**证券客户适当性管理系统的设计与实现总结**

**撰写人：张劲峰**

证券客户适当性管理系统在金融证券行业中，对客户的分类服务在近年来越来越被重视，因此，在对客户分类服务的过程中，对客户的适当性管理是对客户口常管理不可缺少的部分，所以，建立证券客户适当性管理系统，不仅解决了繁重的客户统计、客户分类、客户维护等大量工作，也逐渐成为了口常管理的常用工具之一。

客户服务工作是证券行业的根本，在口益普及的现代化信息技术面前，联网管理某些具体工作己经成为必然和流行的事情，以前那些手工模式即人工管理，己经与当前企业的发展相背离和显得滞后，为了改变这种不相应和落后，必然要向智能的、效率的新模式发展，这也是必然和趋势，所以传统的必将被代替，证券客户服务这一券商常规工作也是一样的。这样，证券公司的企业管理者就要不断地最求和探索如何为了保证企业的发展，提高客户的服务质量，优化客户服务工作，解放劳动力，实现最大人力功效。因此，营业部研发出了这套系统，这套系统基于WEB，该软件按着证券行业的服务模式，完成从客户录入到客户分类服务的所有工作。通过本系统，强化了和细分了证券客户服务工作的新模式，也使服务落地有声，将无形的服务进行了保存，使证券客户服务工作管理得到了极大地提升和优化，服务和管理都变得高效与快捷，使客户服务工作进入了现代化、信息化和网络化。

根据有关管理制度的规定，各家券商应该明确建立健全客户适当 性管理制度，为客户提供适当的服务，并根据客户所在地区、财务状况、专业知识、投资经验、年龄及投诉情况等进行分类，每年评估一次客户的风险承受能力.。

账户风险有风险认知和承受能力风险还有反洗钱风险。现在国家大力加强金融业反洗钱风险，尤其是证券业反洗钱。证券领域由于自身具有的交易复杂、流动性高等特点，能够为不法分子的洗钱活动提供较好的掩护而逃避执法机关的打击，受到洗钱者的青睐，已经成为我国洗钱犯罪新的重点领域。

我们应综合考虑客户可能涉嫌洗钱和恐怖融资的各类风险因素，对所有客户进行风险等级划分。应当在充分了解客户的基础上，提高对客户身份的识别能力，审慎进行客户风险等级评定。应对客户风险等级进行持续关注，根据实际情况适时调整客户风险等级划分标准和客户风险等级。应根据客户不同风险等级，对其采取相应的风险监控措施。应对客户身份资料、资金和交易信息、风险等级信息等予以保密，非依法律法规规定、客户同意或者因客户身份识别的需要，不得向任何单位和个人提供。

证券业洗钱的主要表现形式为资金的不合理流动、异常的证券交易和伪造、变造客户身份。主要特征如下：

1. 机构客户和自然人客户身份异常。我国证券市场在实行第三方存管制度和实名制开户以前，大量存在客户开户信息不完整、不真实，尤其是伪造、变造身份证明文件等现象，这些故意隐瞒客户真实身份的异常行为，其目的主要是为掩盖洗钱活动。反洗钱经验表明，客户身份异常是证券业可疑交易识别的一个重要特征。

2. 账户间的关联交易明显。洗钱者通常会通过控制大量“拖拉机”账户或多层代理关系账户，使其资金转移的方向基本上控制在自己所掌握的账户中，因此，洗钱账户的实际控制人和这些账户的交易对手等指标往往具有一定的关联性。

3.账户盈亏异常。由于洗钱者需要将非法资金转成合法收益，因此，需要操纵多个账户之间的交易，使其中一部分账户保持长期盈利，而相对应的另一些账户则长期亏损。为了加快非法资金的“漂洗”，有些洗钱账户甚至没有亏损记录。有些洗钱账户在开户后短期内大量买卖证券，然后迅速销户，根本不计盈亏。

4.资金流动异常。有些客户的证券账户长期不进行或少量进行证券交易，但资金账户却频繁发生大额资金收付，有的资金收付也没有明显的交易目的或用途。有的长期闲置账户原因不明地突然启用，并在短期内发生大量证券交易，然后迅速销户将资金转出。 我们应建立健全客户业务办理审核程序，对不同风险等级的客户分别建立相应的审核程序，对较高风险等级客户的审核程序应严于较低风险等级客户。对高风险等级客户，应当了解其资金来源、资金用途、经济状况或者经营状况等信息，加强对其金融交易活动的监测分析。应在持续监控过程中，适时调整客户的风险等级。对较低风险等级客户，证券、基金公司在发现客户多次出现风险因素，或出现多种风险因素并存的情况时，应适当提高其风险等级，并实施更为严格的监控。对高、中风险等级客户，证券、基金公司在一段时期内经过严格审核未再次发现风险因素，或通过回访调查等措施发现客户交易特征均正常时，按审核程序可适当降低其风险等级，并采取相应风险等级的监控措施

这一次我们小组做的证券客户适当性管理系统并没有拥有所有的功能，只截取了一部分有用的内容进行设计和实现。其中内容大致上有五个模块:客户录入、客户分类、服务计划生成、服务计划发送、综合查询。在具体实现上，证券客户适当性管理系统主要体现了软件工程的一些具体方法，在进行了可信性分析和需求分析的基础上，按照需求分析的结果，创建了数据层、业务逻辑层和数据表现层的三层体系结构，并在体系结构中的相关位置，按照系统需要实现的核心功能进行了阐述，明确了系统设计初衷。在证券客户适当性管理系统的设计过程中，是按着体系结构作为指导的，对所有设计进行了详细介绍和说明，特别是结构整体、模块和数据库等都介绍的比较详尽。

其中系统采用的是B/S结构。这种结构是：WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser），如Netscape Navigator或Internet Explorer，服务器安装Oracle、Sybase、Informix或 SQL Server等数据库。浏览器通过Web Server 同数据库进行数据交互。

接下来对我们所采用的技术进行总结：AJAX,SQL SERVER 2005,Power Designer以及Mockups。

AJAX:

事实上AJAX是多种原有技术相结合在一起的，它既不是新兴的也不是单一的而是一个技术组合，而这个组合显现出来了巨大的技术功能能力。具体的说它是由下面所罗列的技术组合而成。

1、它是使用了“XHTML”与“CSS”表示的。

2、它是使用“DOM”模型实现交互与动态显示的。

3、它是应用了“XMLHttp Request”与服务器之间进行的非同步通信。

4、它是使用了“Javascript”来实现绑定与调用的。

在这四种技术当中，"XHTML”与“CSS ", "DOM”和“Javascript”的全部技术中，全是基于web标准的，并且被广泛的应用了，而对于“XMLHttpRequest”来说目前虽然还没有被“W3C”采用，但它己经作为一个真实标准而存在了，因为现在它被全部的在用浏览器所支持。

Sql Server:

在数据库系统中，作为关系数据库管理系统的SQL Server，是由Ashton-Tate公司、Sybase公司和Microsoft公司在一起开发出来的，并成功的于一九九八年推出了首个版本OS/2。在推出Windows NT以后，Sybase与Microsoft在SQL Server的研制开发上就分开了，后来SQL Server被微软移植到了Windows NT上，Windows NT版本的SQL Server就这样的被开发和推广了。而Sybase则致力于在UNIX操作系统上SQL Server的开发和安装界面应用。

数据库管理的最重要特点是:1、在线检索;2、数据库镜像;3、在线恢复(数据库超级用户可在SQL服务器正常运行的时候，实施操作用以恢复。它促使了SQL服务器的提高与改良，因当正在执行被恢复过程中的数据是不可以使用的，而数据库的其余部分依然在线、可提供使用[}' } o > 4、在线检索操作;\_5、安全性能的提高;6、快速恢复;7,崭新的SQL Server 200\_5 ; 8、快照隔离;9、增强复制功能;10、数据分割。

Power Designer:

作为Sybase公司的工具集产品来说，它不仅是CASE工具集，也是其建模与设计的解决方案，使用Power Designer能够非常轻松的设计和分析管理信息系统，因为Power Designer所采用的是模型驱动方法，所以设计数据库模型的全部过程几乎都被包含其中。利用它可以制作概念数据模型、物理数据模型和数据流程图，也能够为数据仓库制作结构模型，同时也能对团队的设计模型进行控制。它能够和Delphi大大缩短PowberBuilder, VB等这些流行的数据库设计软件配合使用，使开发时间也使系统的设计变得更加优化。

PowerDesigner核心功能非常多，见如下简单列举:

它能够提供实体式与概念式二阶段(dual level)的反复式资料库模组设计。

它支援正向工程。

它能够设定多维式实体资料库模组，并且使用定义“OLAP”资料库内的结构。

同样它也支援反向工程(reverse engineering)，同正向工程支援一样能够粹取出来的资料库可以超过30种之上。

它提供资料模型的合并功能(Merge model)o

它提供资料模型以及它们子结构互相比对(Check model)的功能。它可生成资料延伸属性(extended attribute)，这个属性是与“PowerBuilder"应用程式相关联的当企业的伙伴或者非工T人员想要了解和观看系统的用途时，它提供显示流程模组设计和工作需求。

它可以产生的EJB版本为EJB 2. 0，提供物件/关联式对应。

模组图提供了XML格式的储存。

产生的报表可储存成RTF或HTML的格式。

能够方便建立多模型报表((Multi-Model Report)，可将不相关的资料模型组合汇集到一起制作成复合式报表，等等。

Balsamiq Mockups:

Balsamiq Mockups是一款共享软件，每个lisence授权是79美刀，对个人用户来说，价格不菲。它推出之后如此受欢迎的原因是在软件产品原型图设计领域，特别是web原型图设计领域，还没有哪款产品有如此丰富的表现形式。使用Balsamiq Mockups画出的原型图都是手绘风格的图像，看上去非常美观、清爽（当然，跟使用者的设计水平也有关系）。它支持几乎所有的HTML控件原型图，比如按钮（基本按钮、单选按钮等）、文本框、下拉菜单、树形菜单、进度条、多选项卡、日历控件、颜色控件、表格、Windows窗体等。除此以外，它还支持目前如火如荼的iPhone手机元素原型图，这为开发iPhone应用程序的软件工程师提供了非常好的设计图。

接下来从开发过程介绍整一个项目的运作。

**了解阶段：**

了解什么是证券，证券交易是怎样进行的，证券交易系统的用户和客户到底是怎么进行分配和需求的。

证券交易一般分为两种形式：一种形式是上市[1]交易，是指证券在证券交易所集中交易挂牌买卖。凡经批准在证券交易所内登记买卖的证券称为上市证券；其证券能在证券交易所上市交易的公司，称为上市公司。另一种形式是上柜交易，是指公开发行但未达上市标准的证券在证券柜台交易市场买卖。

众多的股份有限公司发行了股票，但不是所有的股票都可以自由上市或上柜交易的。股票要上市或上柜交易，必须按一定条件和标准进行审查，符合规定的才能上市或上柜自由买卖。已上市股票如条件变坏，达不到标准，证券交易所可以停止其上市资格。

一般规定：允许交易的证券，必须是依法发行并交付的证券。所谓依法发行并交付，是指证券的发行是完全按照有关法律的规定进行的，符合法律规定的条件和程序，具有法律依据，通过发行程序并将证券已经交付给购买者。也就是说，进行证券交易的当事人依法买卖的证券，是其合法持有的证券。非依法发行的证券，即证券的发行，没有按照法律规定的条件和程序进行，这样的证券，不得买卖。

依法发行的投票或者公司债券及其他证券，都允许依法进行交易。依法发行的证券可以进行交易，但并不排除法律根据证券的性质和其他情况，对某些证券的交易作出限制性规定。在现实中，对某些证券的转让期限作出限制是可能存在的。如有些债券，只允许在发行后满一定期限才可转让。对于股票，如我国《公司法》对股份有限公司发起人持有的股份有限公司的股份的转让限定为3年之内不得转让。再如目前对上市公司中，内部职工股的转让也有期限的限制。凡是法律对转让期限作出限制性规定的，在限定的期限内，该种证券不得买卖。

经依法核准的上市交易的证券在证券交易所挂牌交易，必须采用公开的集中竞价交易方式。公开的集中竞价，是所有有关购售该证券的买主和卖主集中在一个市内公开申报、竞价交易，每当买卖出价相吻合就构成一笔买卖，交易依买卖组连续进行，每个买卖形成不同的价格。公开的集中竞价具有过程公开性、时间连续性、价格合理性和对快速变化的适应性等特点。

证券交易的集中竞价实行价格优先、时间优先的原则。即买方出价高的优先买方出价低的，卖方出价低的优先于卖方出价高的，多数卖方中出价最低的与多数买方中出价最高的优先成效，以此类推，连续竞价。并在出价相同时，由最先出价者优先成交。

证券交易以现货进行交易，即证券交易达成后，按当时的价格进行实物交割的交易方式。买卖双方的资金和股票发生转移，购买者以直接持有股票为目标，相对其他交易形式来讲更具有投资性。

证券交易所、证券公司、证券登记结算机构从业人员、证券监督管理机构工作人员和法律、行政法规禁止参与股票交易的其他人员，在任期或者法定限期内，不得直接或者以化名、借他人名义持有、买卖股票，也不得收受他人赠送的股票。任何人在成为上列人员时，其原已持有的股票必须依法转让。为股票发行出具审计报告、资产评估报告或者法律意见书等文件的专业机构和人员，在该股票承销期内和期满后6个月内，不得买卖该种股票。此外，为上市公司出具审计报告、资产评估报告或者法律意见书等文件的专业机构和人员，自接受上市公司委托之日起至上述文件公开后5日内，不得买卖该种股票。

持有一个股份有限公司已发行股份5%的股东，应当在其持股数额达到该比例之日起3日内，向该公司报告，公司必须在接到报告之日起3日内向国务院证券监督管理机构报告；属于上市公司的，应当同时向证券交易所报告。上述股东将其所持有的股票在买入后6个月内卖出，或者在卖出后6个月内又买入，由此所得收益归该公司所有，公司董事会应当收回该股东所得收益。公司董事会不按规定执行的，其他股东有权要求董事会执行。如董事会不执行，以致公司遭受损害的，负有责任的董事依法承担承带赔偿责任。但是，证券公司从事证券包销业务，因购入售后剩余股票而持有5%以上股份的，卖出该部分股票时不受6个月时间的限制，即可以在6个月之内将该部分股票卖出。

交易方式

早期证券交易主要采取现货交易方式，但随着商品经济及资本市场的发展，证券交易形式呈现出由低级向高级、由简单向复杂、由单一向复合的发展趋势。各国证券交易方式的分类标准出现多元化趋势，既可按单一标准分类，也可兼采多种标准分类，并形成了现货交易、信用交易、期货交易和期权交易等并存的交易形式。

现货交易

现货交易是证券交易双方在成交后即时清算交割证券和价款的交易方式。现货交易双方，分别为持券待售者和持币待购者。持券待售者意欲将所持证券转变为现金，持币待购者则希望将所持货币转变为证券。现货交易最初是在成交后即时交割证券和钱款，为“一手交钱、一手交货”的典型形式。在现代现货交易中，证券成交与交割间通常都有一定时间间隔，时间间隔长短依证券交易所规定的交割日期确定。证券成交与交割日期可在同一日，也可不是同一日期。在国际上，现货交易的成交与交割的时间间隔一般不超过20日。如依现行的T+1交割规则，证券经纪机构与投资者之间应在成交后的下一个营业日办理完毕交割事宜，如果该下一营业日正逢法定休假日，则交割日期顺延至该法定休假日开始后的第一个营业日。

证券交易所为了确保证券交易所和证券公司有合理时间处理财务事宜(包括准备证券交付和款项往来)，都会对证券成交和交割的时间间隔作出规定。但为防止该时间间隔过长而影响交割安全性，交割日期主要有当日交割、次日交割和例行交割。当日交割，也称“T+0”交割，为成交当日进行交割；次日交割则称“T+1”交割，为成交完成后下一个营业日办理交割；例行交割，则依照交易所规定确定，往往是成交后5个营业日内进行交割。

在现货交易中，证券出卖人必须持有证券，证券购买人必须持有相应的货币，成交日期与交割日期相对比较接近，交割风险较低。从稳定交易秩序角度，现货交易应成为主要交易形式。现货交易作为历史上最古老的证券交易方式，适应信用制度相对落后和交易规则相对简单的社会环境，有助于减少交易风险，是一种较安全的证券交易形式，也是目前场内交易和场外交易中广泛采用的证券交易形式。

期货交易

在广义上，期货交易包括远期交易，与现货交易相对应。其特点是：

1)期货交易对象不是证券本身，而是期货合约，即未来购买或出卖证券并交割的合约。期货合约属于证券交易所制订的标准合约。根据期货合约，一方当事人应于交割期限内，向持有期货合约的另一方交付期货合约指定数量的金融资产。

2)期货合约期限通常比较长，有些金融资产的期货合约期限可能长达数月，甚至一年。在合约期限来临前，期货合约持有人可依公开市场价格向他人出售合约，并借此转让期货合约项下权利。所以，在合约期限来临前，合约持有人可因转让期货合约而发生若干变化。

3)在证券交易所制订标准期货合约时，参考了该等证券资产当时的市场价格，但在期货合约期限内，证券资产的实物价格会发生变动，但在交割证券资产时，其期货价格可能已接近实物资产的市场价格。

由于期货交易具有预先成交、定期交割和价格独立的特点，买卖双方在达成证券期货合同时并无意等到指定日期到来时实际交割证券资产，而是企盼在买进期货合约后的适当时机再行卖出，以谋取利益或减少损失，从而出现“多头交易”和“空头交易”。多头交易与空头交易是站在对期货价格走势不同判断的基础上分别作出的称谓，但均属于低买高卖并借此谋利的交易行为。

在期货合约期限届满前，有一交割期限。在该期限内，期货合约持有人有权要求对方向其进行实物交割。证券交易所为保持信誉和交割安全性，会对此提供担保，并同时要求交割方存入需交割的证券或金钱。

期权交易

证券期权交易是当事人为获得证券市场价格波动带来的利益，约定在一定时间内，以特定价格买进或卖出指定证券，或者放弃买进或卖出指定证券的交易。证券期权交易是以期权作为交易标的的交易形式。期权分为看涨期权和看跌期权两种基本类型。根据看涨期权，期权持有人有权在某一确定时间，以某一确定价格购买标的资产即有价证券。根据看跌期权，期权持有人有权在某一确定的时间，以某一确定价格出售标的资产。根据期权交易规则，看涨期权持有人可以在确定日期购买证券实物资产，也可在到期日放弃购买证券资产；看跌期权持有人，可在确定日期出售证券实物资产，也可拒绝出售证券资产而支付保证金。期权交易属选择权交易。

信用交易

对证券信用交易，学术上有多种学说。依据一般观点，信用交易是投资者凭借自己提供的保证金和信誉，取得经纪人信用，在买进证券时由经纪人提供贷款，在卖出证券时由经纪人贷给证券而进行的交易。因此，凡符合以下条件的证券交易，均属于信用交易乙：

1）典型的信用交易必须是保证金交易，即投资者向经纪人交付一定数额的保证金，并在此基础上进行交易，故信用交易也称保证金交易。

2）经纪人向投资者提供借款购买证券或者经纪人提供证券以供出售。据此，可将信用交易分为融资信用交易和融券信用交易，故证券信用交易也可称为“融资融券交易”。

3）信用交易是证券交易所依照法律规定创设的证券交易方式。证券信用交易具有活跃市场、创造公正市场价格和满足投资者需求的优点，但因存在投资风险，须均衡证券信用交易优劣，设置合理和周严的风险控制制度，给证券信用交易以适当的地位，以实现扬长避短之理想。

信用交易可分为融资交易和融券交易两种类型，但这与我国证券交易实践中出现的“融资交易”与 “融券交易”根本不同。首先，信用交易是依照法律和证券交易所规则创设的证券交易方式，具有适法性，实践中出现的融资融券交易则缺乏法律依据。其次，信用交易以投资者交付保证金为基础，实践中的融资融券交易则几乎完全没有保证金交易的性质。另外，信用交易是经纪人向投资者提供信用的方式，但我国实践广泛存在经纪人向投资者借用资金或借用证券的形式，属于反向融资融券行为。加之，有些资金和证券的借用系未经投资者同意，属于非法挪用资金和证券行为。在此意义上，未经法律准许的融资融券行为，属非法交易行为。

在各国证券市场中，除前述四种基本证券交易方式外，还大量存在其他非主要形态的证券交易方式。有些是相对独立于前四种的证券交易方式，如股票指数交易，有的则是附属于前四种基本证券交易方式，如利率期货等。

5、证券上市交易

上市交易，是指已公开发行且经批准上市的股票在证券交易所集中交易挂牌买卖。

6、证券上柜交易

上柜交易，是指已公开发行但未达上市标准的股票在证券柜台交易市场买卖。

7、证券全价交易

全价交易，是指公司债券以全价(含息价)进行报盘、竞价、撮合、成交，并以全价结算。

目前的银行间债券市场和证券交易所债券市场都实行净价交易。

8、证券净价交易

净价交易，是指公司债券以净价(不含息价)进行报盘、竞价、撮合、成交，并以净价结算，默认由持有人享有持有期的利息收入。

以上是我了解的关于计算机之外的知识，对掌握整一个系统的设计很有帮助，不论是在哪一个阶段都非常重要。

**需求分析阶段**：

需求分析是系统开发的基础工作，但什么是需求分析，通俗的说，就是弄明白做什么，怎么做，确定系统需求进而找到解决办法。在软件工程中，是对系统开发目标进行详细分析的步骤，分析出要输入那些数据，对这些数据如何处理，最终得到什么样的结果。

软件需求分析是系统开发过程中的重要步骤，是系统能否顺利开发的前提，只有提出准确、完整、清晰、具体的客户需求，才能够开发出合格的产品，让客户满意。

证券公司的客户是企业发展的根本，是企业提供服务，获取利润的对象，因此客户分类管理、提供适当性服务是企业口常运营的一个重要组成部分。怎样为各类股民提供所需的服务，进而提高股民对营业部的忠诚而不离开，这才是营业部最终目标和发展前提。随着信息技术的发展和普及，各行各业的工作都逐步向信息化、网络化迈进，客户分类服务工作也不例外，先进的技术为提高服务质量提供了便捷的手段，如何充分利用这一技术为企业的发展提供保障，是每一位客户管理人员不断探索的问题。在证券客户服务管理方面，以往的工作模式完全由人工完成，简单的按照客户资产或客户数量等基本信息进行分类，再由不同的客户经理对这些客户提供统一的服务，这种工作方法极其简单，并没有为特定客户提供特定服务。

信息技术的快速发展为客户服务工作带来了新的契机，要想将信息技术手段引入到客户服务工作中，就势必要通过客户适当性管理系统，将客户的分类服务与客户适当性管理工作融为一体，使系统按照客户资产、交易频度、忠诚度、交易风格等信息自动生成客户分类，并对不同类别的客户提供适当性的管理与服务，以此满足客户服务部门的紧迫需求。

证券客户适当性管理系统，它是券商实现“以服务为中心”发展战略的重要手段，从客户信息采集到客户分类的完成，最后到为不同类别客户提供适当性的服务全部实现网络化，将过去的人工管理模式提高到自动管理模式，通过先进的技术手段提高服务的效率和管理水平，降低企业运营成本。证券客户适当性管理系统的开发，就是为了最终实现客户服务工作的电子化和自动化管理目标，它所使用的技术都是为这一目标所服务。基于以上对证券客户适当性管理系统需求的梳理，首先系统要考虑到数据的安全问题。在营业部里，不同岗位的员工他们的服务层次是不同的，他们的客户群体也是不同的，每口访问的数据也是不同的。所以，设计了不同级别权限，不同客户经理分配不同分级权利。通过以上的分析，我们提出证券客户适当性管理系统的主要功能如下:

1、权限分配

系统用户权限分配是证券客户适当性管理系统正常运行，系统各用户按照自己权限进行相应工作的重要组成部分。在系统中需要有一个用户信息权限总库，里面存储着系统所有用户的各项信息，如工号、姓名、部门、用户角色等信息。当用户使用该系统时，在我们的用户信息库查不到的时候，可通过系统管理员新增用户，并赋予该用户适

当的权限。如果存在该用户信息，则该用户可以按照系统赋予的权限进行相应的操作。

2、服务计划

服务计划主要包括两方面，一个是对所有用户的服务项目，是对所有客户提供无差别服务的项目的统计汇总，包括服务项目的详细描述、服务频度等;二是各类别客户的适当性服务项目，根据各类别客户的属性不同，分别提供的各项适当性服务。

3、信息发布

信息发布模块主要实现了客户成功获取相应信息的功能。此功能需要在客户成功归属到适合自己的类别后方可执行。当客户分类工作完成后，以各类别为单位，将该类别客户的服务计划按照时间、频率等条件系统发送到客户的信息接收终端。

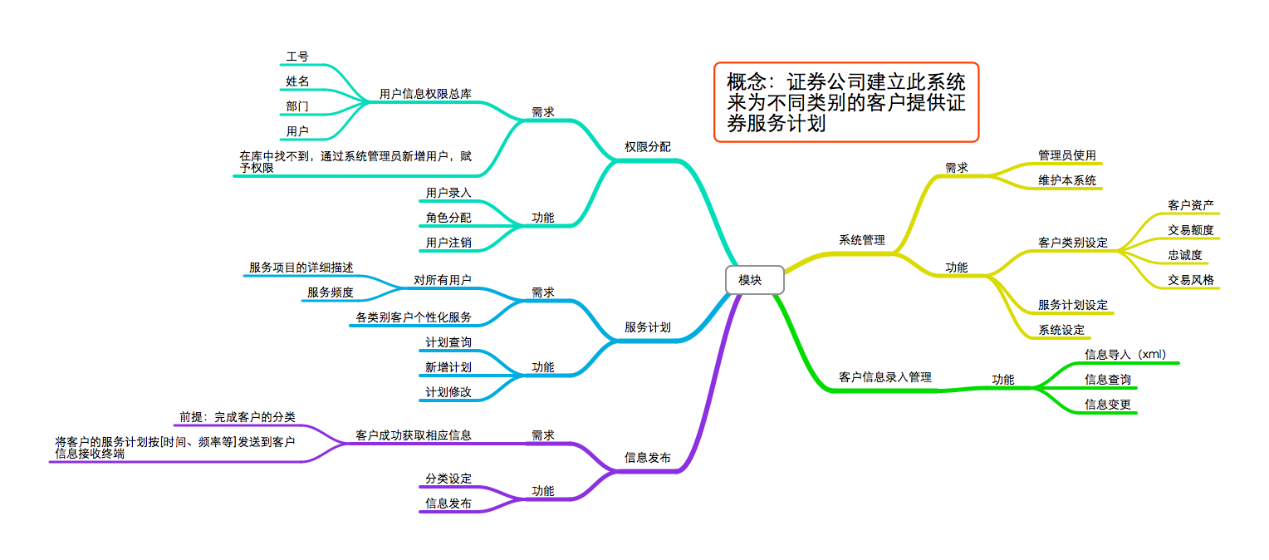
4、系统管理

在本系统中，系统管理模块是证券客户适当性管理系统管理员所使用的模块，通过这个功能组件，系统管理员可以进行对本系统的维护工作，来保证系统的正常运转。

5、客户信息录入管理

客户信息录入管理是对所有客户的基本信息，分类关键信息的管理，可从开户系统中直接导入，也可按分类进行批量处理，为口后提供特定服务提供基础。

**需求分析阶段主要的工作人员是：王翔宇和韩冰，其它人员负责的就是参与其中的讨论工作。**在需求分析阶段,我们制作出了需求分析的图：



**系统的可行性分析：**

证券客户适当性管理系统的可行性分析主要包括了经济可行性分析、技术可行性分析和操作可行性分析。

(1>经济可行性分析

从该系统的开发和效益关系上讲，证券客户适当性管理系统在开发上采用了Visual Studio.net200\_5，它是目前开发高性能Web数据库应用程序最便捷的开发工具，它的开发速度快，系统运行起来也非常稳定，具有其它开发工具所不具有的优势，使用该工具，减少了系统的开发费用。在系统使用上，对硬件要求不高，无需另外购买服务器，只要将该系统与券商的其它应用程序放在一起即可，在使用上也非常经济。另外，通过该系统的使用，将会极大的提高客户服务工作的效率，降低服务人员的劳动强度，减少了管理大量服务人员的成本。通过该系统的使用，将大大降低客服人员的管理成本和服务费用，并将具有长期性，因此，该系统的开发费用与之相比则显得微不足道。

(2)技术可行性分析

这里主要探讨该系统在实现上的技术难度，从系统开发和使用角度分析，主要分为软件和硬件两方面。从目前的计算机网络领域和通信技术领域的发展来看，针对不同系统的开发很多，开发一个基于Web的管理系统可选择的技术很多，不存在软件上的开发瓶颈，与之配套的硬件技术也很成熟，且随着信息技术的发展，费用也越来越低，

因此，在系统开发和使用上，不存在技术上的难题。

**系统的开发目标：**

1、数据稳定性

对任何数据库系统来说，数据的稳定性都是系统的生命。系统的运行离不开数据，数据库中存储了大量的数据，这些数据一旦出现安全和稳定方面的问题，将直接导致系统的崩溃。因此，在系统设计上，首先要考虑的问题就是数据的稳定性和可靠性，只有这个问题解决了，才能够进行下面的开发设计，在系统上线使用后，才会保证系统的平

稳运行。

2、技术的先进性

考虑到后期的升级维护费用，在证券客户适当性管理系统的开发过程中当前的先进技术，升级维护提供便利通过先进技术的使用，来提高系统运行效率和可操作性，应该采用并为后期的。目前，信息技术发展迅猛，技术的更显换代很快，新技术在使用不久、甚至刚刚投入使用就处于被淘汰的境地，因此在技术选择上，一定要紧跟当前最新的技术，通过多方面的评判来选择，以保证系统的先进性。在采用新技术的同时，也面临着所选择的技术是否成熟稳定，这要通过实际的应用来检验，只有经过检验的成熟的技术才是我们所选择的。

3、系统的易维护性

维护性是指系统在投入使用后，能够针对使用中发现的问题便捷的进行修改更新，并随着服务理念和手段的提高，能够迅速升级。这需要在三个方面保证，即时间成本、经济成本、工作量成本，只有这三个方面同时提供了保障，才是具有易维护性的系统。

4、系统经济实用

一个好的系统，不但首先要实现功能上的实用性，能够满足实际工作中的各项需求，而且要降低开发和后期使用的成本，要将系统的实用性和经济行结合起来，缺一不可。

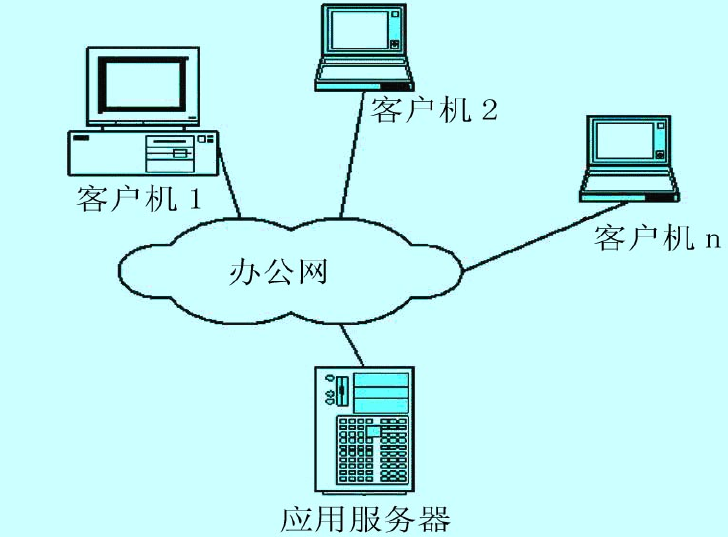
如果只考虑实用性，忽略了系统的经济性，有可能系统的开发和后期维护费用远远大于不使用该系统的成本，则得不偿失;如果只考虑经济型，忽视了系统的实用性，又将对今后的工作设置了许多障碍，降低了客户的服务质量，只有两方面综合考虑，达到一个平衡点，这样开发出来的系统才是具有高性价比的系统。

上面提到的四点原则，就是我们在开发和后期使用过程中需要时刻牢记的目标。在本系统的开发过程中，这四点原则贯穿了始终。

**系统的物理结构分析：**

之前已经说明，该系统使用的是B/S结构的构造，我们知道，B/S系统的三层结构就是把程序按照内部分工以及业务逻辑分割成多个相互独立的程序，通常被划分为界面层、数据存储层和业务处理层。而且业务处理层又可根据需要进一步的分割，能够使程序彼此的关系更加清晰、藕合小。

利用证券公司的内部局域网办公网络环境，本系统采用了三层B/S模式的系统架构，和当前流行的一样传输协议使用了HTTP，客户经理在办公电脑上使用浏览器来访问服务器。工作人员在证券营业部内部使用浏览器登陆系统，系统会给不同的用户分配不同的使用权限，从而实现客户数据查询、录入、修改和生成报表，并打印报表和导出报表，其物理结构图如下图所示：



**系统功能模块设计：**

在系统功能模块设计上，按照需求分析结果，对证券客户适当性管理系统进行功能模块划分，系统共分为以下五大功能模块:权限管理模块、信息发布模块、服务计划模块、系统管理模块和客户信息录入管理这几个部分。

1、权限管理

权限管理模块中包括用户录入、角色分配、用户注销功能模块。在系统管理员权限下，还包括了对所有用户的查询，角色权限的设定等功能。

权限管理是本系统的基本功能，系统用户通过权限管理赋予相应的角色后，即可按照角色功能从事相应的操作，由最低级别的查询功能到高级别的数据修改，都通过本模块实现。

如果用户经过一段时间的工作后，不再使用本系统，可通过注销功能注销用户账号，账号注销后，将在用户库中删除该用户信息，且一会注册的用户也不能使用该用户曾经使用过的ID。

2、服务计划

服务计划模块包括三个功能，最基本的就是服务计划的查询，包括涵盖所有客户的服务计划及特定类别客户的服务计划，在查询过程中可按照限定条件进行查询，并附有打印功能。第二个功能是新增计划功能，指在系统中录入新增加的服务计划，在录入过程中不必设定类别，具体类别在使用中进行设定。第三个功能是服务计划修改，包括对现有服务计划的修改或者删除。

3、信息发布

信息发布实际上就是服务计划的执行，因为券商的服务主要体现在信息的提供上。

在信息发布模块中，包括两个方面的功能，第一个是信息的分类设定，即具体服务计划是向所有客户发布，还是向特定客户发布。第二个是信息发布，在这个模块中可选择是将信息发布到客户的手机上，还是客户的邮箱中，在业务的具体开展上，还可针对券商服务模式的改进增加新的信息发布途径。

4、系统管理

管理模块是系统的管理核心，通过这个模块，管理员可以进行权限分配等维护工作，来保证系统的顺利运转，在功能设定上，将券商服务人员口常服务的管理业务进行提炼总结，归纳出如下功能:客户类别的设定、服务计划的设定及系统的一些基本设定。在客户类别设定中，可以按照用户资产状况、用户资金周转频率、风险偏好等关键信息进行设定，条件设定好后，系统将根据条件信息进行自动比对，完成客户分类工作。计划设定主要包括服务计划的设定，可按券商服务模式的不同，进行不同的设定。

5、客户信息管理

客户信息管理是对所有客户信息的管理，在功能实现上，可通过系统导入功能将券商的所有现有客户信息导入证券客户适当性管理系统的数据库内，导入后可根据用户的条件进行查询、打印等功能，也是系统管理中客户分类的基础。在客户销户后，也可通过客户信息管理模块对客户信息进行删除。

**该系统功能模块设计的主要设计人员是：王翔宇和陈博，其他人员也一并参与其中。**

**数据库系统设计：**

在系统设计上，数据库设计只是系统设计的开端，一个合理的数据库模型，将会减少客户端和服务器的开发和维护难度，并在系统的实际运行中起着关键的作用，因此，数据库设计的好坏，将直接影响系统的性能。一般来讲，在系统的开发初期，设计人员和测试人员注意力大多集中在系统的功能实现方面，忽略了数据量的因素，这样在开发过程中很容易忽略系统的薄弱之处，等到系统正式运行时才发现性能上的缺陷。

数据库是其应用系统的核心和基础，在数据库设计中，数据库模式的构建要根据它的具体应用环境来建立系统，通过数据库模式的优化，来提高系统数据存储的效率，满足应用上的需求。在具体设计中可分为以下几个阶段:

通过对证券客户服务工作的流程分析，归纳提炼出核心服务内容，将各种服务数据进行合理存储，通过对客户基础信息，服务计划信息的设计，并辅以其它信息，最终确定数据库结构。

另外，在数据库设计中，我们还要考虑它的安全性、稳定性、完整性和可恢复性等方面，虽然这些性能的完善总是以牺牲效率为代价，但通过对效率和功能的合理划分，找出它们之间的平衡点，最终提供完美的解决方案，这才是系统开发人员的首要任务。

**数据库系统的设计人员主要是：张劲峰、孙伟然和韩冰。其他人也参与了指导和讨论。**该部分设计的一部分重要数据表如下：

|  |  |  |
| --- | --- | --- |
| 客户信息 | | |
| id | int | <pk> |
| 客户姓名 | varchar(50) |  |
| 客户年龄 | int |  |
| 开户时间 | varchar(50) |  |
| 资金账号 | int |  |

|  |  |  |
| --- | --- | --- |
| 用户设定 | | |
| id | int | <pk> |
| 用户姓名 | varchar(50) |  |
| 工号 | int |  |
| 部门 | varchar(50) |  |

|  |  |  |
| --- | --- | --- |
| 客户类别 | | |
| id | int | <pk> |
| 类别名称 | varchar(50) |  |
| 类别描述 | varchar(50) |  |
| 条件 | varchar(50) |  |

|  |  |  |
| --- | --- | --- |
| 客户计划信息 | | |
| 计划编号 | int | <pk> |
| 计划描述 | varchar(100) |  |
| 计划类别 | varchar(100) |  |
| 计划内容 | varchar(1000) |  |

|  |  |  |
| --- | --- | --- |
| 客户资产状况 | | |
| 资金账号 | int | <pk> |
| 开户资产 | double |  |
| 投资中的资产 | double |  |
| 当前资产 | double |  |
| 盈亏情况 | double |  |

由于我们的系统并不是一个完善的证券客户适当性管理系统，所以数据库的设计也从简。

在证券客户适当性管理系统中，数据库设计通过资金账号和所属类别为主要索引项，以发布方式、资产信息、资产周转率和系统用户信息管理为辅助项，以这些关键信息为主导，结合数据库设计的原则和方法，实现证券客户适当性管理系统的数据库总体设计。

由于我主要的工作是数据库的设计，所以在总结中我的大部分内容会围绕数据库的设计来进行说明。

数据库设计(Database Design)就是按照用户提出的某些要求，在特定的数据库管理系统上，通过设计，建立起数据库的结构，来形成完整数据库的前后过程。

在设计过程中，要建立各种数据表，并规定表内数据是以什么方式存储的，各个表之间是什么关系，构建出一个性能最佳的模式来，这样才能有效地存储所有数据，便于整个系统的开发，但是简化问题也是必须要考虑的问题。由于系统的复杂性，需求多样性，数据库设计往往就变得非常复杂，所以最优的设计不是一气呵成的，而是不断探索、反复修改、当完成精益求精的过程。数据库设计以后，还要生成相关的表等等，也就是要对其进行物理实现。

数据库中的数据表，我们也习惯称为表，它是最重要的在数据库中。数据库就是架构，表才是内容。按所有不同信息的分类情况，数据库中应当包含有很多个数据表。这些各自独立的数据表通过建立关系被联接起来，成为可以交叉查阅、一目了然的数据库。

用户信息:该表存储了用户的一般信息情况，主要包括用户姓名、工号、部门。

|  |  |  |
| --- | --- | --- |
| 用户设定 | | |
| id | int | <pk> |
| 用户姓名 | varchar(50) |  |
| 工号 | int |  |
| 部门 | varchar(50) |  |

客户信息：该表储存了客户的一些信息情况。

|  |  |  |
| --- | --- | --- |
| 客户信息 | | |
| id | int | <pk> |
| 客户姓名 | varchar(50) |  |
| 客户年龄 | int |  |
| 开户时间 | varchar(50) |  |
| 资金账号 | int |  |

客户资产状况信息：

|  |  |  |
| --- | --- | --- |
| 客户资产状况 | | |
| 资金账号 | int | <pk> |
| 开户资产 | double |  |
| 投资中的资产 | double |  |
| 当前资产 | double |  |
| 盈亏情况 | double |  |

等等图。

**系统用例图设计：**

在面对简单的系统时，程序人员往往可以很轻松的给出解决问题的方案，并且通常情况下这些解决方案都是行得通的。这是因为问题集(问题域)的关系比较简单，它的内部结构和相互联系都是很容易就能解释的。但是面对当前逐步变得更加复杂的系统来说，其问题集也变得更加复杂，并且其内部关系也非常不易解释，某些大的系统往往超出了人们能够解决的能力范围。遇到这样的情况，曾经使用的面对过程这一解决方法就不能再达到逐步增加的繁杂系统的分析要求，所以面向对象这一分析方法就表现出来了它的重要性。

在面向对象开发的过程中，系统分析与系统设计都需要按照以下原则进行:应该从客户的角度来考虑问题;比如系统开发生命周期等等之类的产品更新换代机构应该在所有的信息系统开发项目中建立起来;信息系统在开发过程中允许步骤的倒转和重叠;如果系统成功开发的可能性受到了极大的限制时，应该立即取消和停止整个项目;需要重视文档资料。

这一部分的主要设计人员是：王翔宇，陈乐华，其他人也参与了其中的部分工作。

**系统界面的设计：**

**该界面的设计工作是：陈乐华，王翔宇**完成的。其中应用了工具Balsamiq Mockups。

**代码部分的设计：**

**该部分的工作主要是：韩冰完成。**

其中几段重要代码：

修改：

</div>

</div>

</div>

+ <div class="formSep">

+ <div class="row-fluid">

+ <div class="span4">

+ <label for="user\_number">Number <span class="f\_req">\*</span></label>

+ <input id="user\_number" type="text" name="user[number]" class="span12" ng-model="user.number" />

+ </div>

+ <div class="span4">

+ <label for="user\_name">Name <span class="f\_req">\*</span></label>

+ <input id="user\_name" type="text" name="user[name]" class="span12" ng-model="user.name" />

+ </div>

+ </div>

+ </div>

+ <div class="formSep">

+ <div class="row-fluid">

+ <label for="user\_desc">Description </label>

+ <textarea id="user\_desc" name="user[desc]" cols="10" rows="3" class="span8" ng-model="user.desc"></textarea>

+ </div>

+ </div>

<div class="form-actions">

<button class="btn btn-inverse" type="submit" ng-click="update(user)">Save changes</button>

- <!-- <button class="btn">Cancel</button> -->

+ <a href="#/users" class="btn">Cancel</a>

</div>

</form>

添加客户：

UserNewCtrl.$inject = ['$scope', '$location', 'Users']

-UserCtrl = ($scope, User, $routeParams) ->

- $scope.user = User.show

+UserCtrl = ($scope, $routeParams, User) ->

+ $scope.user = User.show({userId: $routeParams.userId})

-UserNewCtrl.$inject = ['$scope', 'User']

+UserNewCtrl.$inject = ['$scope', '$routeParams', 'User']

UserEditCtrl = ($scope) ->

{}

-<h1>Users#show</h1>

-<p>Find me in app/views/users/show.html.erb</p>

+<div class="row-fluid">

+ <div class="span12">

+ <h3 class="heading">{{user.email}}</h3>

+ <div class="row-fluid">

+ <div class="span12">

+ <div class="vcard">

+ <img class="thumbnail" src="http://www.placehold.it/80x80/EFEFEF/AAAAAA" alt="" />

+ <ul>

+ <li class="v-heading">

+ User info

+ </li>

+ <li>

+ <span class="item-key">Email</span>

+ <div class="vcard-item">{{user.email}}</div>

+ </li>

+ <li class="v-heading">

+ Recent activity <span>(latest 24 hours)</span>

+ </li>

+ </ul>

+ </div>

+ </div>

+ </div>

+ </div>

+</div>

+

Update Rail:

-# source 'https://rubygems.org'

-source 'http://ruby.taobao.org'

+source 'https://rubygems.org'

+# source 'http://ruby.taobao.org'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'

-gem 'rails', '4.0.0.rc2'

+gem 'rails', '~> 4.0.0'

# Flexible authentication solution for Rails with Warden

gem "devise", "~> 3.0.0.rc"

@@ -12,7 +12,7 @@ gem "devise", "~> 3.0.0.rc"

gem 'sqlite3'

# Use SCSS for stylesheets

-gem 'sass-rails', '~> 4.0.0.rc2'

+gem 'sass-rails', '~> 4.0.0'

# Use Uglifier as compressor for JavaScript assets

gem 'uglifier', '>= 1.3.0'

@@ -49,6 +49,11 @@ end

group :development, :test do

# Test your JavaScript without any framework dependencies, in any environment, and with a nice descriptive syntax.

# gem 'jasmine', :git => "https://github.com/pivotal/jasmine-gem.git"

+

+ # Debugger

+ gem 'pry-debugger', "~> 0.2.2"

+ gem 'pry-rails', "~> 0.3.1"

+

GEM

- remote: http://ruby.taobao.org/

+ remote: https://rubygems.org/

specs:

- actionmailer (4.0.0.rc2)

- actionpack (= 4.0.0.rc2)

+ actionmailer (4.0.0)

+ actionpack (= 4.0.0)

mail (~> 2.5.3)

- actionpack (4.0.0.rc2)

- activesupport (= 4.0.0.rc2)

+ actionpack (4.0.0)

+ activesupport (= 4.0.0)

builder (~> 3.1.0)

erubis (~> 2.7.0)

rack (~> 1.5.2)

rack-test (~> 0.6.2)

- activemodel (4.0.0.rc2)

- activesupport (= 4.0.0.rc2)

+ activemodel (4.0.0)

+ activesupport (= 4.0.0)

builder (~> 3.1.0)

- activerecord (4.0.0.rc2)

- activemodel (= 4.0.0.rc2)

+ activerecord (4.0.0)

+ activemodel (= 4.0.0)

activerecord-deprecated\_finders (~> 1.0.2)

- activesupport (= 4.0.0.rc2)

+ activesupport (= 4.0.0)

arel (~> 4.0.0)

activerecord-deprecated\_finders (1.0.3)

- activesupport (4.0.0.rc2)

+ activesupport (4.0.0)

i18n (~> 0.6, >= 0.6.4)

minitest (~> 4.2)

multi\_json (~> 1.3)

@@ -30,6 +30,7 @@ GEM

atomic (1.1.9)

bcrypt-ruby (3.0.1)

builder (3.1.4)

+ coderay (1.0.9)

coffee-rails (4.0.0)

coffee-script (>= 2.2.0)

railties (>= 4.0.0.beta, < 5.0)

@@ -37,7 +38,14 @@ GEM

coffee-script-source

execjs

coffee-script-source (1.6.2)

+ columnize (0.3.6)

daemons (1.1.9)

+ debugger (1.6.0)

+ columnize (>= 0.3.1)

+ debugger-linecache (~> 1.2.0)

+ debugger-ruby\_core\_source (~> 1.2.1)

+ debugger-linecache (1.2.0)

+ debugger-ruby\_core\_source (1.2.2)

devise (3.0.0.rc)

bcrypt-ruby (~> 3.0)

orm\_adapter (~> 0.1)

@@ -47,7 +55,7 @@ GEM

eventmachine (1.0.3)

execjs (1.4.0)

multi\_json (~> 1.0)

- font-awesome-rails (3.2.1.0)

+ font-awesome-rails (3.2.1.1)

railties (>= 3.2, < 5.0)

hike (1.2.3)

i18n (0.6.4)

@@ -61,38 +69,49 @@ GEM

mail (2.5.4)

mime-types (~> 1.16)

treetop (~> 1.4.8)

+ method\_source (0.8.1)

mime-types (1.23)

- minitest (4.7.4)

+ minitest (4.7.5)

multi\_json (1.7.7)

orm\_adapter (0.4.0)

polyglot (0.3.3)

+ pry (0.9.12.2)

+ coderay (~> 1.0.5)

+ method\_source (~> 0.8)

+ slop (~> 3.4)

+ pry-debugger (0.2.2)

+ debugger (~> 1.3)

+ pry (~> 0.9.10)

+ pry-rails (0.3.1)

+ pry (>= 0.9.10)

rack (1.5.2)

rack-test (0.6.2)

rack (>= 1.0)

- rails (4.0.0.rc2)

- actionmailer (= 4.0.0.rc2)

- actionpack (= 4.0.0.rc2)

- activerecord (= 4.0.0.rc2)

- activesupport (= 4.0.0.rc2)

+ rails (4.0.0)

+ actionmailer (= 4.0.0)

+ actionpack (= 4.0.0)

+ activerecord (= 4.0.0)

+ activesupport (= 4.0.0)

bundler (>= 1.3.0, < 2.0)

- railties (= 4.0.0.rc2)

+ railties (= 4.0.0)

sprockets-rails (~> 2.0.0)

- railties (4.0.0.rc2)

- actionpack (= 4.0.0.rc2)

- activesupport (= 4.0.0.rc2)

+ railties (4.0.0)

+ actionpack (= 4.0.0)

+ activesupport (= 4.0.0)

rake (>= 0.8.7)

thor (>= 0.18.1, < 2.0)

- rake (10.0.4)

+ rake (10.1.0)

rdoc (3.12.2)

json (~> 1.4)

sass (3.2.9)

- sass-rails (4.0.0.rc2)

+ sass-rails (4.0.0)

railties (>= 4.0.0.beta, < 5.0)

sass (>= 3.1.10)

sprockets-rails (~> 2.0.0)

sdoc (0.3.20)

json (>= 1.1.3)

rdoc (~> 3.10)

+ slop (3.4.5)

sprockets (2.10.0)

hike (~> 1.2)

multi\_json (~> 1.0)

@@ -133,8 +152,10 @@ DEPENDENCIES

font-awesome-rails

jbuilder (~> 1.2)

jquery-rails

- rails (= 4.0.0.rc2)

- sass-rails (~> 4.0.0.rc2)

+ pry-debugger (~> 0.2.2)

+ pry-rails (~> 0.3.1)

+ rails (~> 4.0.0)

+ sass-rails (~> 4.0.0)

sdoc

sqlite3

thin (~> 1.5.1)

**数据设计：**

**该部分的主要工作由张劲峰，陈博，王翔宇完成。**主要完成的内容是构建3个表，以供网页显示。表的内容如下：

用户信息：

见上表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用户姓名 | 单位 | 工号 | 部门 | 角色 | 职务 | 备注 |
| Carly Rae Jepsen | 国泰君安证券股份有限公司陕西分公司 | 6523231967 | 销售交易部 | 营销总监 | 经纪业务部业务主办 |  |
| Jacek Kaspszyk | 国泰君安证券股份有限公司甘肃分公司 | 2105051989 | 销售交易部 | 营销总监 | 经纪业务部业务主办 |  |
| Stuart Zagnit | 国泰君安证券股份有限公司新疆分公司 | 3708011971 | 销售交易部 | 营销总监 | 经纪业务部业务主办 |  |
| Eder Quartett | 国泰君安证券股份有限公司黑龙江分公司 | 4408231982 | 销售交易部 | 营销总监 | 经纪业务部业务主办 |  |
| Tommy Flanagan | 国泰君安证券股份有限公司吉林分公司 | 2305221967 | 证券投资部 | 营销总监 | 保荐代表人 |  |
| Keith Martin | 国泰君安证券股份有限公司辽宁分公司 | 1405211976 | 证券投资部 | 营销总监 | 保荐代表人 |  |
| Wolfgang Petry | 国泰君安证券股份有限公司内蒙古分公司 | 4412261979 | 证券投资部 | 营销总监 | 保荐代表人 |  |
| Chuck Cooper | 国泰君安证券股份有限公司广东分公司 | 5303811979 | 证券投资部 | 营销总监 | 保荐代表人 |  |
| Jonathan Biss | 国泰君安证券股份有限公司福建分公司 | 4512241976 | 零售客户部 | 客户经理 | 高级客户经理 |  |
| Mose Allison | 国泰君安证券股份有限公司海南分公司 | 3503001986 | 零售客户部 | 客户经理 | 高级客户经理 |  |
| Neal Wright | 国泰君安证券股份有限公司广西分公司 | 5304261984 | 零售客户部 | 客户经理 | 高级客户经理 |  |
| Barney Martin | 国泰君安证券股份有限公司重庆分公司 | 3209011976 | 零售客户部 | 客户经理 | 高级客户经理 |  |
| Richard Hawley | 国泰君安证券股份有限公司四川分公司 | 3212841983 | 信息技术部 | 系统管理员 | 信息技术员 |  |
| Tex Williams | 国泰君安证券股份有限公司云南分公司 | 4407011971 | 信息技术部 | 系统管理员 | 信息技术员 |  |
| Gigi Gryce | 国泰君安证券股份有限公司贵州分公司 | 3506281989 | 信息技术部 | 系统管理员 | 信息技术员 |  |
| Dave Douglas | 国泰君安证券股份有限公司河北分公司 | 6401221985 | 信息技术部 | 系统管理员 | 信息技术员 |  |
| Don Patterson | 国泰君安证券股份有限公司河南分公司 | 4502261963 | 人力资源部 | 系统管理员 | 人力资源专员 |  |
| Donald Peers | 国泰君安证券股份有限公司山东分公司 | 3412001968 | 人力资源部 | 系统管理员 | 人力资源专员 |  |
| Norman Large | 国泰君安证券股份有限公司山西分公司 | 1102001974 | 人力资源部 | 系统管理员 | 人力资源专员 |  |
| Stanley Jordan | 国泰君安证券股份有限公司湖北分公司 | 2207001968 | 人力资源部 | 系统管理员 | 人力资源专员 |  |
| Holland Davis | 国泰君安证券股份有限公司湖南分公司 | 3715261973 | 人力资源部 | 系统管理员 | 人力资源专员 |  |

客户信息：

|  |  |  |  |
| --- | --- | --- | --- |
| 客户姓名 | 资金账号 | 股东账号 | 性别 |
| Alicia Keys | 1702022929024910000 | 1108021319000020000 | female |
| Bill Withers | 1702022929024910000 | 32001598636052500000 | male |
| Celine Dion | 443066405018010000000 | 1102021119002120000 | female |
| Daniel Powter | 212060120010141000000 | 535258190661 | male |
| David Guetta | 441162518018001000000 | 258500622924 | male |
| Hans Zimmer | 3602000119200030000 | 41001521010050000000 | male |
| Jennifer Lopez | 3202016429200040000 | 1001173209004610000 | female |
| Keren Ann | 3400203509000020000 | 1103020919200050000 | female |
| Kylie Minogue | 35101588001052500000 | 310066056017911000000 | female |
| Lana Del Rey | 16001301040001600 | 4301012929100610000 | female |
| Lene Marlin | 44201581500059100000 | 3400203509014410000 | female |
| Leona Lewis | 826609796108098000 | 200003629200045000 | female |
| Louis Armstrong | 3602000109000090000 | 1702022929024900000 | male |
| Mariah Carey | 1001173229024930000 | 1105020329001010000 | female |
| Ricky Martin | 44058301040004100 | 327001301040000000 | male |
| Robbie Williams | 3400203529000010000 | 320006600018170000000 | male |
| Ronan Hardiman | 310066056010020000000 | 327001301040002000 | male |
| Sarah Connor | 411060700010141000000 | 10113401040008600 | female |
| Shayne Ward | 44001581301050200000 | 411060700010141000000 | male |
| Sofia Jannok | 42001868608053000000 | 1702022929024900000 | female |
| Wiz Khalifa | 1001173209006510000 | 411060700018001000000 | male |

接上表右边：

|  |  |  |
| --- | --- | --- |
| 身份证号码 | 联系地址 | 工作单位 |
| 431128199002278000 | 山西省晋中市太谷县 | 山西省电力公司电力建设第一公司 |
| 370300198004050000 | 山西省大同市天镇县 | 山西国际招标进出口贸易有限公司 |
| 150781197508233000 | 山西省长治市壶关县 | 山西顺达胜业通信工程有限公司 |
| 410601197402132000 | 山西省大同市灵丘县 | 山西省机械设备进出口公司 |
| 533122198807151000 | 山西省太原市杏花岭区 | 山西中旭国际贸易实业有限公司 |
| 411100197111152000 | 山西省太原市阳曲县 | 太原刚玉国际贸易有限公司 |
| 469038197603263000 | 山西省晋中市和顺县 | 山西洲联科贸有限公司 |
| 611025197201154000 | 山西省晋中市左权县 | 南风化工集团股份有限公司 |
| 330183196610276000 | 山西省晋中市祁县 | 大同煤矿集团有限责任公司 |
| 130582199010010000 | 山西省大同市广灵县 | 山西中艺国际贸易有限公司 |
| 231222197408259000 | 山西省长治市襄垣县 | 山西冶金岩土工程勘察总公司 |
| 360881198412300000 | 山西省长治市沁源县 | 山西省电力公司送变电工程公司 |
| 370200198306262000 | 山西省太原市晋源区 | 太原田和食品集团公司 |
| 511400198806015000 | 山西省长治市长治县 | 山西康博人才就业服务有限公司 |
| 330922196703135000 | 山西省太原市迎泽区 | 山西远翔煤焦有限公司 |
| 433130198711062000 | 山西省大同市浑源县 | 山西三联正丰国际贸易有限公司 |
| 421125198809287000 | 山西省大同市大同县 | 山西中瑞泰克斯泰尔贸易有限公司 |
| 469025198504267000 | 山西省大同市阳高县 | 山西大民国际贸易有限公司 |
| 131125198010012000 | 山西省太原市尖草坪区 | 山西怡康健身器材有限公司 |
| 210702196908017000 | 山西省晋中市灵石县 | 太原路桥建设有限公司 |
| 350102197010180000 | 山西省太原市娄烦县 | 山西省新时代进出口股份有限公司 |

接上表右边：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 手机号码 | 座机号码 | 出生年月 | 佣金类别 | 资产属性 | 利率类别 | 邮政编码 |
| 13250637196 | 03546862345 | 19820902 | 浮动 | 住宿餐饮 | 活期 | 308000 |
| 18719082881 | 03526862345 | 19711204 | 浮动 | 制造业 | 活期 | 382000 |
| 13138321055 | 03556109333 | 19830428 | 浮动 | 综合 | 活期 | 473000 |
| 13829923615 | 03528614666 | 19820704 | 浮动 | 房地产 | 活期 | 344000 |
| 15018656295 | 03512593344 | 19820305 | 固定 | 金融业 | 定期 | 300000 |
| 13502257690 | 03512525258 | 19701104 | 固定 | 金融业 | 定期 | 301000 |
| 13928376193 | 03544956789 | 19830714 | 浮动 | 制造业 | 活期 | 327000 |
| 15627316786 | 03544956789 | 19811112 | 浮动 | 住宿餐饮 | 活期 | 326000 |
| 15119091861 | 03546952333 | 19771117 | 浮动 | 住宿餐饮 | 活期 | 309000 |
| 13829925121 | 03526955553 | 19790630 | 浮动 | 制造业 | 活期 | 375000 |
| 13250635859 | 03556150222 | 19740919 | 浮动 | 综合 | 活期 | 462000 |
| 13250632535 | 03557999991 | 19780228 | 浮动 | 综合 | 活期 | 465000 |
| 18319611111 | 03512597755 | 19770430 | 固定 | 金融业 | 定期 | 300200 |
| 13007522766 | 03554394777 | 19800715 | 浮动 | 住宿餐饮 | 活期 | 471000 |
| 15113318895 | 03512731314 | 19791015 | 固定 | 金融业 | 定期 | 300000 |
| 13928379621 | 03528212222 | 19840927 | 浮动 | 房地产 | 活期 | 374000 |
| 13794519938 | 03522724567 | 19780819 | 浮动 | 房地产 | 定期 | 373000 |
| 15207522550 | 03526564888 | 19730213 | 浮动 | 制造业 | 活期 | 381000 |
| 13680879606 | 03512525775 | 19720514 | 固定 | 金融业 | 定期 | 300000 |
| 13794577735 | 03546097555 | 19751219 | 浮动 | 制造业 | 活期 | 313000 |
| 13719643968 | 03512595999 | 19760509 | 固定 | 房地产 | 定期 | 303000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 计划名称 | 计划编号 | 计划描述 | 计划类别 | 计划内容 |
| Zeus | A000101 | Take it | A | Zeus is the god of sky and thunder in Greek mythology. |
| Apollo | B002548 | Just do | B | Apollo is the son of Zeus and Leto. |
| Ares | C125741 | When need | C | Ares is the Greek god of war. |
| Poseidon | D598742 | Think the | D | Poseidon's main domain is the ocean, and is called the "God of the Sea". |
| Athena | D547123 | What need | D | Athena is the goddess of wisdom, courage, inspiration. |
| Hera | B007052 | With it | B | Hera is the wife of Zeus in the Olympian pantheon of Greek mythology. |
| Aphrodite | C125231 | Be care | C | Aphrodite is the Greek goddess of love, beauty, pleasure. |
| Hades | A541247 | Who need | A | Hades is the ancient Greek god of the underworld. |
| Eros | B213546 | Know it | B | Eros , in Greek mythology, is the Greek god of love. |
| Prometheus | C778412 | Be good | C | Prometheus is a Titan, culture hero,in Greek mythology. |
| Lucifer | D031450 | Cant wait | D | Lucifer is the morning star,in Greek mythology. |
| Daphne | A999854 | Go straight | A | Daphne is a minor figure in Greek mythology known as a Naiad. |
| Hestia | A862145 | Turn around | A | Hestia is a virgin goddess of the hearth, architecture. |
| Demeter | C039872 | Where need | C | Demeter is the goddess of the harvest. |

服务计划信息：

**系统测试：**

**该部分的主要工作由杨洋，韩冰，和其他人共同完成。**

在软件工程中，往往把系统测试看的非常重要，因为它是不可或缺的重要步骤，在系统被使用前，系统测试必须予以实施，避免使用中出现任何差错，系统的负载最大程度也要在测试中完成，以及检测使用效果。系统中存在的问题也是在这个过程中被检测出来，并加以解决，经过反复的测试修改，系统质量和功能不断得到提高，这样，使用者才能得到最满意的产品。

在系统测试中，主要对19名证券公司营业部存量客户的相关信息为用例来检测的，通过这些客户信息的全面测试，测试了该系统的功能和性能。

测试环境：

本系统的测试环境主要有硬件、软件、网络等环境。

(1)系统硬件环境

主要硬件有:

1、一台服务器;Intel 2.6 x2, \_SOOG硬盘，1G内存。

2、八台终端机;Intel P4 2.0G, 1\_50G硬盘，1G内存。

(2)系统软件环境

主要软件有:

服务器端操作系统:Windows Server 2003

数据库:SQL Server200\_5

终端机安装:Windows 7

(3)网络环境

服务器和终端机全部处在局域网环境中。

测试总结

虽然在测试初期，发现了一些小问题，但完善后，运行稳定，没有差错，有非常好的正确性。

系统具有非常好的性能，当多名用户一起对系统操作和使用时，系统能够运行得非常稳定，无系统死机、等待、堵塞、出错等症状，网络占用率处在40%之下，CPU的占用率处在3\_5%之下。

系统具有很好的安全性，无权限用户无法使用和访问该系统。

**该系统的结论**

本课题在系统开发上使用的主要工具是ASP.NET ( C#语言)，通过系统开发，进一步掌握了该工具开发信息系统的方法，对它的关键技术进行了运用和研究，具体涉及如何利用配置文件连接数据库，提高程序的可维护性;通过通用的程序开发技术的运用，提高系统的可扩展性，为今后系统的升级和相关信息系统的开发奠定了一定的基础。本系统使用了SQL SERVER 200\_5作为后台数据库，它是当下使用很普遍的数据库，对中档规模的信息管理系统比较适用，安全性、易用性都很好。使用POWER DESINGER作为数据模型设计软件，方便设计和维护。

本系统在系统开发上使用的主要工具是ASP.NET ( C#语言)，并对它的关键技术进行了运用和研究，提高程序的可维护性;系统使用了SQL SERVER 200\_5作为后台数据库，它是当下使用很普遍的数据库，对中档规模的信息管理系统比较适用，安全性、易用性都很好;使用POWER DESINGER作为数据模型设计软件，方便设计和维护。对证券公司的客户适当性管理系统进行了系统的数据建模，采用B/S模式进行，最后实现了证券公司客户适当性管理系统。并给出了开发过程。其实现了三方面内容:

1、实现了证券公司客户信息的自动导入功能，并按照录入的客户信息，按照关键信息进行分类工作。证券客户服务人员登陆本系统后，即可对不同类别的客户进行查询。

2、实现了服务计划的生成功能，为了给客户提供适当性的管理和服务，实现了对不同类别的客户自动生成适合该类客户的服务计划。

3、开发中应用了AJAX，使使用效果和运行速度最大化的提高。

**个人感想：**

为期五个星期的证券客户适当性管理系统实训终于结束了，总算松了一口气，在这五周时间内承受了很大的压力现在终于能够得到“释放”了，感觉到很轻松。回顾这几周所做过的工作和努力，感觉到收获不少，这里面除了实习本省的内容外，还包括许许多多超乎实习本身意义之外的东西。

说实话本来我多程序编写这相关的都不怎么感兴趣，甚至可以说是害怕，因为大一的时候曾经有过程序设计的大作业，但是包括之前的一学期的学习我感觉到自己学的很差，就这样懵懵懂懂的学过来了，虽然最后大作业的时候我对程序设计有了一知半解，也学到了一些知识，但是这也只是些皮毛。

但是这次的实训不一样，可以说我觉得实训给我带来的最大的帮助就是使我对这门课程有了兴趣，有时候我自己都会感到很奇怪为什么自己会突然迷上了编程，近乎痴迷的爱，以前下了课堂书本就扔的远远的，但是自从这学期开始自学B/S变成后我就开始期盼着最后的实习，可以自己独立设计出一套系统之类的程序，所以我上课充满了激情，下了课有时我还喜欢自己在电脑上捣鼓捣鼓一些程序，去做一些小小的钻研。从以前的“惧怕”上课到现在的“期盼”上课，我自己都说不上来是什么原因，如果非要找出一点理由的话，我想应该是从这学期的程序设计的课程中我已经渐渐地开始有点“入门”，相对于以前我只是站在“程序设计”的大门外徘徊，可以说现在我们能够独立完成这次的实习，变化是巨大的，这种变化还主要原因还在于程序设计给我带来了一种信息，一种前所未有的成就感。

此次我们实训的内容是基于基于HTML、CSS和sql server的数据库应用系统开发一个管理系统，根据证券客户适当性管理系统的需求设计出管理系统的具体功能然后再编写代码。但唯一不同的是，这次证券客户适当性管理系统的开发是由我们每个人独立开发完成的，一开始我们有些不理解，不是说实际过程中真正开发一个软件不可能完全靠哪一个人自己的能力去完成，而是要靠一个团队的分工协作，最终才有可能设计出一个完美的程序吗？那现在靠我们一个人能做什么呢？最终会来得及吗？

后来这几周的实训下来，我才发现这种怀疑和担忧是没有必要的。因为我们这次编写的证券客户适当性管理系统虽然是一个系统，但其实证券客户适当性管理系统所要实现的功能都是平常我们练习过的，所以我们独立完成起来并不是特别的困难，最主要的是我们现在必须先要把个人的能力提高起来才有能力将来和别人一起协作，我们这次的实训主要是锻炼个人的能力和让我们自己去发现问题和解决问题。

实习训开始我先安排了一下自己的进度计划，因为“磨刀不误砍柴工”，虽然证券客户适当性管理系统的开发不是很难，但是对于我们这些初级学者来说，还是有必要做好规划和时间进度的安排，只有这样最后才能够按时完成任务。但是由于实训中有一些小小的变化，所以我也及时调整了自己事先安排好的计划，虽然如此，但是最后还是能够完成了任务。当然不可避免的我们所开发设计的证券客户适当性管理系统肯定还存在方方面面的缺陷和问题，可能并没有那么完美，但是最基本的功能都实现了，最起码能够给我一种莫大的鼓舞，剩下的系统功能的继续的完善就需要在今后的时间内我自己去不断的学习和总结经验，去钻研的更深、更透彻。

从一开始我对这次的实训就充满了信心和期望，所以我的积极性一直很高，当然其中也遇到过一些困难，但是最终在我不懈地努力下都将这些困难一一克服了。所以我一直很享受这个过程。当然对于我们来讲，独立完成一个证券客户适当性管理系统虽然难度不是太大，最终我们也能够完成这个任务，但是对于我们还不是很熟练的情况下，编写出这一个系统还是有一定的压力的，需要我们不断努力和抓紧时间去赶进度的。当然虽然没有团队的协作，但是我们凭自己的能力肯定不可能那么顺利，所以还是需要我们去虚心请教，向基础较好的同学或者老师请教，或者上网搜集相关的资料。

就这样在实训的最后时间内，我们终于完成了证券客户适当性管理系统，当看着我们自己设计的证券客户适当性管理系统编译成功和运行时，心里的那种喜悦和无比的成就感简直就是“无与伦比”的。

有的时候我发现实训的作用真的是无可替代的，我一直在说学校里能够安排每学期的实训真的是很正确的一个决定，有的时候我们甚至会有一种感觉，就是感觉到最后几周的实习所学到的知识甚至比我们平常在课堂上学到的知识要多，因为平常我们所学的知识都仅仅是局限于书本或者老师的讲授，而实习的时候我们实际操作过程会遇到的种种问题可能都是课堂上所未涉及到的。还有一个比较明显的收获就是可能平常我们上课学习的时候都只是按照现成的代码直接去输入，所以并没有什么特别大的收获，当然这次我们实训的刚开始我们也是翻书，满世界的找笔记去找代码，但是不同的是，随着实训的深入我们渐渐地对于平常我们所不能理解的代码慢慢的有了认识，逐渐的我们开始不需要书本，不需要笔记了，我们可以自己敲出那一段段代码，再到后来我们可以对于一些平常我们所没有讲过的一些功能实现所需要的代码可以自己去钻研初来，可以说这真的有点“不可思议”。

如果没有这次的实训，我们可能都无法发现这些问题的所在，所以仅仅学习理论的东西只会将我们“束之高阁”，通过这次实训，最主要的是让我们学会去发现问题，在编译发现问题的过程中学会自己去解决问题，最后解决问题所得到的收获包括成功后的喜悦感都将是属于真正我们自己的东西，成为我们的一笔宝贵的财富和经验，是一笔潜在的能源，受益无穷。