**总体设计报告**

目录

[1、引言 2](#_Toc151320665)

[1.1编写目的 2](#_Toc151320666)

[1.2背景 2](#_Toc151320667)

[1.3参考资料 2](#_Toc151320668)

[2、总体设计 2](#_Toc151320669)

[2.1需求规定 2](#_Toc151320670)

[2.2运行环境 3](#_Toc151320671)

[2.2.1硬件环境 3](#_Toc151320672)

[2.2.2软件环境 3](#_Toc151320673)

[2.3基本设计概念和处理流程 3](#_Toc151320674)

[2.4结构 5](#_Toc151320675)

[2.4.1服务端结构 5](#_Toc151320676)

[2.4.2客户端结构 6](#_Toc151320677)

[2.5人工处理过程 6](#_Toc151320678)

[3、接口设计 6](#_Toc151320679)

[3.1用户接口 6](#_Toc151320680)

[3.1.1用户界面描述 6](#_Toc151320681)

[3.1.2用户操作流程 6](#_Toc151320682)

[3.2外部接口 7](#_Toc151320683)

[3.2.1与django后端的接口 7](#_Toc151320684)

[3.3内部接口 7](#_Toc151320685)

[3.3.1小程序前端和后端通信 7](#_Toc151320686)

[4、运行设计 8](#_Toc151320687)

[4.1运行模块组合 8](#_Toc151320688)

[4.1.1小程序前端模块组合 8](#_Toc151320689)

[4.1.2Django后端模块组合 8](#_Toc151320690)

[4.2运行控制 8](#_Toc151320691)

[4.3运行时间 9](#_Toc151320692)

[5、系统数据结构设计 9](#_Toc151320693)

[5.1E-R实体图 9](#_Toc151320694)

[5.2 数据结构表 10](#_Toc151320695)

[6、系统出错处理设计 11](#_Toc151320696)

[6.1出错信息 11](#_Toc151320697)

[6.2补救措施 12](#_Toc151320698)

[7、小组成员分工 12](#_Toc151320699)

[7.1成员分工 12](#_Toc151320700)

[7.2成员评分 12](#_Toc151320701)

# 1、引言

## 1.1编写目的

该总体设计说明书的目的在于根据“爱游”小程序需求说明书提出该系统的概要设计，即系统的大概轮廓，主要包括处理流程，结构，接口设计和运行设计及系统数据结构设计。预期读者为“爱游”设计小组的做详细设计部分的同学和老师。

## 1.2背景

1.小程序系统名称:“爱游”小程序业务

2.项目来源于《软件工程》课，G14组基于目前人们约伴旅游困难以及攻略制作耗时的痛点，开发的社交旅游平台。

## 1.3参考资料

1.《软件设计文档国家标准GB8567-88》

2.已上交的《项目可行性分析报告》与《需求分析说明书》

3.《软件工程导论（第6版）》

# 2、总体设计

## 2.1需求规定

1.搜索：通过标签进行模糊搜索显示地区特色旅游路线攻略。

2.发布攒局：根据已有的旅游攻略发起攒局。

3.加入攒局：可选择是否加入已发起的攒局。

4.攒局聊天功能：可在攒局内进行群体聊天。

5.发布攻略：用户、管理员可以发布攻略。

6.好友列表：可以添加好友进行聊天。

7.个人中心：查看基本资料和功能列表，可以修改资料。

8.攻略评论：用户、管理员可以发布攻略评论。

## 2.2运行环境

### 2.2.1硬件环境

用户端设备：

用于访问微信小程序的用户移动设备，如智能手机或平板电脑。

开发和测试设备：

用于开发和测试微信小程序的开发者设备，通常是个人计算机。

服务器：

如果你的后端部署在服务器上，你需要相应的服务器硬件。云服务提供商如AWS、阿里云等可以提供虚拟机或容器服务，用于托管后端应用。

### 2.2.2软件环境

前端软件环境：

微信小程序开发工具：用于开发、调试和发布小程序。

代码编辑器：用于编辑小程序前端代码，例如 Visual Studio Code。

后端软件环境：

Python：Django 后端框架的运行环境。

Django：后端应用框架，提供了处理请求、路由、数据库等功能。

MySQL：数据库管理系统，用于存储和检索数据。

数据库管理工具：

用于管理 MySQL 数据库的工具，例如 MySQL Workbench。

## 2.3基本设计概念和处理流程

（1）“爱游”的顶层数据流图

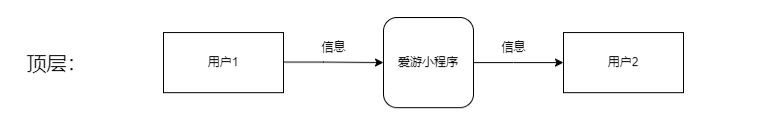


          图 1 “爱游”顶层数据流图

描述：如图1所示，用户1可以通过爱游小程序的平台，将信息转递给用户2。

（2）“爱游”的0层数据流图

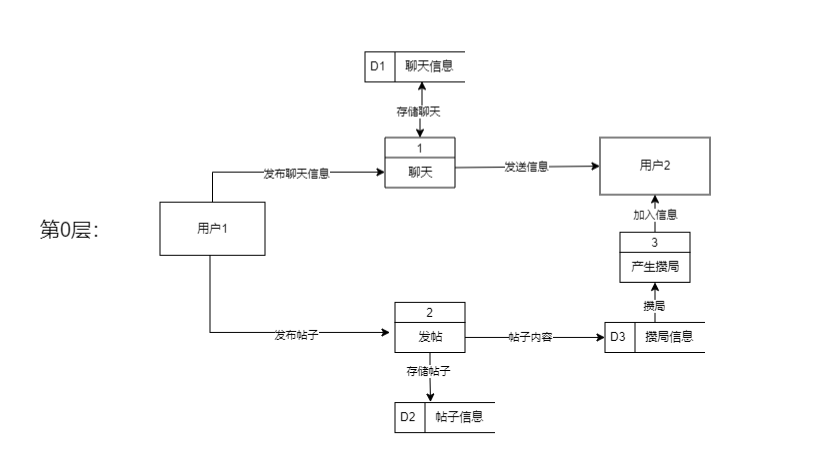


          图 2 “爱游”0层数据流图

描述：

1. 用户可以扮演两种角色—攒局发起者、攒局者。当用户扮演攒局发起者的角色时，可以向爱游系统发送事务，同时攒局发起者可以浏览爱游系统返回的事务，即攒局发起者与爱游系统有双向的数据流动。
2. 当用户扮演攒局者角色时，可以向爱游系统发送事务，同时攒局者可以浏览爱游系统返回的事务，即攒局者与爱游系统也有双向的数据流动。

（3）“爱游”的1层数据流图

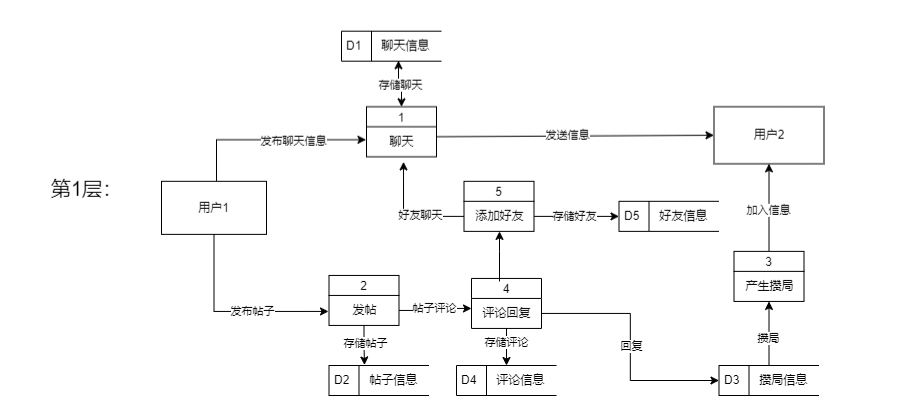


图 3 “爱游”1层数据流图

描述：

1.用户通过提交身份信息向用户登录事务发送请求。

2.用户登录事务从用户信息库中读取相应的用户信息进行匹配判断登录结果。

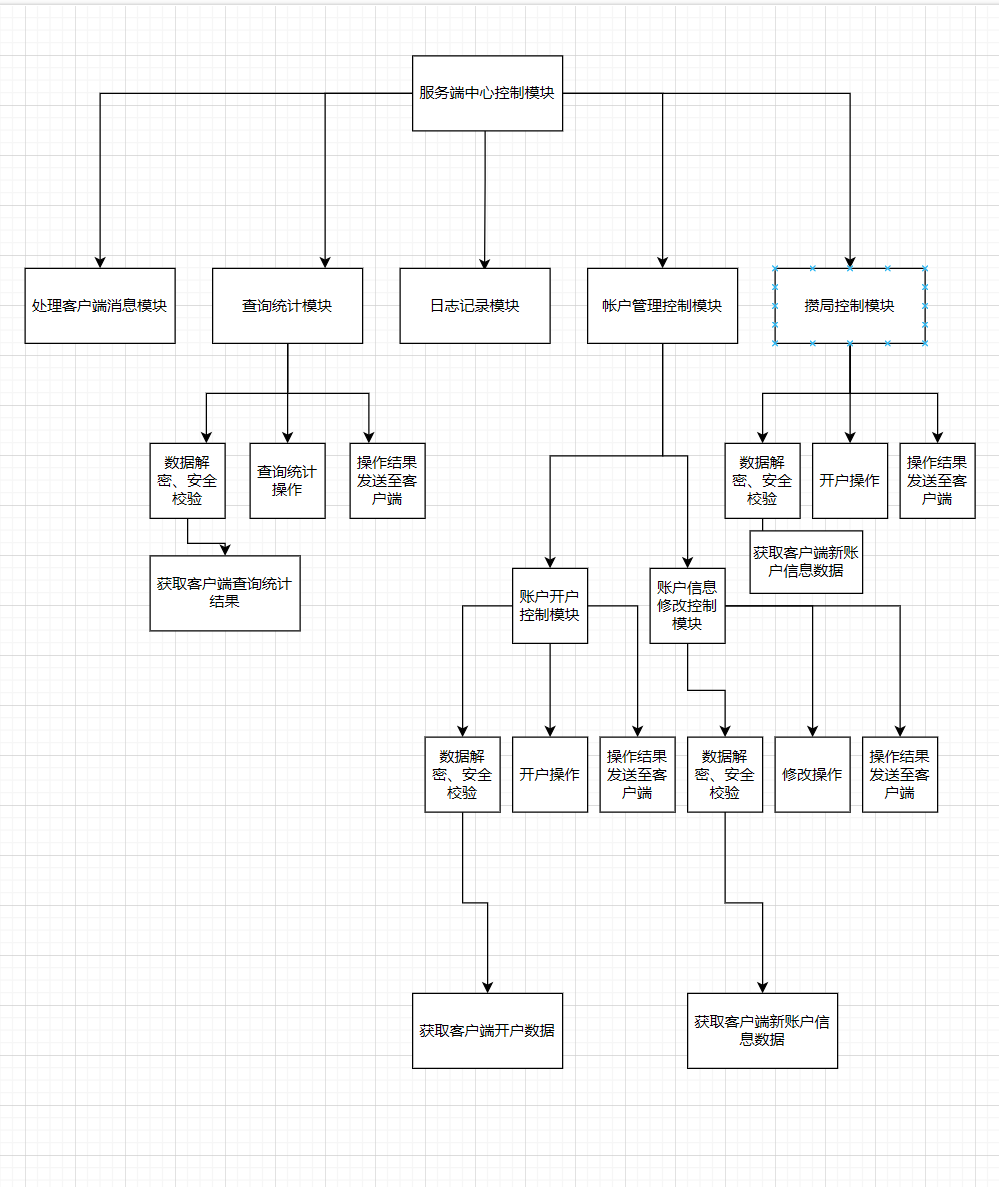
3.用户登录成功后，攻略推送事务将要根据用户信息推送攻略，用户可以进行个人信息管理、搜索攻略等操作。

4.用户进行搜索攻略操作时,用户提供的搜索信息流动到搜索攻略事务，搜索攻略事务对搜索信息进行相应的处理后得到攻略信息。

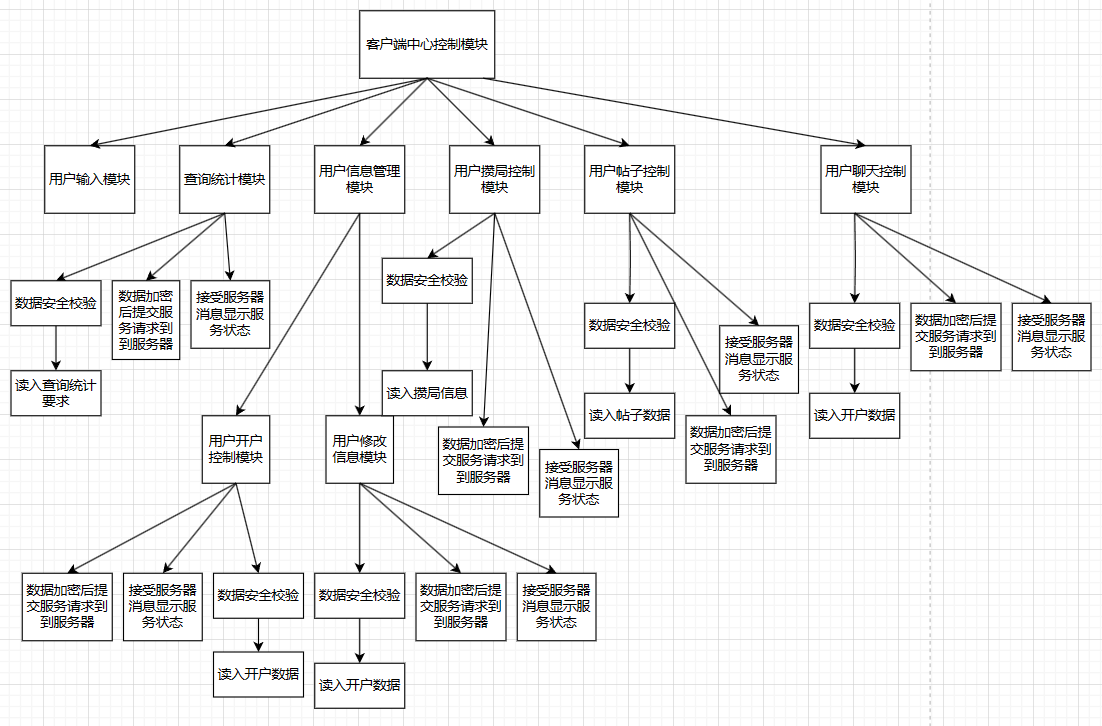
5.系统对用户进行攻略推送时，用户信息流动到攻略推送事务，攻略事务对用户信息进行处理得到请求信息。

## 2.4结构

### 2.4.1服务端结构



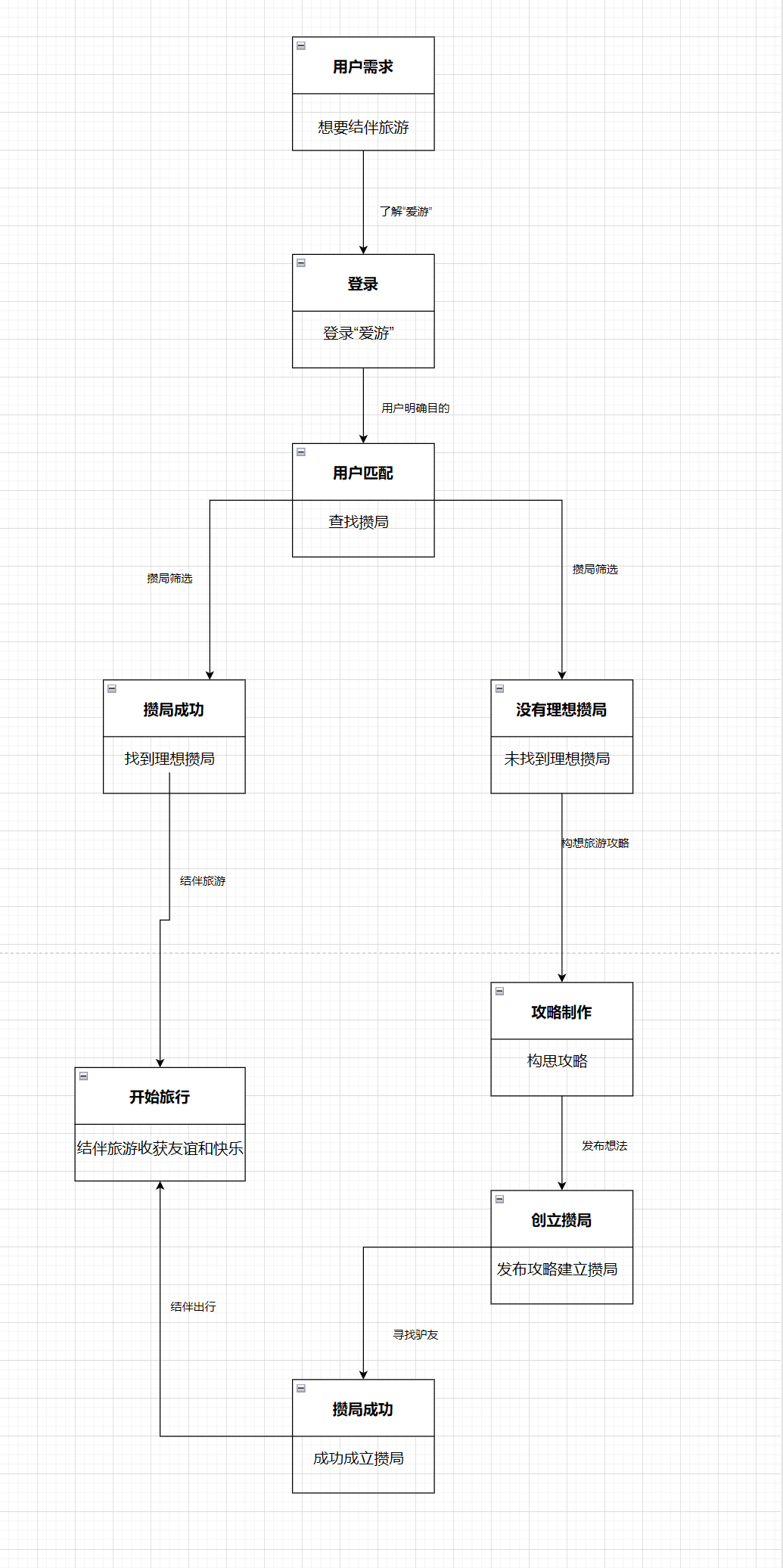
### 2.4.2客户端结构



## 2.5人工处理过程

对用户发布的一系列帖子以及聊天记录进行监管，防止出现违规信息。

## 2.6业务流程图



# 3、接口设计

## 3.1用户接口

### 3.1.1用户界面描述

在微信小程序中，用户界面主要通过 WXML（WeiXin Markup Language）和 WXSS（WeiXin Style Sheets）进行设计。小程序主页包含攒局帖子，每个攒局以卡片形式展示。攒局详情页显示攒局的详细信息，包括标题、评论、点赞、描述、参与人数等。聊天界面采用聊天气泡形式，支持文字和图片消息展示。个人信息界面显示用户头像、昵称，提供编辑功能。

### 3.1.2用户操作流程

1. 用户进入小程序，首次登录时进行微信授权，获取用户信息。

2. 在攒局界面，用户可以浏览攒局列表，点击进入攒局详情页。

3. 进入聊天界面，用户可以与其他参与攒局的用户进行实时聊天。

4. 在个人信息界面，用户可以编辑个人信息，如更换头像、修改昵称等。

## 3.2外部接口

### 3.2.1与django后端的接口

1. 用户登录：

- 接口路径：`/api/login/`

- 请求方法：POST

- 请求参数：微信授权信息，如 code、encryptedData、iv

- 返回结果：用户信息，包括用户ID、昵称、头像等

2. 获取攒局帖子：

- 接口路径：`/api/events/`

- 请求方法：GET

- 返回结果：攒局列表，每个攒局的基本信息

3. 获取攒局详情：

- 接口路径：`/api/events/{event\_id}/`

- 请求方法：GET

- 返回结果：攒局的详细信息，包括标题、描述、参与人数等

4. 参与攒局：

- 接口路径：`/api/events/{event\_id}/join/`

- 请求方法：POST

- 请求参数：用户ID、攒局ID

- 返回结果：成功或失败的状态信息

5. 发送聊天消息：

- 接口路径：`/api/chat/`

- 请求方法：POST

- 请求参数：发送者ID、接收者ID、消息内容

- 返回结果：成功或失败的状态信息

## 3.3内部接口

### 3.3.1小程序前端和后端通信

1. 发起 HTTP 请求：

- 使用微信提供的 `wx.request` 接口。

- 设置请求头、请求体、请求方法等参数。

2. 处理 HTTP 响应：

- 接收后端返回的 JSON 数据。

- 根据返回的状态码判断请求成功或失败。

3. 异步操作：

- 使用微信小程序提供的 `wx.showLoading`、`wx.hideLoading` 等接口处理异步操作。

4. 文件上传：

- 若有头像或图片上传需求，使用 `wx.uploadFile` 接口上传文件。

# 4、运行设计

## 4.1运行模块组合

### 4.1.1小程序前端模块组合

主页模块：展示攒局列表，处理用户导航。

攒局详情模块：展示攒局详细信息，处理用户参与请求。

聊天模块：处理用户聊天请求，显示聊天记录。

个人信息模块：展示和编辑用户个人信息。

### 4.1.2Django后端模块组合

用户管理模块：处理用户登录、注册、信息更新等请求。

攒局管理模块：处理攒局创建、详情获取、参与等请求。

聊天管理模块：处理聊天消息的发送和接收，维护聊天记录。

## 4.2运行控制

运行控制将严格按照各模块间函数调用关系来实现。在各事务中心模块中，需对运行控

制进行正确的判断，选择正确的运行控制路径。

在网络数据传输方面，用户在发送数据请求后，将等待服务器的确认收到信号，收到后，

再次等待服务器发送回答数据，然后对数据进行确认。服务器在接到数据后发送确认信号，在对数据处理、访问数据库后，将返回信息送回用户浏览器完成B/S之间的数据交互操作。

## 4.3运行时间

1. 实时性要求：

聊天模块具有实时性要求，保证用户在聊天界面能够及时接收到新消息。

攒局状态更新具有一定的实时性，确保用户能够实时看到攒局的参与人数变化。

2. 定时任务：

数据清理任务：定期清理数据库中过期的聊天记录、攒局信息等。

活动提醒任务：定时提醒用户即将开始的攒局活动。

# 5、系统数据结构设计

## 5.1E-R实体图

软件的数据库概念模型（ER图）

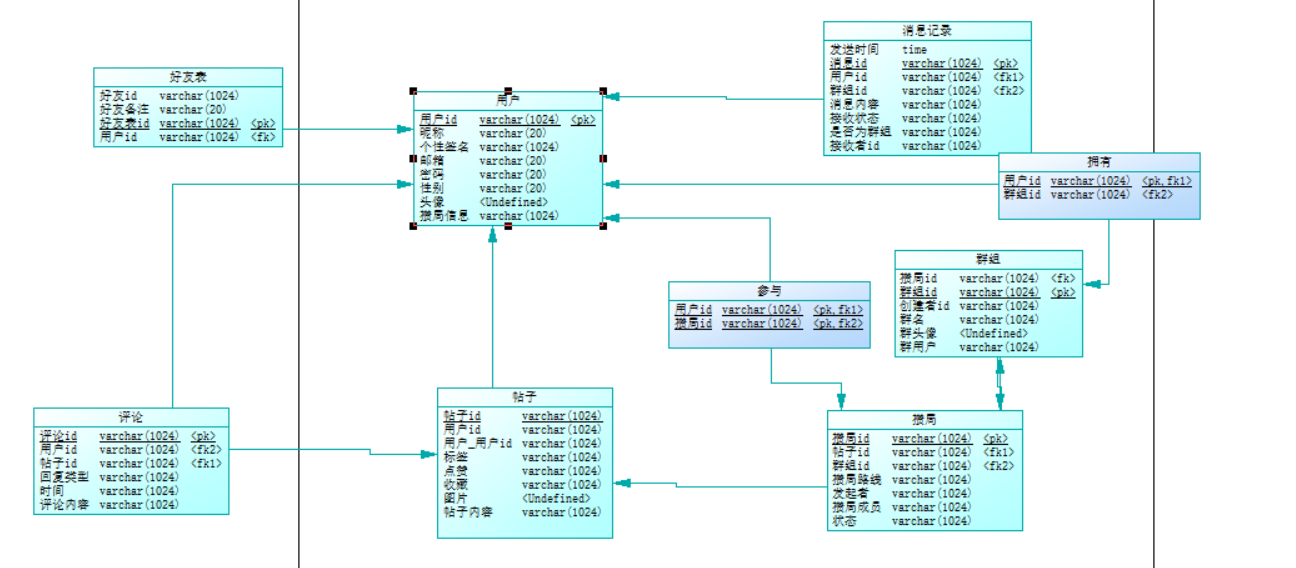


            图 4 “爱游”概念模型

  由图4所示，该ER图中共有7个实体，分别为用户、好友表、消息记录、群组、攒局，评论和帖子，其中用户以用户id为关键字，攒局以其编号为关键字，帖子以其编号为关键字，群组以其编号为关键字，消息记录以其编号为关键字，好友表以好友id为关键字，评论以其编号为关键字。

## 5.2 数据结构表

根据系统需求得出本系统需要的数据表有：用户信息表、帖子表、攒局表、群组表、消息记录表、好友表、评论表7张表。

(1)用户信息表

| 字段名 | 数据类型 | 长度 | 约束条件 |
| --- | --- | --- | --- |
| 用户id | 字符串 | 15 | 主键 |
| 昵称 | 字符串 | 10 | 非空 |
| 个性签名 | 字符串 | 20 | 非空 |
| 邮箱 | 字符串 | 20 | 非空 |
| 密码 | 字符串 | 15 | 非空，限制为大于6位 |
| 性别 | 字符串 | 2 | 非空，限制为'M'或'F' |
| 攒局信息 | 字符串 | 10 | 非空 |

(2)帖子表

| 字段名 | 数据类型 | 长度 | 约束条件 |
| --- | --- | --- | --- |
| 帖子id | 字符串 | 20 | 主键 |
| 标签 | 字符串 | 1024 | 非空 |
| 点赞 | 整数 | 20 | 非空 |
| 收藏 | 整数 |  | 非空 |
| 帖子内容 | 字符串 | 2000 | 非空 |

(3)攒局表

| 字段名 | 数据类型 | 长度 | 约束条件 |
| --- | --- | --- | --- |
| 攒局id | 字符串 | 15 | 主键 |
| 攒局路线 | 字符串 | 1024 | 非空 |
| 发起者 | 字符串 | 15 | 非空 |
| 攒局成员 | 字符串 | 1024 | 非空 |
| 状态 | 字符串 | 15 | 非空 |

(4)群组表

| 字段名 | 数据类型 | 长度 | 约束条件 |
| --- | --- | --- | --- |
| 群组id | 字符串 | 20 | 主键 |
| 创建者id | 字符串 | 15 | 非空 |
| 群名 | 字符串 | 10 | 非空 |
| 群用户 | 字符串 | 1024 | 非空 |

(5)消息记录表

| 字段名 | 数据类型 | 长度 | 约束条件 |
| --- | --- | --- | --- |
| 发送时间 | 时间 |  | 非空 |
| 消息id | 字符串 | 10 | 主键 |
| 消息内容 | 字符串 | 1024 | 非空 |
| 接收状态 | 字符串 | 5 | 非空 |
| 是否为群组 | 字符串 | 15 | 非空 |
| 接收者id | 字符串 | 15 | 非空 |

(6)好友表

| 字段名 | 数据类型 | 长度 | 约束条件 |
| --- | --- | --- | --- |
| 好友id | 字符串 | 15 | 非空 |
| 好友备注 | 字符串 | 10 | 非空 |
| 好友表id | 字符串 | 15 | 主键 |

(7)评论表

| 字段名 | 数据类型 | 长度 | 约束条件 |
| --- | --- | --- | --- |
| 评论id | 字符串 | 20 | 主键 |
| 回复类型 | 字符串 | 20 | 非空 |
| 回复时间 | 时间 |  | 非空 |
| 评论内容 | 字符串 | 1024 | 非空 |

# 6、系统出错处理设计

## 6.1出错信息

程序在运行时主要会出现两种错误:1、由于输入信息，或无法满足要求时产生的错误称为软错误。2、由于其他问题，如网络传输超时等产生的问题，称为硬错误。

对于软错误，须在用户发出的请求操作是否合法及输入的数据验证模块由数据进行数据分析，判断错误类型，并将错误记录到系统错误日志中再返回给用户友好的错误提示信息。

对与硬错误，可在出错的相应模块中输出简单的出错语句，并将程序重置。返回输入阶段。

## 6.2补救措施

1.所有的客户机及服务器都必须安装不间断电源以防止停电或电压不稳造成的数据丢失的损失。

2.定期建立数据库备份，一旦服务器数据库被破坏，可以使用最近的一份数据库副本进行还原。

3.在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网络断路时，还能继续保证数据的正常通信。

4.在硬件方面要选择较可靠、稳定的服务器机种，保证系统运行时的可靠性。

5.为防止服务器故障，预备另外一台服务器，只要主服务器出现故障，可以迅速启动预备服务器运行系统。

6.根据收集和分析用户的反馈信息，筛选用户提出的功能要求，对于合理的予以采纳，系统维修人员根据要求进行更新修改。

# 7、效益成本分析

7.1成本分析

7.1.1研发成本

(1)人力成本:项目组三名成员,按照 chaque成员90-95分计,每人平均成本约为 15000元。

(2)设备成本:研发和测试设备折旧费用约5000元。

(3)软件费用:开发工具和服务器的费用约10000元。

7.1.2运营成本

(1)服务器费用:云服务提供商的费用按流量和容量计费,预计每月1000元。

(2)人工客服成本:2名客服,月薪3000元,共计每月6000元。

7.2效益分析

7.2.1直接效益

(1)会员收入:预计第一年吸引10万会员,年费100元,共计100万元收入。

(2)广告收入:逐步引入商业广告,预计第一年可实现20万元收入。

7.2.2间接效益

(1)用户数据价值:丰富的用户数据和使用行为数据,有利于未来产品 Upgrade 和商业变 现。

(2)品牌影响力提升:“爱游”品牌知名度提高,有利于后续产品推广。

7.2.3投资回收期

开发成本 + 年运营成本 = 40000 + (12000 + 1000×12) = 76000元

第一年总收入 = 100万 + 20万 = 120万元

投资回收期 = 开发+运营成本/年收入 = 76000/120万= 0.63年

初步预测投资可在8个月内收回,项目具有很高的成本效益。

# 详细设计

### 8.1 设计概述

8.1.1 项目简介

爱游小程序是一款基于微信平台的社交旅游应用，旨在为用户提供一个方便、轻松、有趣的方式来组织和参与集体旅游活动。

8.1.2 架构概述

系统分为前端和后端两个主要部分。前端采用微信小程序框架，实现用户界面和交互。后端使用Django框架，通过RESTful API提供数据支持。数据存储在MySQL数据库中。

8.2 运行环境概述

8.2.1 前端微信小程序环境

微信小程序基于微信开发者工具，支持微信版本6.6.6及以上。

使用云开发能力存储用户数据，采用云函数实现一些后台逻辑。

8.2.2 后端Django环境

Django框架运行在Python 3.8以上。

使用Django REST framework构建API，通过Gunicorn和Nginx部署。

采用Docker容器化部署，提高应用的可伸缩性和灵活性。

8.2.3 数据库环境

使用MySQL 8.0作为主要数据库存储用户信息、攒局活动信息、报名信息等。

数据库采用InnoDB引擎，支持事务管理，确保数据的一致性。

8.3 条件与限制

8.3.1 技术限制

微信小程序对每日请求次数有限制，需合理设计API调用。

Django框架在处理大规模并发请求时需要优化，采用缓存等手段提高性能。

8.3.2 安全性考虑

使用JWT（JSON Web Token）进行用户身份验证。

通过HTTPS协议保障数据传输的安全性。

实施合适的权限控制，确保用户只能访问其拥有权限的资源。

8.4 详细设计方法和工具

8.4.1 前端微信小程序设计

设计用户注册、登录、攒局创建、报名等页面。

利用微信云开发存储用户信息，使用云函数实现一些业务逻辑。

8.4.2 后端Django设计

设计Django模型，包括用户模型、攒局活动模型。

编写RESTful API，包括用户注册、登录、攒局创建、报名等接口。

利用Django Signals实现一些触发式操作，如用户报名后发送通知。

8.4.3 数据库设计

设计用户表、攒局活动表、报名信息表，通过外键关联建立数据之间的关系。

考虑数据库索引的使用，提高查询性能。

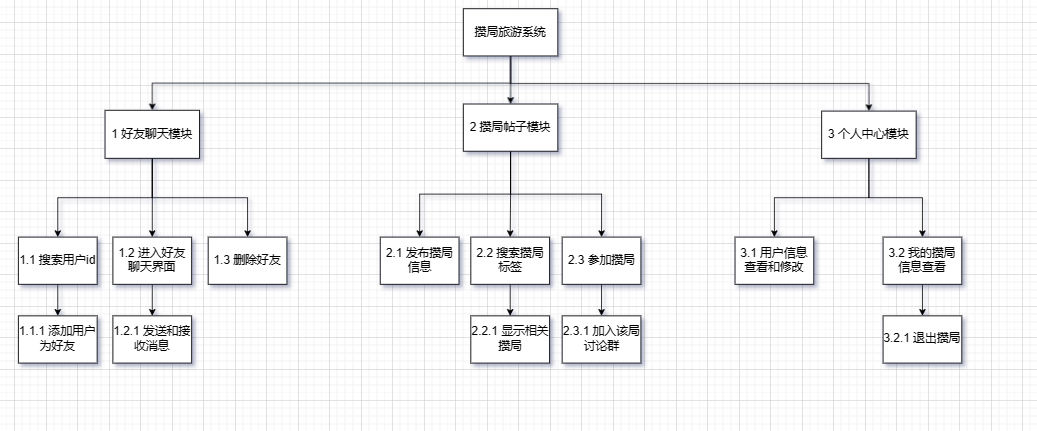
8.4.4 接口设计

定义前后端之间的RESTful API接口，包括请求方法、数据格式等。

实施用户认证机制，保障API的安全性。

使用Docker进行应用的容器化部署，利用Docker Compose管理多个服务的部署。

### 8.2 HIPO图



### 8.3模块详细设计

每个模块的描述说明可参照以下格式：

模块编号：1

模块名称：添加id模块

输入：用户id

处理：用户1点击好友列表的中用户2，系统通过在消息记录表里的用户id项进行精确查询，搜索到该用户，可以选择添加该用户，系统会向该用户发送请求，得到确认后系统会在好友表里生成两条条信息包含两位用户的id，以及好友备注等信息。

算法描述：使用django的orm框架MySQL数据库的用户表、好友表进行查、增等处理

输出：在好友表中生成一条用户id为用户1id和一条用户id为用户1id数据

Pdl：

# 输入

user\_id1 = "" # 用户1的id

user\_id2 = "" # 用户2的id

# 处理

friend\_list = 好友表.objects.filter(用户id=user\_id1) # 在好友表中搜索用户1的所有好友

if user\_id2 not in friend\_list: # 如果用户2不在用户1的好友列表中

# 向用户2发送添加好友请求

request = FriendRequest.objects.create(from\_user\_id=user\_id1, to\_user\_id=user\_id2)

if request.is\_accepted(): # 如果请求被接受

# 在好友表中生成两条记录

好友表.objects.create(用户id=user\_id1,好友id=user\_id2)

好友表.objects.create(用户id=user\_id2, 好友id=user\_id1)

# 输出

# 在好友表中生成两条记录,一条用户1的好友中包含用户2,一条用户2的好友中包含用户1

模块编号：2

模块名称：发送和接收消息模块

输入：消息

处理：用户1在好友界面点击用户2，相当于用户1输入用户2用户id系统通过在用户表里的用户id项进行精确查询，搜索到该用户，系统会将所有发送者为用户1接收者为用户2或者发送者为用户2接收者为用户1并且并非群组消息的消息进行提交到聊天界面，按发送时间进行展示，用户1向用户2发送消息，会先将聊天信息进行保存在消息记录表里生成一条信息，随后进行页面刷新，显示该条记录。

算法描述：使用django的orm框架MySQL数据库的消息记录表进行查、增等处理

输出：在消息记录表中生成一条用户id为用户1id包含消息内容，发送时间等信息的数据

Pdl：

# 输入

sender\_id = "" # 发送者用户id

receiver\_id = "" # 接收者用户id

message = "" # 消息内容

# 处理

# 保存消息到数据库

消息记录.objects.create(

发送者=sender\_id,

接收者=receiver\_id,

消息内容=message

)

# 获取相关消息

messages = 消息记录.objects.filter(

Q(发送者=sender\_id, 接收者=receiver\_id) |

Q(发送=receiver\_id,接收者=sender\_id)

).order\_by("发送时间")

# 输出

# 在消息表中新增一条记录

# 得到排序后的相关消息列表

模块编号：3

模块名称：删除好友模块

输入：删除好友id

处理：用户1在好友界面点击用户2，可以选择用户1输入用户1、用户2用户id系统通过在好友表里的用户id项、好友id项进行精确查询，搜索到该条消息，系统会将该条信息在好友表中删除。

算法描述：使用django的orm框架MySQL数据库的好友表进行查、增等处理

输出：在消息记录表中生成一条用户id为用户1id包含消息内容，发送时间等信息的数据

Pdl：

# 输入

user\_id1 = "" # 用户1的id

user\_id2 = "" # 用户2的id

# 处理

好友表.objects.filter(

Q(用户id=user\_id1, 好友id=user\_id2)

).delete()

# 输出

# 在好友表中删除好友记录

模块编号：4

模块名称：发布帖子模块

输入：帖子消息

处理：用户1在首页界面点击发布帖子，可以跳转发布帖子模块，输入帖子信息，系统自动将在帖子表中生成一个包含帖子内容，发布者id，点赞收藏等的数据

算法描述：使用django的orm框架MySQL数据库的帖子表进行查、增等处理

输出：在帖子表中生成一条用户id为用户1id包含帖子内容，发布时间等信息的数据

Pdl：

# 输入

user\_id = "" # 用户id

content = "" # 帖子内容

# 处理

帖子.objects.create(

用户id=user\_id,

帖子内容=content

)

# 输出

# 在帖子表中新增一条包含发布帖子的用户id和帖子内容的记录

模块编号：5

模块名称：搜索帖子模块

输入：搜索内容消息

处理：用户1在首页界面点击搜索帖子，输入搜索内容，系统自动将在帖子表中对帖子内容进行模糊查询，将搜索到的信息提交到界面进行展示。

算法描述：使用django的orm框架MySQL数据库的帖子表进行查等处理

输出：所以模糊查询所得到的内容

Pdl：  
# 输入

keyword = "" # 搜索关键词

# 处理

posts = 帖子.objects.filter(帖子内容=keyword)

# 输出

# 返回包含关键词的帖子列表

模块编号：6

模块名称：加入攒局模块

输入：帖子id

处理：用户1在首页界面点击帖子，进入帖子，点击加入攒局，系统自动将在攒局表中对帖子id进行精确查询，将搜索到的所有信息提交到界面进行展示，用户选择加入在其中之一的攒局，系统在参与表中生成一条攒局id是该攒局，用户id为该用户的数据。

算法描述：使用django的orm框架MySQL数据库的参与表进行查等处理

输出：在参与表中生成一条攒局id是该攒局，用户id为该用户的数据。

Pdl：

# 输入

user\_id = "" # 用户 id

group\_id = "" # 攒局 id

# 处理

参与.objects.create(

用户id=user\_id,

攒局id=group\_id

)

# 输出

# 在参与表中新增一条记录,记录用户加入了该攒局

模块编号: 7

模块名称: 用户信息查看和修改

输入: 用户id、用户昵称、个性签名、邮箱、性别、头像

处理: 通过用户的id去数据库的用户表中查询，查询出用户的昵称、个性签名、邮箱、性别、头像等信息。再通过新输入的用户昵称、个性签名、邮箱、性别、头像对原信息进行修改。

算法描述: 使用diango的 orm 框架 MySQL 数据库的用户表进行查、改等处理

输出: 在用户表中为该用户id的数据进行修改。

Pdl：

# 输入

user\_id = "" # 用户ID

nickname = "" # 昵称

email = "" # 邮箱

gender = "" # 性别

avatar = "" # 头像

# 处理

user = 用户表.objects.get(id=user\_id)

user.nickname = nickname

user.email = email

user.gender = gender

user.avatar = avatar

user.save()

# 输出

# 用户表中该用户的信息修改为新输入的信息

模块编号: 8

模块名称: 我的攒局信息查看及退出攒局

输入: 用户id、攒局id

处理: 通过用户的id去数据库的参与表中查询，查询出该用户的参与信息，通过参与表中的攒局id查询该攒局的相关信息。通过用户选择退出攒局的攒局id去数据库的参与表中查询，删除该用户id和该攒局id的记录。

算法描述: 使用diango的 orm 框架 MySQL 数据库的参与、攒局表进行查、改等处理

输出: 在参与表中为用户id、攒局id的数据进行删除。

# 输入

user\_id = "" # 用户 id

group\_id = "" # 攒局 id

# 查看用户参与的攒局

user\_groups = 参与.objects.filter(user\_id=user\_id)

# 退出攒局

Participation.objects.get(

用户id=user\_id,

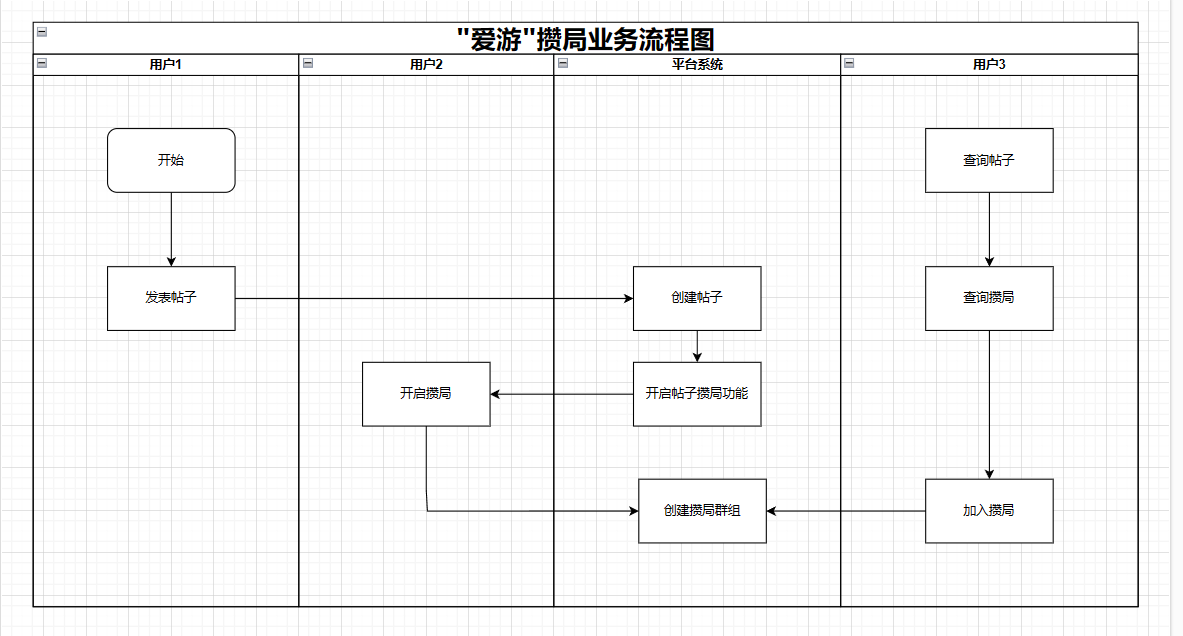
攒局id=group\_id

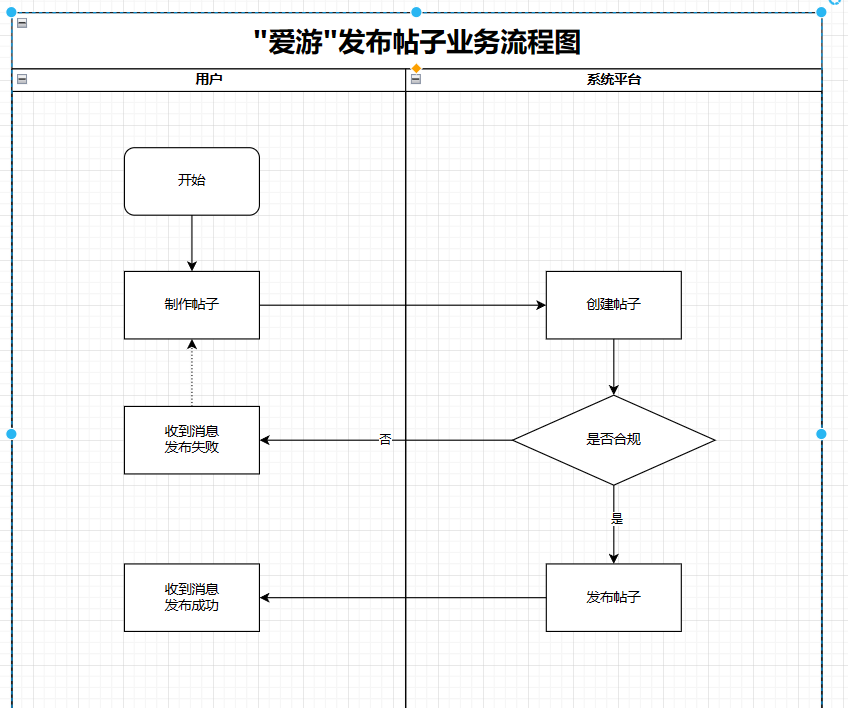
).delete()

# 输出

# 从参与表中删除该用户参与该攒局的记录

### 8.4业务流图





# 9、小组成员分工

## 9.1成员分工

李昊阳：编写总体设计报告的引言、总体设计、成本效益分析部分文档，项目会议整理，界面原型设计，模块结构及pdl语句编写。

黄育锋：编写总体设计报告的系统数据结构设计、系统出错处理设计部分文档，Gantt图，实体联系图设计，业务流图，及ppt制作。

陈芋彤：编写总体设计报告的接口设计、运行设计部分文档，文档整理，hipo图制作以及设计概述部分编写。

## 9.2成员评分

李昊阳：90分

黄育锋：95分

陈芋彤：85分