



南京大學

本科毕业设计

院 系 _____ 软件学院 _____

专 业 _____ 软件工程 _____

题 目 _____ 二手房交易平台的电话转接模块和房屋带看评
价模块的设计与实现 _____

年 级 _____ 2012 _____ 学 号 _____ 121250237 _____

学生姓名 _____ 邹源 _____

指导教师 _____ 王浩然 _____ 职 称 _____ 副教授 _____

论文提交日期 _____

南京大学本科生毕业论文（设计）中文摘要

毕业论文题目：二手房交易平台的电话转接模块和带看评价模块的设计与实现

软件学院 院系 软件工程 专业 2012 级本科生姓名：邹源

指导教师（姓名、职称）：王浩然副教授

摘要：

随着社会的不断进步、经济的快速发展，对购房有需求的客户愈来愈多。由于房屋价格不断提高和房产在学区划分、户籍办理上的重要作用，用户越发需要专业而全面的房产信息检索。与之兴起的二手房交易网站，提供了便捷的房产信息搜索方式，因为省去了门面成本的而具有价格优势，从而迅速冲击着传统二手房中介。而传统的房产代理商只有通过将线下积累的房产信息和经纪人优势延续到线上，通过为用户提供更全面和真实的信息，本地化和专业化的咨询才可以跟上潮流，赢得市场。

本系统主要有三个角色，包括审核人员、购房客户和购房中介经纪人。系统的对外网站主要提供给购房客户使用，主要包括房源信息搜索和查看、经纪人推荐和联系方式查看、经纪人微聊；系统的经纪人站主要提供给内部员工经纪人使用，其功能主要有房源信息录入，房源带看经历录入等。系统的审核人员使用访问管理站点进行房源信息、带看经历和带看评价的审核。

本系统基于 **SpringMVC+MyBatis** 框架进行开发，总体遵循分层的设计思想，系统分别部署在阿里云和上海链家本地服务器，使用的是 **Sqlserver** 数据库。本系统后台用到 **log4j** 框架和缓存技术，消息传递用到了 **FastJson** 和 **ActiveMq** 技术。

本人所在的链家上海外网组负责完成链家上海站给用户使用的对外网站。本人在项目中主要负责带看评价模块和转接号管理模块的设计与实现。

关键词：短链接、数据同步、O2O、SpringMVC、Mybatis

南京大学本科生毕业论文（设计）英文摘要

THESIS: Design and implementation of Telephone adapter module and Watch evaluation module in Secondhand Housing Trading System

DEPARTMENT: Software Institute

SPECIALIZATION: Software Engineering

UNDERGRADUATE: Zou Yuan

MENTOR: Haoran Wang

ABSTRACT:

With the advance of society, the rapid development of economy, more and more customers are in high demand for house-purchase. Because of the increasing of house price and the important role that house plays in census and school district, more and more users need professional and comprehensive housing information retrieval.

Providing a convenient way of house trading information search, cutting cost by cancel of stores, Second-hand housing transactions website impact the traditional second-hand housing intermediary quickly.

And only if the traditional real estate agents continue the superiority in accumulation of housing and brokers, can they keep up with the trend and win the market.

This system includes three roles, namely system administrator, customer, and broker. The foreign website of system is provided for customers to search house information and view house details. The website for broker focus on information entry, including second-hand house entry and watch history entry. The website for system administrator is used to audit information which brokers and customers submit.

Based on SpringMVC + Mybatis framework, this system adheres to the layered design pattern as a whole. The system is deployed respectively on Aliyun server and Lianjia local server and uses sqlserver. The back end adopts log4j framework and the caching technology. Besides, the system uses FastJson and ActiveMQ technology to change information between modules.

When I worked as an intern in Shanghai Lianjia, our team is responsible for completing Lianjia Shanghai foreign customer-oriented web-site. And my job is to complete the design and implementation of Telephone adapter module and Watch evaluation module in Secondhand Housing Trading System.

KEY WORDS: LianJia Secondhand Housing Trading System, ShortLink, Data Sync, O2O, SpringMVC, Mybatis

目 录

图目录	III
表目录	IV
第一章 引言	1
1.1 项目背景	1
1.2 二手房交易系统现状研究	1
1.3 论文的主要工作和组织结构	2
第二章 技术概述	3
2.1 Spring 框架	3
2.1.1 Spring 简介	3
2.1.2 Spring 结构	3
2.2 MyBatis	4
2.3 ActiveMQ	4
2.4 FastJSON	5
2.5 Maven 依赖管理	5
2.6 本章小结	5
第三章 系统需求分析与概要设计	6
3.1 二手房交易网站带看评价和电话转接模块概述	6
3.1.1 系统用户角色	6
3.1.2 系统功能需求	6
3.1.3 系统非功能需求	7
3.2 二手房带看评价模块系统需求分析	9
3.2.1 带看模块用例描述	9
3.2.2 带看模块实体关系分析	10
3.2 转接号模块系统需求分析	11
3.2.1 转接号模块用例描述	11
3.4 系统概要设计	13
3.3.1 总体设计	13
3.3.2 带看评价模块设计	16
3.3.3 转接号模块设计	20
3.4 本章小结	21
第四章 详细设计与实现	22
4.1 带看评价模块概述	22
4.2 带看评价模块的详细设计	22
4.2.1 设计部模块详细类图和顺序图	22
1.2.2 带看评价短链接功能设计	27
4.3 带看评价模块实现	29
4.3.1 带看评价模块数据层实现	29
4.3.2 带看评价模块服务层实现	34
4.3.2 带看评价模块控制层实现	36
4.3.4 带看评价模块展示页面	41

4.4 电话转接模块概述	45
4.5 电话转接模块详细设计	45
4.6 电话转接模块实现	49
4.6.1 电话管理模块发起的数据同步	49
4.6.2 转接号模块实现的数据同步。	52
4.7 本章小结	55
第五章 总结与展望	56
5.1 总结.....	56
5.2 展望.....	56
参考文献	58
致谢	59

图目录

图 2.1 Spring 框架图	3
图 3.1 二手房交易系统模块图	7
图 3.2 带看评价模块用例图	10
图 3.3 带看评价模块实体关系图	11
图 3.4 二手房交易系统模块部署	14
图 3.5 通用模块程序结构图	15
图 3.6 带看评价逻辑架构	16
图 3.7 短信发送接口	17
图 3.8 带看评价反馈接口	17
图 3.9 带看评价已经提交、返回推荐经纪人接口	18
图 3.10 带看评价详细信息浏览接口	20
图 3.11 转接号逻辑架构	20
图 4.1 生产部模块详细类图-面向用户	23
图 4.2 生产部模块顺序图-面向用户	24
图 4.3 带看评价模块详细类图-面向经纪人服务器	25
图 4.4 带看评价模块顺序图-面向经纪人服务器	26
图 4.5 带看评价模块 Model 包类图	27
图 4.6 配置文件 myBatis-config	30
图 4.7 配置文件 jdbc.properties	30
图 4.8 EmpDao 和 takeWatchDao 接口类代码	31
图 4.9 TakeWatchDao.xml 关键代码	32
图 4.10 EmpDao.xml 中的经纪人推荐算法	34
图 4.10 ListenMqService.onMessage()代码	35
图 4.11 phoneService.sendMessage()代码	35
图 4.12 Base62 类关键代码	36
图 4.14 Spring-mvc 配置文件	38
图 4.14 TakeWatchController 类主要代码	41
图 4.15 浏览带看评价内容	42
图 4.16 提交带看反馈	43
图 4.17 查看提交结果	45
图 4.18 电话管理服务详细类图	46
图 4.19 转接号服务详细类图	47
图 4.20 电话管理的 Model 层类图	48
图 4.20 转接号服务的 Model 层类图	49
图 4.20 TransferService 类代码	50
图 4.21 TempDataCleaner 类代码	51
图 4.22 TransferController 类代码	51
图 4.23 mergeBatch 方法代码	52
图 4.24 jdbc.properties 配置文件	54
图 4.25 KeyExtNumberUpdate 类代码	54
图 4.26 askForUpdate 方法代码	55

表目录

表 3.1 非功能性需求列表.....	8
表 3.2 转接号码生成用例.....	12
表 3.3 转接号码查询用例.....	12
表 3.4 转接号码同步用例.....	13
表 3.5 TakeWatch 接口定义.....	16
表 3.6 DesignService 接口方法表.....	20
表 4.1 短链接算法优越点比较.....	29

第一章 引言

1.1 项目背景

在我国，住房问题一直属于重要的民生问题，随着各级城市房价持续走高，越来越多的购房者将目光投向了总价较低、配套成熟、交通便利的二手房。由于传统的二手房中介提供给用户的浏览信息有限，为了让购房者更方便得获取二手房信息，了解候选房屋环境、价格、条件是否满足自身需求，用二手房交易网站取代传统的二手房交易门店成为一种趋势。在二手房交易市场趋于火热的背景下，涌现了大量的二手房信息服务网站，小型中介网站更是不计其数。在为购房者提供便利的同时，大量质量参差不齐的二手房信息同时也带来了“信息爆炸”的问题。这使缺乏经验的初次购房者在面对海量信息时感到无所适从。因此短时间内二手房交易还无法脱离房产交易经纪人进行。

链家网是为消费者打通线上与线下所有环节，提供高品质、高效率、标准化服务的房产 O2O 公司^[1]。借着近几年二手房交易的发展浪潮，公司规模逐步扩大，原本由北京总部提供的外网服务无法给北京、上海、深圳等不同城市的用户提供一致的本地化服务、无法保证稳定的网络访问。故而链家上海将原本北京提供的上海站服务转移到上海外网部门独立开发和运行。为了提供给用户更好的经纪人服务。链家的二手房系统除了基本的经纪人联络系统外，还加入了经纪人推荐系统，带看经纪人评价系统。

本文介绍的二手房交易系统，很大程度上提高了生产销售工作的效率，解决了智造链公司传统业务模式存在的诸多问题。

1.2 二手房交易系统现状研究

O2O 二手房交易网站在国内有很多，比如搜房网、爱屋吉屋、58 同城等。由于房源信息需要审核的原因，数据同步技术在这类网站中都得到了广泛的应用。链家网的带看评价数据同步模块参考了房源数据同步的方式，在房源信息同步时，房源信息常用的方法涉及到了增量同步，数据冲突与一致性，分布式数据等问题。

由于看房时携带用户随身携带的通常只有手机，所以为了方便用户看房时进行二手房信息查询，二手房经纪人查询和进行带看评价。所以二手房交易系统通常涉及到移动互联网技术。本系统的带看评价网页便以 wap 技术提供给用户使用。同时移动端设计到短信验证，短信发送和短信 URL 链接，带看评价中的短

信 URL 采用了业界常用的短链接技术，这一技术后来反馈到了链家的房源详情短链接功能中。具体设计见文章后面部分。

1.3 论文的主要工作和组织结构

本文介绍了二手房交易系统的带看评价模块、转接号管理模块的设计与开发细节。

第一章：概述和前言部分，主要介绍了二手房交易网站的项目背景，并描述了该论文的组织结构。

第二章：将主要介绍系统在开发过程中使用到的相关技术及类库，并简单阐述使用这些技术的原因。

第三章：将会描述系统的需求分析和概要设计过程。在需求分析阶段，将介绍领域相关概念，分析和总结功能性和非功能性需求。从项目概要设计方面，描述项目整体的设计框架以及通过功能划分对项目进行包设计和接口设计。

第四章：

对二手房交易系统详细设计和实现进行阐述，包括设计思想及各个模块的类设计，同时结合详细的代码和说明，描述了本系统中关键功能的实现。

第五章：总结本阶段中所完成的工作，探讨系统存在的缺点和不足，并指出该系统未来的扩展和发展方向。

第二章 技术概述

2.1 Spring 框架

2.1.1 Spring 简介

Spring 框架是 Rod Johnson、Juergen Hoeller 等开发的，用于支持 JavaBean 构件运行的容器。该框架提供了依赖注入方式的构件组装机制和基于 AOP 技术的事务和日志管理等功能。基于 Spring 框架的轻量级 J2EE 架构能够发挥上述两种架构方案的优势，避免它们存在的缺陷，是一种成熟的 J2EE 应用开发方案^[2]。

2.1.2 Spring 结构

Spring 框架由 7 个模块组成，如图 2.3 所示。Spring 模块构建在核心容器上，核心容器定义了创建、配置和管理 bean 的方式。用户的系统不必使用所有的模块，可以自由挑选。

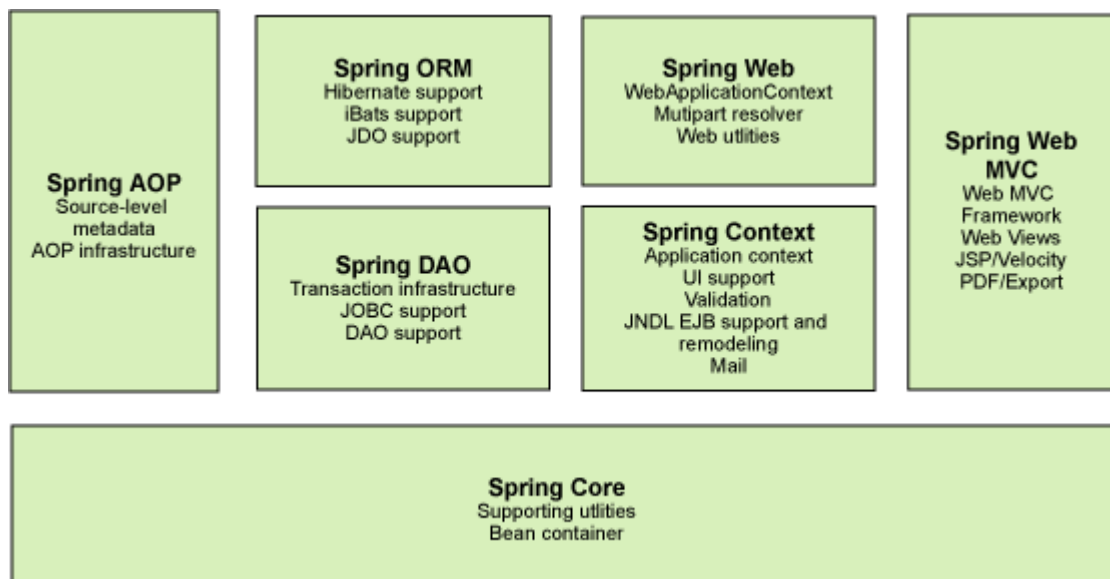


图 2.1 Spring 框架图

- **核心容器**：提供 Spring 框架的基本功能。核心容器主要组件是 BeanFactory，它是工厂模式的具体实现。应用程序的配置和依赖性规范与实际的应用程序代码的分开由控制反转（IoC）模式实现。
- **Spring 上下文**：Spring 上下文向 Spring 框架提供上下文信息，是一个配置文件。
- **Spring AOP**：Spring AOP 模块通过配置管理特性直接将面向方面的编程功能集成到了 Spring 框架中。所以，可以很容易地使 Spring 框架管理的任何对象支持 AOP。Spring AOP 模块为基于 Spring 的应用程序中的对象提供了事务管理服务。
- **Spring DAO**：提供了有意义的异常层次结构，该结构可用来管理异常和处理不同数据库供应商抛出的错误消息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量。
- **Spring ORM**：Spring 框架插入了若干个 ORM 框架提供了 ORM 的对象关系工具，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。
- **Spring MVC 框架**：通过策略接口，MVC 框架变成高度可配置的，MVC 容纳了大量视图技术，其中包括 JSP、Tiles、iText 和 POI 等 [3]。

2.2 MyBatis

MyBatis 本是 apache 的一个开源项目 iBatis，2010 年这个项目由 apache software foundation 迁移到了 google code，并且改名为 MyBatis。

MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀持久层框架，避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis 可以对配置和原生 Map 使用简单的 XML 或注解，将接口和 Java 的 POJOs (Plain Old Java Objects, 普通的 Java 对象) 映射成数据库中的记录 [4]。

2.3 ActiveMQ

ActiveMQ 是 Apache 出品，最流行的，能力强劲的开源消息总线。完全支持 JMS1.1 和 J2EE 1.4 规范（持久化，XA 消息，事务）。ActiveMQ 支持 Spring，可以很容易内嵌到使用 Spring 的系统 [5]。通过了常见 J2EE 服务器的测试可以让 ActiveMQ 可以自动的部署到任何兼容 J2EE 1.4 商业服务器上。

在实际使用时，链家已有单独的项目部署和配置了内部的 ActiveMQ 服务器，在外网项目中只需引入配置即可使用。

2.4 FastJSON

阿里巴巴 FastJson 是一个 Json 处理工具包，包括“序列化”和“反序列化”两部分，项目使用 FastJson 主要因为它具备如下特征^[6]：

- 1、速度最快，测试表明，fastjson 具有极快的性能，超越任其他的 Java Json parser。包括自称最快的 Jackson。
- 2、功能强大，完全支持 Java Bean、集合、Map、日期、Enum，支持泛型，支持自省；
- 3、无依赖，能够直接运行在 Java SE 5.0 以上版本。

2.5 Maven 依赖管理

Maven 项目对象模型(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的软件项目管理工具^[7]。

Maven 除了以程序构建能力为特色之外，还提供高级项目管理工具。由于 Maven 的缺省构建规则有较高的可重用性，所以常常用两三行 Maven 构建脚本就可以构建简单的项目。

2.6 本章小结

本章介绍了二手房交易系统中使用到的相关技术的概述，主要为 Spring 的框架及结构，信息交换工具 FastJSON 和 ActiveMQ 以及数据持久化工具 Mybatis 的结构和优点。

第三章 系统需求分析与概要设计

3.1 二手房交易网站带看评价和电话转接模块概述

本人在上海链家实习期间参与了二手房交易对外网站项目，并负责实现了带看评价和电话转接模块。该二手房交易网站分为两个系统，对外网站用于展示房产信息，推荐中介经纪人和提供经纪人联系方式，同步经纪人录入的看房记录信息，对经纪人的服务进行反馈和评价。中介经纪人系统用于中介经纪人录入带看房源的记录，审核经纪人带看信息和房源信息是否一致。

3.1.1 系统用户角色

本系统的使用者包括购房用户、房产交易经纪人、审核人员。购房用户需要查找浏览房产信息、经纪人信息，联系经纪人，对看房服务进行评价。交易经纪人主要负责和用户联系，录入和用户看房的经历。审核人员负责审核经纪人录入的信息和用户评价是否一致。

3.1.2 系统功能需求

根据不同的服务要求，二手房交易对外服务系统在分为 8 个模块，其中与转接号管理、带看评价管理相关的模块有 5 个：

- (1) 电话转接模块：包括同步网服务器上发布的转接号数据，转接号与真实号码的转换。
- (2) 电话管理模块：提供经纪人转接号查询，生成经纪人转接号。
- (3) 二手房后台模块：提供网页定位和生成，短链接定位。
- (4) 用户中心模块：负责用户关注房源，看房记录，账号管理。
- (5) 带看评价模块：负责接收经纪人系统发布的房源信息，发送包含带看评价网址的短信给用户录入带看评价，推荐新的中介经纪人。

模块划分如图 3.1 所示：

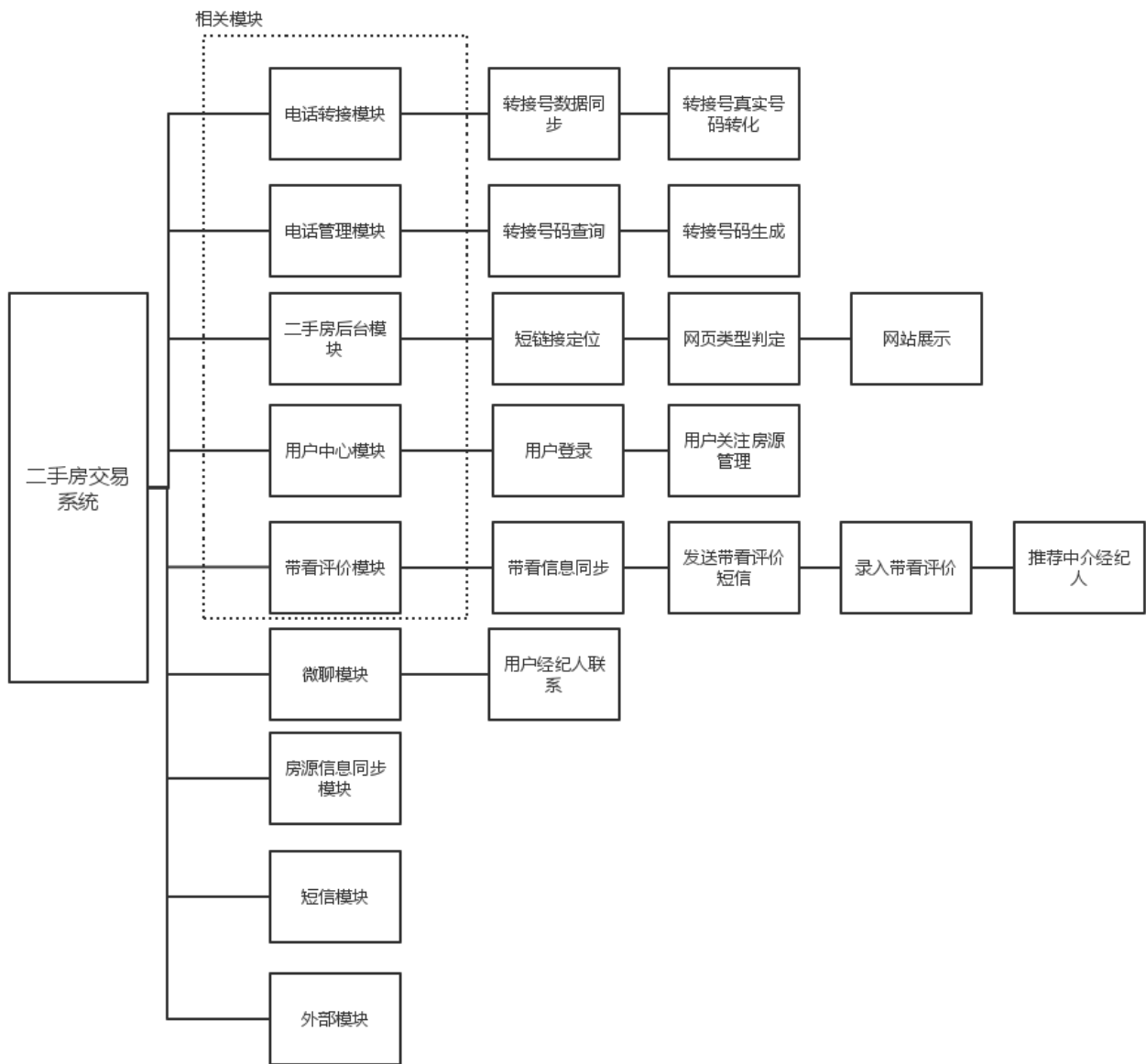


图 3.1 二手房交易系统模块图

3.1.3 系统非功能需求

除了基础的功能性需求，性能需求、质量属性、可用性等非功能性需求在判定一个系统是否是受用户欢迎的好产品上也有着重要的作用。本系统对性能和质量提出了以下的要求：

容量大：

二手房系统目前遗留的房源数量为 10 万以上，经纪人信息数据为 5 万条，

结合经纪人、带看用户以及房源的带看信息数目近亿条，这些数据除了增删查改外，在经纪人推荐、关注房源推送等算法中也是重要的比较对象，故而在存储容量上有着一定的要求。

速度快：

一方面，用户希望能在最短时间内打开网站，查询和浏览信息。另一方面，经纪人希望用户尽快反馈带看评价，因为及时提供的评价更准确而参考价值，且便于审核人员审核。故而系统需要能快速打开网站，也能在经纪人录入带看经历后快速发送评价短信息。

健壮性：

由于转接号系统分别部署在两个服务器上，使用不同的数据库。由于网络问题可能存在同步故障。为保证两边的数据库一致，系统应处理网络故障带来的不同步影响。

可用性：

系统面对的用户群体复杂而多样，所以系统显示的界面应该简单易懂，操作应对户友好，交互方便。

可拓展性：

对外网站的不同功能之间应该避免互相干扰，在增加新功能时避免对原有模块的变动。由于部分模块暂时需要使用一部分北京链家的服务，应当考虑到将来服务独立开的情况，能够及时修改和拓展。

安全性：

用户的私人信息不应当被其他用户无意中浏览。

根据以上分析，总结出本系统的非功能性需求，具体可见下表 3.1 中。

性能需求	PR1：网站的所有页面都必须在 5s 内打开
	PR2：系统至少能够存储一条万条带看数据
质量属性	QA1：系统的可用性要达到 98%
	QA2：系统在网络故障的情况下，将发送失败的数据暂存直到重新发送成功
	QA3：如果系统发送数据失败，接收端需要主动拉回同步的数据
	QA4：如果系统要变更使用的服务接口，要能够在 7 个人日内完成
	QA5：系统需要保证用户信息，带看信息等资料的私密性。

表 3.1 非功能性需求列表

3.2 二手房带看评价模块系统需求分析

3.2.1 带看模块用例描述

二手房带看模块包括四个主要用例:发送二手房带看评价短信、评价带看经历、反馈带看经历、推荐经纪人。其用例关系如图 3.2 所示

1. 发送带看二手房评价短信:

二手房交易经纪人录入带看经历后,需要及时向用户发送一条短信,短信内容需要包括带看经纪人信息、带看时间以及进行带看评价的短链接网址,短信需要信息清晰,内容简短。短信内容不得超过 60 字且短信中的网址长度不得超过 30 字符。短信内容中需要包含上海链家的标签。

2. 评价带看经历:

当用户打开带看评价网站后,可以浏览带看过的房源信息,带看经纪人信息,可以直接联系带看经纪人。在评价信息中,用户可以有选择地填写经纪人专业能力、服务态度以及补充说明。当经纪人录入的带看信息和实际情况不符合时,可以打开带看反馈界面。

3. 反馈带看经历:

用户打开反馈带看经历的页面后,可以选填问题种类,包括没有看过录入的带看房源、带看房源信息与实际房源不符、更多吐槽。

4. 推荐经纪人:

用户评价或者反馈评价带看经历的内容显示用户对带看经历不满意,系统应针对性地重新推荐经纪人,重新推荐的经纪人应当与原经纪人同属一个地域板块但不属于同一个营业分部,拥有相同房源或者同小区房源带看经历的经纪人拥有更高推荐优先级。

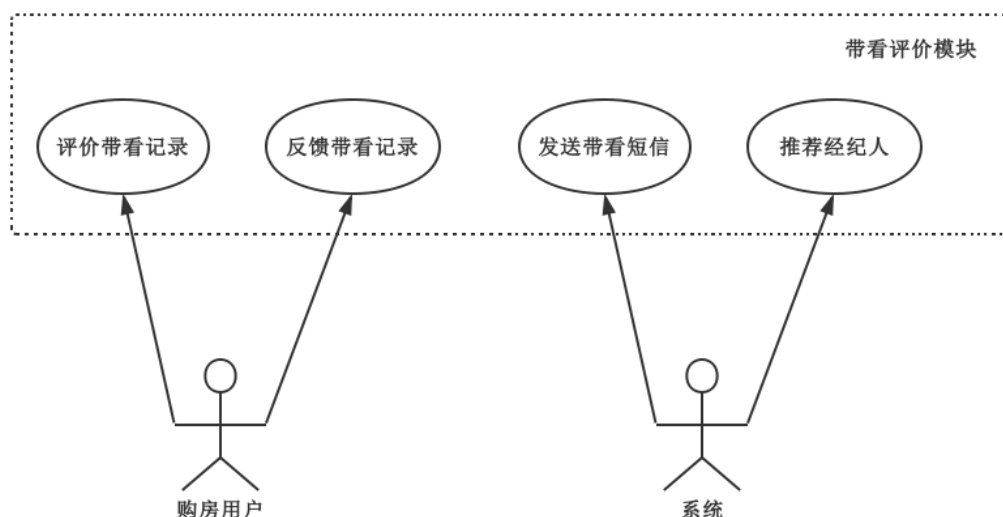


图 3.2 带看评价模块用例图

3.2.2 带看模块实体关系分析

带看评价模块的实体关系图如图 3.3 所示，涉及的实体包括带看经历实体、带看评价实体、经纪人实体、客户实体、部门实体、板块实体、转接号实体、主机号实体、带看房源实体、房屋销售记录实体、推荐经纪人实体。

每个经纪人每次带看只能接待一个客户，故带看经历与客户的对应关系为 1:1。

同理，带看经历与经纪人的对应关系为 1:1。

每个带看经历只能添加一次带看评价，带看经历与带看评价的关系为 1:1。

每次带看可能包含很多房源，故带看经历与带看房源关系为 1:N。

每个经纪人可能有多种渠道被推荐，故经纪人与转接号对应关系为 1:N。

每个经纪人会在不同情况下被联系，不同情况经历的主机号不同，故经纪人与主机号对应关系为 1:N。

每个经纪人只属于一个部门，所以经纪人与部门对应关系为 1:1。

每个部门只属于一个板块，所以部门与板块的对应关系为 1:1。

每个部门只属于一个父部门，所以部门与父部门的对应关系为 1:1。

每个房屋可能经过多次销售，所以房源和房屋销售记录对应关系为 1:N。

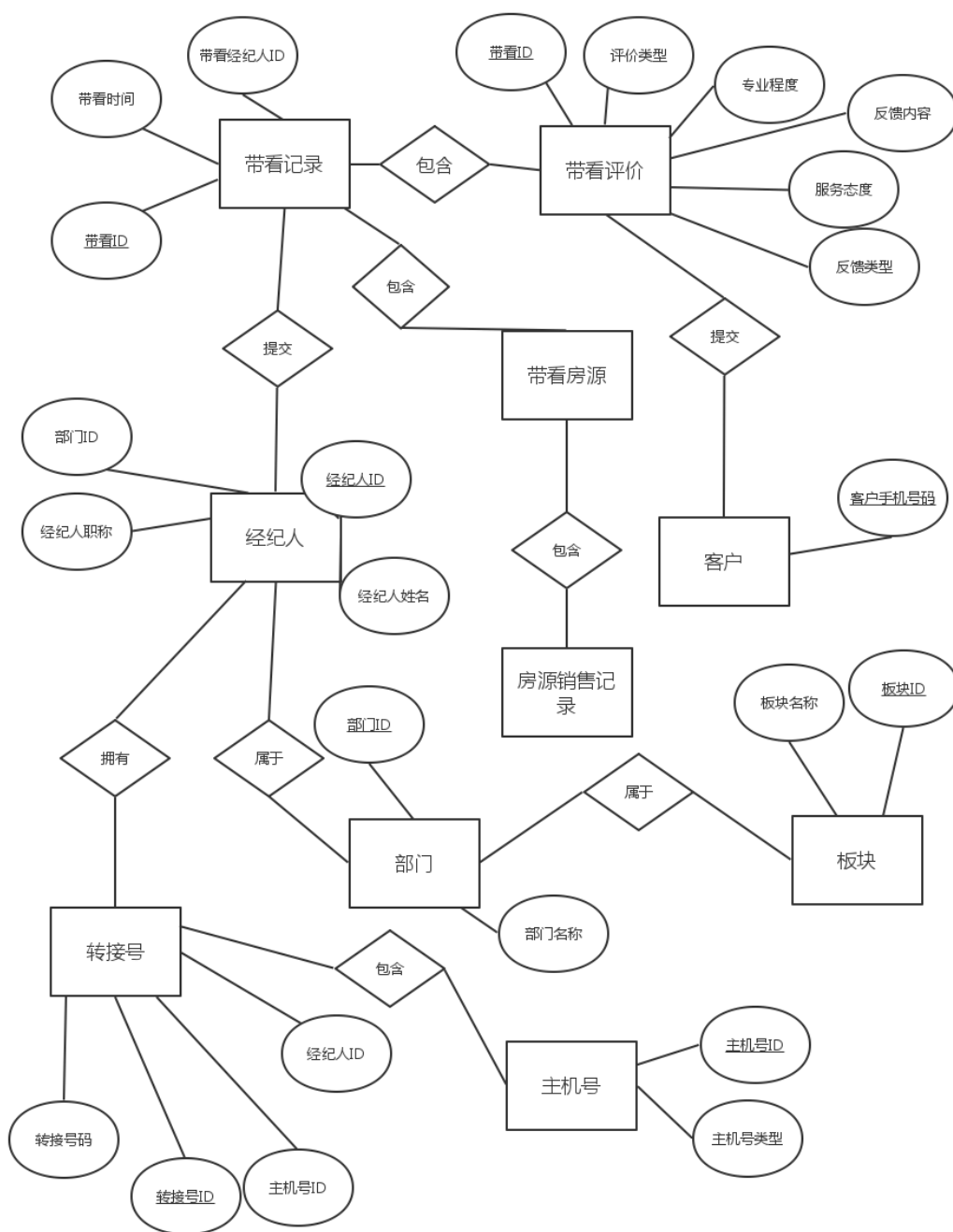


图 3.3 带看评价模块实体关系图

3.2 转接号模块系统需求分析

3.2.1 转接号模块用例描述

当用户需要电话联系经纪人时，需要拨号从链家的电话系统进行转接。

转接时号码分为主机号和转接号，主机号用于区别用户得到号码的方式，比如从移动端网页获取的主机号和从 pc 网页获取的主机号不同。通过区分主机号的方式，系统可以记录用户获得经纪人推荐的方式，便于以后的推广方式扩展。转接号为逻辑号码，用于区分实际联系的经纪人。通过主机号和转接号即可确定联系经纪人的实际号码。

转接号模块分为号码管理和电话转接两部分，都属于后端项目。电话管理的功能为转接号自动生成，转接号码获取服务。电话转接部分为真实号码获取，转接号数据同步。这几个功能在符合条件的情况自动执行从而取代了复杂的人工操作，因此都可以不面向用户独立执行。因此他们的用例描述如下表所示：

用例标识号	UC_01	用例名称	转接号码生成
参与者			
简要描述	房源信息变更时自动生成推荐经纪人转接号		
优先级	高		
触发条件	新房源上架或者新经纪人上任		
前置条件			
后置条件	同步数据给转接号模块		
正常流程	<ol style="list-style-type: none"> 1. 房源数据同步模块同步到新房源或者同步新经纪人数据 2. 检查是否有空余转接号，若有，将空转接号 and 对应主机号分给经纪人、若没有则批量生成空余转接号重新进行 2 3. 将添加的转接号同步给电话转接模块 		
拓展流程			
特殊需求			

表 3.2 转接号码生成用例

用例标识号	UC_02	用例名称	转接号码查询
参与者			
简要描述	用于其他模块查询转接号码		
优先级	高		
触发条件	用户希望得到经纪人的联系方式		
前置条件	进入按时间查询数据比较结果页面		
后置条件	显示数据比较结果		
正常流程	<ol style="list-style-type: none"> 1. 根据经纪人 ID 和房源信息和查询渠道获取转接号码接口 2. 根据转接号码获取经纪人实际号码 		

表 3.3 转接号码查询用例

用例标识号	UC_03	用例名称	转接号码同步
参与者			
简要描述	保证内网电话转接模块的数据和外网电话管理模块一致		
优先级	高		
触发条件	外网转接号数据变更		
前置条件	电话转接模块配置好外网链接信息		
后置条件	显示数据比较结果		
正常流程	<ol style="list-style-type: none"> 1 电话管理模块通过调用号码同步模块接口传送转接号数据给电话转接模块.如果发送失败，进入 1.a 2 电话管理模块每隔 10min 重新发送同步失败的临时数据 3 电话转接模块每隔 3 小时自动从电话管理模块抓取变更数据进行同步。 		
拓展流程	1a.将同步失败数据存入 temp 数据库保存		
特殊需求	新数据必须在 24 小时内同步成功		

表 3.4 转接号码同步用例

3.4 系统概要设计

3.3.1 总体设计

根据 3.2 中的需求分析可以知道，二手房对外网站系统面向购房用户，故而与之相关的模块应部署在对外服务器。经过考虑，二手房项目的外网系统主要部署在阿里云服务器，主要因为房产信息和交易信息存储数据量很大，使用阿里云的分布式服务可以一站式地完成数据的整合、清洗和加工。同时解决了 IP 规划、负载均衡的问题，有利于提升用户体验。

而电话转接模块和短信模块部署在本地，是他们依赖部署在本地的电信运营商的系统。中介人项目属于遗留系统，由固定的人员进行开发和维护故而保留原本的服务部署。

如图 3.4 所示为系统模块结构图，在该系统结构下，每块服务共享同一数据库，各司其职。

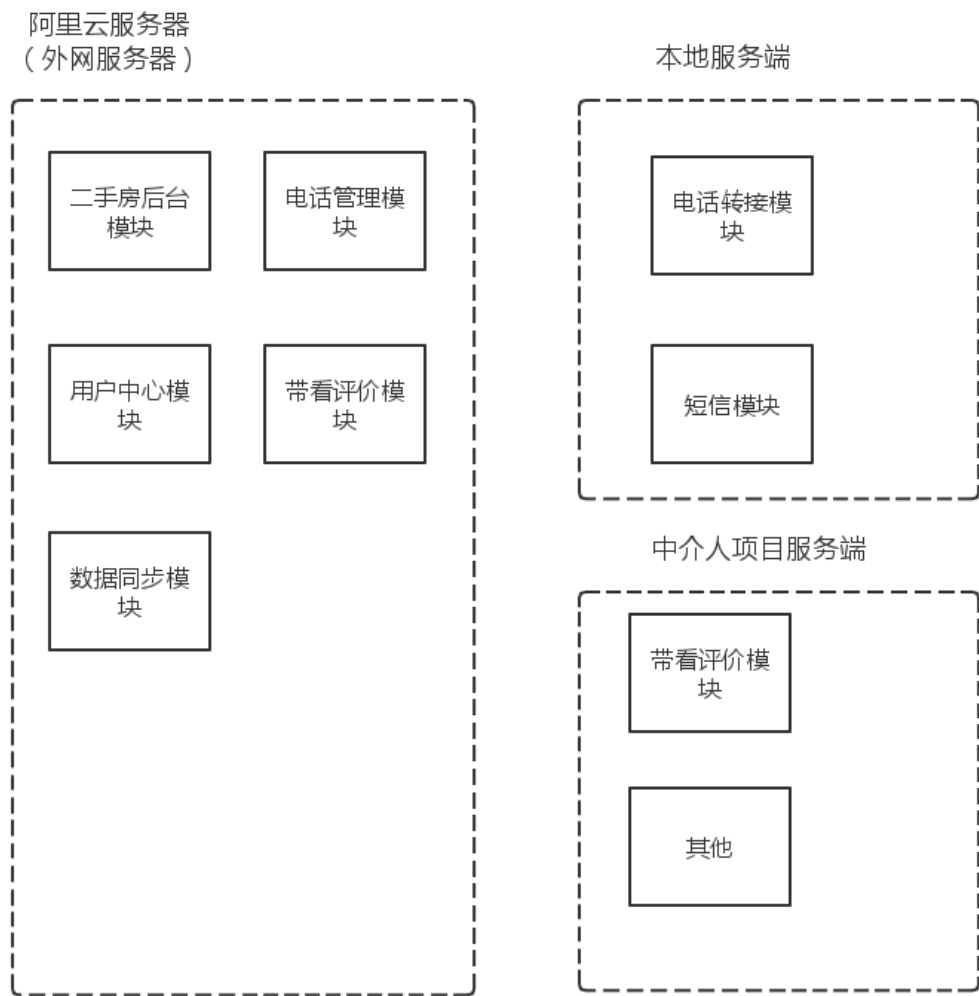


图 3.4 二手房交易系统模块部署

如图 3.5 所示，模块内使用分层结构。Controller 层负责接口和网址定位，Service 层负责处理业务逻辑，Dao 层定义数据库访问函数，根据 Mybatis 框架，由同名的 xml 文件实现数据库访问函数。

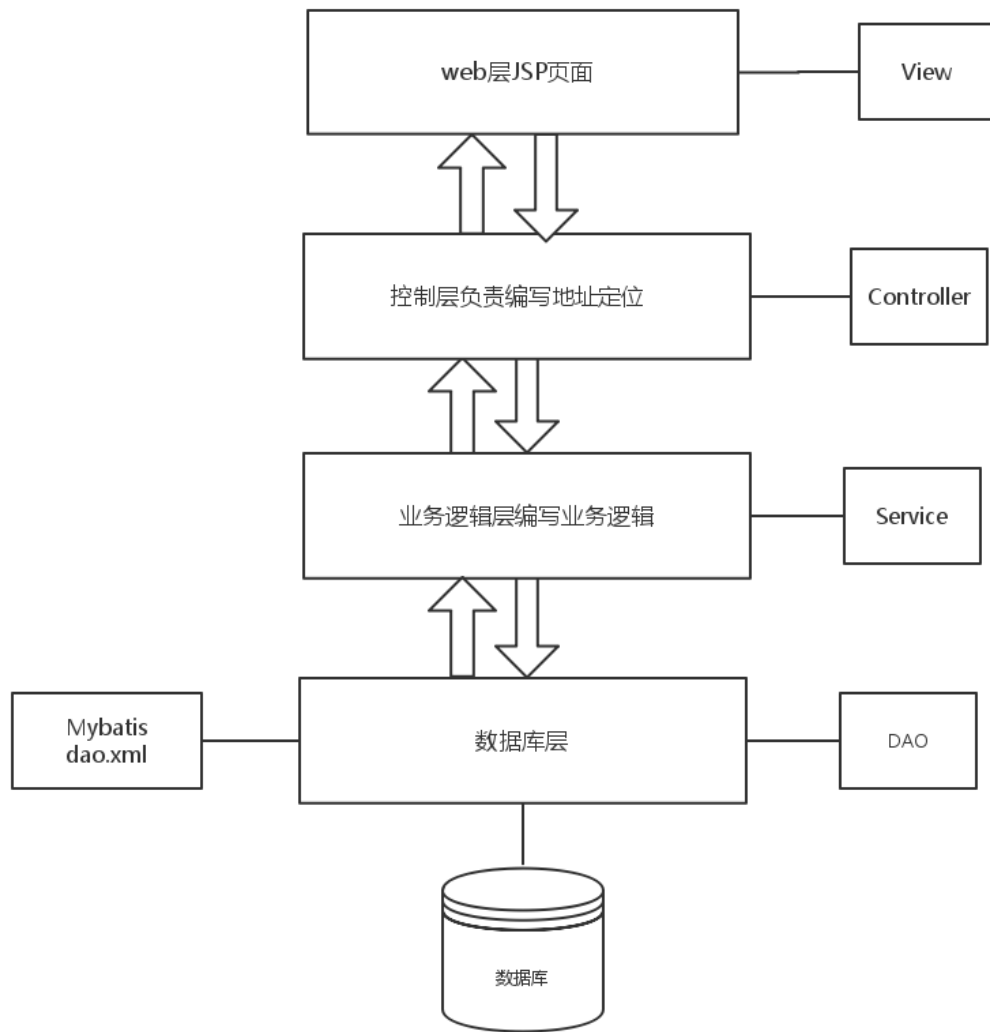


图 3.5 通用模块程序结构图

3.3.2 带看评价模块设计

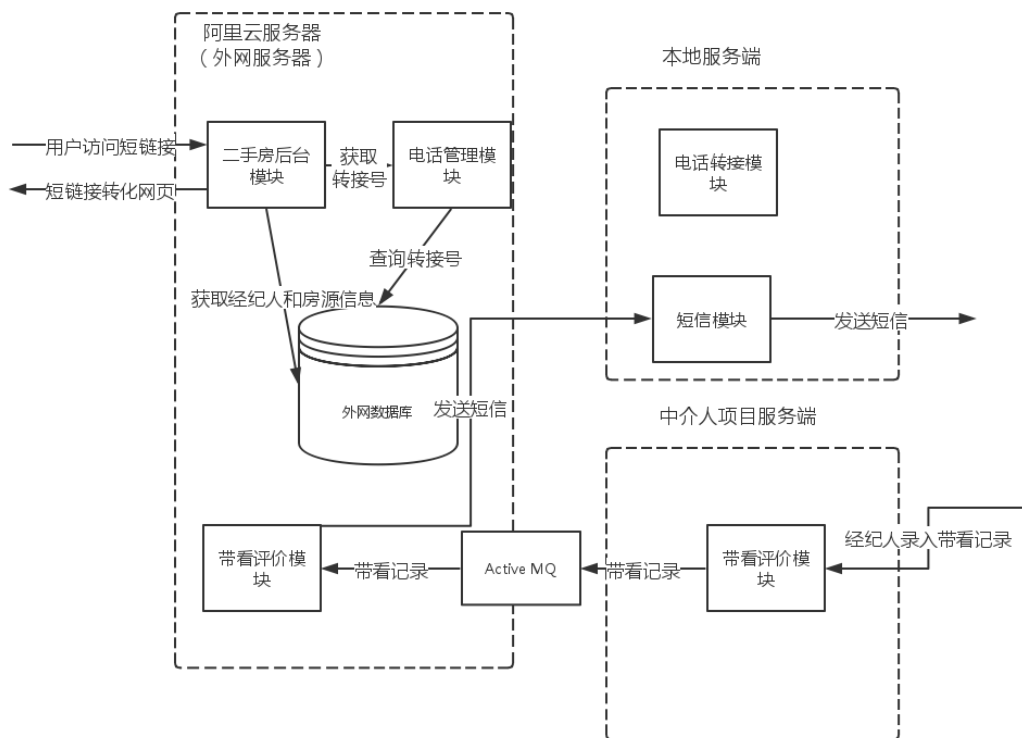


图 3.6 带看评价逻辑架构

带看评价模块和其他模块间关系如图 3.6 所示，由于使用 MQ 消息与中介服务器传递带看记录，故而需要定义带看记录传递格式。其余需要定义的接口还有调用短信模块的短信发送接口，给前端查看和评价带看记录的接口。具体定义见表 3.5：

表 3.5 TakeWatch 接口定义

方法名	描述
postWatchRemark	提交带看评价
postAccuracyRemark	提交带看反馈
getWatchInfo	获取带看信息
ListenMQService. onMessage	接收带看记录
SendMQService. SendMessage	发送带看记录

对应提供给前端的接口定义和返回消息体如图 3.7-3.10 所示

短信-短信发送

接口维护人员：常雨	
http://duanxin.dooioo.org/api/tools/sendMsg	
所需授权	系统管理员
请求方法	GET POST
请求URL	http://open.dooioo.org/api/v4/duanxin/tools/sendMsg
响应消息体示例	<p>请求参数：</p> <p>phone 必需 手机号(多个手机用，分隔)</p> <p>msg 必需 短信内容</p> <p>project 必需 项目名称(首次使用接口，需向短信中心负责人申请)</p> <p>sendTime 非必需 定时发送时间，无定时需求则不填</p> <p>userCode 非必需 经纪人工号</p> <p>remark 非必需 短信备注</p> <p>channel 非必需 发送通道</p> <p>company 非必需 所属公司，若非德佑公司，必须加上此字段</p> <p>请求响应</p> <pre>{ "status": "ok", "message": "" }</pre>

图 3.7 短信发送接口

用户中心-带看举报反馈

接口维护人员：邹源

http://192.168.3.85:8100/openApi/takeWatch/postAccuracyRemark.json

所需授权

系统管理员

请求方法

POST

请求URL

http://192.168.3.85:9000/api/v4/online/favor/ershoufang/postAccuracyRemark

请求参数

名字		类型	必填	默认值	备注
watchId	带有编号	long	是		
accuracyCode	举报类型	string	是		
accuracyRemark	举报描述	string	是		

响应消息体示例

```
{"errno":0,"status":"ok"}
```

```
{"errno":0,"error":"你已经对带看情况做出过反馈","status":"fail"}
```

图 3.8 带看评价反馈接口

用户中心-带看评价					
接口维护人员：邹源					
http://192.168.3.85:8100/openApi/takeWatch/postWatchRemark.json					
所需授权	系统管理员				
请求方法	POST				
请求URL	http://192.168.3.85:9000/api/v4/online/favor/ershoufang/postWatchRemark				
请求参数	名字	类型	必填	默认值	备注
	watchId	带看编号long	是		
	knowledge	专业度int	是		
	attitude	服务态度int	是		
	suggestion	建议string	是		
响应消息体示例					
{ "errno": 0, "status": "ok" }					
{ "errno": 0, "error": "你已经对带看情况做出过反馈", "status": "fail" }					

图 3.9 带看评价已经提交、返回推荐经纪人接口

用户中心-获取带看详情

接口维护人员：邹源

http://192.168.3.85:8100/openApi/takeWatch/getWatchInfo.json

所需授权

系统管理员

请求方法

GET

请求URL

http://192.168.3.85:9000/api/v4/online/favor/ershoufang/getWatchInfo

请求参数

名字	类型	必填	默认值	备注
watchId	带看编号 long	是		

响应消息体示例

```
{
  "errno": 0,
  "hasComment": 1,
  "status": "ok",
  "agents": [
    {
      "avatar": "http://fdfs.dooioo.com/fetch/vp/touxiang/photos/81916_150x200.jpg",
      "commentCount": 0,
      "extNumber": "10982",
      "hostNumber": "4008444008",
      "lookCount90": 0,
      "orgName": "幸福店",
      "phone": "4008444008,10982",
      "plateName": "徽桥",
      "soldCount": 0,
      "takeWatchId": 0,
      "userCode": "81916",
      "userName": "黄乔贵",
      "workYear": 1
    }
  ]
},
{
  "errno": 0,
  "data": {
    "agent": {
      "avatar": "http://fdfs.dooioo.com/fetch/vp/touxiang/photos/82248_150x200.jpg",
      "commentCount": 0,
      "extNumber": "10988",
      "hostNumber": "4008444008",
      "lookCount90": 0,
      "orgName": "静安新城二店",
      "phone": "4008444008,10988",
      "soldCount": 0,
      "takeWatchId": 0,
      "userCode": "82248",
      "userName": "郭洪涛",
      "workYear": 1
    },
    "houseList": [
      {
        "acreage": 70.44,
        "allowedPhoto": "/static/img/default_block.png",
        "chineseTags": "地铁房",
        "englishTags": "is_subway_house",
        "face": "",
        "floorTypeName": "低层",
        "hall": 1,
        "houseSellId": "100000",
        "houseServerUrl": "http://10.8.1.42:8020",
        "isKeyHouse": false,
        "isNew": false,
        "isSole": false,
        "mainPhotoUrl": "http://fdfs.dooioo.com/fetch/vp/fy/gi/2013/1023/8DE8BD6DF63241D5895F078B8B3FC733.jpg_200x150.jpg",
        "metroRemark": "距离9号线南城路站928米",
        "propertyName": "久阳滨江公廨",
        "putAway": 2,
        "room": 1,
        "showPrice": 205,
        "soldPrice": 205,
        "tags": ["is_subway_house"],
        "unitPrice": 30565
      }
    ],
    "hasComment": 0,
    "lookDate": "三月22日 星期二",
    "lookCount": 1,
    "status": "ok"
  }
}
```

图 3.10 带看评价详细信息浏览接口

3.3.3 转接号模块设计

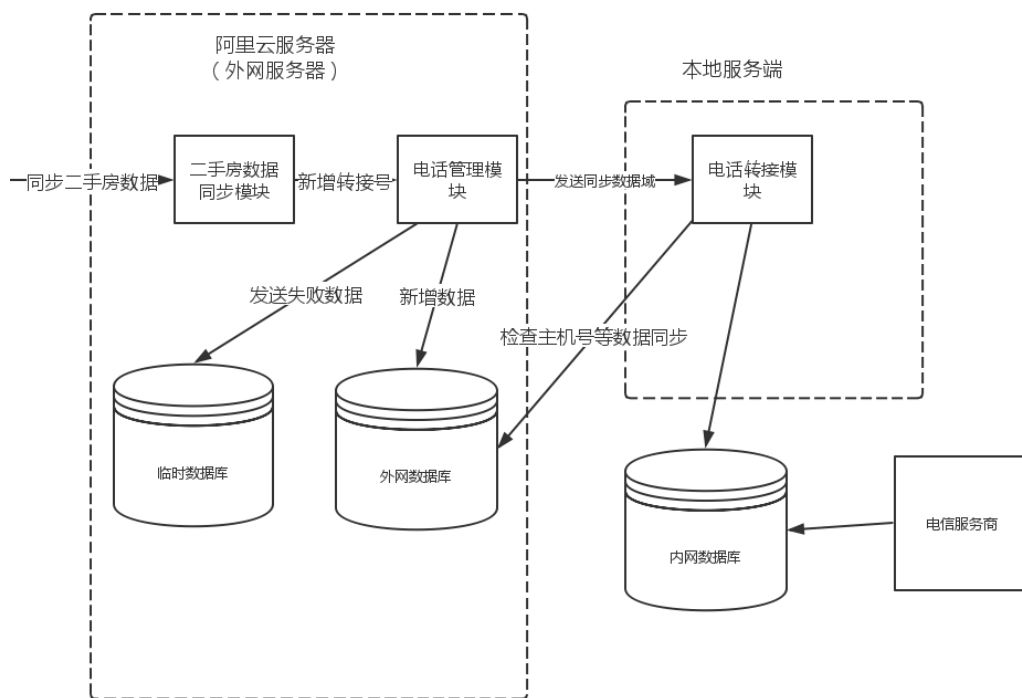


图 3.11 转接号逻辑架构

转接号模块和其他模块间关系如图 3.11 所示，电话管理模块需要提供一个接口给二手房数据同步模块用来调用自动增加转接号的服务，电话转接模块需要提供一个接口给电话管理模块用来提交同步数据。对应接口设计见表 3.6：

表 3.6 DesignService 接口方法表

方法名	描述
getHostNumber	根据访问类型获取主机号
getBrokersExt	获取根据 key, model, source, keyType 获取分机信息 的拦截器
checkAndCreateEmpExtPool	批量生成经纪人转接号码并同步进 online 数据库
postExtNumber	接收提交的转接号数据

3.4 本章小结

本章主要对二手房交易系统的电话转接模块和带看评价模块进行了详细的需求分析和概要设计。对于整个系统详细描述了其模块划分，接着针对电话转接模块属于后端模块、带看评价模块面向用户的不同特点使用用例图、系统顺序图和活动图等 UML 图对该系统进行需求分析，包括功能性需求和非功能性需求，使用 ERD 图描述了系统的数据库设计。另外，本章还介绍了相关模块之间的接口，为后续系统的详细设计与实现做好了充分的准备。

第四章 详细设计与实现

4.1 带看评价模块概述

二手房交易网站的带看评价功能分为外网部分和经纪人部分。外网部分是基于 Spring 框架开发的移动网页系统，只有看房者一类用户，其余功能均由后台程序独立自动进行。成如 3.2 部分所述，看房者的操作为打开短信内的 url，查看带看记录，提交带看评价或者带看反馈，查看重新推荐的经纪人。后台自动进行的操作有：自动接收录入的带看记录，发送带看短信，同步带看记录给经纪人服务器，生成推荐经纪人。

4.2 带看评价模块的详细设计

4.2.1 设计部模块详细类图和顺序图

带看评价模块面向用户的核心类是控制层中的 `TakeWatchController` 类，与之相关的关键类图如图 4.1 所示，前后端的异步数据使用 JSON 传递，`ApiJsonResult` 为返回给前端的标准类型。`Ok` 和 `Fail` 方法分别对应成功和失败。`ERRNO` 用于标记错误类型，如 `ERRNO_NOT_FIND = -100` 表示未找到相关数据，`ERRNO_NOT_LOGIN = -102` 表示未登录。结合 http 的错误代码，前端可以更好得进行异常处理。`TakeWatchController` 通过调用 service 层的方法实现业务逻辑。其中 `TakeWatchService` 用于查询和修改 `TakeWatch` 数据，`SendMqService` 用于将更新后的数据发送给经纪人服务器，`EmpService` 用于查询经纪人数据和生成推荐经纪人。两个 Dao 文件定义了数据库函数，由同名的 xml 文件实现函数。具体的流程如图 4.2 所示。

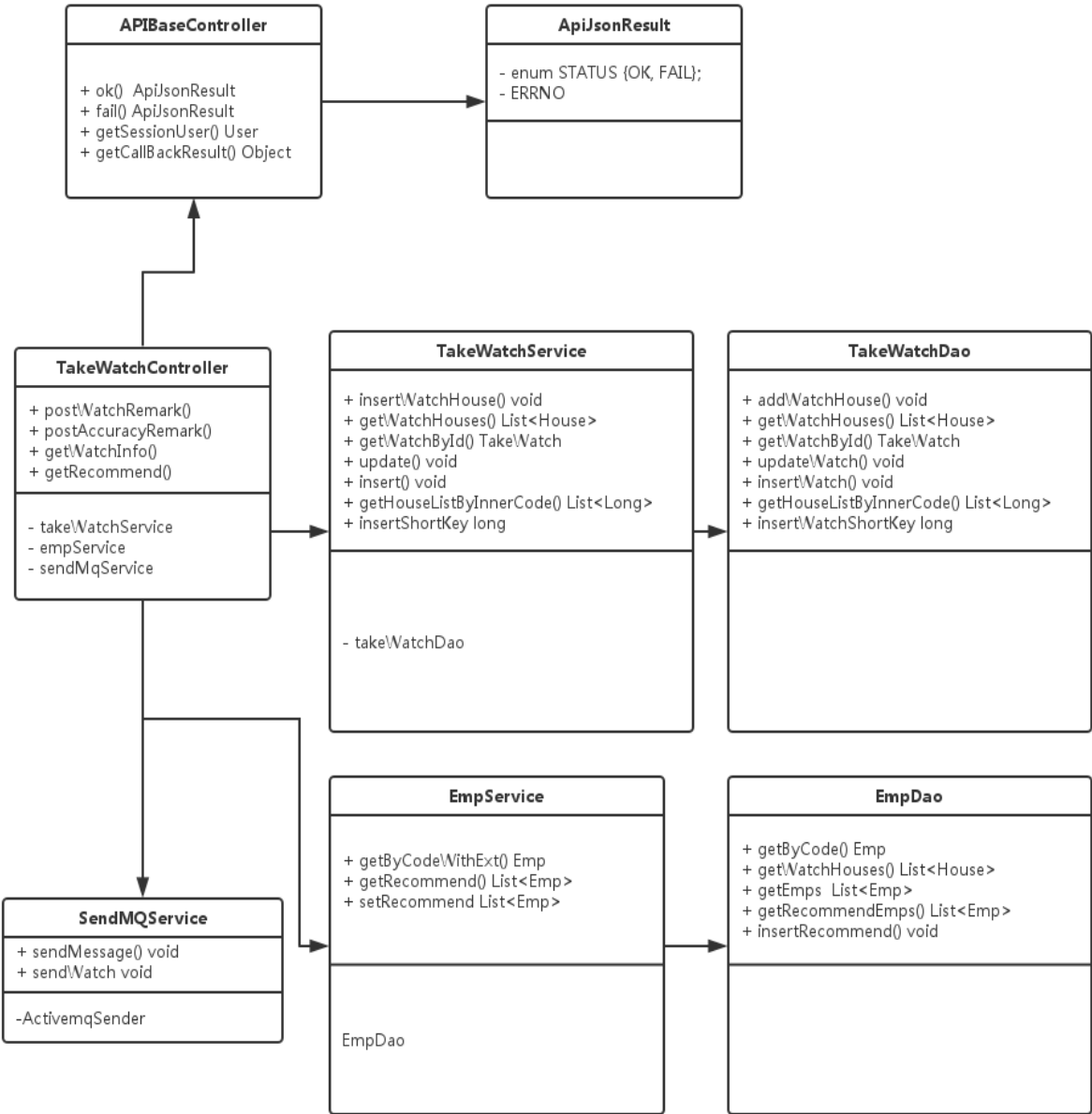


图 4.1 生产部模块详细类图-面向用户

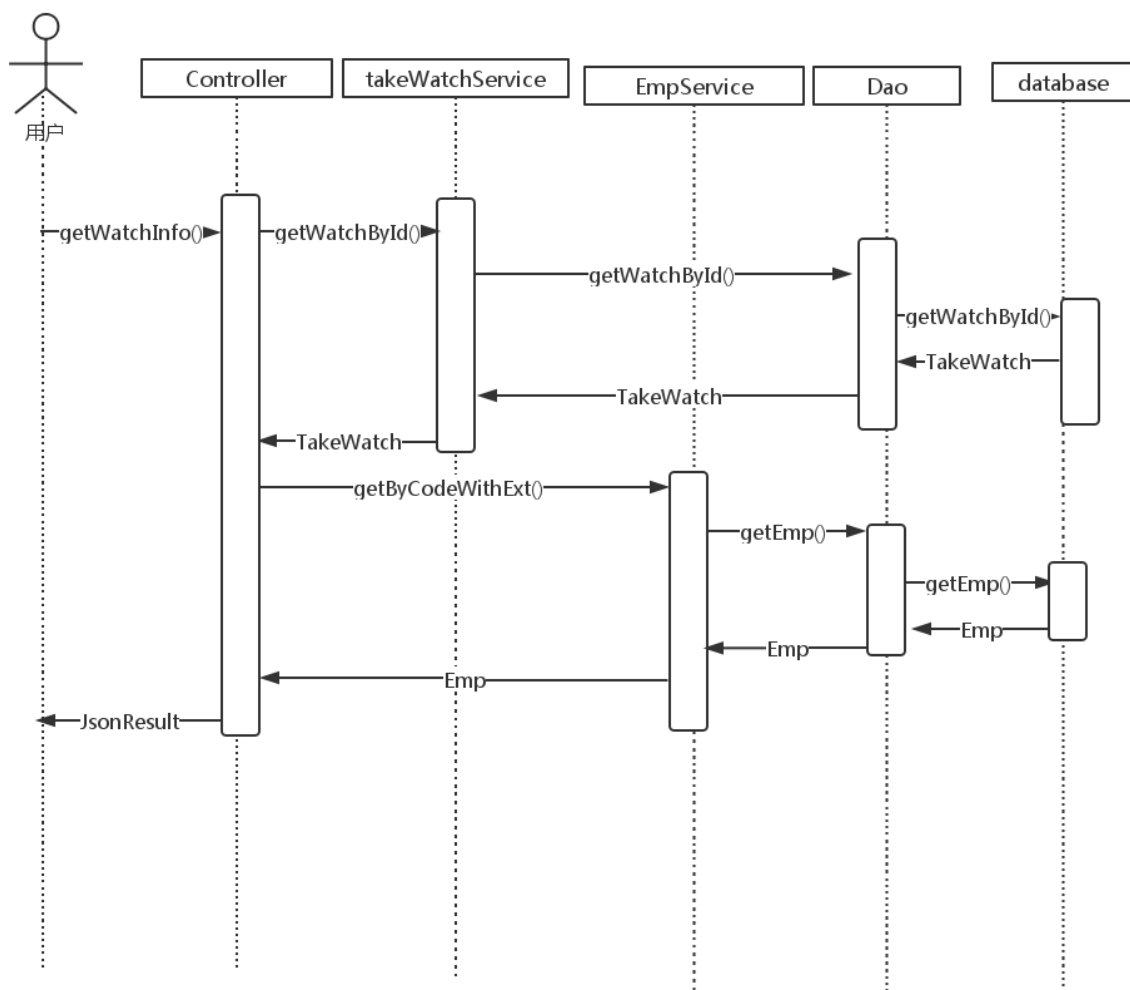


图 4.2 生产部模块顺序图-面向用户

图 4.3 以 ListenMqService 为核心展示了接收新的带看信息的关联类图。ListenMqService 用 onMessage 方法接收和处理经纪人服务器送入队列的新带看记录,调用 TakeWatchServicef 服务将数据存入本地数据库,调用 PhoneService 生成短信并发送给用户。发送给用户的短信需要将初始链接转换成短链接从而减少短信长度,短链接转换由 Base62 实现。具体的流程见图 4.4

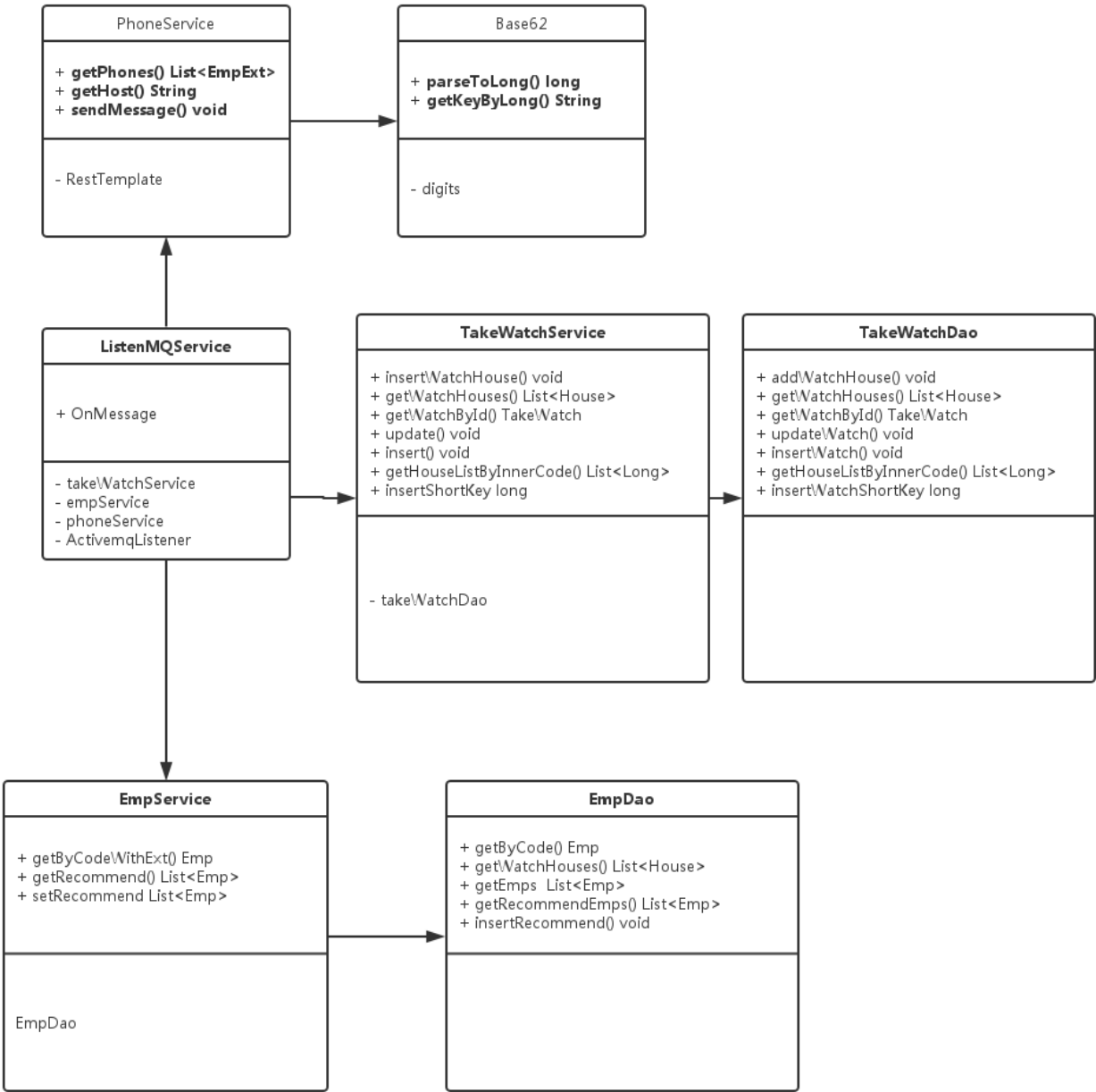


图 4.3 带看评价模块详细类图-面向经纪人服务器

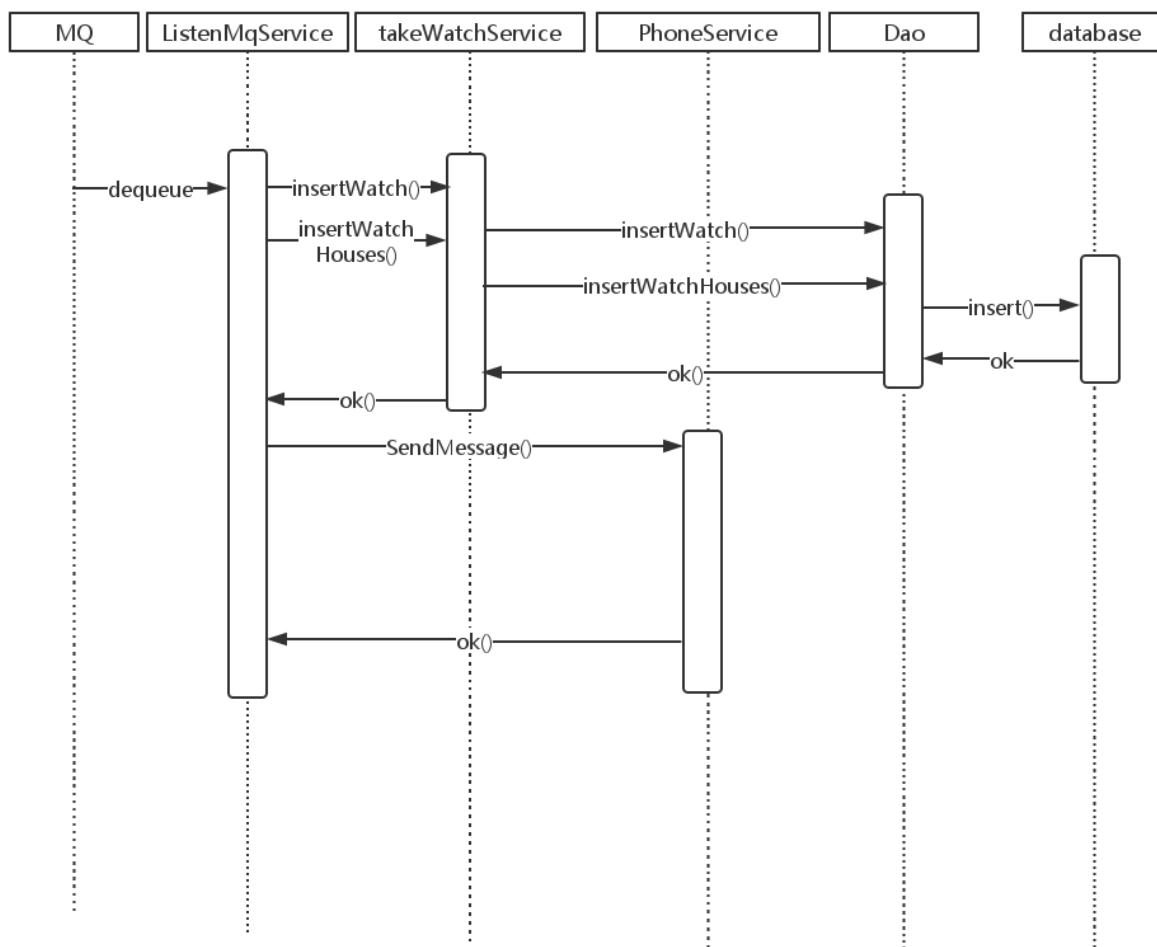


图 4.4 带看评价模块顺序图-面向经纪人服务器

最后是实体类模块的详细设计，由于实体类是为便利业务逻辑层中的模块同数据访问模块间的调用而设计的，这里借鉴对象关系映射的设计思想，给出与数据库表相对应的实体类，如下图 4.5 所示，到它们有 House 类、Emp 类、TakeWatch 类和 User 类

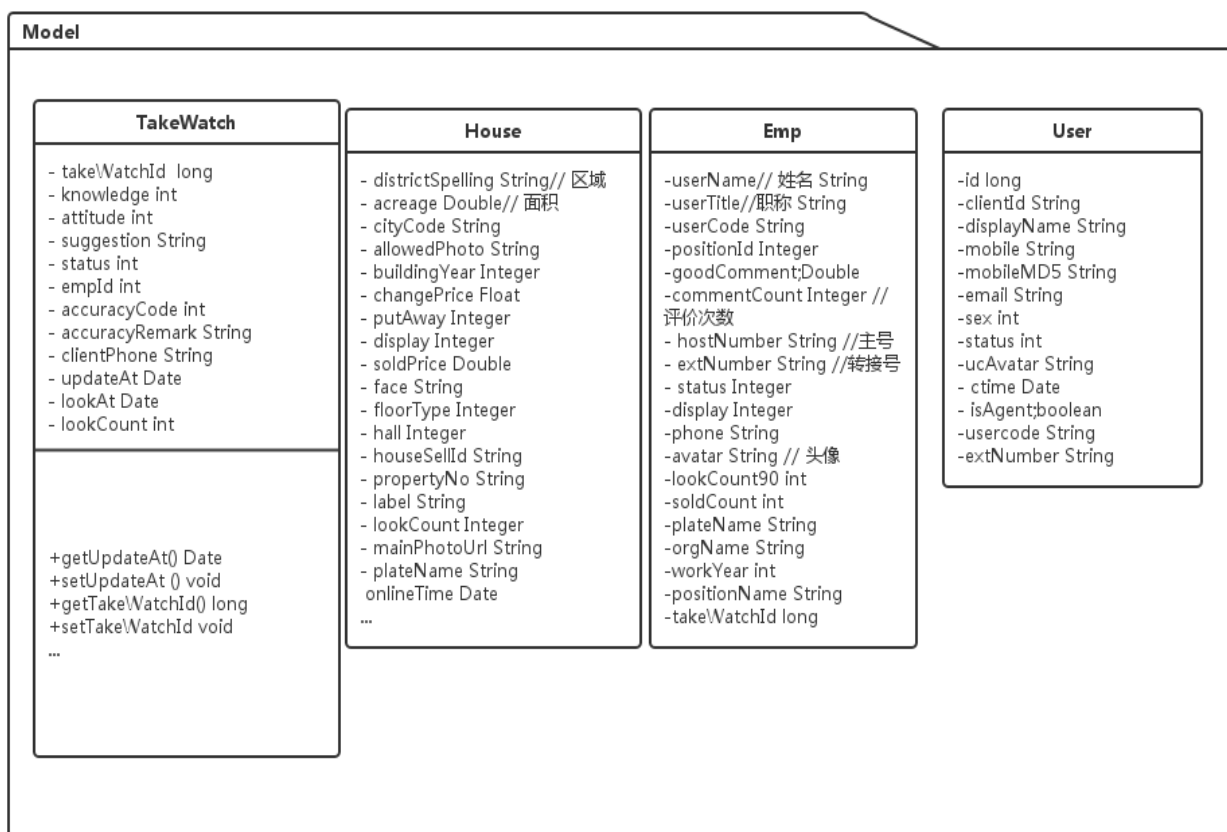


图 4.5 带看评价模块 Model 包类图

1.2.2 带看评价短链接功能设计

短链接（ShortLink）顾名思义就是长度比较短的网址，比如 <http://is.gd/w>。早期，短链接主要应用在网站的图片上传中，上传网站通过特定的算法缩短 URL（UniformResourceLocator，统一资源定位符）链接地址，达到减少代码字符的目的^[8]。

在二手房交易系统发送给用户的带看评价短信中使用短链接有以下几个原因：

- (1) 通信服务商短信发送字数限制
- (2) 给带看经纪人、时间、房源信息留下给多的空间
- (3) 避免用户恶意访问其他用户的带看评价
- (4) 规范 URL 格式，使其阅读起来更正式和规范

为了实现端连接功能，结合业界常用的思路，初步设计出的实现方法如下：

MD5 加密+Hash 方法+Base62:

- (1) 使用 MD5 加密将长链接转化为 32 位字符串
- (2) 将长度为 32 位的字符串分割为四段长度为 8 的字符串
- (3) 将每段视为 32 位的 16 进制数，取后 30 位，分割为 6 段，每段长度为 5
- (4) 取每段长度为 5 的字段模 62 的余数，利用 Base62 加密组合成一个长度为 6 的字符串

MD5 加密+Hash:

- (1) 使用 MD5 加密将长链接转化为 32 位字符串
- (2) 将长度为 32 位的字符串分割为四段长度为 8 的字符串
- (3) 将每段视为 32 位的 16 进制数，取后 30 位，分割为 5 段，每段长度为 6
- (4) 取一段长度为 6 的字符串并进行 Hash

Base62 加密:

- (1) 将永不重复的带看评价 ID 利用 Base62 加密转换为 62 进制 (A-Z, a-z, 0-9) 的字符串 key。
- (2) 将带看评价 ID 与 key 的对应关系记入数据库。
- (3) 利用 key 重组 url

时间+带看增量:

- (1) 利用每天带看评价的数量有限，将日期时间 (2 位) 作为关键字段前缀，用当日第几次带看作为后缀生成 key。
- (2) 将带看 ID 与 key 对应关系记入月数据库。
- (3) 默认超出一个月的带看记录失效，定期将月数据库中超过 30 天的信息存入总库。
- (4) 利用 key 重组 url，url 定位仅从月数据库中查询。

不同方法的优缺点比较见表 4-1:

方法	实现复杂度	短链接格式	安全性
MD5 加密+Hash 方法+Base62	短链接生成算法 复杂，耗时长	短链接格式规则 性不强	安全性最高

MD5 加密+Hash	计算方法简单,但是碰撞概率高,短链接生成时间不固定	短链接格式规则性不强	安全性高
Base62	生成方法简单,无碰撞,计算时间稳定	规则性较弱	用户很难恶意修改带看评价中的key
时间+带看增量	生成算法简单,无碰撞,计算时间稳定	规则性很强	可以通过修改 url 中的key 访问其他用户的带看评价

表 4.1 短链接算法优越点比较

在算法稳定性和效率上 Base62 和时间结合带看增量上具有优势。由于审核部门认定,带看评价只有在用户及时评价的情况下才真正有效,一般认定超过三天的带看评价不具备价值。所以在安全性上只需要防范用户恶意修改 url 查看其他用户的带看评价即可,Base62 和 MD5 加密都符合要求。而使用 Base62 格式可以有效的规范访问 URL,使短链接更简洁清晰,避免用户担心链接指向黑网站,故而最终采用 Base62 生成短链接。

4.3 带看评价模块实现

4.3.1 带看评价模块数据层实现

带看评价数据层 Mybatis 配置文件如图 4.6 所示。数据库连接参数由 jdbc.properties 设置,如图 4.7 所示,为保证数据库安全,其密码为 md5 加密字符串,在图中已经略去。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>
  <settings>
    <setting name="lazyLoadingEnabled" value="false" />
  </settings>

  <typeAliases>
    <typeAlias alias="integer" type="java.lang.Integer" />
    <typeAlias alias="string" type="java.lang.String" />
  </typeAliases>
</configuration>
```

```

<typeAlias alias="bigDecimal" type="java.math.BigDecimal" />
<typeAlias alias="long" type="java.lang.Long" />
<typeAlias alias="date" type="java.util.Date" />

<package name="com.lianjia.user.model" />
</typeAliases>

</configuration>

```

图 4.6 配置文件 myBatis-config

```

#SqlServer
#development
development.online.jdbc.url=jdbc:sqlserver://10.8.1.112:1433;DatabaseName=lianjiaOnline;sendStringParametersAsUnicode=false
development.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
development.online.jdbc.username=online
development.online.jdbc.password=//密码

#test
test.online.jdbc.url=jdbc:sqlserver://10.8.1.112:1433;DatabaseName=lianjiaOnline;sendStringParametersAsUnicode=false
test.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
test.online.jdbc.username=online
test.online.jdbc.password=//密码

#integration
integration.online.jdbc.url=jdbc:sqlserver://192.168.3.66:1433;DatabaseName=lianjiaOnline;sendStringParametersAsUnicode=false
integration.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
integration.online.jdbc.username=online
integration.online.jdbc.password=//密码

#production
production.online.jdbc.url=jdbc:sqlserver://10.22.20.90:53435;DatabaseName=lianjiaOnline;sendStringParametersAsUnicode=false
production.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
production.online.jdbc.username=online
production.online.jdbc.password=-//密码

```

图 4.7 配置文件 jdbc.properties

Dao 层中的 TakeWatchDao、EmpDao 定义了对数据库操作的函数,代码如图 4.8 所示。其中@CheckCache 使用了缓存技术保存搜索结果,60*60*4 表示 cache 有效时间为四小时。其对应的实现由 Mybatis 中的 TakeWatchDao.xml 和 EmpDao.xml 实现,其中 TakeWatchDao.xml 的关键代码如图 4.9 所示。图中展示了插入短链接关键字、查找带看房源列表更新用户信息等方法。图 4.10 展示了推荐数据库函数实现的挑选经纪人算法,其排序规则为: 90 天经纪人成交量*100+同房源带看次数*100+同小区房源带看次数*1+近 14 天带看房源次评分(带看 5 次以下次数-带看 6 次以上数目)*10-近 30 天内评价被推荐次数*20,对经纪人的约束条件为来自初始带看经纪人的同一板块的不同部门,不处于离职、休息状态。推荐经纪人最多为三人,不存在符合要求的经纪人时不推荐。

```
public interface TakeWatchDao {
    public TakeWatch getTakeWatchByPhone(@Param("clientPhone") String clientPhone);
    public TakeWatch getTakeWatchById(@Param("takeWatchId") Long takeWatchId);
    public void insertWatch(TakeWatch watch);
    public void updateWatchInfo(TakeWatch watch);
    public void addWatchHouse(@Param("takeWatchId")Long
        takeWatchId,@Param("houseId") Long houseId);
    public List<House> getWatchHouses(@Param("takeWatchId")Long takeWatchId);
    public void insertWatchShortKey(Map<String,Object> data);
    public List<Long> getHouseListByInnerCode(@Param("list") List<String> list);}

    public interface EmpDao {
        @CheckCache(timeToLive = 60*60*4)
        public Emp getByCode(Integer userCode);
        public List<Emp> getEmps(Emp takeWatchEmp);
        public List<Emp> getRecommendEmps(TakeWatch watch);
        public void insertRecommend(@Param("takeWatchId")long
            takeWatchId,@Param("userCode") String userCode);}
    }
```

图 4.8 EmpDao 和 takeWatchDao 接口类代码

```
<mapper namespace="com.lianjia.user.dao.TakeWatchDao">

    <select id="getTakeWatchById" resultType="TakeWatch">
        select top 1 * from t_user_take_watch
        where takeWatchId = #{takeWatchId}
    </select>

    <update id="updateWatchInfo" parameterType="TakeWatch">
        update t_user_take_watch set
            suggestion=#{suggestion},status=#{status},updateAt=getdate(),knowledge=#{
            knowledge},attitude=#{attitude},accuracyCode=#{accuracyCode},accuracyRe
```

```

        mark=#{accuracyRemark}
        where takeWatchId=#{takeWatchId}
</update>

<insert id="insertWatchShortKey" useGeneratedKeys="true" parameterType="map">
    insert into t_user_take_watch_shortKey (takeWatchId,mobile,createAt)
    values (#{takeWatchId},#{mobile},getDate())
    <selectKey resultType="long" order="AFTER" keyProperty="returnId">
        SELECT @@IDENTITY as returnId
    </selectKey>
</insert>

<select id="getWatchHouses" resultType="House">
    select d.acreage,d.cityCode,e.face,e.hall,e.id as
houseSellId ,e.mainPhotoUrl,b.propertyName,
    e.room,e.showPrice,e.label,e.putAway,s.soldPrice,e.title,d.cityCode,
    replace(rtrim(CONCAT((case when b.metroRemark!=" then
    'is_subway_house ' else " end)
    ,(case when b.schoolRemark!=" then 'is_school_house ' else "
    end),isNull(e.label,"))),','') as englishTags
    ,e.title,b.metroRemark,b.schoolRemark
    from t_user_take_watch_house a
    inner join t_house d with(nolock) on d.id=a.houseId
    inner join t_property b with(nolock) on b.propertyNo=d.propertyId
    inner join t_house_sell e with(nolock) on e.houseId=a.houseId
    left join t_house_sold s with(nolock) on e.id=s.houseSellId
    where a.takeWatchId=#{takeWatchId}
    order by a.id
</select>

</mapper>

```

图 4.9 TakeWatchDao.xml 关键代码

```

select top 3
b.userCode,b.userTitle,b.userName,b.extNumber,DATEDIFF(YEAR ,b.newJoinDate,getDate()) as workYear,b.houseSoldCount as soldCount,
    e.orgName,q.plateName,ISNULL(g.lookCount90,0)as lookCount
from t_emp b with(nolock)
left join t_organization d with(nolock) on b.orgId=d.id
left join t_organization e with(nolock) on e.id=d.parentId
left join t_emp_major_plate p with(nolock) on p.userCode=b.userCode
left join t_plate q with(nolock) on p.plateId=q.id
left join (SELECT t.empId,SUM(t.lookCount)as lookCount90 from
t_user_take_watch t WHERE DATEDIFF(DAY ,t.lookAt,getDate()) <= 90 GROUP BY

```

```

t.empld) g
    on g.empld=b.userCode
    left join (SELECT tt.empld,SUM(tt.lookCount)as lookCount14 from
t_user_take_watch tt WHERE DATEDIFF(DAY ,tt.lookAt,getDate()) < 14 GROUP BY
tt.empld) gg
    on gg.empld=b.userCode
    left join (select max(ttt.soldDate)as soldDate,ttt.soldUserCode from
t_house_sold ttt GROUP BY ttt.soldUserCode) st
    on st.soldUserCode = b.userCode
    left join (SELECT tg.userCode,count(tg.takeWatchId)as recommendCount30
from t_emp_recommend_take_watch tg WHERE
DATEDIFF(DAY ,tg.updateDate,getDate()) < 30 GROUP BY tg.userCode) ggg
    on ggg.userCode=b.userCode
    left join ( select count(bg.takeWatchId) as takeHouseCount,bg.empld
    from
    (SELECT t_h.houseId,t_w.empld,t_h.takeWatchId from
t_user_take_watch_house t_h
    left join t_user_take_watch t_w on t_w.takeWatchId = t_h.takeWatchId
    where EXISTS (SELECT ttthhh.* from t_user_take_watch_house ttthhh
where ttthhh.houseId=t_h.houseId and ttthhh.takeWatchId=#{takeWatchId})
    )bg
    GROUP by bg.empld
    ) thouse
    on thouse.empld = b.userCode

    left join ( select count(bg2.takeWatchId) as takeHouseCount,bg2.empld
from
    (SELECT t_h2.houseId,t_w2.empld,t_h2.takeWatchId from
t_user_take_watch_house t_h2
    left join t_user_take_watch t_w2 on t_w2.takeWatchId = t_h2.takeWatchId
    inner join t_house t_pro on t_pro.id = t_h2.houseId
    where EXISTS (
        SELECT ttthhh2.* from
        t_user_take_watch_house ttthhh2
        inner join t_house t_pro2 on t_pro2.id = ttthhh2.houseId
        where t_pro2.propertyId=t_pro.propertyId and
ttthhh2.takeWatchId=#{takeWatchId}
    )
    )bg2
    GROUP by bg2.empld
    ) tproperty
    on tproperty.empld = b.userCode

where b.positionId=1 and b.status!=-1 and not EXISTS (SELECT hh.* from

```



```
t_black_emp hh where hh.userCode=b.userCode)
    and e.orgName!=#{orgName}
    order by
100*(ISNULL(thouse.takeHouseCount,0))-20*ISNULL(ggg.recommendCount30,0)
+10*(5-abs(ISNULL(gg.lookCount14,0)-5))+ISNULL(tproperty.takeHouseCount,0)
DESC, st.soldDate DESC ;
```

图 4.10 EmpDao.xml 中的经纪人推荐算法

4.3.2 带看评价模块服务层实现

带看评价模块的service层主要包括：SendMQService, ListenMQService, TakeWatchService, EmpService，他们通过调用Dao类提供的方法实现逻辑层次的功能。图4.10中的Listen- MqService.OnMessage方法完成了从Mq中接收到带看评价后的逻辑操作。在从Mq队列中取出字符串后，方法先使用阿里的FastJson工具将字符串转换成JSON对象，然而根据Json对象生成takeWatch实例和带看房源的List,调用takeWatchService将数据保存到本地数据库，调用Base62的方法加密，调用phoneService接口发送短信。图4.11展示了phoneService的短信发送方法，图4.12展示了Base62的加密方法，通过调用Base62的getKeyByLong()方法，可以将long类型的不重复id进行62进制转换，每位对应为预设的一个字符。通过预设的长度可以使生成的字符串长度规范。

```
public void onMessage(String advJson) throws JMSEException {
    JSONObject obj = JSONObject.parseObject(advJson);
    JSONArray arr = obj.getJSONArray("watchPropertyList");
    TakeWatch watchInfo = new TakeWatch();
    watchInfo.setTakeWatchId(obj.getLong("id"));
    watchInfo.setEmpId(obj.getIntValue("empNo"));
    watchInfo.setLookAt(obj.getDate("lookTime"));
    watchInfo.setClientPhone(obj.getString("clientPhone"));
    List innerList = new ArrayList<String>();
    for(int i=0;i<arr.size();i++){
        innerList.add(arr.getJSONObject(i).getString("propertyId"));
    }
    List l;
    takeWatchService.insert(watchInfo);
    if(innerList.size()>0){
        l = takeWatchDao.getHouseListByInnerCode(innerList);
        takeWatchService.insertWatchHouse(l,watchInfo.getTakeWatchId());
        takeWatchService.insertShortKey(watchInfo);
    }
}
```

```

        Emp emp = empService.getByCodeWithExt(watchInfo.getEmpId());
        DateFormat df = new SimpleDateFormat("MM 月 dd 日 EEEE");
        String time = df.format(watchInfo.getLookAt());

        phoneService.sendMessage(watchInfo.getClientPhone(),emp.getUserName(),time,
                                Base62.getKeyByLong(watchInfo.getTakeWatchId()));
    }

}

```

图 4.10 ListenMqService.onMessage()代码

```

public void sendMessage(String mobile,String empName,String time, String message) {
    String url = ResourceConfig.getString("api.sendsmsurl");
    message = "您对经纪人"+empName+" "+time + "陪同的看房还满意么？鼓励或吐
槽经纪人请进："+ResourceConfig.getString("wap.short")+"/w/"+message;
    JSONObject p = new JSONObject();
    p.put("content", message);
    JSONObject param = new JSONObject();
    param.put("version", "1.0");
    param.put("method", "sms.sent");
    param.put("group", "ehrproject");
    param.put("auth", "4yzY20VkkPLVq70VNdbmIxt06nDDlwP5");
    param.put("template", "ehrtemplate");
    param.put("phone", mobile);
    param.put("params", p);
    try {
        logger.info("phone "+mobile);
        logger.info("content "+param.toJSONString());
        restTemplate.postForObject(url, param, JSONObject.class).toJSONString();
    } catch (Exception e){
        logger.info("SendError "+e.getMessage());
    }
}
}

```

图 4.11 phoneService.sendMessage()代码

```

public class Base62 {
    private static String data =
        "nW012F3456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMN0PQRSTUVWXYZ";
    private static char[] digits = data.toCharArray();
    private static int digitSize = digits.length;
    public static long parseToLong(String key){
        long result = 0L;
        for (int i=0;i<key.length();i++){

```

```

        result=result*digitSize+getIndex(key.charAt(i));
    }
    return result;
}
public static String getKeyByLong(long data,int strLen){
    String result = "";
    for(int i=0;i<strLen;i++){
        result=digits[(int) (data%digitSize)] + result;
        data = data/digitSize;
    }
    return result;
}

public static String getKeyByLong(long data){
    return getKeyByLong(data,6);
}

private static int getIndex(char A){
    for(int i=0;i<digitSize;i++){
        if(A==digits[i]) return i;
    }
    return -1;
}
}

```

图 4.12 Base62 类关键代码

4.3.2 带看评价模块控制层实现

带看评价模块的 controller 层实现了提供给前段的接口，对应 url 解析的配置文件为 Spring-mvc.xml，其配置如图 4.14 所示，分别设置了对 Controller 自动扫描的范围，静态资源路径的设置和登陆拦截设置。图 4.15 展示了 TakeWatchController 提供给前端的三个主要接口，分别用于查看带看评价，提交带看评价，提交带看反馈，对应的接口说明见第三章接口设计。

```

<?xml version="1.0" encoding="UTF-8"?>
    <!-- 自动扫描且只扫描@Controller 注：如果 base-package=com.lianjia 则注解事务不起作用 -->
    <context:component-scan base-package="com.lianjia"
        use-default-filters="false">
        <context:include-filter type="annotation"

```

```

        expression="org.springframework.stereotype.Controller" />
    <context:include-filter type="annotation"
        expression="org.springframework.web.bind.annotation.ControllerAdvice" />
</context:component-scan>

<bean

class="org.springframework.web.servlet.mvc.annotation.DefaultAnnotationHandlerM
apping" />
<!-- 文件上传适配器 -->
<bean id="multipartResolver"

class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <!-- set the max upload size100MB -->
    <property name="maxUploadSize" value="104857600" />
    <property name="maxInMemorySize" value="4096" />
</bean>
<!-- json,jsp 适配器 -->
<bean

class="org.springframework.web.servlet.view.ContentNegotiatingViewResolver">
    <property name="mediaTypes">
        <map>
            <entry key="html" value="text/html" />
            <entry key="json" value="application/json" />
        </map>
    </property>
    <property name="viewResolvers">
        <list>
            <bean
class="org.springframework.web.servlet.view.BeanNameViewResolver" />
            <bean

class="org.springframework.web.servlet.view.InternalResourceViewResolver">
                <property name="prefix" value="/pages/" />
                <property name="suffix" value=".jsp" />
            </bean>
        </list>
    </property>
</bean>
<mvc:annotation-driven>
    <mvc:message-converters>
        <bean

```

```

class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter" />
</mvc:message-converters>
</mvc:annotation-driven>

<!-- 静态资源路径 -->
<!--<mvc:resources mapping="/favicon.ico"-->
    <!--location="/static/img/favicon.ico" order="0" />-->
<mvc:resources mapping="/static/**" location="/static/" />
<!-- 登录拦截器 -->
<mvc:interceptors>
    <!-- app、wap、web 登陆状态验证 -->
    <mvc:interceptor>
        <mvc:mapping path="/**" />
        <mvc:exclude-mapping path="/static/**" />
        <mvc:exclude-mapping path="/locallogout" />
        <mvc:exclude-mapping path="/openApi/**" />
        <mvc:exclude-mapping path="/user/account/**" />
        <bean class="com.lianjia.common.LoginInteceptor" />
    </mvc:interceptor>
    <!-- 经纪人不能进用户中心 -->
    <mvc:interceptor>
        <mvc:mapping path="/**" />
        <mvc:exclude-mapping path="/static/**" />
        <mvc:exclude-mapping path="/locallogout" />
        <mvc:exclude-mapping path="/openApi/**" />
        <mvc:exclude-mapping path="/api/**" />
        <mvc:exclude-mapping path="/user/account/**" />
        <mvc:exclude-mapping path="/login/**" />

        <bean class="com.lianjia.common.AgentInteceptor" />
    </mvc:interceptor>
</mvc:interceptors>

</beans>

```

图 4.14 Spring-mvc 配置文件

```

@Controller
@RequestMapping(value = "/openApi/takeWatch")
public class TakeWatchController extends ApiBaseController {

    Logger logger = Logger.getLogger(this.getClass());
    @Autowired
    private TakeWatchService watchService;
    @Autowired
    private EmpService empService;
}

```

```

@Autowired
private SendMqService sendMqService;

/**
 *提交带看评价
 * @param watchId
 * @param knowledge
 * @param attitude
 * @param suggestion
 * @return
 */
@ResponseBody
@RequestMapping(value = "/postWatchRemark.json")
public Object postWatchRemark(@RequestParam long watchId,
@RequestParam(required = false ) Integer knowledge,@RequestParam(required = false)
Integer attitude,
                                @RequestParam(required = false,defaultValue="")
String suggestion) throws Exception {
    logger.info("suggestion:"+suggestion);
    TakeWatch watch = watchService.getWatchById(watchId);
    if(watch==null) return fail(-100).put("error","带看记录不存在");
    if(watch.getStatus()>0) return  hasComment(watch.getStatus(),false);
    if(knowledge==null) knowledge=0;
    if(attitude==null) attitude=0;
    watch.setKnowledge(knowledge);
    watch.setAttitude(attitude);
    watch.setSuggestion(suggestion);
    watch.setStatus(1);
    watchService.update(watch);
    if(knowledge<=3||attitude<=3){
        //empService.setRecommend(watch);
    }
    sendMqService.sendWatch(watch);
    return hasComment(1,true);
}

/**
 *举报带看评价
 * @param watchId
 * @param accuracyCode
 * @param accuracyRemark

```

```

    * @return
    */
    @ResponseBody
    @RequestMapping(value = "/postAccuracyRemark.json")
    public Object postAccuracyRemark(@RequestParam long watchId,
    @RequestParam(required = false) String accuracyCode,
    @RequestParam(required = false,defaultValue="")
    String accuracyRemark) throws Exception {

        String remarkCode[] ={"我没看过","带看房源不对"};
        TakeWatch watch = watchService.getWatchById(watchId);
        if(watch==null) return fail(-100).put("error","带看记录不存在");
        if(watch.getStatus()>0) return  hasComment(watch.getStatus(),false);
        int i;
        for(i=0;i<remarkCode.length;i++){
            if (remarkCode[i].equals(accuracyCode)) break;
        }
        if(i<remarkCode.length)  watch.setAccuracyCode(i);
        watch.setAccuracyRemark(accuracyRemark);
        watch.setStatus(2);
        watchService.update(watch);
        sendMqService.sendWatch(watch);
        return hasComment(2,true);
    }

    private Object hasComment(int status,boolean ok){

        JSONObject data = new JSONObject();
        data.put("hasComment",status);
        if(ok) return ok().put("data",data);
        else return fail(0).put("data",data);
    }

    /**
     * 根据手机号获取看房记录
     *
     * @param watchId
     * @return
     */
    @ResponseBody
    @RequestMapping(value = "/getWatchInfo.json")
    public Object getWatchInfo(@RequestParam long watchId) {

        List houseList = watchService.getWatchHouses(watchId);
    }

```

```

        TakeWatch watchInfo = watchService.getWatchById(watchId);
        if(watchInfo==null) return fail(-100).put("error","带看记录不存在");
        if(watchInfo.getStatus()==1) return getRecommend(watchInfo);
        else if(watchInfo.getStatus()==2)return hasComment(2,false);
        JSONObject obj = new JSONObject();
        DateFormat df = new SimpleDateFormat("MM 月 dd 日 EEEE");
        obj.put("hasComment",watchInfo.getStatus());
        obj.put("houseList",houseList);
        obj.put("lookCount",houseList.size());
        obj.put("lookDate",df.format(watchInfo.getLookAt()));
        obj.put("agent",empService.getByCodeWithExt(watchInfo.getEmpId()));
        return ok().put("data",obj);
    }

    public Object getRecommend(TakeWatch takeWatchInfo){
        JSONObject obj = new JSONObject();
        obj.put("hasComment",takeWatchInfo.getStatus());
        //obj.put("agents",empService.getRecommend(takeWatchInfo));
        return ok().put("data",obj);
    }
}

```

图 4.14 TakeWatchController 类主要代码

4.3.4 带看评价模块展示页面

如图 4.15 展示的带看信息浏览页面，页面上方是带看经纪人信息，左侧为经纪人头像，右侧为联系方式，电话号码为主机号+0+分机号。下方是提交带看评价的内容。专业能力，服务态度和补充评价都是可选项。没有问题点击下方的提交评价即可完成评价，存在问题可以点击左下的我没看过进入带看反馈(举报)页面。带看举报页面如图 4.16 所示。提交正常评价或者举报反馈后则可以浏览提交结果，如图 4.17 所示。

链家带看评价



张丹

03月23日 星期三

带您看了2套房源



致电

请评价本次带看（您的评价对经纪人匿名）

专业技能

服务态度

hi~我是年哥

7/130

您看的房源（2）



配套完善，房型正气，赢在教育，经典好房

1室1厅 70m²

久阳滨江公寓

205万

地铁房



配套完善，房型正气，赢在教育，经典好房

4室3厅 315m² 南

万科提香别墅

1,770万



我没看过

10.8.1.42:8010/ershoufan/ershoufang/100000.html

提交评价

图 4.15 浏览带看评价内容

42

返回

反馈内容



郭洪涛

02月28日 星期六

带您看了0套房源

致电

请选择您面临的问题

我没看过

带的房源不对

更多吐槽

/200

确认提交

图 4.16 提交带看反馈



图 4.17 查看提交结果

4.4 电话转接模块概述

电话转接模块是基于 j2ee 的后台程序，主要服务由定时任务完成。电话转接模块由部署在阿里云服务器的电话管理服务和部署在本地服务器的转接号管理服务两部分组成。

4.5 电话转接模块详细设计

电话管理模块的详细类图如图 4.18 所示，其控制层主要有两个功能类，extController 向其他服务提供 getBrokenExt 接口用于获取转接号，getHost 接口用于获取主机号。JobController 用户定制定时任务，checkExtBumber 用于检查空转接号库存，数量过少则自动生成，getExtNumber 定时清空 Temp 数据库（temp 数据库用于存储由于网络故障向转接数据库同步失败的数据）。ExtNumberGenerator 负责批量生成生成不重复的转接号和检查剩余转接号数量的逻辑实现，TempDataCleaner 则负责检查发送 temp 数据库内容的逻辑实现。HostNumberService 和 ExtNumberService 分别调用 Dao 层接口管理主机号和转接号。

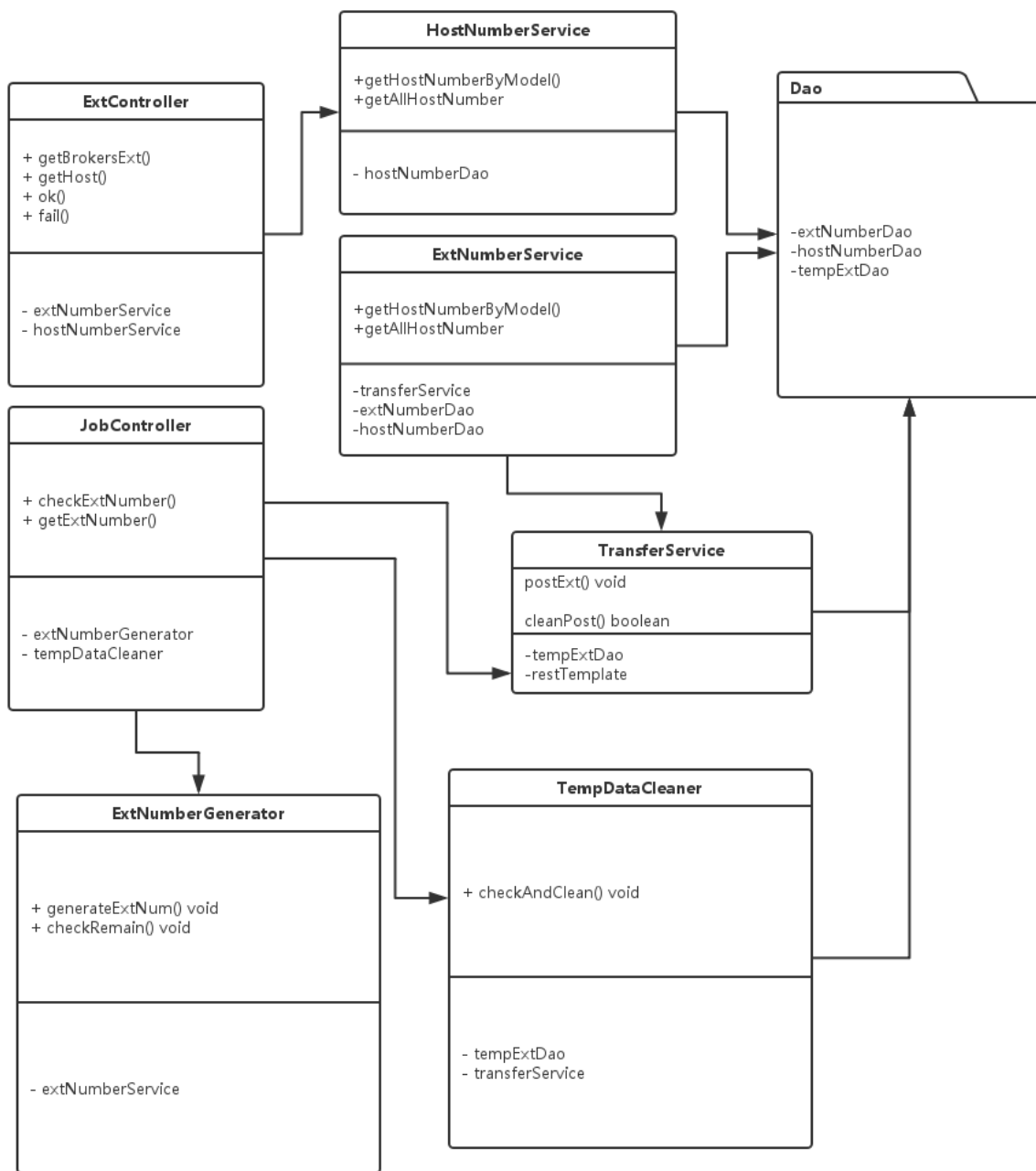


图 4.18 电话管理服务详细类图

转接号管理模块详细类图如图 4.19 所示，JobController 定制定时任务，SyncKeyExt()负责同步转接号，SyncEmp()用于同步经纪人信息，SyncData()用于同步主机号等其他变动小的数据。TransferController 给电话管理模块提供了 postExtNumber 接口，用于提交变动的转接号信息,convertExtNumber 用于将电话管理模块发送来的信息解析成本地类 ExtNumber。而 service 层几个类负责实现

对应的逻辑。`mergeBatch()`批量新增和更新转接好数据，`askForUpdate()`由转接号模块主动更新转接号数据。`GetExtNumber` 用于查询转接号对应的真实号码给通信服务商使用。

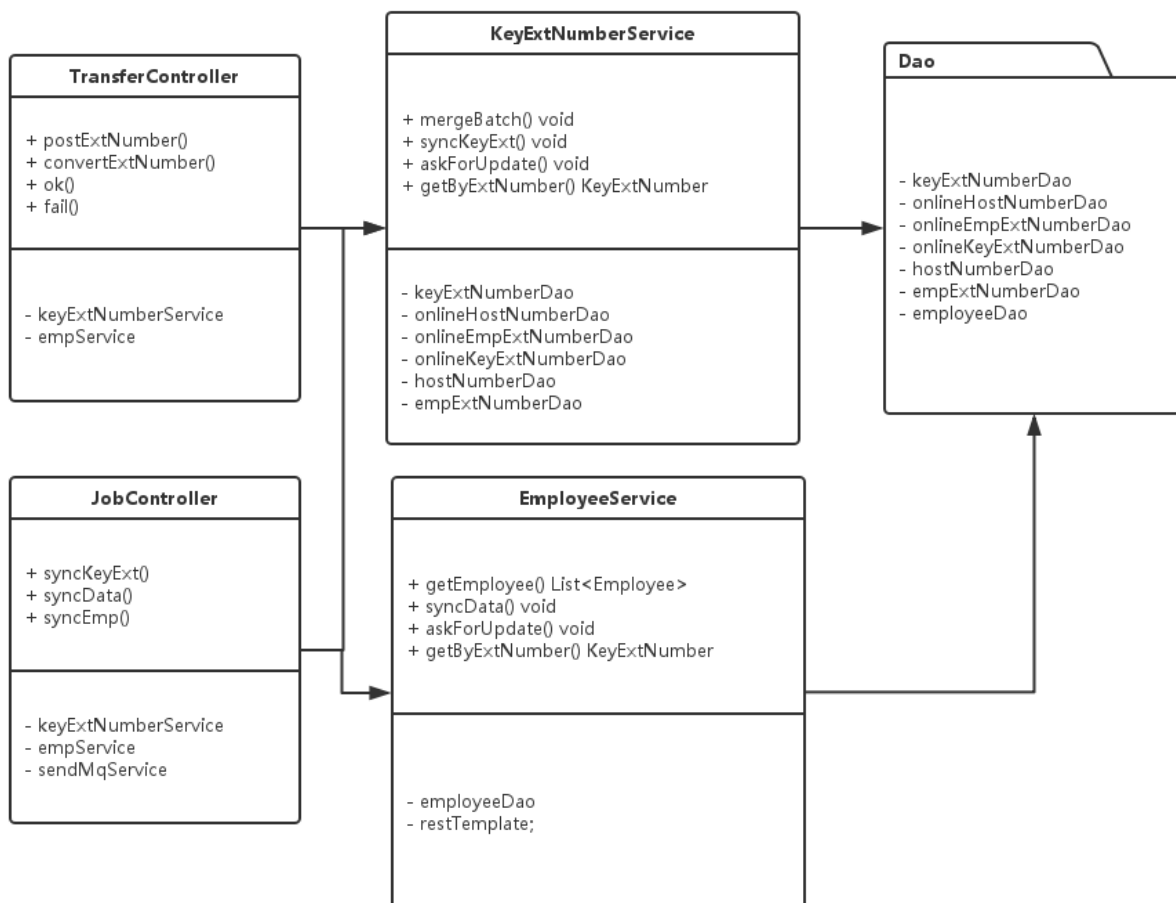


图 4.19 转接号服务详细类图

最后，电话管理的 Model 层类图如图 4.20 所示，转接号服务的 Model 层类图如图 4.21 所示。可以注意到他们的 `EmpExtNumber` 类中有些许不同，其原因是转接号系统存在遗留的其他服务需要为电信服务商服务，而电话管理模块不需要，所以电话管理模块将重复的数据剥离开存储，减少了某些表段的数据。

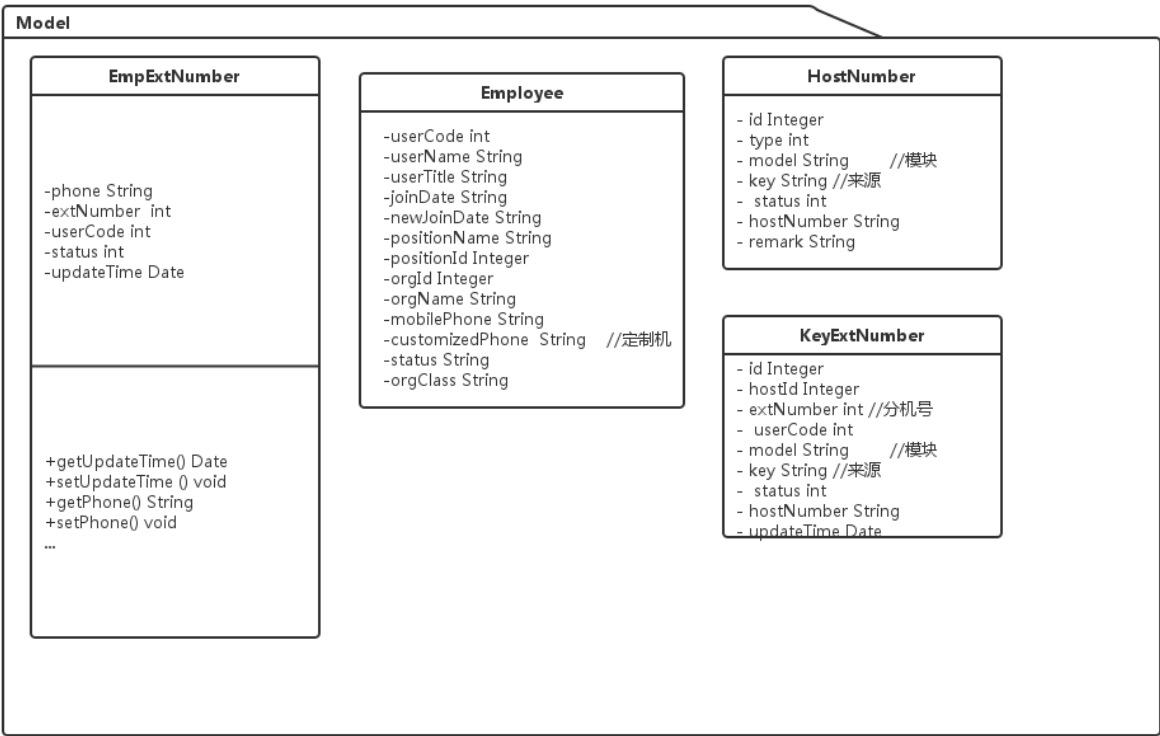


图 4.20 电话管理的 Model 层类图

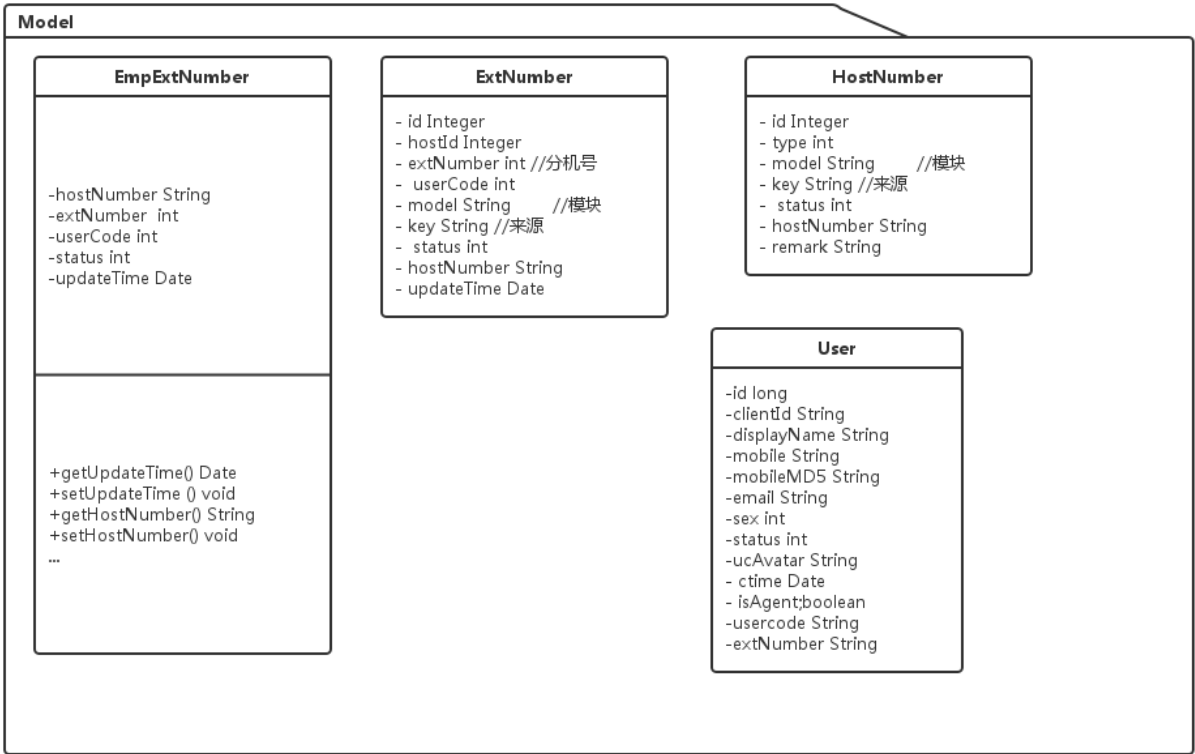


图 4.20 转接号服务的 Model 层类图

4.6 电话转接模块实现

在电话管理模块和转接号模块同步数据时，存在着两种同步方式。

一种由电话管理模块发起，将数据库中变动的转接号通过 **Http** 接口发送给转接号模块，由转接号模块负责更新。在网络故障的情况下将发送失败的数据存入 **temp** 数据库，每隔一段时间重新发送，以保证数据同步的及时性，安全性。

第二种是由转接号主动发起的更新，由于转接号模块部署在本地，同时拥有访问部署在阿里云服务器上数据库的访问权限，所以在转接号模块可以通过 **onlineDb** 和 **innerDb** 分别对两个数据库进行操作，在不涉及 **controller** 层接口的情况下高速更新数据。在主动更新数据时，更新方式也分为增量更新和全量更新，如转接号数据的 **key** 值变动较多，所以采用增量更新。由于主动更新定时完成，故可以利用 **updateTime** 进行数据筛选，减少 **merge** 操作的时间。而诸如 **hostNubmer** 这类变动较小且难以通过筛选判定是否有变更的数据则需要全量更新，将整张数据表进行同步。由于此类数据量很小变动也很少，同步定时工作时间设立在凌晨对用户基本不存在影响。

由于电话转接号的同步使用了 **http** 传送同步数据，所以两者除了 **DAO** 和 **Service** 层外也涉及到了 **controller** 层提供的数据上传接口。

4.6.1 电话管理模块发起的数据同步

电话管理模块的同步数据代码在 **TransferService** 中实现，在转接好更新或生成时调用。其代码如图 4.21 所示。在发送失败时数据会调用 **mergeBatch()** 方法存入临时数据库，等待定时任务重新发送。数据同步定时任务写在 **job** 包中的 **TempDataCleaner** 中，用于清空临时数据库中同步失败的数据，代码如图 4.xx 所示，**@Scheduled(cron="0 */5 * * * ?")** 表示每五分钟执行一次。

```
@Service
public class TransferService {

    @Autowired
    private TempExtDao tempExtDao;

    @Autowired
    private RestTemplate restTemplate;
```



```

Log log = LogFactory.getLog(TransferService.class);
@Async
public void postExt(List list) {
    boolean sendOK = cleanPost(list);
    if (!sendOK)
        try {
            tempExtDao.mergeBatch(list);
        } catch (Exception ex) {
            log.error("存入临时数据失败！" + ex.getMessage());
        }
}

public boolean cleanPost(List list){
    JsonResult result = null;
    try {
        String url= ResourceConfig.getString("transfer.host");
        url = url + "/transfer/post/extnum";
        JSONArray jsonArr = new JSONArray(list);
        result = restTemplate.postForObject(url,jsonArr.toJSONString(),
JsonResult.class);
    } catch (Exception e) {
        log.error("/post/extnum]==>报错详情: " + e.getMessage());
    } finally {
        if (result == null || result.containsKey("error") )
            return false;
        else return true;
    }
}
}

```

图 4.20 TransferService 类代码

```

@Service
public class TempDataCleaner {

    @Autowired
    private TempExtDao tempExtDao;
    @Autowired
    private TransferService transferService;

    @Scheduled(cron="0 */5 * * * ?")

```

```

public void checkAndClean() {
    for(;;){
        List<ExtNumber> extList = tempExtDao.getAllExtNum();
        if(extList.size() <=0) break;
        boolean result = transferService.cleanPost(extList);
        if(result){
            tempExtDao.deleteBatch(extList);
        }
    }
}
}

```

图 4.21 TempDataCleaner 类代码

电话管理模块发送的数据由转接号模块的 TransferController. postExtNumber() 负责处理。其代码如图 4.22 所示。postExtNumber 会调用 convert 方法生成本地对应的转接号类型。在 Dao 层使用了 merge 方法将新增数据和更新数据批量变更，其实现如图 4.23 所示。

```

@RequestMapping(value = "/post/extnum", method = RequestMethod.POST)
public @ResponseBody JsonResult postExtNumber(@RequestBody String postedString)
{
    try {
        List<KeyExtNumber> postedExtNumber = convert(postedString);

        if(!postedExtNumber.isEmpty())keyExtNumberService.mergeBatch(postedExtNumbe
r);

        return ok();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return fail();
}

private List<KeyExtNumber> convert(String postedString) {
    List<KeyExtNumber> list ;
    list = JSONArray.parseArray(postedString,KeyExtNumber.class);
    return list;
}

```

图 4.22 TransferController 类代码

```
<insert id="mergeBatch" parameterType="list">
```

```

merge into t_tel_extnumber a
using(
  <foreach collection="list" index="index" item="item" separator="union all">
    (select #{item.id} as id,#{item.key} as [key],#{item.extNumber} as
extNumber,
      #{item.userCode} as userCode,#{item.model} as model,#{item.hostId} as
hostId,
      #{item.updateTime} as updateTime, #{item.status} as status)
  </foreach>) b
on
(a.id = b.id)
when not matched then
insert
(id,hostId,extNumber,userCode,model,[key],status,updateTime) VALUES
(b.id,b.hostId,b.extNumber,b.userCode,b.model,b.[key],b.status,b.updateTime)
when matched then
update
set a.[key] = b.[key],a.userCode = b.userCode,a.model = b.model
,a.status = 1 ,a.updateTime =
b.updateTime,a.hostId=b.hostId,a.extNumber=b.extNumber;
</insert>

```

图 4.23 mergeBatch 方法代码

4.6.2 转接号模块实现的数据同步。

由于转接号模块需要配置有两个服务器的链接权限，其数据库连接配置如图 4.24 所示。转接号模块每小时都会检查主动进行一次数据更新，定时任务设定如图 4.25 所示，其中的关键方法 askForUpdate()主动更新方法如图 4.26 所示。

其中的 setTime(now.getTime()-1* 2 * 60 * 60 * 1000)用来设定筛选最近更新时间在两小时前的数据，可以减少更新数量。配合每小时更新一次的设定可以保证数据更新。

```

#sqlserver integration
development.inner.jdbc.url=jdbc:sqlserver://10.8.1.112:1433;DatabaseName=lianjiaOnlineTel
development.inner.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
development.inner.jdbc.username=dooiooadmin
development.inner.jdbc.password=密码

```

```
#sqlserver integration
test.inner.jdbc.url=jdbc:sqlserver://10.8.1.112:1433;DatabaseName=lianjiaOnlineTel
test.inner.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
test.inner.jdbc.username=dooiooadmin
test.inner.jdbc.password=密码

#sqlserver integration
integration.inner.jdbc.url=jdbc:sqlserver://192.168.3.66:1433;DatabaseName=lianjiaOnlineTel
integration.inner.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
integration.inner.jdbc.username=onlinetel
integration.inner.jdbc.password=密码

#sqlserver integration

production.inner.jdbc.url=jdbc:sqlserver://192.168.4.114:53435;DatabaseName=lianjiaOnlineTel
production.inner.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
production.inner.jdbc.username=onlinetel
production.inner.jdbc.password=密码

#sqlserver integration
development.online.jdbc.url=jdbc:sqlserver://10.8.1.112:1433;DatabaseName=lianjiaOnlineExt
development.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
development.online.jdbc.username=dooiooadmin
development.online.jdbc.password=密码

#sqlserver integration
test.online.jdbc.url=jdbc:sqlserver://192.168.3.66:1433;DatabaseName=lianjiaOnlineExt
test.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
test.online.jdbc.username=dooiooadmin
test.online.jdbc.password=密码

#sqlserver integration
integration.online.jdbc.url=jdbc:sqlserver://192.168.3.66:1433;DatabaseName=lianjiaOnlineExt
integration.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
integration.online.jdbc.username=onlineext
integration.online.jdbc.password=密码

#sqlserver integration
```

```
production.online.jdbc.url=jdbc:sqlserver://139.196.82.230:53435;DatabaseName=lianjia
OnlineExt
production.online.jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
production.online.jdbc.username=onlineext
production.online.jdbc.password=密码
```

图 4.24 jdbc.properties 配置文件

```
@Service
public class KeyExtNumberUpdate {

    private Logger log = LoggerFactory.getLogger(getClass());
    @Autowired
    private KeyExtNumberService extNumberService;

    @Scheduled(cron="0 0 */1 * * ?")
    @TaskOnOff(canRunIpProp = "${canRunIp}")
    public void updateExtNum(){
        extNumberService.askForUpdate();
    }

    @Scheduled(cron="0 30 0 * * ?")
    @TaskOnOff(canRunIpProp = "${canRunIp}")
    public void syncKeyExt(){
        DateTime date = DateTime.now().minusDays(1).minusMillis(30);
        extNumberService.syncKeyExt(date.toString("yyyy-MM-dd HH:mm:ss"));
    }

}
```

图 4.25 KeyExtNumberUpdate 类代码

```
public void askForUpdate() {
    List<HostNumber> list = onlineHostNumberDao.getAllHostNumbers();
    if(list.size() > 0){
        hostNumberDao.mergeBatch(list);
    }
    Date now = new Date();
    now.setTime(now.getTime()-1* 2 * 60 * 60 * 1000);
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss.sss");
    String timeStr = dateFormat.format(now);
    List<EmpExtNumber> list2 =
    onlineEmpExtNumberDao.selectEmpExtNumbers(timeStr);
}
```

```
int count = list2.size();
if(count > 0){
    //进行分页处理
    for(int index=0;index<count;){
        int endIndex = index + 200;
        List<EmpExtNumber> subList =
list2.subList(index,(endIndex>count?count:endIndex));
        empExtNumberDao.mergeBatch(subList);
        index = endIndex;
    }
}
}
```

图 4.26 askForUpdate 方法代码

4.7 本章小结

本章主要内容为二手房交易系统的带看评价模块和转接号管理模块的详细设计和实现细节。带看评价模块设计介绍了模块的详细类图、顺序图以及关键数据包设计，带看评价模块实现细节部分通过项目的配置，系统的运行页面，以及分三层展现的部分方法代码描述了一些关键功能的实现细节。转接号管理模块则以电话管理和转接 hao 服务两部分交互的转接号数据同步为核心阐述了两个服务的设计和实现。

第五章 总结与展望

5.1 总结

本项目是本人在链家外网开发部门实习期间参与的项目，带看评价模块和转接号管理模块是我主要编写和维护的两个模块。本项目针对了二手房交易中存在的电话联络问题，和带看后的后续跟进问题设计和实现了带看评价与转接号管理模块。

本文首先在第一章里介绍了二手房交易系统的项目背景，同时阐述了论文的主要工作和组织结构。

在第二章，主要介绍了系统中使用到的相关技术的概述，包括 Spring 框架、数据持久化工具 MyBatis,ActiveMq 和 FastJson 工具等。

第三章对系统进行了需求分析，对系统的功能需求和面向的用户进行了详细的分析，并通过用例图和用例描述对系统进行了用例建模，同时还分析了系统的非功能需求。

第四章介绍了本系统的带看评价模块和转接号管理模块的详细设计和部分关键功能的代码实现细节。

在项目从需求分析到代码实现的整个过程中，本人运用了大学本科中学到的相关知识，积极参与，了解了软件开发生命周期，对软件工程有了更加深入的理解。在实线项目的过程中，通过本人的对 Spring、MyBatis 和 fastJson 等技术，以及在部门同事和主管的帮助下，对已有经纪人推荐系统、数据同步系统的学习，我成功总结和设计了带看评价的推荐系统和转接号同步方式，最终完成了本系统的开发。

5.2 展望

目前，本文介绍的带看评价模块和电话管理模块均已在测试后发布上线，在发布上线的过程中还发现了一些项目设计上存在的问题，好在在同事和主管的帮助下得以及时修复。文中设计良好的一些模块也被公司的其他模块吸收，比如带看评价短链接功能通过简单的修改就发布在了房源详情短链接中。在实际用户的反馈中，带看评价的体验方面做的还不够好，由于没有对异常数据进行仔细的处理，有些地方会造成用户的不解，在帮助用户更好的使用系统上还有着很大的改进空间。

本系统是以实际使用为目的的，因此，系统的后期维护、可扩展性和高性能都很重要。在后续的版本中，我们会加强这方面的开发和维护工作。在电话管理模块中进一步加强同步数据的准确度和速度，在电话管理模块和带看评价模块进一步脱离目前依赖的一部分外部系统接口，保持系统的独立和稳定。

参考文献

- [1]链家网-微头条, <http://www.baidu.com/link?url=8HuPt>
- [2] Johnson R.Hoeller J.J2EE Development without EJB[M].JavaEye, 译.北京: 电子工业出版社, 2005
- [3]百度百科: Spring, <http://baike.baidu.cn/view/23023.htm>
- [4]MyBatis 官方文档, <http://www.mybatis.org/mybatis-3/zh/index.html>
- [5]邹伟,《基于 j2ee 的 rfid 中间件的研究与实现》,硕士硕博学位论文, 湖南大学,2009。
- [6]FastJson 解析博客, http://blog.sina.com.cn/s/blog_7ffb8dd501013qas.html
- [7] Maven 构建 Hadoop 工程博客, http://www.cnblogs.com/Leo_wl/p/4862820.html
作者,译者,书名(版本),出版地:出版社,出版时间,引用部分起止页。
- [8] 成亦陈, 黄淑华,《恶意短链接欺骗的防护对策研究》出版自《信息网络安全》,2013(7):32-35

致谢

在这里，我首先要感谢我的导师王浩然教授对我的悉心指导，在论文完成期间，不仅监督了我毕业论文的按时完成，同时也对我的工作提出了许多宝贵的意见。

其次，感谢我在链家实习期间的同事给予我的指导和鼓励，尤其要感谢我的部门主管，在我毕业设计的完成过程中，给我提供了很多的建议，给予了我很大的帮助。在链家我得到了很好的学习和锻炼机会，三个月的实习不仅使我了解到了工业界最前沿的技术，还锻炼了我的沟通能力和团队合作能力。在此，我谨向他们表示最衷心的感谢。

感谢学院所有的老师和辅导员，你们的授道解惑和悉心叮咛让我在收获知识的同时也明白了很多人生道理，这些宝贵的财富将会使我终身受益。

感谢一直以来陪伴在我身边的同学和朋友，我们一同迷茫、一同探讨、一同成长，一路同行，你们的帮助和陪伴，是我青春路上最珍贵的礼物。还有一直以来在我背后默默支持我的家人，你们的肯定是我最大的动力，感谢你们。

最后，感谢所有参与论文评审和答辩的老师，你们辛苦了。