

斗量车联

软件设计报告

选题： 校园电动车互联平台 .

组号： 第七组 .

队员： 曾庆聪、蔡万霖、林靖淞、黄舒俊、郑捷
林莲巧、尚思佳、黄玺琚 .

目录

第一部分 概述	4
1.1 编写目的	4
1.2 编写依据	4
第二部分 软件概述	4
2.1 软件总体描述	4
2.2 软件设计约束及有关说明	4
2.3 使用者特点	5
第三部分 开发和运行环境	5
3.1 硬件环境	5
3.2 软件环境	5
3.3 接口	6
3.3.1 系统外部接口	6
3.3.3 系统内部接口	6
3.4 控制和操作	6
第四部分 体系结构设计	6
4.1 体系结构风格	6
4.1.1 原型设计	6
4.1.2 软件体系结构风格	10
4.2 系统的环境-定义外部接口	10
4.3 体系结构化精细为顶层构件模型	10
4.4 描述系统实例	10
第五部分 构件设计	11
5.1 标识功能类	11
5.2 确定基础设施类	13

5.3 细化设计类	13
5.3.1 主页导航	13
5.3.2 社区模块	13
5.3.3 用户模块	14
5.3.4 好友中心	15
第六部分 数据设计	15
6.1 概念结构设计（E-R 图）	15
6.2 逻辑结构设计	16
6.3 物理结构设计	16
6.3.1 数据的存储结构	16
6.3.2 数据的存取方法	17
第七部分 系统出错处理设计	17
7.1 出错信息	17
7.2 故障处理设计	17
7.3 系统维护设计	17
第八部分 安全性设计	18

第一部分 概述

1.1 编写目的

编写此文档的目的是进一步定制软件开发的细节问题，希望能使本软件开发工作更具体。为了使用户、软件开发者及分析和测试人员对该软件的初始规定有一个共同的理解，它说明了本软件的系统体系结构设计、构件设计及数据设计等，探讨每个模块该如何具体实现，每个实现中需要哪些算法、属性、参数、数据结构、接口，确定了系统的详细功能模块和数据结构，为下阶段开发工作提供依据。

1.2 编写依据

“斗量车联”选题报告：

<https://files.cnblogs.com/files/blogs/704299/%E7%AC%AC%E4%B8%83%E7%B8%B4%E6%96%97%E9%87%8F%E8%BD%A6%E8%81%94%E9%80%89%E9%A2%98%E6%8A%A5%E5%91%8A.zip>

“斗量车联”需求分析报告：<https://docs.qq.com/pdf/DTktjeEJtbEZXY2xz>

第二部分 软件概述

2.1 软件总体描述

该软件是微信小程序，一个支持多平台的软件程序。

它是一款针对于大学生而设计的小程序，是一个带有车辆绑定，车牌识别，车辆绑定的车辆信息联络平台。它帮助大学生更为安全便捷地管理自己的车辆信息。

2.2 软件设计约束及有关说明

- 开发环境：
 - 后端：Python v3.7、Golang v1.16、
 - 前端：微信小程序基础库 v2.16.0、Vue v3.3.0
 - 数据库：MySQL v5.7.30、MongoDB v4.4.6、Redis v4.0.9
- 遵循的规范：软件的设计和开发过程根据《软件设计规范说明书》来进行。软件开发过程应遵循软件工程规范，对过程和版本进行管理和控制；

- 测试环境：安装有微信的 Android / iOS 系统，对硬件网络无特殊要求；
- 软件交付形式：微信小程序软件包
- 软件交付日期：2021.11.14 前（V1.0）

2.3 使用者特点

本软件目前主要在福州大学城附近使用，使用者包括具有电动车等的大学生。

第三部分 开发和运行环境

3.1 硬件环境

"斗量车联"小程序运行环境为 windows 系统

需求名称	详细要求
CPU	i3-6255 及以上
内存	2g 以上
硬盘	至少有 2g 空闲的存储空间

3.2 软件环境

需求名称	详细要求
iOS 微信微信客户端	版本 8.0.15 及以上
Android 微信客户端	版本 8.0.15 及以上
Harmony 微信客户端	版本 8.0.15 及以上

3.3 接口

3.3.1 系统外部接口

- 外部系统

通过远过程调用 RPC 来实现同步的数据交换，面向大众服务。外部实体通过接口与系统进行通信。

- 用户接口

用户界面部分按小程序用户界面的规范来设计，界面设计风格与微信环境保持一致，采用导航栏以及对话框等方式作为用户界面，降低用户使用成本。

3.3.3 系统内部接口

- 构件内部接口

通过构件之间的类进行数据交换，数据交换是同步的。

- 构件间接口

通过远过程调用实现数据交换，采用 CORBA 框架，基于 JSON 格式，完成数据间的交互。

3.4 控制和操作

- 该小程序最终交付方式为微信小程序软件包。
- 控制该软件运行的方法为通过屏幕点击微信中的小程序启动。
- 该小程序提供的各功能项的设置及使用应符合人们使用手机的操作习惯，控制信号均由屏幕传感器传入

第四部分 体系结构设计

4.1 体系结构风格

4.1.1 原型设计

简约大体，采用主题采用蓝白色调。

1. 主页



2. 用户信息模块



3. 车辆绑定

9:41 AM

100%

亲友车辆绑定

绑定亲友车辆后，可代收亲友消息

* 亲友手机号：

* 电动车品牌：

* 电动车颜色：

* 车牌号：

省市

区域

牌号

拍照识别

绑定成功后可代收亲友消息

绑定

4. 亲友车辆

9:41 AM

100%

亲友车辆

小椰子的亲友1号

蓝色 雅迪 福州·上街Z1664

解除绑定

小椰子的亲友2号

黑色 绿源 福州·上街Z4321

解除绑定

新增亲友

5. 消息列表



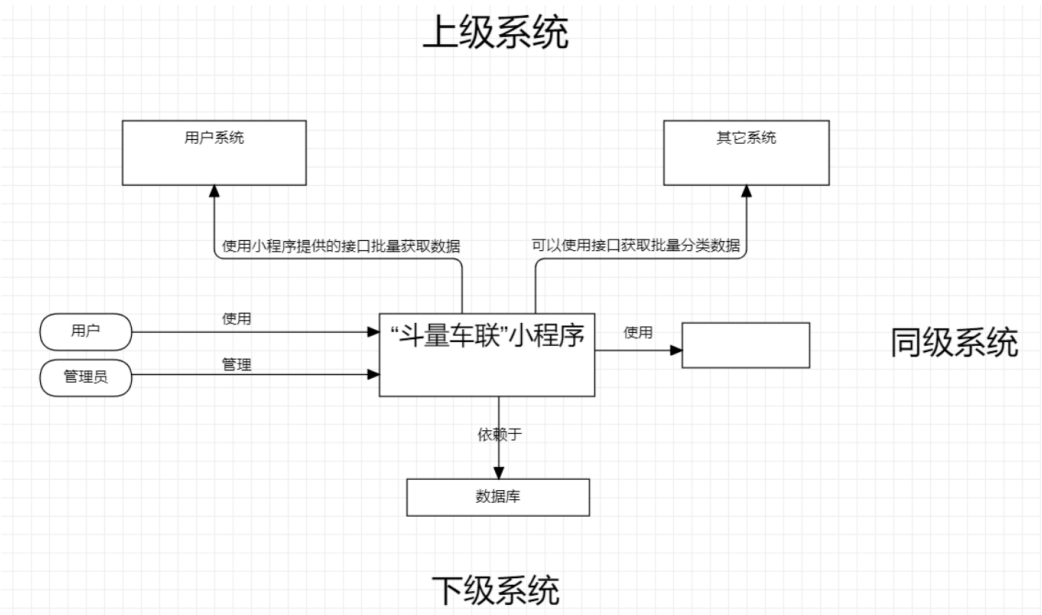
6. 消息详情



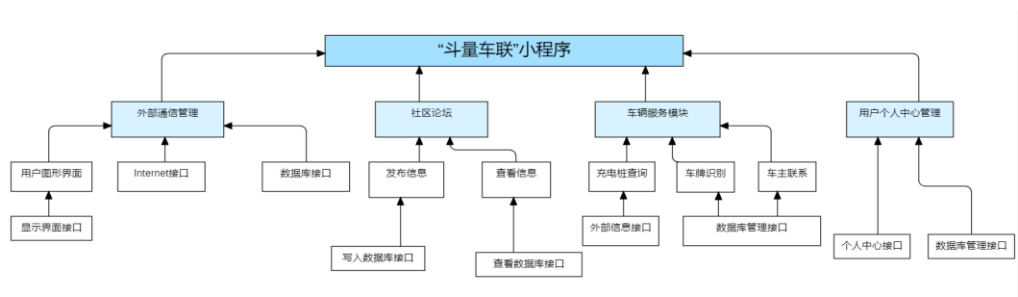
4.1.2 软件体系结构风格

“斗量车联”小程序采用面向对象体系结构，系统的构件封装了数据和应用到该数据上的操作，构件间通过消息传递进行通信和合作。

4.2 系统的环境-定义外部接口



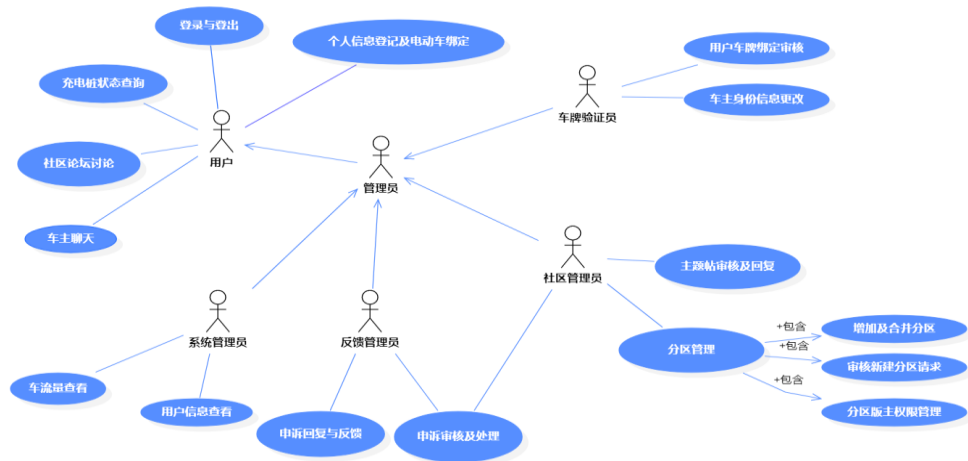
4.3 体系结构化精细为顶层构件模型



4.4 描述系统实例

操作	描述
用户认证个人资料	编辑自身认证信息（学号、班级等等）、个人信息（涉及隐私部分仅管理员可见）等。
绑定信息	用户提供车辆信息（车牌名称等），手机号码及相应凭证等证明自身可靠性。
社区模块	可以通过平台发放各种帖子，比如买卖二手车等。
交流板块	可以通过车牌号识别出车主信息，然后联系上车主，当然会对车主的隐

	私进行合理的保护。
管理员权限	管理员管理车辆、用户、社区，核对信息是否相符及其可靠性完整性，出现违规情况时，可将违规用户永久拉入黑名单。交流板块管理员管理交流版块，核查不当言论，对主题帖及相关发言用户账号进行处理。



第五部分 构件设计

5.1 标识功能类

- 1. 普通用户类
- 2. 车类
- 3. 评论类
- 4. 帖子类
- 5. 反馈类
- 普通用户类

User
+ UserID: uint
+ Relations: []user
+ UserName: string
+ OpenID: string
+ Avatar: string
+ Login(): return user
+ UserInfo(id): return user
+ Show(id): return user
+ Create(id, u_id):return user
+ Delete(id, u_id):return int

- 车类

Car
+ CarName: string
+ CarNum: string
+ CarImages: string
+ CarBossId: uint
+ CarId: car
+ Create(userID): return car
+ Delete(CarId): return int
+ Show(userID): return car
+ Search(CarNum): return car

- 评论类

Comment
+ Content: string
+ Children: []Comment
+ UserID: uint
+ SocialId: uint

- 帖子类

Society
+ SocietyID: uint
+ Title: string
+ Content: string
+ Picture: string
+ UserID: uint
+ CategoryID: uint
+ Create(): return Society
+ Show(): return Society
+ Delete(): return int
+ Update(): return Society
+ Search(): return Society

- 话题分类帖子

Category
+ CategoryID: uint
+ CategoryName: string
+ EnglishName: string

5.2 确定基础设施类

login 类、register 类、binding 类、delete 类、update 类、upload 类、update 类、image 类、scroll-view 类，input 类，普通用户类、认证用户类、管理员类、数据库管理类（单例）

5.3 细化设计类

5.3.1 主页导航

主页统一风格，简洁明了，降低用户学习成本。

5.3.2 社区模块

1.社区主页面

顶部使用一个 view 容器固定，使用流式布局，分为上下两个容器，上容器再通过流式布局分为左右两个部分，左边使用 icon，右侧容器为定位功能的文本，点击可以展开选择定位的 view，下容器包含一个 input 表单组件。然后创建一个 scroll-view，在其中创建 3 个 view 容器。

第一个使用 cover-view 通过横向 flex 布局三个小 view，分别存放三个功能 logo，并设置点击事件。

第二个 view 为筛选功能栏，右侧为一个下拉 view，并为其设置多个 block 来进行筛选选项的选择。

第三个 view 为信息区，通过当前的数据来设置多个信息栏容器，其中左侧为 image，来展示用户想要展示的图片，右侧文字区。

2.发布帖子

发布职位的页面类似问卷式交互，整体通过一个纵向的 `scroll-view` 封装，然后涵盖了多个 `view`，并细化其功能。其中有使用 `image`、`swiper`、`text`、以及 `input`、`checkbox`、`button`、`textarea` 等多种表单组件。

3.帖子详情页面

首先顶部使用一个 `view` 容器容纳 5 个横向 `flex` 布局的 `view`，通过调整样式控制五个 `view` 容器的位置。接着使用 `image` 以及 `text` 来完善外观，并用 `bindtap` 属性来编写点击的效果代码。接着主题内容使用 `scroll-view`，在其内部设置多个 `view`，分别展示基础信息。其中职位描述部分通过 `wx-if` 设置多个标签 `view`。接下来也是通过设置多个嵌套 `view` 的方式完成家庭信息的排版。最后在底部使用一个 `button` 组件，完成整个简历的提交。

4.评论部分

用户可以通过 `input` 类输入框对帖子进行评论，也可以点赞或是举报评论。

5.反馈意见

反馈意见的主页面主要通过两个 `view`，右侧设置 `icon` 并绑定 `wx.navigateTo` 的 API 实现页面的滑动跳转。

6.举报界面

使用 3 个 `view` 容器，其中第一个与第三个的内部使用 `input` 处理单行输入，第二个供用户进行举报的相关截图的上传，实现原理同产品建议。

5.3.3 用户模块

1.用户个人界面

采用毛玻璃的形式进行底部填充，然后再用透明 `scroll-view` 对内容进行填充。该界面采用简单的表单编辑布局，多个 `view` 垂直居中排列，每一个的左侧使用 `text` 描述表单头，右侧使用 `input` 组件与 `button` 组件。其中 `button` 控件主要引导图片上传的相机调用动作与性别的选择。底部 `button` 组件绑定确定的弹窗，弹窗使用 `cover-view`，内容包含 `button`、`text` 组件。

2.绑定车辆&邮箱&手机号码

该界面属于一般的表单编辑功能页，主要采用 `scroll-view` 试图容器内部设置多个 `flex` 布局的 `view` 容器，并在每一个 `view` 里使用 `input`、`picker-view`、`text-area`、`button` 等表单组件。

3.编辑其他信息

主要使用 **view** 对三个子功能进行封装，子功能均使用一个 **view** 对 **text** 以及 **image** 进行封装，并对该子 **view** 设置 **bandtap** 属性跳出到各自的子页面。

5.3.4 好友中心

1.好友信息

好友信息相比于个人中心界面，主要采用 **view** 容器装在 **input** 与 **text-area** 两种表单组件。

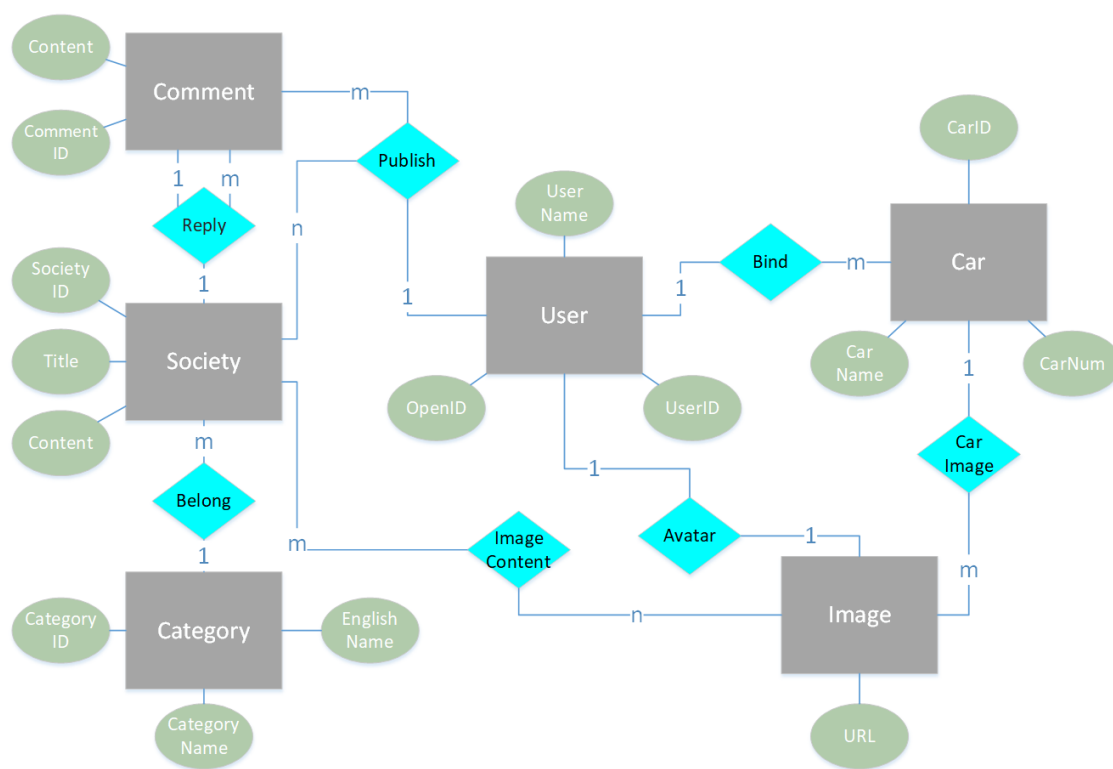
2.聊天模块

主要采用 **ws** 进行长连接的建立

第六部分 数据设计

6.1 概念结构设计（E-R 图）

产品的开发主要是为了解决大学生管理电动车的需求，为学生提供充电桩信息，帮助广大学生群体能安全的找到空闲的充电桩。通过需求分析，将概念模型用如下的 E-R 图来表示。



6.2 逻辑结构设计

完成概念结构结构设计后，进行逻辑结构设计。

主要的关系模式如下：

申诉&举报（对象 ID，内容）

交流区管理员（交流区管理员 ID，所属板块）

主题帖（主题帖 ID，用户 ID，文本内容，文本格式，主题帖状态，主题帖类型）

用户（用户 ID，邮箱，联系方式，学号，性别，姓名，昵称）

车辆（车辆 ID，用户 ID，颜色，品牌，车牌号，购买年份）

将一些关系转为关系模式如下：

管理（交流区管理员 ID，主题帖 ID，文本内容，文本格式，.....）

用户认证（用户 ID，权限，学号，.....）

车辆绑定（用户 ID，车辆 ID，车牌，颜色，品牌，.....）

好友关注（用户 ID，好友 ID）

6.3 物理结构设计

6.3.1 数据的存储结构

1. 存放位置

采用多磁盘存放，将表和索引等分开进行存放，加快存取速度。进行数据备份和日志文件备份，可以把这部分放在磁带或磁盘上，有利于数据库的转储和恢复。

2. 系统配置

数据库连接数设置为 500-1000

联机日志文件组数设置为 6 组以上，大小 1G

内存配置 `sga_max_size=1200M,sga_target=1200M`

更多的配置参数就不一一列出

6.3.2 数据的存取方法

采用 B+树索引方法存取方法，在关系的属性上建立索引。例如在用户 ID 的属性上建立索引，能更有效的实现查询。

第七部分 系统出错处理设计

7.1 出错信息

故障情况	处理方法
同时使用人数过多，导致服务器过载	重新设置负载均衡，或是后端增加并发量
网络延迟过高	重新加载小程序
数据量过大导致数据库故障	利用数据库备份恢复数据
连续大量发帖，或是刷新充电桩信息的恶意行为	启用保护机制，同时封禁用户

7.2 故障处理设计

- 1、页面长时间无响应，提出警告，要求用户刷新。
- 2、页面载入时间过长，要有提示，防止用户认为系统有问题。
- 3、设置数据备份模块，防止数据库出现故障导致数据丢失。
- 4、经济允许的条件下，尽量使用性能更好的服务器，防止用户量过大导致的故障。
- 5、经济允许的条件下，可以购买备用服务器，出现紧急故障仍能提供服务。

7.3 系统维护设计

1. 设置有软件日常运行管理员，负责软件的日常检查维护。
2. 同时在论坛模块保留了拓展的接口，日后想要进行更新迭代，可以在论坛模块进行拓展。
3. 设置维护文档模块，将每次的维护、更新进行记录，以便需要时查看。

第八部分 安全性设计

1. **账户管理：**保证用户权限的正确设置，防止出现越权行为，保证平台的正常良好运行。账户的产生、修改、变更、删除以及身份认证应采用统一的身份认证平台来实现。
2. **加密：**使用 AES 对称加密方法，使用正确的算法和密钥大小，确保加密密钥的安全。要有足够的防御能力，防止用户信息被不良分子窃取。
3. **敏感数据保护：**应用软件应包含数据安全设计：包括数据库的安全、数据采集、数据传输、数据处理、数据存储、数据备份和恢复的安全。对重要的、敏感数据应进行加密和完整性保护。处理诸如地址、档案等用户私人信息的应用程序应该采取专门的步骤，来确保这些数据的保密性，并确保其不被修改。
4. **审计：**未来软件应该支持审计日志，应包含以下事件由于超出尝试次数的限制而引起的拒绝登录；成功或失败的登录；用户权限的变更；用户密码的变更等。