# 项目规格说明

## 1.1 引言

### 项目背景

在花样繁多的社交软件中，各有各的特色。不然也不能在日新月异的互联网发展潮流中站稳脚跟，但在我看来，在这之中缺少一种符合国人特色的社交软件。中国自古以来便以含蓄为美，从赠人玫瑰，手留余香到桃李不言，下自成蹊。再从桃之夭夭灼灼其华，到窈窕淑女，寤寐求之。都很好的剖析了中国自古以来的人群社交特点。欲交往，止于含蓄。往往只能点到即止。

我之前看到的网上的一则消息中，引发了我的思考。这则消息以共伞为切入点。想要找到合适的“伞友”，甚至可以斥资以达到他们的目的。让我看到了他们躲在“共伞”外表背后的欲拓宽自己的社交圈子亦欲求偶的赤子之心。使我萌发了开发一款以“共伞”打开社交尬境的野心。我相信，它也会以异军突起之势杀入社交市场。在僧多肉少的社交app之中抢得一席之地。

### 项目概述

本项目是以一个基于微信开发的“伞友”社交小程序。实现了个人注册登陆，

首页推荐、查看附近的人，附近的娱乐设施、社交圈、发布动态、点赞、评论、关注、私信等功能。为年轻人社交提供了一步到位的服务。它能在一定程度上缓解当下年轻人的社交障碍。填补了社交市场上此类软件的空白

## 可行性分析

### 技术可行性

（1）技术架构

总体架构系统将采用经典的三层架构模型（表示(view)层、业务(Controller)层、数据(Model)层），即系统至少分为三个大的模块，表示(view)层负责界面设计、业务(Controller)层负责业务计算，数据(Model)层负责数据库的访问。三层架构基于经典的体系模式MVC（Model、View、Controller）模式，该模式是当今设计交互应用程序的一个先行标准，可以直接借鉴。

（2）团队技能

掌握数据结构、算法设计与分析等相关课程，具有良好的编程能力和编程思维。开发人员多次参加暑假ACM集训、程序设计大赛，具有一定的项目开发能力。前端技术栈熟练掌握html、css、javascript、vue、JQuery,后端技术栈node，express，数据库掌握mongoDB，此外还熟练使用两栏、三栏、自适应、绝对定位、浮动、flex布局方式、熟悉MVC设计模式和前后端交互细节。

### 社会可行性

互联网发展可谓日新月异，不进则退。软件开发者想在软件市场分一杯羹不可谓不难。要么就要有庞大的资金链支撑。要么就要用一种独一无二的方式直击用户需求打开市场。饿了么就是很好的正面例子。“伞友”小程序可以在一定程度上缓解当下年轻人的社交障碍，拓宽用户的社交圈。在之后提供了附近的消费场所。在这种层面上也促进了国家GDP发展。

### 结论意见

在社会层面，本项目既促进了国家GDP发展又可以缓解年轻人社交障碍；在技术上，项目采用的技术更贴合市场和用户需求，同时，项目成员是一个有干劲、有想法、有创造力的优秀大学生，再加上有优秀的教师资源进行指导，项目开发成功只是时间问题。综上所述，该项目开发的可行性是极高的。

# 需求规格说明

## 引言

### 项目目的

本项目是在年轻人普遍患社交障碍的大环境下设计与实现的一个微信社交类小程序，项目直接目的是给当代年轻人一种更为含蓄的方式去与他人社交。项目使用了腾讯地图api来实现附近的人，基础设施等功能、网络免费接口实现天气功能。流行框架和自建服务器数据库实现用户互动实时通讯功能。间接目的拉动国家GDP发展。

### 作用范围

以用户登陆位置为中心辐射到周围可见用户范围

### 用户对象

喜爱交友、酷爱社交、或有社交障碍的用户

## 数据描述

### 数据库描述

本项目采用MySQL数据库，MongoDB是一个基于分布式文件存储 的数据库。由[C++](https://baike.baidu.com/item/C%2B%2B" \t "_blank)语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。它是一个介于[关系数据库](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93" \t "_blank)和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似[json](https://baike.baidu.com/item/json" \t "_blank)的[bson](https://baike.baidu.com/item/bson" \t "_blank)格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立[索引](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95" \t "_blank)。

### 数据流图

（1）顶层数据流图

顶层数据流图由车主和易出行平台实体组成，车主可查看限行区域，绑定车辆并提供行驶记录，查询出行日期和出行路线，易出行平台可根据车主提供的数据提供行驶日期规划和出行路线，自动获取车主的位置，并提供附近的生活服务点，在车主出行的前一天发送出行消息提醒。

顶层数据流图如图2-1所示：



图2-1 顶层数据流图

（2）0层数据流图

0层数据流图包含许多数据流条目，包括用户登陆、附近的人、附近的基础设施、地点导航、发布动态、用户点赞、用户评论、关注用户、私信用户等。

0层数据流图如图2-2所示：

图2-2 0层数据流图



## 功能性需求

### 用户界面功能描述

1. 用户登陆功能：用户在“登陆”页面输入个人信息，前端获取用户授权之后自动获取用户登陆的唯一性标识openid并传入后台，前端表单对用户输入信息进行有效性验证。后台对用户传入的信息进行判断，如果没有此用户则后台自动注册并登录，如果存在该用户则判断密码是否正确，正确直接登录、错误给出提示。注册的用户将写入数据库以其他用户的登录功能。
2. 首页推荐功能：根据社交分类的每一个分类的第一个动态的组成的首页推荐功能。
3. 地图功能：地图功能集成了附近的人、附近的基础设施、地点搜索、位置导航、多方式导航、天气等功能。
4. 社交功能：社交功能下又细分了很多子分类。主要有摄影，音乐、舞蹈、小哥哥、小姐姐等。用户自行选择感兴趣的分类进行信息的浏览或发布自己的动态。用户还可以评论和点赞其他用户的动态。关注之后可以私信其他用户。
5. 个人信息功能：个人信息展示用户微信头像，微信昵称。用户在此可以管理自己的关注。查看自己与其他用户的会话。

### 后台管理功能描述

1. 用户信息管理功能：使用moogose在moogoDB中建立用户数据集合，保存用户登录唯一id，头像信息、用户昵称、个人简介、性别、关注、粉丝、用户兴趣爱好、对话详情。
2. 用户动态基本信息，接口使用restful接口规范，使用path路径同时规范接口以及作为用户动态分类的唯一标识以便moongose操作动态。数据字段包括动态唯一id，动态内容，接口路径。其中动态内容又包括了发表用户头像、用户昵称、个人简介、动态内容描述、动态图片、点赞数、评论数、评论。
3. 首页信息表，保存着首页轮播图图片地址，推荐动态详情以及用户发表的全部动态详情。

## 非功能性需求

### 性能需求

为了保证项目能够长期、安全、稳定、可靠、高效的运行，小程序应该满足以下的性能需求：

（1）服务器处理的准确性和及时性

服务器处理的准确性和及时性是小程序的必要性能。在小程序设计和开发过程中，要充分考虑系统当前和将来可能承受的工作量，使项目的处理能力和响应时间能够满足用户对信息处理的最基本需求。

（2）微信小程序的开放性和扩充性

微信预约小程序在开发过程中，应该充分考虑以后的可扩充性。因为用户查询的需求也会不断的更新和完善。所有这些，都要求小程序提供足够的手段进行功能的调整和扩充。

（3）小程序的易用性

微信出行小程序是直接面对车主的，而车主往往对操作流程并不是非常熟悉。这就要求小程序能够提供良好的用户接口，易用的人机交互界面。要实现这一点，就要求系统应该尽量使用用户熟悉的术语和中文信息的界面；针对用户可能出现的使用问题，要提供足够的在线帮助，缩短用户对系统熟悉的过程。

### 属性需求

1. 可用性

界面UI设计简洁美观，系统功能便捷易操作，能正确无误的处理系统业务，数据加载速度快，避免用户等待时间过长，整体可使用性较好。

1. 安全性

小程序要保存用户的个人信息，不可轻易被不良商家或个人窃取。

1. 可维护性

使用Github进行源码管理，便于维护与修复。

### 运行环境

1. 软件

操作系统： 能正常运行微信客户端的所有手机操作系统

数据库：mongoDB

服务器：阿里云

开发软件：webstorm、微信小程序开发者工具

开发语言：html、css、JavaScript 、node.js、

客户端平台软件：微信移动客户端

网络平台：通信协议TCP/IP协议，网络要求：要求联网使用

使用方式：微信扫一扫或者关注使用小程序

1. 硬件

手机微信版本至少iOS 6.5.1版本或者Android 6.5.2版本，在微信手机客户端搜索或扫一扫小程序二维码即可直接打开使用，无需安装。无须卸载，真正实现了“轻应用”

# 概要设计说明书

## 3.1 引言

### 3.1.1 编写目的

本概要设计文档主要用来指导“伞友兴趣社交”小程序的详细设计工作，为详细设计提供统一的参照标准，其中包括系统的设计方法、总体设计，内外部接口、系统架构和功能以及其他各种主要问题的解决方案。

在详细设计的过程中，如果发现需要添加新的概要设计标准或者约束指导详细设计工作，必须在此文档进行更新和评审，以确保各模块详细设计的一致性和正确性。

本文档主要描述的是“伞友兴趣社交”微信小程序的概要设计，其中包括定义系统的内外部接口、相关的系统架构和设计标准，不会涉及系统业务逻辑实现的细节。

在对用户进行需求调研的基础上，根据需求分析说明书编写了本概要设计，以作为详细设计及系统实现的依据，本文档主要阅读对象为相关技术人员和项目负责人。

### 3.1.2 背景

1. 项目名称：伞友兴趣社交出行助手
2. 用户:喜爱交友、酷爱社交、或有社交障碍的用户
3. 项目由来：基于当代年轻人普遍患有的社交障碍、以及中国人具有的含蓄性格特点的现状，湖南工商大学计算机与信息工程学院学生王鑫晟提出设想，自发组织，欲设计与实现一个基于微信平台的，为了方便用户的社交，利用腾讯地图API接口来提供针对限行区域及周边特色区域服务的出行助手平台，利用用户个人兴趣分类以及发布动态的可互动性来实现便利用户的社交。该微信小程序功能实用，能为用户带来便捷，清爽的社交体验。

### 3.1.3 定义

1. Windows操作系统

Windows中文是窗户的意思，微软公司推出的视窗电脑操作系统，windows操作系统，即Microsoft Windows，采用的是图形化模式GUI，使用起来更加简单，方便，同时安全性也有一定的保证。

1. MongDB数据库

MongoDB是一个基于分布式文件存储 [1]  的数据库。由[C++](https://baike.baidu.com/item/C%2B%2B" \t "_blank)语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。

MongoDB是一个介于[关系数据库](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93" \t "_blank)和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。它支持的数据结构非常松散，是类似[json](https://baike.baidu.com/item/json" \t "_blank)的[bson](https://baike.baidu.com/item/bson" \t "_blank)格式，因此可以存储比较复杂的数据类型。Mongo最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立[索引](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95" \t "_blank)。

1. 微信小程序

简称小程序，Mini Program，不需要下载安装即可使用的应用，实现了应用“触手可及”的梦想，用户扫一扫或者搜一下即可打开应用。

小程序依托于微信应用程序，有大量的潜在用户，同时小程序开发环境和开发者生态也在逐步发展，开发者可以更高效的开发，同时用户也有很好的使用体验。

## 3.2 总体设计

### 3.2.1 设计前提和约束条件

1. 系统用户已成功登录小程序，通过用户信息基本认证；
2. 有合理的通知功能；
3. 遵循项目开发文档所规定的规范；

### 3.2.2 基本设计思想

中国人在表达自己的情感时是间接的含蓄的。要说男欢女爱，偏道关关雎鸠，要说情断义绝，却道“不及黄泉，无相见也”。不同于外国友人时时将I love you挂在嘴边。这样的性格特点限制了他们社交的圈子大小以及质量。原本志趣相投的两个人经常会因为他们的含蓄与内敛而互相错过。此时需要一个良好的契机来帮助此类用户相识相知。此时“伞友社交小程序”给他们提供了一个良好的平台。一定程度上缓解了此类用户的社交恐惧综合症。此类社交软件可以为他们提供恰到好处的服务来帮助他们拓宽社交，打开视野。

### 3.2.3 系统架构

系统架构如图1所示，系统主要由接入层的微信小程序客户端和应用层的服务提供模块组成，微信小程序客户端为用户提供浏览地图数据, 智能规划路线和规划日期显示, 获取导航和语音的界面, 并且和服务提供模块实时通信。用户获得界面代码来自微信服务器, 而界面上数据则从服务提供模块获得。服务提供模块采用分布式架构, 以应对微信小程序客户端带来的海量服务请求。服务提供模块是整个系统的核心, 包括首页动态推荐、地图数据提供、用户互动、，导航与语音、个性化周边信息提供等多个功能。

该系统体系以计算机硬件与网络通信为依托，以标准和制度为保障，以空间和属性数据为中心，以计算机网络技术、地理信息技术等为支撑构建业务系统，提供服务，系统架构说明如下：

1. 接入层主要包括用户使用的小程序界面以及供小程序界面与服务提供模块交互的网络接口组成。
2. 应用层与程序界面相对应，为小程序界面的各个服务模块提供后台分析支持。
3. 数据资源库以用户信息、动态信息、周边信息数据为核心，包含了地图服务、动态分类、动态详细信息、周边信息等多个周边专题数据库，建立了空间数据和非空间数据相结合的数据库。
4. 基础层包括网络环境、软硬件设施，在系统建设中，需要充分利用现有数据资源、网络资源、软件资源和硬件资源，例如阿里云、腾讯云等云服务器，有效提升工作效率。

图3-1 系统构架示意图



### 3.2.4 项目主要技术

Web前端：“开四停四”出行助手基于微信平台，用户直接进入小程序端即可使用，通过微信小程序框架进行前端动态的展示，地图的展示、其他用户的动态、以及用户个人信息的展示。以及相关服务的UI界面显示和相应的处理，后台管理系统采用node开启一个http server服务端来进行业务逻辑的处理，在前端和服务器之间采用ajax数据交互的方式，数据格式采用json。采用html，css，javascript，前端构建工具：Webpack，gulp，前端安装包工具：NPM；

后端：后端使用的是node，开发工具：webstorm，使用MVC框架，界面层，业务逻辑层，数据访问层的三层架构模式。

数据库：采用主流的关系型数据库mongoDB，数据库连接工具：数据库windows可视化工具mongoDBcompass。

### 3.2.5 平台应用技术路线

技术路线如图3-2所示：

图3-2 技术路线图

技术路线说明：

1. 数据的预处理，用前端表单提供的基础性验证以及正则表达式进行用户注册信息的校验，并通过小程序开发者工具提供的基础ui来提示用户登录结果。并将用户信息写入进数据库中。
2. 资源的获取，在腾讯位置服务中申请成为开发者，拿到开发者key，并在微信公众平台申请使用腾讯地图API。
3. 算法的设计与实现，算法的设计与实现本质上可以归纳为数据库的增删改查。用户登入小程序，首先就是将用户信息写入数据库。用户点赞与评论其他用户也是在数据库中动态的点赞数和评论数进行刷新。用户发布动态，前端提供表单给用户填写，用户需要填写动态标题，动态内容，动态显示图片和动态内容图片。前端整合表单数据和用户信息发给后台，后台将这些数据根据用户选择的分类写入数据库。地图方面则是直接调用的腾讯地图api来进行地图选点，地图标识，附近的人，附近娱乐场所。市场上现有的地图api在此方面以及足够完善，经过引包后能在项目中发挥很高的效用。
4. 数据交互，根据用户登录所提供的信息，自动获取用户周围娱乐设施。即附近咖啡馆、电影院、麦当劳等信息，并将用户当前地理位置信息通过视窗展示给用户。通过后端和网络api提供数据接口地图信息，通过前端显示给用户。

### 3.2.6 平台建设目标

本项目研究年轻人的社交障碍以及国人性格含蓄的特点为背景，依托微信小程序优势，利用腾讯地图API、web等技术搭建一个针对年轻用户的基于兴趣分类的社交平台，实现用户的快速、轻量社交，缓解年轻人的社交障碍，同时也促进了国家GDP增长，为用户提供特色化，个性化的社交新体验。

### 3.2.7 性能要求

1. 处理能力

系统面对多用户任务并发性，应保证正常的处理运行能力。

1. 稳定性

系统资源利用合理，保证系统前后台数据操作效率。

1. 传输能力

本系统业务能力包括地图、日历的展示，应提高文件传输速度。

1. 管理能力

有序的对信息内容和模块进行合理的管理。

## 3.3 功能设计

### 3.3.1 功能总体设计

本兴趣社交平台共分为四个子系统，首页推荐子系统，交友地图子系统，兴趣社区子系统和个人信息子系统。

首页推荐子系统主要由一个轮播图和一个动态推荐构成。动态推荐取的是兴趣社交子系统每一个分类的第一条用户动态组成的数组。将其渲染到首页。形成首页推荐。

地图寻友子系统主要采用的是腾讯位置服务提供的网络接口以及微信小程序提供的map地图标签，先行市场上三大位置服务提供商（腾旭、百度、高德）提供的网络位置服务已经相当完善，直接利用其提供的网络服务api足以实现该项目的全部需求。如附近的人，附近的麦当劳、咖啡馆、电影院。当前位置详情，天气，地图选点、地图多方式导航（包括步行、公交、驾车）等功能。

兴趣社区子系统。此子系统根据兴趣分类。分为设计、摄影、小哥哥、小姐姐、音乐、舞蹈分类。每个分类中是用户发布的真实动态，其他用户可以点赞，评论，关注用户。在此子系统下还提供有动态发布功能。用户需填写动态标题、动态简介、动态展示图（使用正则表达式判断是否为url地址，如果不是则会抛出提示框），内容正文、选择动态分类，前端整合用户id发给后台、后台写入数据库。发表成功后用户便可以在对应的分类中看到自己发布的动态、其他用户看到动态可以进行点赞、评论、关注、私信等操作。

个人信息子系统通过微信登陆获取用户头像和昵称显示在前端，同时可以看自己关注的人和消息记录。点击其他用户头像可以发起聊天。聊天页面实现了即时通信。用户发送的消息会及时写入数据库并且前端会及时渲染。

### 3.3.2 系统功能模块图

图3-3 系统功能模块图



### 3.3.3 系统业务流程图

图3-4 系统业务流程图



## 3.4 接口设计

### 3.4.1 用户接口

（一）搜索位置：

1.小程序列表中搜索栏：通过关键词搜索，会直接展示搜索关键词下的所有小程序

2.顶部搜索页：搜索界面会直接展示常用小程序

3.顶部搜索结果页：搜索结果会展示小程序

4.添加好友搜索框的搜索结果页：搜索结果会展示小程序

（二）小程序二维码

通过二维码扫描，可直接进入小程序

### 3.4.2 外部接口

外部接口主要涉及腾讯地图API，天气API，自定义后台接口通过相关的API发送请求，来获取地图以及实时天气情况，再通过小程序展示出来。

腾讯位置服务api：腾讯旗下的用于处理位置服务的api。

天气接口：聚合数据网站提供的网络免费数据api

自定义后台接口。使用node.js写的后台代码。放在阿里云轻量应用服务器中部署好node环境。运行代码。再配置好nginx反向代理将接口部署到https域名上。小程序添加域名请求白名单即可实现正常数据通信。

### 3.4.3 内部接口

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传递，接口传递的信息将是以数据结构封装的数据，以参数传递或者返回值的形式在各模块间传输。

## 3.5 运行设计

### 3.5.1 运行模块组合

在用户进入小程序内部后，启动用户登陆模块，进行实名认证，以及个人信息的填写，读入数据并对输入进行验证处理。

接收数据模块得到充分的数据时，将调用网络传输模块，将数据通过网络送到服务器，并等待接收服务器返回的信息；通过地图模块发送请求到服务器，来进行用户实时地图位置的更新，服务器返回用户位置信息，通过位置信息等数据，向服务器请求实时天气情况，返回相应信息，接收到返回信息后随即调用数据输出模块，对信息进行处理，产生相应输出。

### 3.5.2 运行控制

运行控制按照函数调用关系来实现，在事物中心模块，对运行控制进行正确的判断，选择正确的运行控制路径。当用户想获取位置、天气相关信息时，发送相应的请求，获取对应的信息返回展示。

在网络传输方面，客户端发送请求后，将等待服务器的确认收到反馈，收到后，再次等待服务器发送应答数据，对数据进行确认。服务器在接收到数据后发送确认信息，在对数据处理、访问数据库后，将返回信息送回客户端，并等待确认。

## 3.6 系统出错处理设计

### 3.6.1 出错信息

|  |  |  |
| --- | --- | --- |
| 错误类型 | 原因 | 解决方法 |
| 数据库连接错误 | 数据库设置不正确或SQL Server异常 | 取消本次操作，提醒用户检查数据库 |
| 输入错误 | 输入不规范 | 通过前端正则表达式等对用户输入进行规范 |
| 不可预知错误 | 未知异常 | 进行数据库备份，帮组开发者完善程序 |

### 3.6.2 补救措施

对本系统的几种可能的错误进行了分析，分别进行了不同的处理。

主要的错误可能由：

1.数据库连接错误:这类错误主要是数据库设置不正确，或mysql异常引起的，我们只要取消本次操作，提醒检查数据库问题即可。

2.输入错误：这主要是用户输入不规范造成的，我们尽量在前端使用一些正则表达式来规范用户的输入，当用户输入不符合规范时，直接提醒用户重新输入，而不用提交到服务器进行处理。

3.其他操作错误：对于用户的不正当操作，有可能使程序发生错误，我们主要是中止操作，并提醒用户中止原因和操作的规范。

4.其他不可预知的错误：程序也会有一些我们无法预知或没考虑完全的错误，我们对此不可能作出安全的异常处理，这时我们主要要保证数据的安全，所以要经常进行数据库备份，并能及时的和我们联系，以逐步完善我们的程序。

## 3.7 系统维护设计

软件的维护主要包括数据库的维护和子系统服务器的维护。对于数据库的维护，需要提供数据库的备份和恢复功能，方便的实现数据库的维护和管理。对于子系统服务器的维护，由于每个模块之间的独立性较高，对于服务器的维护带来了很大的方便。对于功能的添加，只需要再添加菜单项内容即可，我们将根据客户的要求和反应，定期对软件进行维护和改进。

# 4. 详细设计说明

## 4.1 设计概述

编写详细设计说明书时软件开发过程中必不可少的部分，其目的是为了使开发人员在完成概要设计说明书的基础上完成概要设计规定的各项模块的具体实现的设计工作。同时也是开发人员和最终客户进行需求交流的有效手段。

## 4.2 总体结构

### 4.2.1 数据视图

图4-1 数据视图



### 4.2.2 用例图

（1）个人信息模块

图4-2 车辆管理模块用例图



功能说明：用户可通过个人信息模块查看个人昵称、头像、个人关注、个人私信等功能

参与者：用户

（2）地图寻友模块

图4-3 日期规划模块用例图



功能说明：查看附近用户、附近男生、附近女生、麦当劳、电影院、天气、地图选点、位置导航。

参与者：用户

1. 兴趣社区模块

图4-4 路线规划模块用例图



功能说明：用户在此可以发布自己的动态、查看他人的动态。点赞，评论、私信等。

参与者：用户

## 4.3 模块设计说明

### 4.3.1 用户免登录模块

（1）模块描述

该模块主要实现用户的登录功能，用户首次进入小程序判断用户是否允许登录，用户成功登录后，即可使用小程序。

（2）功能

* 用户允许登录后，小程序访问用户信息，将信息传给服务器，服务器解析获取用户唯一标识openID。
* 不需要用户进行注册登录，只需要用户使用微信访问即可。

（3）输入项

用户信息 object

（4）设计方法（算法）

用户初始进入小程序，小程序调用方法访问用户是否允许登录

* 用户允许小程序登录，小程序将访问到的用户信息传给服务器；
* 服务器解析信息，得到用户唯一标识ID；

### 4.3.2 个人信息模块

（1）模块描述

该模块主要实现用户对个人信息的管理功能，用户输入正确的个人信息后，即可登陆小程序，可对已注册的个人信息进行修改、删除操作。

（2）功能

* 用户名默认为微信名，该模块将数据发送请求到服务器，服务器调用业务层判断用户输入数据的合法性及类型，并给出相应的提示信息。
* 用户可对注册的个人信息进行修改、删除操作，数据库中的数据也随之改变。

1. 输入data项

个人登陆唯一ID号 string

用户名 string

密码 string

性别 female/male

1. 设计方法（算法）

用户输入车牌号后，点击绑定：

* 小程序先判断用户提交的数据是否为空，若为空则提醒用户；
* 使用小程序自带api，wx.login获取用户唯一ID数据传给服务器做处理；
* 先判断手机号格式是否正确，正确则将数据保存到数据库中，其他情况将会给出相应的处理和提示。

### 4.3.3地图社交模块

（1）模块描述

地图功能集成了附近的人、附近的基础设施、地点搜索、位置导航、多方式导航、天气、用户地理位置信息展示等功能。

（2）功能

* 用户在地图上选择伞友的位置，发送请求并将数据传到腾讯地图接口；
* 由腾讯地图提供多种到达伞友位置的路线。

（3）输入项

自身位置 string

伞友位置 string

1. 设计方法（算法）

* 将用户在地图上选择的位置传到腾讯地图接口做处理；
* 将腾讯地图提供的路线结果渲染到页面上显示。

### 4.3.4兴趣社区模块

（1）模块描述

此子系统根据兴趣分类。分为设计、摄影、小哥哥、小姐姐、音乐、舞蹈分类。每个分类中是用户发布的真实动态，其他用户可以点赞，评论，关注用户。在此子系统下还提供有动态发布功能。用户需填写动态标题、动态简介、动态展示图（使用正则表达式判断是否为url地址，如果不是则会抛出提示框），内容正文、选择动态分类，前端整合用户id发给后台、后台写入数据库。发表成功后用户便可以在对应的分类中看到自己发布的动态、其他用户看到动态可以进行点赞、评论、关注、私信等操作

（2）功能

* 用户可发布自己的动态；
* 用户可查看其他人的动态；
* 用户可通过点赞、评论、私信、来实现与其他用户的互动。

# 5. 数据库设计说明

## 5.1 设计概述

数据库设计是指对于一个给定的应用环境，提供一个确定最有数据模型与处理模式的逻辑设计，以及一个确定数据库存储结构与存取方法的物理设计，建立起既能反映显现实世界信息和信息联系，满足用户数据要求和加工要求，又能被某个数据库管理系统所接受，同时，能实现系统目标，并且有效存取数据的数据库。

本说明书将从数据库设计、结构情况方面对易出行项目进行综合概述，使用数据库对象关系图来描述项目所涉及到的数据库对象和对象之间的关系，并对具体对象进行详细的规格描述。

软件开发人员可依照此说明书的数据库结构设计进行软件程序开发，实施部署人员可依照此说明书进行数据库的部署实施。同时也为后续的数据库维护工作提供了良好的使用说明，也可以作为未来版本升级时的重要参考资料。

## 5.2 外部设计

### 5.2.1 标识符和状态

数据库库名：um

用户名：wxs

密码：wang.123

数据库集合：group、home、users

权限：全部

### 5.2.2 使用它的程序

“伞友社交”小程序

### 5.2.3 约定

命名富有意义英文词汇或拼音，要避免使用缩写。

除数据库名称长度为1-8个字符，其余为1-30个字符不等。字符集采用utf-8，请注意字符的转换。

命名只能使用英文字母、数字和下划线。

各表之间相关列名尽量同名，以便记忆。

除特别说明外，所有日期格式都采用datetime格式。

除特别说明外，所有整型都采用int格式。

## 5.3 结构设计

### 5.3.1 概念结构设计

本项目主要由车辆信息与行驶记录表两个实体，车辆信息的属性包括车主名和车牌号，行驶记录表的属性包括车牌号、行驶天数、本次行驶日期、行驶是否合法。车辆行驶产生行驶记录，车辆信息与行驶记录表的联系类型是1:n，即一辆车可产生多条行驶记录。

车辆信息与行驶记录表的E-R图如下图所示：

图5-1 车辆信息与行驶记录表的E-R图



### 5.3.2 逻辑结构设计

1. 表名：uesrs

描述：用户信息表

表4-1 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数值类型 | 非空 | 默认值 | 具体描述 |
| open\_id(PK) | varchar | Y | 无 | 用户唯一id |
| name | varchar | Y | 无 | 用户昵称，即微信用户名 |
| password | varchar | Y | 无 | 用户密码 |
| sex | varchar | Y | 无 | 用户性别 |

1. 表名：group

描述：动态类别表

表4-2 动态类别表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数值类型 | 非空 | 默认值 | 具体描述 |
| id(PK) | int | Y | 无 |  |
| cate | varchar | Y | 无 | 类别名称 |

1. 表名：home

描述：动态信息表

表4-3 动态信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数值类型 | 非空 | 默认值 | 具体描述 |
| id(PK) | int | Y | 无 |  |
| name | varchar | Y | 无 | 用户昵称 |
| good\_num | int |  | 0 | 点赞数 |
| comment\_num | int |  | 0 | 评论数 |
| cate | varchar | Y |  | 外键 |

1. 表名：relate

描述：用户关系表

表4-4 用户关系表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数值类型 | 非空 | 默认值 | 具体描述 |
| openid(PK) | varchar | Y | 无 | 用户openid |
| star | varchar |  | 无 | 关注的用户openid |
| follow | varchar |  | 无 | 粉丝openid |

# 6. 用户界面设计说明

## 6.1 设计概述

本文档对系统界面的布局、设计风格、交互部分进行描述，包括界面原型的展示。本文档的编制是为了让用户和软件开发者双方对该开发软件的UI（User Interface）初始规定有一个共同的理解，定义索要开发的“易出行”的用户界面布局、风格等设计目标。是整个项目中软件产品开发设计与实现的重要根据，也是软件产品的测试与验收的依据。

## 6.2 界面设计规范

### 6.2.1 用户界面设计原则

本系统坚持图形用户界面（GUI）设计相关原则，界面直观、对用户透明；用户接触软件后对界面上的功能一目了然，不需要多少培训就可以方便使用本应用系统。

强调“以用户为中心的设计”原则，界面设计应该明确软件中用户是所有处理的核心，不应该由应用程序来决定处理过程，所有用户自己的意愿把操作流程强加给用户。

### 6.2.2 界面一致性

在本系统的界面设计中应该保持界面的一致性。一致性既包括各子界面使用标准的控件以及一致的框架结构，也指使用相同的信息表现方式。

### 6.2.3 向导使用原则

对于应用中某些部分的处理流程是固定的，用户必须按照指定的顺序输入操作信息，为了使用户操作得到必要的引用应该使用向导，使用户使用功能时比较轻松明了，但是向导必须用在固定处理流程中，并且处理流程应该不少于3个处理步骤。

### 6.2.4 出错信息和警告

出错信息和警告是指出现问题时系统给出的坏消息，本系统对于出错信息和警告应该遵循以下原则：

1. 信息以用户可以理解的术语描述；
2. 信息应提供如何从错误中恢复的建设性意见；
3. 信息应指出错误可能导致哪些不良后果，以便用户检查是否出现了这些情况或帮助用户进行改正；
4. 信息应伴随着视觉上的提示，如特殊的图像、颜色或信息闪烁；
5. 信息不能带有判断色彩，即任何情况下不能指责用户；

### 6.2.5 数据输入原则

本系统数据输入遵循以下原则：

1. 尽量减少用户输入动作的数量；
2. 维护信息显示和数据输入的一致性；
3. 交互时应该灵活，对键盘和鼠标输入的灵活性提供支持；
4. 让用户控制交互流，用户可以跳过不必要的动作、改变所需动作的顺序（如果允许的话）以及在不退出系统的情况下从错误状态中恢复；

## 6.3 用户界面设计

### 6.3.1 首页推荐页面界面



图6-1 首页推荐界面图

表6-1 首页导航关系表

|  |  |  |
| --- | --- | --- |
| 元素名称 | 导航 | 备注 |
| 首页 | pages/home/home | 首页推荐 |
| 地图 | pages/index/index | 地图寻友页面、附近设施、选择起点和终点，查询路线 |
| 兴趣社区 | pages/group / group | 兴趣社区页面，发布动态，用户互动 |
| 个人信息 | pages/my/ my | 个人信息展示、关注详情、私信功能 |

### 6.3.2 地图寻友界面



图6-2 地图寻友界面图

表6-2 地图寻友界面页面跳转关系表

|  |  |  |
| --- | --- | --- |
| 元素名称 | 跳转页面 | 备注 |
| 搜索框 | 地点搜索页面 | 无 |
| 页面图标 | 跳转到相应功能 | 无 |
| 地图导航 | 导航页面 | 无 |

### 6.3.3 兴趣社区界面

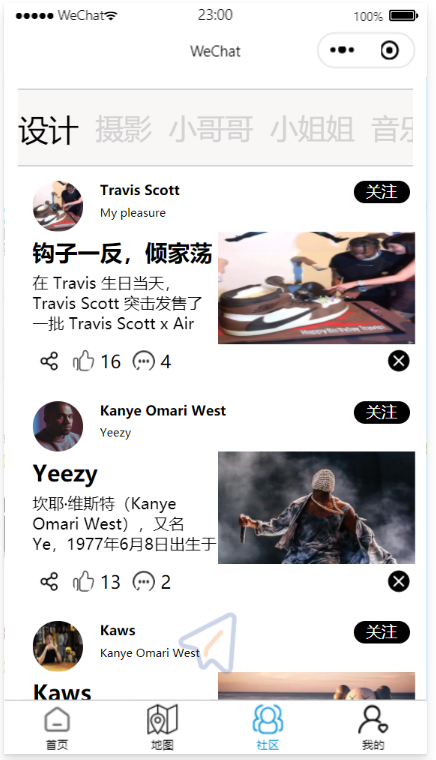


表6-4 兴趣社区界面页面跳转关系表

|  |  |  |
| --- | --- | --- |
| 元素名称 | 跳转页面 | 备注 |
| 发布动态 | 发布动态表单页面 |  |
| 动态图 | 动态详情 |  |

### 6.3.4 个人信息界面



图6-6 个人信息界面图

表6-7 个人信息界面页面跳转关系表

|  |  |  |
| --- | --- | --- |
| 元素名称 | 跳转页面 | 备注 |
| 我的关注 | 用户关注页面 | 无 |
| 我的消息 | 用户私信页面 |  |

### 6.3.5 附近娱乐设施界面

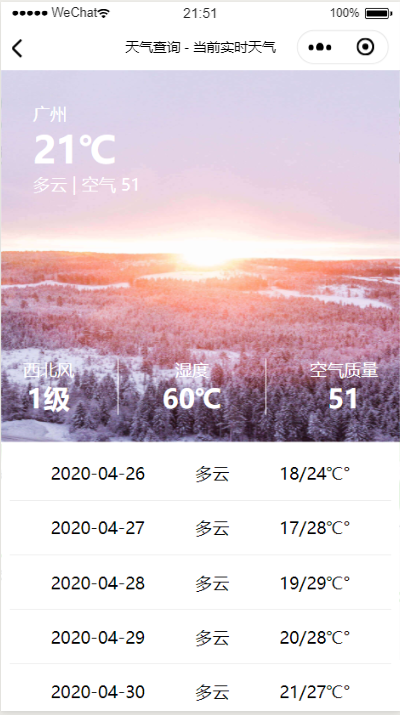


图6-9 附近娱乐设施及天气

表6-10附近娱乐设施界面页面跳转关系表

|  |  |  |
| --- | --- | --- |
| 元素名称 | 跳转页面 | 备注 |
| 天气查询 | 天气详情页面 | 自动获取用户所在地 |
| 出行路线 | 多种出行方案详情页面 |  |
| 选择地点 | 地图选点页面 |  |

表6-11 路线规划界面页面元素表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 元素名称 | 功能 | 元素描述 | 必须/可选 | 数据校验 |
| 目的地 | 记录目的地 | 文本框 | 必须 | 不能为空 |
| 出发地 | 记录出发地 | 文本框 | 可选 | 不能为空（默认为自身位置） |
| 麦当劳图标 | 查询附近麦当劳 | 图标 | 可选 | 无 |
| 电影院图标 | 查询附近电影院 | 图标 | 可选 | 无 |
| 饮料图标 | 查询附近饮料店 | 图标 | 可选 | 无 |
| 天气图标 | 查询用户所在地的天气 | 图标 | 可选 | 无 |

表6-12 路线规划界面功能元素表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 功能描述 | 校验 | 异常 |
| 查询出行路线 | 根据用户的起点和终点提供出行方案 | 起点和终点不能为空且不可相同 | 当起点和终点为空或相同时，不可查询路线 |

# 7. 用户操作手册

## 7.1 概述

本手册的目的在于告诉“伞友社交”小程序的使用者，本系统软件提供了哪些功能，以及如何正确、有效地来使用这些功能。

## 7.3 软件概述

### 7.3.1 功能

软件的基本功能如下：

1. 用户授权登录：该模块主要实现用户的登录功能，用户首次访问小程序时，用户选择是否授权登录小程序，用户选择登录后即可使用小程序。
2. 查看用户动态：用户成功登陆后，在首页和社区中都能查看到其他用户的信息。
3. 发布动态功能：该模块主要实现用户动态发布的规划功能，用户按要求填写好全部表单、后台写入数据库、前端进行渲染。
4. 互动功能：用户可以对其他用户发布的动态进行互动操作。包括点赞、评论、私信。

### 7.3.2 性能

时间特性：本系统响应时间取决于数据库的长度和硬件性能指标；

灵活性：本系统可以在支撑汉字的操作系统下运行；

开放性和扩充性 ：能针对用户需求的变化以后有较好的可扩充性，对用户需求不断更新和完善；

易用性：能够提供良好的用户接口，易用的人机交互界面。

## 7.4 运行环境

（1）软件

操作系统： Android/IOS

数据库：node.js

服务器：阿里云轻量应用服务器

开发软件：webstorm、微信小程序开发者工具

开发语言：html、css、Java、JavaScript、node.js

客户端平台软件：微信

网络要求：要求网络

使用方式：微信扫一扫或者关注使用小程序

（2）硬件

手机微信版本至少iOS 6.5.3版本或者Android 6.5.3版本，在微信手机客户端搜索即可直接打开使用，无需安装。

## 7.5 使用说明

### 7.5.1 打开方式

在微信客户端首页的搜索窗口，通过搜索小程序名字获取一个小程序，然后在微信的“发现”栏中就可以看到“小程序”这个栏目了，可在栏目内搜索本小程序，也可以朋友分享打开，或者扫描二维码。

### 7.5.2 操作方法

点击进入“伞友社交”小程序，分别分为首页、地图、社区、我的四个页面。

1. 个人管理功能：用户授权登录小程序后，可在我的页面查看我的关注以及与其他用户的聊天记录。可对已关注的人取消关注。
2. 首页动态推荐功能：用户在首页可查看每个动态分类中最新的一条动态信息，方便知晓最近的动态更新情况。
3. 发布动态功能：用户在登录后，可以在社区页面浏览不同分类的动态内容，亦可点赞评论，点击页面下方纸飞机图标即可发布动态，输入标题、内容、图片等必填项，并选择动态类别，点击提交即可发布内容。
4. 寻找伞友功能：用户在地图页面，点击男女图标选择男生或女生，地图上将会显示对应的伞友位置，选择某个伞友位置，在地图下方会显示伞友具体位置，点击路线图标，将会显示到达伞友位置的不同的路线方案，包括公交、步行、骑行、驾车不同的交通方式。
5. 周边娱乐设施功能：用户在地图页面可通过点击页面右侧的娱乐设施图标，包括电影院、麦当劳、茶饮点、天气，点击相应的图标，地图上将会显示附近对应的娱乐设施点的位置。点击天气图标，将会显示用户所处地的实时天气。

# 8. 测试计划

## 8.1 引言

本文档主要阐述”易出行“测试过程中的一些细节，为”易出行“的测试工作提供一个框架和规范：

## 8.2 测试目标

（1）计划的测试用例已全部执行并达到设计要求。

（2）经确定的所有缺陷都已得到了商定的解决结果，并没有发现新的缺陷。

## 8.3 测试种类及测试标准

### 8.3.1 功能测试

表8-1 功能测试类型描述

|  |  |
| --- | --- |
| 测试目标 | 验证软件提供的功能是否都可以实现 |
| 测试方法和技术 | 检验在输入正确数据时结果能否与设计期望相符合；  检验在输入错误数据时软件能否报错并正常运行。 |
| 完成标准 | 所有功能都经过测试，且达到目标。 |

### 8.3.2 业务测试

表8-2 业务测试类型描述

|  |  |
| --- | --- |
| 测试目标 | 与该项目的业务流程一致 |
| 测试方法和技术 | 测试业务流程的各个方向 |
| 完成标准 | 数据能正确从软件的一个模块流入到另一个模块中 |

### 8.3.3 容量测试

表8-3 容量测试类型描述

|  |  |
| --- | --- |
| 测试目标 | 增大软件输入数据量，以确认软件在处理大量数据时运行正常 |
| 测试方法和技术 | 对设计时预计的平均数据处理量提供一个数量级 |
| 完成标准 | 大数据输入时软件可以正常完成功能 |

## 8.4 测试重点

### 8.4.1 功能测试

（1）绑定车辆

* 是否能成功进入绑定车辆页面
* 是否能在输入框中输入数据
* 是否能成功发送请求和接收数据
* 页面是否能成功渲染
* 数据传递是否正确
* 输入框为空时提交数据是否会有提示
* 是否能正确验证车牌号
* 对于不同类型的车牌号是否由不同的提示
* 绑定车辆成功后是否能够跳到我的页面

（2）限行区域查看

* 页面是否能成功渲染
* 限行区域范围是否和”开四停四“政策所设定的范围一致

1. 日期规划

* 能否能选择出行日期
* 显示的可行驶日期是否符合”开四停四“政策的规定
* 是否能成功发送请求和接收数据
* 页面是否能成功渲染
* 数据传递是否正确

1. 出行路线

* 是否能正确获取当前位置
* 是否能成功选择终点位置
* 是否能成功发送请求和接收数据
* 页面是否能成功渲染
* 数据传递是否正确
* 所给出的路线的起点和终点是否与选择的终点一致

1. 天气页面

* 城市是否定位准确
* 天气的数据是否正确
* 是否能成功发送请求和接收数据
* 页面是否能成功渲染
* 数据传递是否正确

1. 周边服务

* 点击图标，地图上是否有显示
* 是否能正确获取周边服务点的位置
* 是否能成功发送请求和接收数据
* 页面是否能成功渲染
* 数据传递是否正确

### 8.4.2 业务测试

* 日历、路线、我的页面之间相互转换
* 绑定车辆成功后是否能跳转到我的页面
* 车牌号有误是否能正确提醒
* 点击天气图标是否能跳转到天气详情页面
* 地图正确且合理地放大与缩小
* 点击绑定车辆是否能跳转到绑定车辆页面并正确渲染