateService.do?id=${service.ID}';" />
                    <input type="button" value="删除" class="btn_delete" onclick="deleteAccount();" />
                </c:when>
                <c:otherwise>
                    </c:otherwise>
                </c:choose>
            </td>
        </tr>
    </c:forEach>
</table>
<p>业务说明: <br />
1、创建即开通，记载创建时间; <br />
2、暂停后，记载暂停时间; <br />
3、重新开通后，删除暂停时间; <br />
4、删除后，记载删除时间，标示为删除，不能再开通、修改、删除; <br />
5、业务账号不设计修改密码功能，由用户自服务功能实现; <br />
6、暂停和删除状态的账务账号下属的业务账号不能被开通。</p>
</div>
<!--分页-->
<div id="pages">
    <a href="javascript:to_page(1);">首页</a>
    <c:choose>
        <c:when test="${servicePage.currentPage==1 }">
            <a href="#">上一页</a>
        </c:when>
        <c:otherwise>
            <a
href="javascript:to_page(${servicePage.currentPage-1});">上一页</a>
        </c:otherwise>
    </c:choose>
    <c:forEach begin="1" end="${servicePage.totalPage }" var="p">
        <c:choose>

```

```
<c:when test="${p==servicePage.currentPage }">
    <a href="javascript:to page(${p});" class="current_page">${p}</a>
</c:when>
<c:otherwise>
    <a href="javascript:to page(${p});">${p}</a>
</c:otherwise>
</c:choose>
</c:forEach>

<c:choose>
    <c:when
test="${servicePage.currentPage==servicePage.totalPage }">
        <a href="#">下一页</a>
    </c:when>
    <c:otherwise>
        <a href="javascript:to_page(${servicePage.currentPage+1});">下一页</a>
    </c:otherwise>
    </c:choose>
    <a href="javascript:to page(${servicePage.totalPage});">
末页</a>
    </div>
</form>
</div>
<!--主要区域结束-->
<div id="footer">
    <p>[源自北美的技术，最优秀的师资，最真实的企业环境，最适用的实战项目]</p>
    <p>版权所有(C)加拿大达内 IT 培训集团公司 </p>
</div>
</body>
</html>
```

