

BusHome

软件设计规格说明书

所属学院：计算机学院

小组名称：生煎包队

小组成员：林博程 郭志伟 吴豪 胡熙

王奂博 王振豪 陈上委 邓浩龙

张伟凌 林景峰

指导老师：程永利

目录

- 1. 引言 1
 - 1.1 编写目的 1
 - 1.2 项目背景 1
 - 1.3 参考资料 1
- 2. 产品概述 1
 - 2.1 总体描述 1
 - 2.2 用户特点 1
 - 2.3 功能需求 1
 - 2.3.1 产品主要功能说明表 1
 - 2.3.2 产品功能结构图 2
 - 2.4 软件环境 2
 - 2.5 接口 2
 - 2.5.1 系统外部接口 2
 - 2.5.2 系统内部接口 2
 - 2.6 性能需求 3
 - 2.7 其他需求 3
- 3. 总体设计 3
 - 3.1 系统体系结构设计 3
 - 3.1.1 体系结构风格 3
 - 3.1.2 定义外部接口 3
 - 3.1.3 顶层构件模型 3
 - 3.1.4 系统实例 3
 - 3.2 构件设计 4
 - 3.2.1 主要功能类 4
 - 3.2.2 基础设施类 5
 - 3.3 数据设计 5
 - 3.3.1 概念结构设计 5
 - 3.3.2 逻辑结构设计 5
 - 3.3.3 数据字典 6
 - 3.4 其他设计 7
 - 3.4.1 安全性设计 7
 - 3.4.2 故障处理设计 7
 - 3.4.3 系统维护设计 7
- 附：小组本次分工情况 8

1 引言

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 项目背景

- (1) 软件名称：BusHomie；
- (2) 项目开发者：软件工程-2021-C 班（福州大学）——生煎包组；
- (3) 项目组长：林博程；
- (4) 开发人数：10 人；
- (5) 开发目标：开发一款方便学生群体返乡拼车、规范打车流程的订票小程序；
- (6) 项目面向用户：有返乡拼车需求的大学生群体、老乡车车主、旅游公司；

1.3 参考资料

- (1) 选题报告：<https://www.cnblogs.com/rakanXayan/p/15367691.html>
- (2) 需求分析报告：<https://www.cnblogs.com/rakanXayan/p/15420995.html>

2 产品概述

2.1 总体描述

步入大学,大学生经常会选择与同乡的同学在假期一起回家。但由于存在返乡返校转车多、时间长、车站离家离学校远等问题，学生每次回家返校都疲惫不堪。许多旅游公司因此发现商机，建立了老乡车路线。因为老乡车可以直接往返学校与家乡，中途无需转车，且具有高速直达，方便快捷等特点，有一小部分的大学生在返乡时都会优先选择老乡车。但其中存在着一些问题诸如：订票不方便、支付机制不完善、无实体票据，还容易滋生电信诈骗等问题。面对如此庞大的学生群体，老乡车的便捷与缺陷是显而易见的，因此我们决定研究一款方便大学生订购返乡车票的软件。

该软件属于一款依赖于微信平台的 APP。它作为一款针对大学生而设计的小程序，是一个带有发布车辆信息、发布车辆需求以及实时支付功能的信息整合平台。它帮助大学生更为安全便捷地找到自己需要的车辆路线信息，解决了订票中的许多问题。

2.2 用户特点

目标用户：有返乡拼车需求的大学生群体、老乡车车主、旅游公司；

(1) **大学生群体**（之后简称“**乘客**”）：特指有返乡拼车需求的大学生群体。此类用户作为老乡车的乘车人和需求者，有查看老乡车信息、订票的需求，喜好便捷、实惠、可信度高的平台。有部分群体不习惯做大巴，与好友、老乡等几人的小群体包车返乡。

(2) **老乡车车主、旅游公司**（之后简称“**车主**”）：此类用户作为老乡车的车主，发车信息提供者，由于老乡车的传统的方式是以微信群为主要平台，订票、验票不方便，也很难有方便有效的凭证和手段，还有误乘、一票多用、超载等问题，用户体验较差，有寻求大学生市场、老乡车平台的需求，喜好简单、便捷、顾客多的平台。

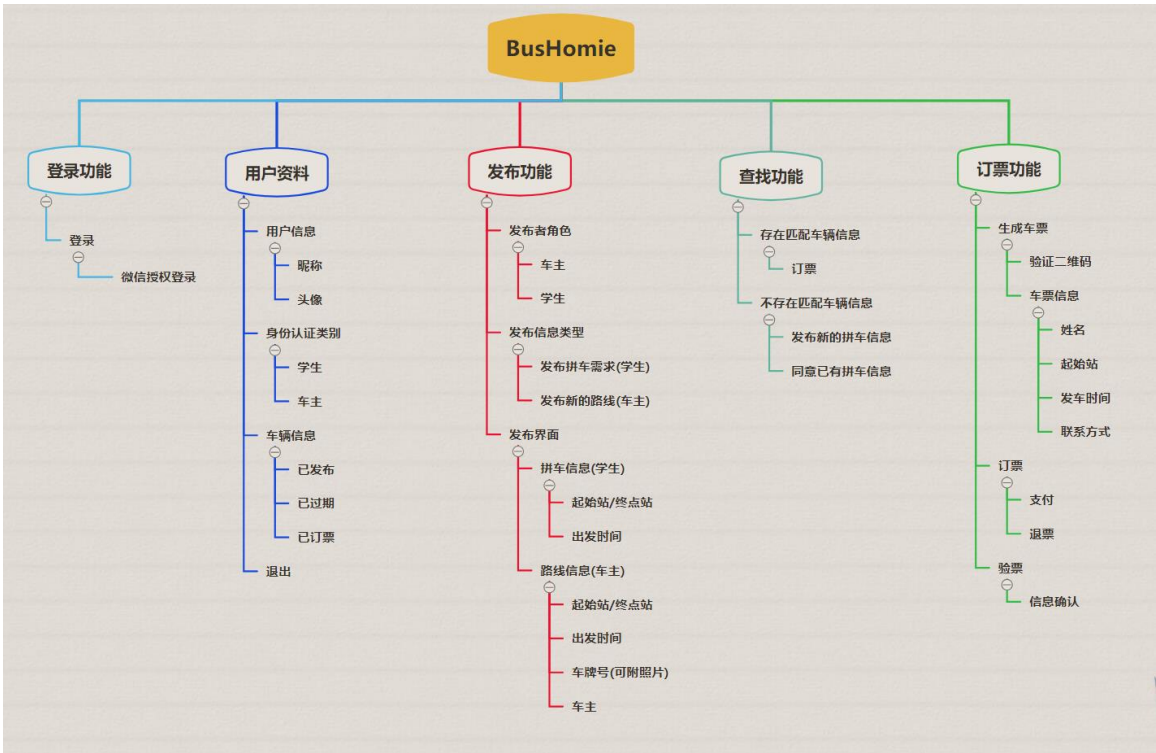
2.3 功能需求

2.3.1 产品主要功能说明表

序号	功能名称	功能需求标识	优先级	简要描述
1	登录	L1	高	进入用户界面的入口
2	信息查找	L2	高	根据起始点、发车时间查找符合的车辆路线信息

3	查看详情	L3	高	查看车辆信息的具体内容
4	订退票	L4	高	进行订票或退票
5	发布	L5	高	车主发布车辆路线信息，或是学生发布拼车的需求信息
6	生成电子车票	L6	高	生成电子车票做为上车凭证
7	验票	L7	高	扫描验证电子车票以确认身份
8	联系	L8	中	与车主电话联系或微信联系
9	用户资料	L9	低	查看(修改)用户信息，查看已发布的车辆信息/拼车信息

2. 3. 2 产品功能结构图



- （1）**发布功能**：对于车主可以随时发布拼车信息，需包括起始站与终点站，发车时间等信息；对于乘客可以发布拼团信息，意为寻求与自己有相同路线需求者，当人数达到一定数量时该条信息可由车主响应并提供车票支付，这样满足了本来无车可乘的乘客的需求。
- （2）**查找功能**：呼应发布功能，区别在于当乘客发现无匹配自己需求的路线时，可选择发布一条拼团信息或是支持他人的拼团信息。
- （3）**订票功能**：功能包括退、订、验票，实现了便捷与安全。

2. 4 软件环境

需求名称	详细要求
iOS 微信客户端	版本 6. 5. 8 及以上
Android 微信客户端	版本 6. 5. 7 及以上

2. 5 接口

2. 5. 1 系统外部接口

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。

2. 5. 2 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，方便用户使用。

2.6 性能需求

由于此小程序针对学生群体,且具有使用高峰时间段(寒、暑假),因此使用性要求比较高。为防止对信息资料和管理程序的破坏,要求有较为可靠的安全性能。总之,要求稳定、安全、便捷,易于管理和操作。

- (1) 查询速度:不超过 10 秒;
- (2) 其他所有交互功能反应速度:不超过 3 秒;
- (3) 可靠性:平均故障间隔时间不低于 200 小时。

2.7 其他需求

能快速恢复系统和故障处理,方便系统升级和扩充,故障恢复时间不超过 5 个小时。

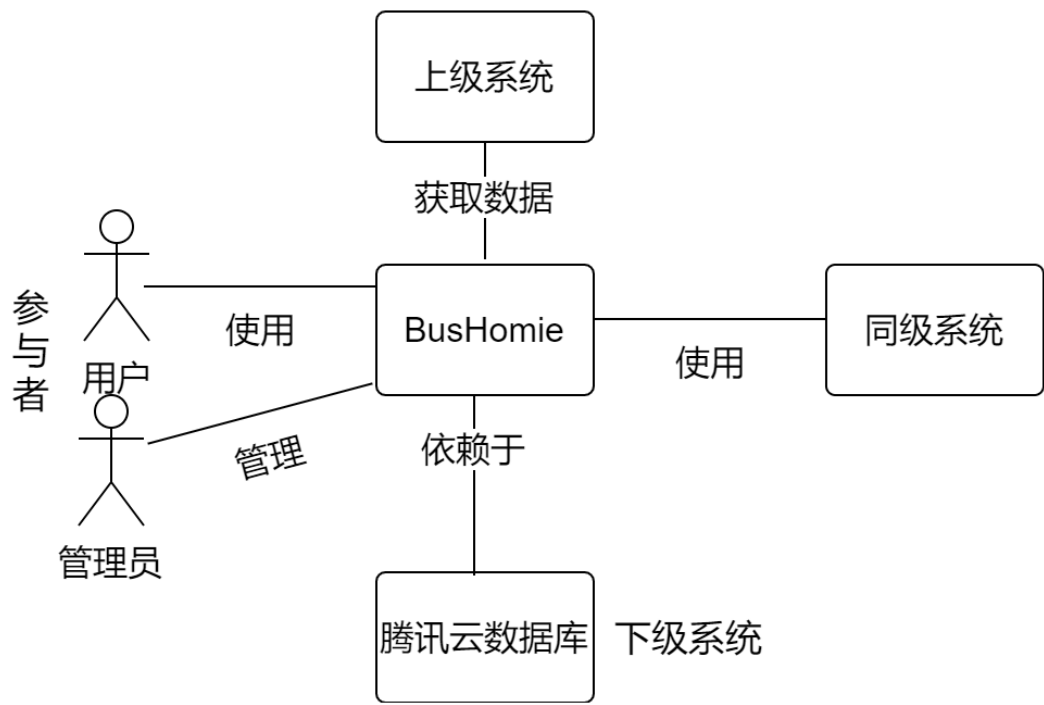
3 总体设计

3.1 系统体系结构设计

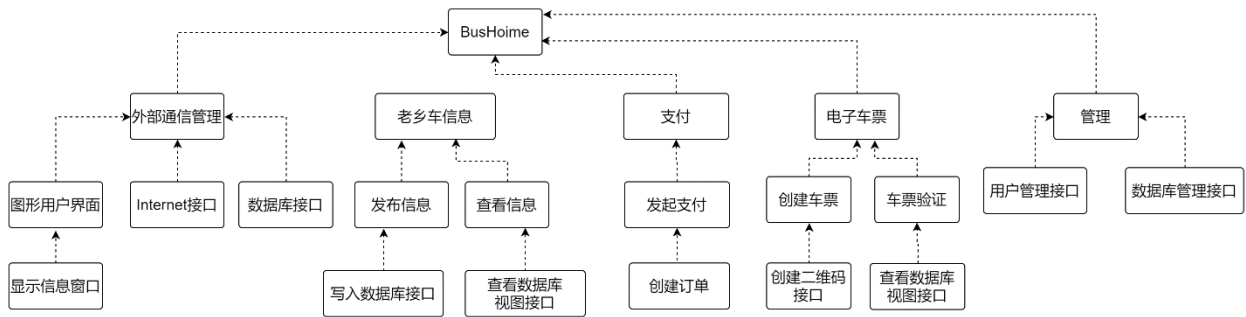
3.1.1 体系结构风格

BusHomie 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。

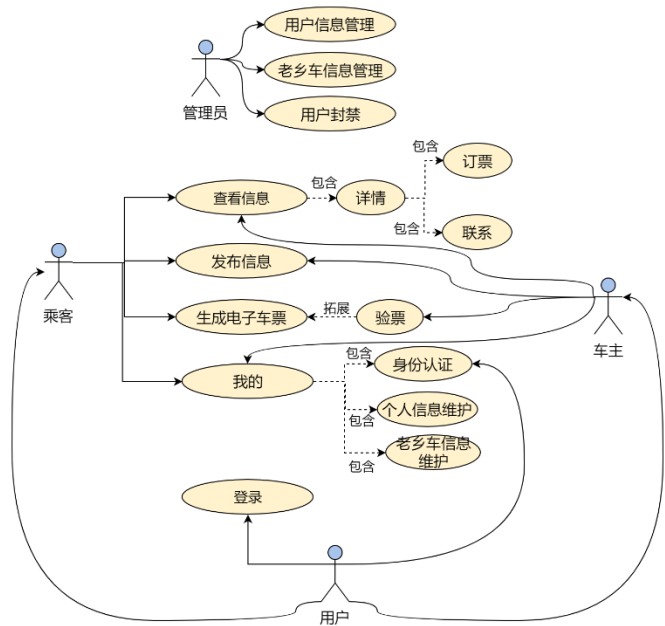
3.1.2 定义外部接口



3.1.3 顶层构件模型



3.1.4 系统实例

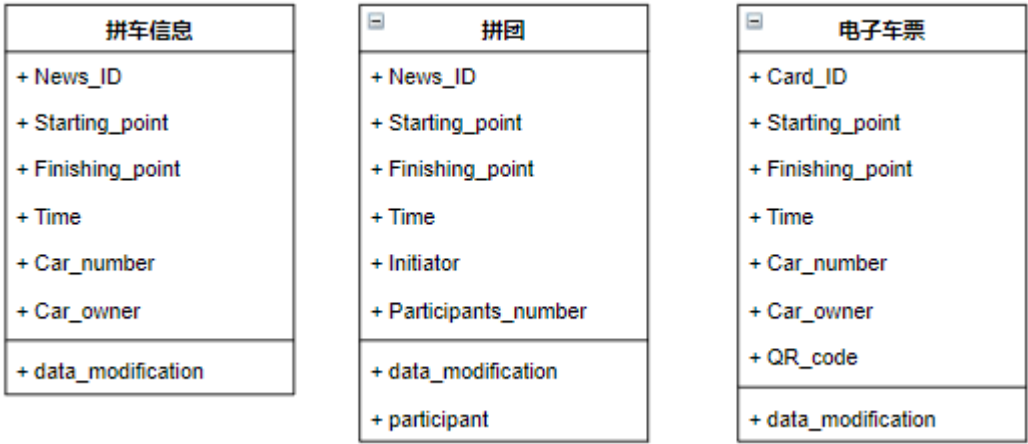


1. 用户登录
- 用户在未登录状态使用软件时，需要登录才能使用。
2. 用户身份认证
- 用户提供自身信息（如姓名、电话号码、车牌号）等身份信息进行身份认证，成为乘客或车主。
3. 信息管理
- 用户可以在“我的”界面管理个人信息和发布、参与过的老乡车信息。
4. 发布信息
- 车主在平台上发布拼团信息，说明起始点、发车时间等，等待乘客加入与联系。
乘客在平台上发布拼车信息，说明起始点、发车时间等，等待其他乘客的加入和车主的响应。
5. 查看信息
- 车主或乘客根据发车时间、起始点等条件查找需要的老乡车信息。
6. 乘客订票
- 当信息得到车主和乘客的参与后，乘客可以进行订票。
7. 乘客生成电子车票
- 在乘客订票后，乘客可以生成带有二维码和个人信息的电子车票，做为身份信息供车主识别。
8. 车主验票
- 车主可以扫描电子车票的二维码，进行验票，验票成功后会使被识别的电子车票失效。
9. 管理员权限
- 管理员管理用户的个人信息，核对信息是否相符及其可靠性、完整性，同时管理已发布的老乡车信息，核查违规信息和违规操作，出现违规情况时，可将违规用户永久拉入黑名单。

3.2 构件设计

3.2.1 主要功能类

用户	乘客	车主	管理员
+ ID + name + nickname + telephone + data_modification + post + certificate	+ ID + name + nickname + telephone + data_modification + post + certificate	+ ID + name + nickname + telephone + data_modification + post + certificate	+ ID + power + post + user_management + news_management + news_management + user_ban

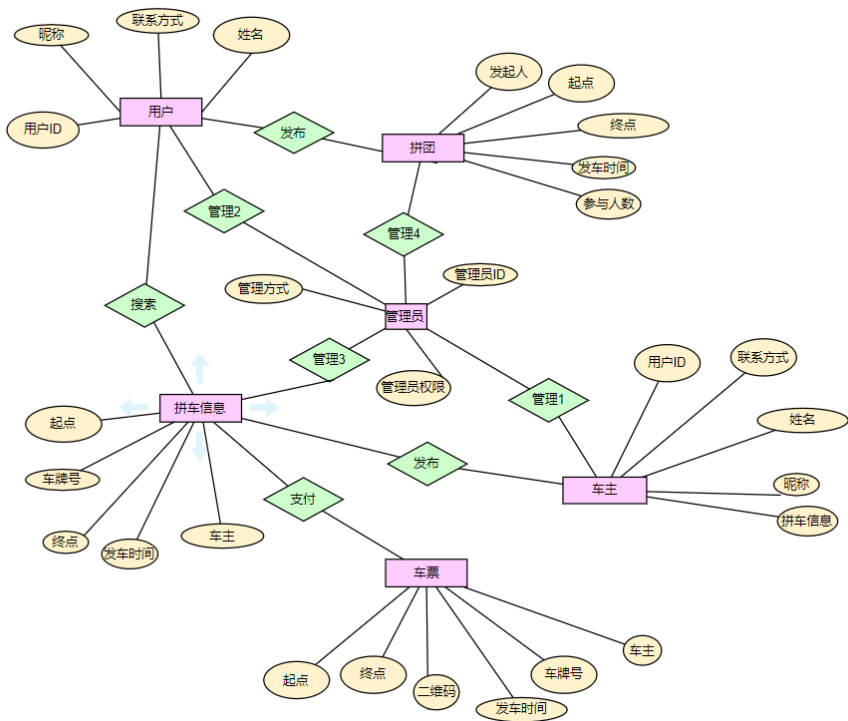


3. 2. 2 基础设施类

- Input 类（输入框）
- scroll-view 类（滚动视图）
- view 类（视图）
- text 类（文本）
- navigator 类（浏览器）
- image 类（图片）
- QR Code 类（二维码）
- button 类（按钮）
- icon 类（图标）
- form 类（表单）
- picker-view 类（滚动选择器）
- switch 类（开关选择器）
- radio-group 类（单项选择器）
- checkbox-group 类（多项选择器）
- 用户类
- 乘客类
- 车主类
- 管理员类
- 数据库管理类（单例）

3. 3 数据设计

3. 3. 1 概念结构设计



3. 3. 2 逻辑结构设计

用户（用户 ID，联系方式，姓名，昵称）
车主（用户 ID，联系方式，姓名，昵称）
拼车信息（信息 ID，车牌号，车主，起点，终点，发车时间）
拼团（信息 ID，发起人，起点，终点，发车时，参与人数）
车票（车票 ID，二维码，车牌号，车主，起点，终点，发车时间）
发布（用户 ID，信息 ID，起点，终点，发车时间，发布时间，截止时间）
搜索（用户 ID，信息 ID，起点，终点，发车时间）
管理员（ID，权限）

3.3.3 数据字典

用户			
名	类型	非空	说明
ID	int	√	主键
Name	varchar	√	
Nickname	varchar		
Telephone	int		

乘客			
名	类型	非空	说明
ID	int	√	主键
Name	varchar	√	
Nickname	varchar		
Telephone	int		

车主			
名	类型	非空	说明
ID	int	√	主键
Name	varchar	√	
Nickname	varchar		
Telephone	int		

拼车信息			
名	类型	非空	说明
News_ID	Int	√	主键
Car_name	varchar	√	外码
Car_number	varchar		
Starting_Point	varchar		
Finishing_Point	varchar		
Time	char		

拼团			
名	类型	非空	说明
News_ID	Int	√	主键
Initiator	varchar	√	外码
Participants_number	Int		
Starting_Point	varchar		
Finishing_Point	varchar		
Time	char		

车票			
名	类型	非空	说明
Card_ID	Int	√	主键
QRcode	Image		
Car_number	int		外码

Car_owner	varchar		外码
Starting_Point	varchar		
Finishing_Point	varchar		
Time	char		

发布关系			
名	类型	非空	说明
Owner_ID	Int	√	主键、外码
News_ID	int	√	主键、外码
Starting_Point	varchar		
Finishing_Point	varchar		
Time	char		
ReleaseTime	char		
Deadline	char		

搜索关系			
名	类型	非空	说明
Owner_ID	Int	√	主键、外码
News_ID	int	√	主键、外码
Starting_Point	varchar		
Finishing_Point	varchar		
Time	char		

管理员			
名	类型	非空	说明
ID	int	√	主键
Power	char		

3.4 其他设计

3.4.1 安全性设计

系统具有较高的安全性。系统对不同的用户提供不同的功能模块，只有具有一定权限的管理员用户才有查询车辆线路、拼车需求信息并且对其进行增、删、调配任务的权限，有添加新线路和新需求的权限，查询乘客、车主信息和执行乘客退票的权限，有管理车辆调度的权限；而一般的普通用户只能查看车辆线路信息和相关拼车信息。

同时应具有一定的保护机制，防止系统被恶意攻击、信息被恶意修改和窃取。有完善的备份机制，如果系统被破坏应能快速恢复。具体包含以下四方面：

（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：不使用自创加密方法，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。

3.4.2 故障处理设计

页面长时间无响应，提出警告，要求用户刷新。

- （1）页面载入时间过长，要有提示，防止用户认为系统有问题。
- （2）设置数据备份模块，防止数据库出现故障导致数据丢失。
- （3）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。
- （4）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。

3.4.3 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）保留了拓展的接口，日后想要进行更新迭代，可以进行拓展。
 - （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。

附：小组本次分工情况

原型设计优化：林博程

前端制作基本页面：陈上委、张伟凌

软件设计规格说明书编写：邓浩龙、胡熙、王奂博、林景峰

（组织者、结构框架、技术支持：胡熙

排版、体系结构设计、构件设计：邓浩龙

数据设计：林景峰

其他文字：王奂博）

问题准备：王振豪

PPT 制作：吴豪

答辩：郭志伟、邓浩龙、陈上委