软件工程管理

软件需求工程

**软件需求工程-住宅维护系统**

系统设计计划

|  |  |
| --- | --- |
| **组 号：** | G09 |
| **组 长：** | 江 号 |
| **组 员：** | 沈子衿 宋宇杰 王优 |
|  | 林宇翔 徐正杰 |

2018年12月14日

**修改历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **作者** | **修改内容** |
| 2018.12.14 | 1.0 | 江号、沈子衿、宋宇杰、王优、林宇翔、徐正杰 | 初稿 |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 4](#_Toc532611250)

[1.1 编写目的 4](#_Toc532611251)

[1.2 项目背景 4](#_Toc532611252)

[1.3 项目目的 4](#_Toc532611253)

[1.4 参考资料 5](#_Toc532611254)

[2. 总体设计 6](#_Toc532611255)

[2.1 需求规定 6](#_Toc532611256)

[2.1.1 用户需求 6](#_Toc532611257)

[2.1.2 性能需求 8](#_Toc532611258)

[2.1.3 权限与安全需求 9](#_Toc532611259)

[2.1.4 可视化需求 10](#_Toc532611260)

[2.1.5 防护性需求 10](#_Toc532611261)

[2.1.6 其他需求 11](#_Toc532611262)

[2.2 运行环境 11](#_Toc532611263)

[2.2.1 软件层面 11](#_Toc532611264)

[2.2.2 硬件层面 11](#_Toc532611265)

[2.3 设计和实现上的约束 12](#_Toc532611266)

[2.3.1 数据存储 12](#_Toc532611267)

[2.3.2 网络服务吞吐 12](#_Toc532611268)

[2.3.3 数据安全 12](#_Toc532611269)

[2.4 系统体系结构设计 13](#_Toc532611270)

[2.4.1 B/S体系结构 13](#_Toc532611271)

[2.4.2 体系结构的实现 14](#_Toc532611272)

[2.5 系统用例设计 15](#_Toc532611273)

[2.5.2 教师、学生用例图 15](#_Toc532611274)

[2.5.2 维修商、供应商用例图 15](#_Toc532611275)

[2.5.3 管理员用例图 16](#_Toc532611276)

[2.6 系统实现时间表与里程碑 16](#_Toc532611277)

[2.6.1 任务安排 16](#_Toc532611278)

[2.6.2 重要里程碑及里程碑评审 17](#_Toc532611279)

[3. 附录 18](#_Toc532611280)

[3.1 主要技术分析 18](#_Toc532611281)

[3.1.1 前端技术 18](#_Toc532611282)

[3.1.2 后端技术 19](#_Toc532611283)

[3.2 数据字典 19](#_Toc532611284)

[3.2.1 数据流定义表 19](#_Toc532611285)

[3.2.2 数据元素定义表 21](#_Toc532611286)

[3.2.3 数据精度表 22](#_Toc532611287)

[3.3 数据库设计 23](#_Toc532611288)

[3.3.1 核心数据表定义 23](#_Toc532611289)

[3.3.2 AUTH\_USER表 23](#_Toc532611290)

[3.3.3 OWNER表 24](#_Toc532611291)

[3.3.4 RENTER表 24](#_Toc532611292)

[3.3.5 ADMINISTRATOR表 25](#_Toc532611293)

[3.3.6 MAINTAINER表 25](#_Toc532611294)

[3.3.7 SUPPLIER表 25](#_Toc532611295)

[3.3.8 HOUSE表 26](#_Toc532611296)

[3.3.9 FACILITIES表 26](#_Toc532611297)

[3.3.10 SMART\_HOME表（以AC\_TABLE为例） 27](#_Toc532611298)

[3.3.11 OWNER\_HOUSE表 27](#_Toc532611299)

[3.3.12 MAINTAIN\_ORDER表 28](#_Toc532611300)

# 引言

## 编写目的

系统设计计划主要为使软件产品和软件项目满足规定的软件规格要求而确定软件系统的体系结构、组成部分、数据组织、模块、内外部接口。主要任务有：

1. 建立软件产品和软件项目目标系统的总体结构；
2. 总体设计；
3. 接口设计；
4. 运行设计；
5. 系统数据结构设计；
6. 系统出错处理设计。

## 项目背景

随着移动互联网与物联网技术的不断发展和人们生活水平的提高，越来越多的传统行业开始走向数字化和信息化。但是，在某些领域，仍然保留着一些传统的服务策略。如，在美国的房地产市场，管理不动产及其配套设施的相关服务之间仍然是相互孤立的。住户常常苦于房屋设施的老化与家居的损坏，却缺少一个能为其进行房屋综合维护的解决方案。

因此，本项目立足美国不动产租赁与家居市场，面向美国本土为数众多的房屋租客、房屋业主，旨在为当下有购置新房及购买新家具需求的人群提供一套基于物联网的房屋管理与维护解决方案，以应对物联网时代日益增长的智能化家居管理需求。其主要功能包含房源查找、房屋租赁、家居保险（包括家居维修、家居维护、维护警报系统和家居换购服务等）及智能家居管理。此外，该系统还将支持用于提供线上支付功能的电子钱包以及面向特殊人群的小额度贷款服务

预计本项目在落地后将会覆盖数十万的用户群，大大造福广大租客与业主。

## 项目目的

如前所述，本平台立足美国不动产租赁与家具市场，向美国本土为数众多的房屋租客、房屋业主，旨在为当下有购置新房及购买新家具需求的人群提供一套基于物联网的房屋管理与维护解决方案。经过多次需求评估以及同甲方的深入交流，我们确定将实现如下功能模块：

1. 房屋管理模块（House Management）
   1. 房产管理（添加房屋，房屋认证，删除房屋等）；
   2. 普通家居管理（添加家居，家具分类，删除家居等）；
   3. 基于IoT的智能家居管理（动态调整智能家居的实时参数）。
2. 房屋维护模块（House Maintenance）
   1. 房屋状况检测与报告上传；
   2. 家居维修与维护（家居报修机制，维护警报系统等）；
   3. 家居换购服务（供应商入驻）；
   4. 维修工具和材料购买服务（供应商入驻）；
   5. 维护与维修人员服务平台（普通用户发起维修订单，维修商旗下维修工接收订单）。
3. 云端服务模块（Cloud Service）
   1. 用户信息管理；
   2. 角色与权限管理；
   3. 用户行为分析；
   4. 数据存储与整合。
4. 电子钱包模块（E-Wallet）
   1. 电子钱包与基础交易逻辑；
   2. 房屋贷款与房屋租赁贷款；
   3. 房屋保险；
   4. 房屋还款。

## 参考资料

1. 《软件需求》
2. 《软件过程——面向对象和传统的方法》
3. [G9]项目可行性报告
4. [G9]软件质量保证计划
5. [G9]项目章程
6. [G9]项目总体计划
7. [G9]愿景和范围
8. [G9]系统变更控制会规程
9. [G9]软件需求规格说明书
10. SEM&SRE课程资料
11. CMMI教材

# 总体设计

## 需求规定

### 用户需求

1. **房东/业主需求**

* 系统要有房屋信息发布平台，可以发布出租房的详细地址、附近的公共施、卧室/浴室数量、面积大小、户型、配套家具与设备状态、房屋年龄、出租日期、停车位、价格、水电费和物业费、房屋及周边的照片的信息。可以在以后添加自动将出租房发布到其他各大租房平台并由本系统帮助房东统一管理的功能；
* 系统能够查看所有房屋的状态，查看在上一次检查中上传的最新维护报告，包括房屋内各种设备的运转情况、已使用时长、预计使用年限等。在之后可以和智能家居结合，使用传感器记录设备的实时状况。系统提供家具以旧换新和折价收购的接口；
* 系统要可以查看租户的详细信息，包括个人信用、租房历史、背景调查等。个人信用先用第三方的信用积分，之后当数据足够时可以开发系统内部的信用计算；
* 系统上能够根据关键信息搜索符合自己要求的租客，进行联系；
* 系统要有辅助定制租赁协议的功能，按步骤顺序引导建立协议并提供一些各部分的模板，在此基础上允许房东自主修改；
* 系统要有电子钱包，可以绑定银行卡/信用卡，进行付账和收款行为。在之后可以在钱包内加入贷款功能；
* 系统能够绑定自己的所有房产，并能查看每一处房产的收租情况与租金到期时间；
* 系统提供家具以旧换新和折价收购的接口；
* 系统上能够制定自定义的检查计划，并且在初次使用计划制定功能时，会按照房屋的状况提供默认的推荐计划。计划包括部分例行检查（白蚁检查、屋顶维修等）的时间间隔、部分设备（水龙头、灯泡、出水口等）的检查/更换时间间隔。在计划制定完成后，平台提供各个合作维护公司的选择，并根据选择自动计算计划实施价格。用户可以选择进行付款，使用我们提供的服务，也可以仅保留计划，自己寻找维修公司。系统会根据计划推送提醒消息；
* 系统提供上传房屋数据的接口，上传的数据内容和之前可以查看的内容是相同的（除开智能家居自动检测的部分），一个房子的房东、租客和维修公司都可以上传房屋的维修报告；
* 系统应当提供一个维修服务平台，在房东/业主需要对房屋的特定部分进行维修或对房屋状况进行总体评估时，可以在此平台上发起订单，由平台合作维护公司指定维修人员上门服务；房东/业主可以对维修人员的服务进行评分。

1. **租客需求**

* 类似于房东发布房屋信息，租客可以发布个人信息，包括理想中房屋的价格、位置、个人介绍、联系方式等等，可以让房东主动搜索联系；
* 系统给租客提供上传房屋检查报告的接口，房东可以和租客商量，由租客来进行日常的检查，将报告上传后再由房东来根据计划付款找公司维修；
* 系统提供按照输入的关键字和选择的要求搜索匹配房屋的功能，对于搜索到的房屋，可以查看房东提供的所有相关信息（即房东发布在房屋信息发布平台的内容）和房东个人的背景。这是一个双向的过程，房东和租客可以互相查看对方的背景；
* 系统要有辅助定制租赁协议的功能，提供的功能和房东端一样；
* 可以直接使用系统内的电子钱包来进行租房的付款。
* 系统应当提供一个维修服务平台，在租客需要对房屋的特定部分进行维修或对房屋状况进行总体评估时，可以在此平台上发起订单，由平台合作维护公司指定维修人员上门服务；租客可以对维修人员的服务进行评分。

1. **管理员需求**

* 能够对房东/业主发布的房源进行审核，决定其是否可以发布到本平台上；
* 能够删除任何非管理员用户发布的信息、资源；
* 能够恢复七天内删除的信息、资源；
* 能够编辑系统界面上的相关参数和链接；
* 能够禁止、允许某一类用户发布特定信息、资源；
* 能够赋予某一类用户新的权限，或者从某一类用户那里收回权限；
* 能够调整一个用户在某一特定租赁关系中的角色及其全局角色，但原则上不可改变租赁关系的本质；
* 能够编辑平台主页的信息流和及其他相关元素。

1. **其他通用需求**

* 系统要有用户指南，可以引导各个功能的使用；
* 系统界面简洁明了，能够直观的展现出所有主要功能；
* 系统提供注册、登录、密码找回的功能；
* 系统可以提供一个专门的论坛版块来供用户进行交流讨论，并且提供向运营方反馈问题、提出建议的入口；
* 系统应当提供一个简易的用户间即时通讯机制，作为社交软件、即时通讯手段和拨号通话的补充；
* 系统要有足够的安全性，可以避免用户信息泄露，保证用户电子钱包的安全；
* 系统能提供一定的资料共享功能（如特定位置由上传下载附件功能。但对附件的大小以及并行文件传输载荷是有限制的）；
* 系统对列表项（如家居列表、房屋列表、维护报告表项等）以及文章标题应当有基本的搜索和排序功能；
* 对于网页中的多媒体资料，特别是音频和视频，网站能提供基本的下载和在线观看功能；
* 系统应当充分考虑受众的需要，对于不同版本的运行环境有较好的兼容性；

### 性能需求

* 系统应保证运行稳定，避免出现崩溃；
* 主流浏览器均能正常访问本系统；
* 系统应能保证至少100人的并发访问；
* 当用户登录以及进行任何操作时，系统能及时进行反应，反应的事件子啊3s以内；
* 系统应该能及时检测出各种非正常情况，如与设备的通信终端无法连接数据库服务器等，避免长时间等待；
* 用户提交查询操作后响应事件不超过5s；
* 每个页面一般情况下应在2s内加载完毕，高峰期应在6s内加载完毕；
* 系统保证在一周内不超过一次的维护与重启。

### 权限与安全需求

* 所有涉及功能信息或个人信息的网络事务，都应进行加密操作；
* 除浏览菜单外，用户必须登录后才能完成其他操作；
* 不同角色原则上登录入口不同；
* 除游客外每个用户在注册时都必须指定一个初始密码，并且该密码应当符合一定强度指标，如必须8位以上，且为数字与英文的组合；
* 用户在找回密码时必须重置密码。而不能使用原来的密码；
* 用户修改密码时必须进行验证；
* 用户无法非法修改数据库；
* 只有授权为管理员的用户，才能进行上面描述的管理员操作；
* 特定房屋/设备只能由添置该房屋/设备的用户编辑；
* 除公有资源外，特定用户只能访问其所绑定房屋对应的相关资源；
* 只有系统管理员有权查看系统日志；
* 任何人都无权修改或删除日志；
* 本系统对重要的数据应进行加密，如用户口令、重要参数等；
* 允许系统管理员进行数据的备份和恢复，以防止数据的破坏和丢失；
* 本系统应该能够记录系统运行时候所发上的所有错误，包括本机错误和网络错误是、这些错误记录便于查找错误的原因。日志同时记录用户的关键性操作信息；
* 系统应当提供一定的限制功能，即只允许在局域网内特定机器上运行用户管理功能；
* 系统应当具备加密登录、数据加密传输、数据存储等安全方面的保障，以确保系统的安全性；
* 系统是居于开放的操作系统平台和数据库上的，因此，要求建立操作系统和数据库的安全保障体系保证操作系统和数据库的安全；
* 对可能发生严重后果的操作要有补救措施。通过补救措施用户可以回到原来的正确状态。对可能造成等待时间较长的操作应该提供取消功能；
* 对一些特殊符号和计算机代码的输入，与系统使用的符号相冲突的字符等进行判断并阻止用户输入该字符；
* 对错误操作支持可逆性处理，如取消系列操作。在输入有效性字符之前应该阻止用户进行只有输入之后才可进行的操作。

### 可视化需求

* 系统管理员修改导航栏后能立刻看到预期效果；
* 用户上传信息和资源时可以看到进度条，显示上传进度；
* 用户下载信息和资源时可以看到进度条，显示下载进度；
* 用户在调整智能家居状态后，更改后状态将回显在界面上；
* 用户可以生成房屋报告并进行在线浏览。
* 用户在加载资源、界面时总是能看到一个旋转中的环形图标，表示“加载中”；
* 对于一些较为抽象的数据（如维修剩余天数），采用扇形饼图或倒计时进度条的方式增加直观性；
* 对于一些处于危险区域的数据和涉及危险操作的按钮/链接，使用醒目的颜色（如红色）进行标注。

### 防护性需求

* 文件格式错误时，系统提出警告，保持数据库数据不变；
* 数据库误删除时，可以使用撤销删除修复；
* 重复操作导致卡死时，系统提出警告；
* 访问无权限时，系统发出提示并禁止用户访问；
* 上行文件出错时，系统应提供自主覆盖功能；
* 系统应该提供验证码防止恶意登录；
* 系统应该及时信息备份防止病毒攻击；
* 系统应提供密保方式防止人为破解密码；
* 系统应该能检测到恶意操作；
* 当检测到恶意重复操作时，系统应提出警告并在一段时间内不允许操作。

### 其他需求

* 软件必须严格按照设定的安全权限机制运行，并有效防止非授权用户进入本系统；
* 软件必须提供对系统中各种码表的维护、补充操作；
* 软件必须按照需求规定记录各种日志；
* 软件对用户的所有误操作或不合法操作进行检查，并给出提示信息。

## 运行环境

### 软件层面

住宅维护系统网站客户端使用现代网页浏览器进行访问及操作，浏览器需Internet Explorer 11、Google Chrome 62、Mozilla Firefox 57、Safari 10 以及更高的版本以获得更好的体验。

住宅维护系统网站服务端需运行在一个单独的服务器中（阿里云），网页服务由Apache 或 Nginx 以及 PHP 提供， 同时服务器使用MySQL数据库系统软件提供住宅维护系统网站后台数据存储。

### 硬件层面

硬件层面对于所使用的服务器有以下要求：

* CPU：主频大于 2.0GHz
* 内存：大于等于 2GB
* 硬盘：硬盘容量大于 200GB、硬盘转速大于等于 5400 转/分钟
* 网卡：百兆网卡
* 网线：具有良好的数据传输能力即可
* 键盘：可以满足正常使用即可
* 鼠标：可以满足正常使用即可
* 显示器：可以满足正常使用即可

## 设计和实现上的约束

系统的设计、编码、以及维护将遵照所提交《项目可行性分析报告》、《项目总体计划》、《项目章程》、《软件质量保证计划》、《项目愿景和范围》、《需求工程计划》共计五个文档进行。

在具体设计和实现上，按照以下约束进行：

* + 1. 数据存储

项目产品使用标准 MySQL 数据库系统作为引擎，按照数据产生、转换和存储的策略，通过将数据导入数据库的方式进行数据的存储操作。

### 网络服务吞吐

根据项目要求，本项目要求提供对外服务的能力，以确保同时为至少 300 名用户进行日常服务的要求。

### 数据安全

保证以下完整性、保密性以及可用性三个特性来保护用户的数据安全：完整性要求数据未经授权不得进行修改，确保数据在传输和存储过程中不被篡改，盗用和丢失。通过利用安全的框架，在加密的基础上，运用多种方案和技术实现。

保密性要求对数据进行加密，只有授权者才能使用。这一特性要求加密技术必须自动、实时、精确、可靠。

可用性要求做到避免因为系统数据泄露而使得合法使用者无法接触可用数据，通过对使用者身份的验证，为合法使用者提供更加安全便捷的使用。

## 系统体系结构设计

### 2.4.1 B/S体系结构



图2-1 B/S体系结构示意图



图2-2 B/S三层架构模型图

在系统中使用浏览器-服务器（Browser/Server，B/S）体系结构，采用 B/S 的三层体系结构，将系统的整个业务应用划分为表示层、业务逻辑层和数据访问层，这样有利于系统的开发、维护、部署和扩展。B/S 结构的基本原则是将计算机应用任务分解成多个子任务，由多台计算机分工完成，即采用“功能分布”原则。客户端完成数据处理，数据表示以及用户接口功能；服务器端完成 DBMS 的核心功能。这种客户请求服务、服务器提供服务的处理方式是一种新型的计算机应用模式。

### 2.4.2 体系结构的实现

对于这一种系统，采用如下方式进行实现：

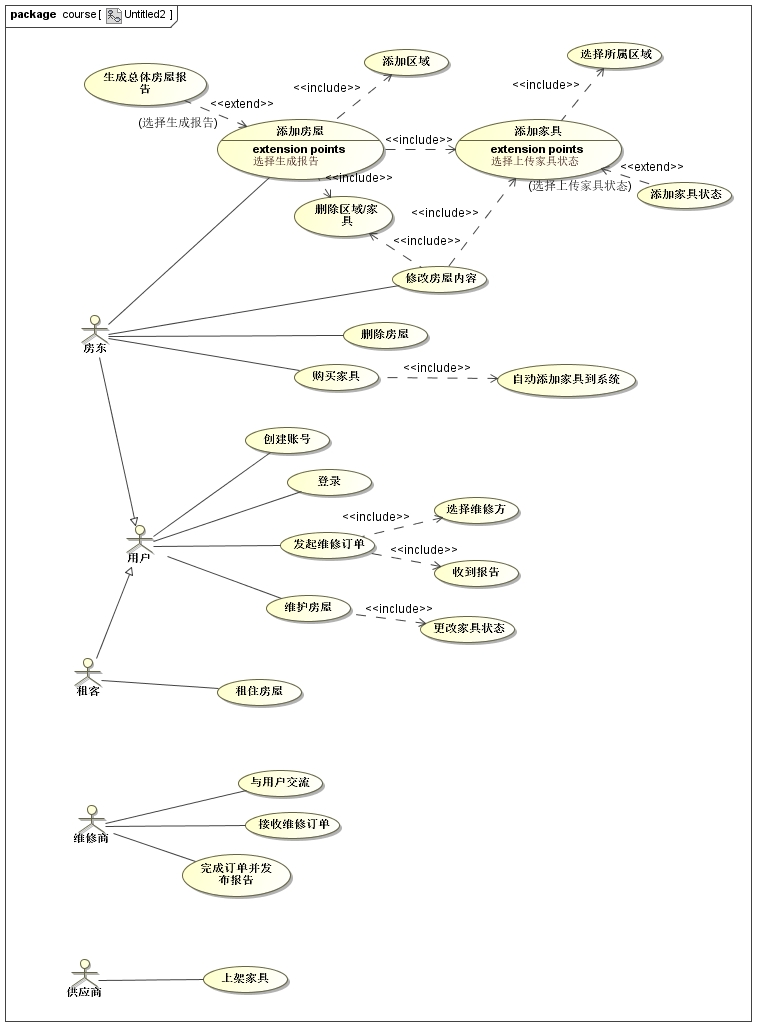
（1）采用如下的服务端、客户端要求：住宅管理系统网站客户端使用现代网页浏览器进行访问及操作，浏览器需Internet Explorer 11、Google Chrome 62、Mozilla Firefox 57、Safari10以及更高的版本以获得更好的体验。住宅管理系统网站服务端需运行在一个单独的服务器（阿里云）中。网页服务由 Apache 或 Nginx 以及 PHP 提供，同时服务器使用 MySQL数据库系统软件提供住宅系统网站后台数据存储。

（2）采用如下的服务端、客户端配置：

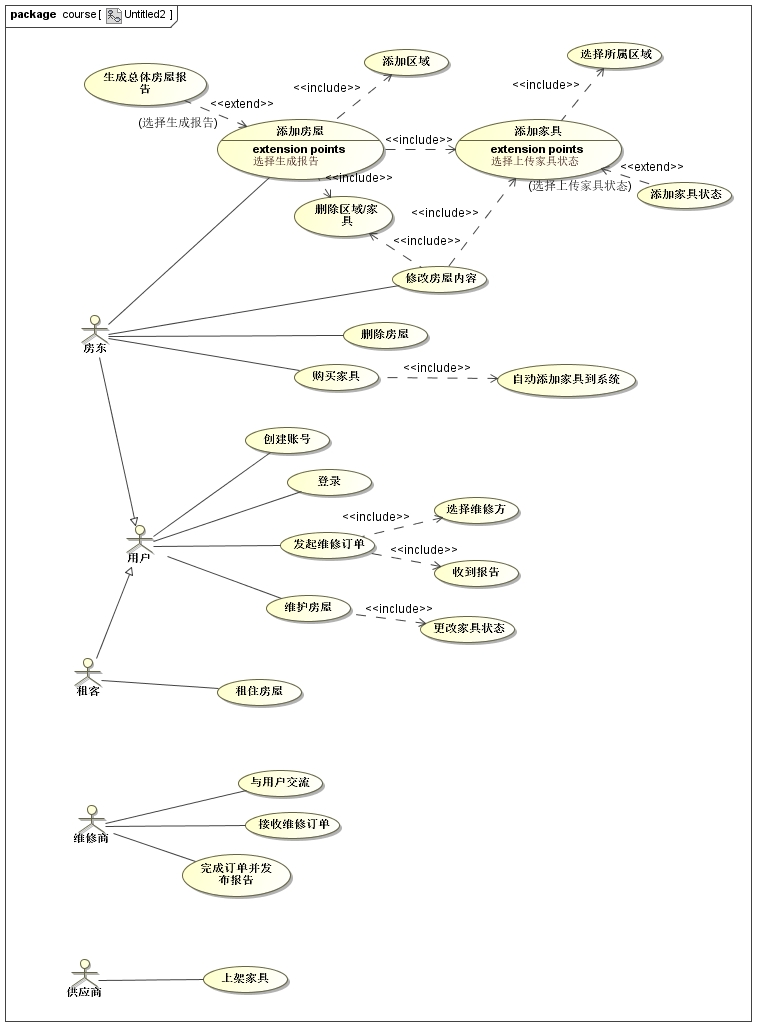
* 服务器：以 Apache 或 Nginx 为 Web 服务器，服务器端采用 PHP 语言编写， 数据库采用 MySQL。
* 客户端：浏览网页主要采用Internet Explorer 11、Google Chrome 62、 Mozilla Firefox 57、Safari 10 以及更高的版本，通过 AJAX 技术进行网页的动态交互，同时增加服务器带宽利用率，并使用 Vue和Element框架对客户端进行美化。

## 系统用例设计

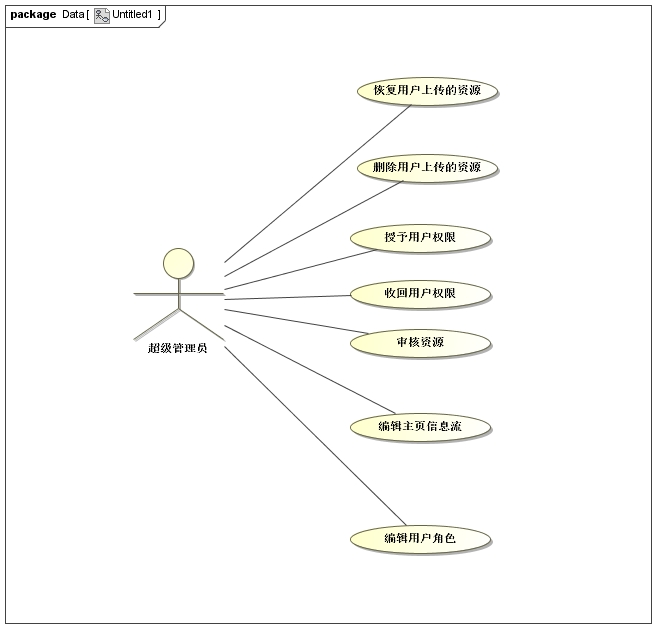
### 2.5.2 教师、学生用例图



### 2.5.2 维修商、供应商用例图



### 2.5.3 管理员用例图



## 2.6 系统实现时间表与里程碑

### 2.6.1 任务安排

|  |  |
| --- | --- |
| **关键时间点** | **进度描述** |
| 10月21日～10月28日 | 进行需求分析，画出用例图、数据流图，形成软件构架，形成需求报告。 |
| 10月29日～11月05日 | 数据库设计，主界面的设计。 |
| 11月06日～11月15日 | 住宅维护模块的设计与实现，与客户进行沟通。 |
| 11月16日～11月28日 | 住宅管理模块的设计与实现，与客户进行沟通。 |
| 11月29日～12月8日 | 云端模块的设计与实现，与客户进行沟通。 |
| 12月9日～12月16日 | 电子钱包模块的设计与实现，与客户进行沟通。 |
| 12月17日～12月23日 | 界面修整、美化。组内成员进行用户体验、测试。 |
| 12月24日～12月31日 | 提交用户进行体验，用户反馈问题，组内成员解决问题。 |
| 1月1日 | 产品正式发布。 |

### 2.6.2 重要里程碑及里程碑评审

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑名称** | **里程碑**  **时间** | **重要交付物** | **评审方式** |
| 制定项目章程 | 10.14 | 项目章程 | 评估章程的合理性 |
| 可行性分析 | 10.14 | 项目可行性报告 | 评估可行性报告的合理性 |
| 编写项目总体计划 | 10.14 | 项目总体计划 | 评估进度安排、项目规划情况 |
| 编写质量保证计划 | 10.21 | 质量保证计划 | 评估QA计划的合理性 |
| 需求调研 | 10.28 | 需求工程计划 | 评估需求是否合理可行 |
| 编写需求说明书 | 11.5 | 软件需求规格说明书 | 评估需求是否符合客户要求 |
| 编写总体设计、  概要设计 | 11.10 | 系统设计计划 | 评估计划是否具有阶段性、  全面性、客观可行性 |
| 集成测试规范编写 | 11.15 | 概要设计说明 | 评估系统结构的合理性 |
| 编码 | 11.18 | 住宅使用与维护管理系统代码 | 评估代码是否符合预先计划，  以及程序是否可以正常运行 |
| 单元测试 | 12.7 |
| BUG修改 | 12.15 |
| 集成计划与方案 | 12.23 | 编码与系统实现计划 | 评估集成计划的全面性与合理性 |
| 进行集成 | 12.26 | 集成代码 | 评估集成代码是否能经过基本测试 |
| 集成测试 | 12.27 | 测试计划 | 评估测试计划是否具有完整性与有效性 |
| BUG修改 | 12.28 |  | 代码是否能通过测试计划 |
| 集成测试报告 | 12.31 | 测试报告 | 评估测试情形的合理性 |
| 编写用户手册 | 1.3 | 用户手册 | 评估用户手册的可使用性 |
| 编写工程部署计划 | 1.4 | 工程部署计划 | 评估部署计划的合理性 |
| 编写培训计划 | 1.5 | 培训计划 | 培训受众的反馈 |
| 编写系统维护计划 | 1.6 | 系统维护计划 | 评估维护计划的合理性 |
| 编写验收测试计划 | 1.9 | 项目总结计划 | 评估总结计划是否完整全面 |
| 完成验收 | 1.14 |  | 验收是否顺利 |

# 附录

## 3.1 主要技术分析

### 3.1.1 前端技术

**（1）HTML 语言**

超文本标记语言，即 HTML（HyperText Markup Language），是标准通用标记语言下的一个应用，用于描述因特网上的网页文档。浏览器通过对其内容进行阅读，根据标记符解释和显示其标记的内容，显示出网页页面的展示效果。

**（2）CSS 样式表**

层叠样式表，即 CSS（Cascading Style Sheets），是一种用来表现 HTML 等文件样式的计算机语言，在网页中能够对网页中元素位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力。 通过 CSS 的设定可以实现在不同分辨率下网页一定程度上的响应式显示。

**（3）JavaScript 编程语言**

JavaScript 一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。因为其解释器，即 JavaScript 引擎是现代浏览器的一部分， 因此其可以广泛用于客户端，配合 HTML 和 CSS 在网页上使用，可以用来给网页增加动态功能，实现响应式布局。

1. **Vue.js**

是一个构建数据驱动的 web 界面的渐进式框架。Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。它不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与单文件组件和 Vue 生态系统支持的库结合使用时，Vue 也完全能够为复杂的单页应用程序提供驱动。

1. **Element框架**

Element是由阿里旗下饿了么技术团队开发的前端UI框架，界面美观，上手容易，是时下流行的前端UI框架之一。

### 3.1.2 后端技术

**（1）Web 服务器**

Web服务器是一种驻留于因特网上某种类型计算机的程序，它用于放置网站的部分或全部文件，同时基于 HTTP协议向Web客户端（较典型的如客户机浏览器）提供各种类型的文档。除此之外，Web 服务器还可以通过在其将为客户端提供信息的基础上运行服务端脚本和程序。目前较常见的三款 Web服务器为跨平台的开源软件Apache和Nginx，以及 Windows 平台独占的IIS。

**（2）数据库系统**

数据库是按照数据结构来组织、存储和管理数据的仓库，它基于计算机硬件的特性进行设计，用于快速地对于计算机数据进行存储以及读取操作，同时提供了针对计算机的软硬件提供了较强的鲁棒性；数据库系统是由数据库及其管理软件组成的系统，是为适应数据处理的需要而发展起来的一个为实际可运行的存储、 维护和应用系统提供数据的软件系统，是存储介质、处理对象和管理系统的集合体。

**（3）PHP编程语言**

超文本预处理器语言，即 PHP（PHP: Hypertext Preprocessor），是一种通用开源脚本语言，主要使用于 Web开发领域。PHP是通过将程序代码嵌入到 HTML文档中，在客户端进行请求时根据由 Web服务器根据其请求的内容进行脚本的处理，通过这一方法为客户端提供动态网页。PHP支持所有流行的数据库以及操作系统，并可以通过C语言以及 C++语言进行扩展，可以为网页后端的搭建提供较大的便利。

## 3.2 数据字典

### 3.2.1 数据流定义表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 数据流名 | 来源 | 去向 | 组成 | 说明 |
| L1 | 添加房屋 | 用户 | 系统 | E1+E2+E4+E5 | 房东添加房屋。 |
| L2 | 编辑房屋设施 | 用户 | 系统 | E1+E2+E4+E6  +E8+E9+E10 | 在某栋房屋的某间房间(位置)添加设施 |
| L3 | 编辑分组 | 用户 | 系统 | E1+E2+E4+E6  +E7 | 删除/添加/更改房屋分组 |
| L4 | 编辑房屋信息 | 用户 | 系统 | E1+E2+E4+E5 | 修改房屋信息 |
| L5 | 删除房屋 | 用户 | 系统 | E1+E2+E4+E5 | 房东删除房屋 |
| L6 | 导入维修报告 | 用户 | 系统 | E1+E2+E4+E5  +E6+E7+E8+E9  +E10 | 用户导入维修报告 |
| L7 | 生成维修报告 | 系统 | 用户 | E4+E5+E6+E7  +E8+E9+E10 | 系统获取房屋/房间/设施的信息，生成维修报告给用户 |
| L8 | 查看房屋情况 | 用户 | 用户 | E4+E5+E6+E7  +E8+E9+E10 | 系统获取房屋/房间/设施信息展现给用户看 |
| L9 | 申请维护 | 用户 | 系统 | E1+E2+E4+E6  +E8+E9 | 用户申请维护 |
| L10 | 更新房屋情况 | 用户 | 系统 | E1+E2+E4+E6  +E8+E9+E10 | 用户（更新/添加/删除）房屋设施信息 |
| L11 | 提交维修订单 | 用户 | 系统 | E1+E2+E4+E6  +E8+E9+E10  +E11+E12+E13  +E17 | 用户提交维修订单 |
| L12 | 取消维修申请 | 用户 | 系统 | E1+E12+E11  +E14+E13+E17 | 用户取消维修申请 |
| L13 | 评价维护服务 | 用户 | 系统 | E1+E12+E15  +E16+E17 | 用户评价维修服务 |
| L14 | 维修商接受订单 | 用户 | 系统 | E1+E2+E11  +E12+E17 | 修商接受用户订单 |
| L15 | 维修商完成订单 | 用户 | 系统 | E1+E12+E17 | 维修商完成用户订单 |

### 3.2.2 数据元素定义表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 数据元素名 | 类型 | 值域 | 说明 |
| E1 | 用户编号 | 字符 | "00000000".."99999999" |  |
| E2 | 用户类型 | 字符 | {"Landlord","Tenant","Repairer"} |  |
| E3 | 用户信息 | 字符 |  | 1-200位英文字符 |
| E4 | 房屋编号 | 字符 | "00000000".."99999999" |  |
| E5 | 房屋信息 | 字符 |  | 1-200位英文字符 |
| E6 | 房间编号 | 字符 | "000".."999" |  |
| E7 | 房间信息 | 字符 |  | 1-200位英文字符 |
| E8 | 设施编号 | 字符 | "00".."99" |  |
| E9 | 设施信息 | 字符 |  | 1-200位英文字符 |
| E10 | 设施名 | 字符 |  | 1-30位英文字符 |
| E11 | 上门维修时间 | 时间 | YYYY-MM-DD hh:mm:ss |  |
| E12 | 订单编号 | 字符 | "00000000".."99999999" |  |
| E13 | 订单价格 | 数字 | "0.00".."999999.99" |  |
| E14 | 当下时间 | 时间 | YYYY-MM-DD hh:mm:ss |  |
| E15 | 评价分数 | 数字 | "1".."5" |  |
| E16 | 评价内容 | 字符 |  | 1-1000位英文字符 |
| E17 | 订单状态 | 字符 | {"completed","accepted","cancel",  "not accepted"} |  |

### 3.2.3 数据精度表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据 | 类型 | 精度要求 | 说明 | 示例 |
| 用户编号 | 字符 | 8位数字字符 | 8位 | 12345678 |
| 用户类型 | 字符 | 6-8位英文字符 | {"Landlord","Tenant","Repairer"} | Tenant |
| 用户信息 | 字符 | 1-200位英文字符 |  | Rice 18 yunnan  18145214578 |
| 房屋编号 | 字符 | 8位数字字符 | 8位 | 00000001 |
| 房屋信息 | 字符 | 1-200位英文字符 |  | \*\*\*Road 123 |
| 房间编号 | 字符 | 3位数字字符 | 3位 | 012 |
| 房间信息 | 字符 | 1-200位英文字符 |  | Two floor  First room |
| 设施编号 | 字符 | 2位数字字符 | 2位 | 11 |
| 设施信息 | 字符 | 1-200位英文字符 |  | Broken |
| 设施名 | 字符 | 1-30位英文字符 |  | Air conditioning |
| 上门维修时间 | 时间 | YYYY-MM-DD hh:mm |  | 2018-12-2  23:59 |
| 订单编号 | 字符 | 8位数字字符 | 8位 | 77777777 |
| 订单价格 | 数字 | 2位浮点数 |  | 356.58 |
| 当前时间 | 时间 | YYYY-MM-DD hh:mm:ss |  | 2018-12-2  21:50:50 |
| 评价分数 | 数字 | 整数 | 1-5 | 4 |
| 评价内容 | 字符 | 1-1000位英文字符 |  | Good |
| 订单状态 | 字符 | 6-12位英文字符 | {"completed","accepted","cancel",  "not accepted"} | completed |

## 3.3 数据库设计

### 3.3.1 核心数据表定义

|  |  |
| --- | --- |
| 表名 | 功能说明 |
| AUTH\_USER | 用户组 |
| OWNER | 房东 |
| RENTER | 租客 |
| ADMINISTRATOR | 管理员 |
| MAINTAINER | 维修商 |
| SUPPLIER | 供应商 |
| HOUSE | 房屋 |
| FACILITIES | 房屋内设施 |
| SMART\_HOME | 智能家居（空调）\* |
| OWNER\_HOUSE | 房东与房屋的关联信息 |
| MAINTAIN\_ORDER | 维修与维护订单 |

\* 原则上每种智能家居都应当有单独的表，此处为简化模型，一切智能家居均暂以空调指代

### 3.3.2 AUTH\_USER表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | AUTH\_USER | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key |  |
| Password | varchar(128) | not null |  |  |
| last\_login | datetime(6) | not null |  |  |
| is\_superuser | tinyint(1) | not null |  |  |
| username | varchar(150) | not null |  |  |
| first\_name | varchar(30) | not null |  |  |
| last\_name | varchar(30) | not null |  |  |
| email | varchar(254) | not null |  |  |
| Phone | varchar(254) | not null |  |  |
| Role | Int | not null |  | 表征角色 |
| is\_active | tinyint(1) | not null |  | 是否处于登录中 |
| date\_joined | datetime(6) | not null |  |  |

### 3.3.3 OWNER表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | OWNER | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key |  |
| is\_auth | int(11) | not null |  | 是否有权限发布房源 |

### 3.3.4 RENTER表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | USER | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key |  |
| is\_auth | int(11) | not null |  | 是否有权限租房 |
| house\_id | Varchar(254) |  |  | 租住的房屋ID |

### 3.3.5 ADMINISTRATOR表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | ADMINISTRATOR | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key |  |

### 3.3.6 MAINTAINER表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | MAINTAINER | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key |  |
| company\_id | varchar(254) | not null |  | 公司注册id（不同于用户id） |
| company\_name | varchar(254) | not null |  | 公司注册名称 |
| address | varchar(254) | not null |  | 公司地址 |

### 3.3.7 SUPPLIER表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | SUPPLIER | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key |  |
| company\_id | varchar(254) | not null |  | 公司注册id（不同于用户id） |
| company\_name | varchar(254) | not null |  | 公司注册名称 |
| address | varchar(254) | not null |  | 公司地址 |

### 3.3.8 HOUSE表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | HOUSE | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key | 房屋的唯一id |
| description | varchar(254) | not null |  | 房屋简单描述（名称等） |
| address | varchar(254) | not null |  | 房屋地址 |
| Score | int(11) | not null |  | 房屋风评（得分） |
| price | decimal(10,2) | not null |  | 房屋租金单价（以月计） |

### 3.3.9 FACILITIES表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | FACILITIES | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | primary key | 设施id |
| buyer\_id | int(11) | not null |  | 所有者 |
| source\_id | int(11) | not null |  | 设施来源（供应商id） |
| description | varchar(254) | not null |  | 设施简单描述（可视为设施的名称） |
| house\_id | int(11) | not null | foreign key | 对应的房屋id |
| group | varchar(254) | not null |  | 默认分组名称 |
| cluster | varchar(254) | not null |  | 设施类别 |
| last\_check | datetime(6) | not null |  | 上次维护时间 |
| status | int(11) | not null |  | 设施状态 |
| life | int(11) | not null |  | 上次维护后该设施预计寿命（min） |
| is\_smart | int(11) | not null |  | 是否为智能家居 |

### 3.3.10 SMART\_HOME表（以AC\_TABLE为例）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | SMART\_HOME(AC\_TABLE) | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| id | int(11) | not null | foreign key | 设施id |
| brand | varchar(254) | not null |  | 空调品牌 |
| temp | int(11) | not null |  | 空调温度 |
| mode | int(11) | not null |  | 空调模式 |

### 3.3.11 OWNER\_HOUSE表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | OWNER\_HOUSE | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| owner\_id | int(11) | not null | foreign key | 房东id |
| house\_id | int(11) | not null | foreign key | 房屋id |

### 3.3.12 MAINTAIN\_ORDER表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | MAINTAIN\_ORDER | | | |
| 列名 | 数据类型 | 空/非空 | 约束条件 | 其他说明 |
| time | datetime(6) | not null |  | 订单提交时间 |
| time\_finish | datetime(6) | not null |  | 订单结束时间 |
| type | int(11) | not null |  | 订单类型（检测还是修复，全局还是单个） |
| launch\_id | int(11) | not null |  | 发起人id |
| house\_id | int(11) | not null | foreign key | 房屋id |
| faci\_id\_list | varchar(300) | not null |  | 被维修设施id列表 |
| mat\_id | int(11) | not null | foreign key | 接单维修商id |
| description | varchar(1000) | not null |  | 修理事由描述 |
| status | int(11) | not null |  | 订单状态（接单、修理中、完成） |
| score | int(11) | not null |  | 订单评分 |
| new\_life | int(11) | not null |  | 维护后设施的使用寿命 |
| price | decimal(10,2) | not null |  | 花销 |