**前言**

在等地铁时，你是否经常感觉无聊，没有事情可以打发时间？想玩一局游戏却害怕到站后还没结束？你是否会因为地铁出行平台功能过于单调而不想花时间下载平台APP？随着大大小小的城市都开通了地铁，近年来随着我国乘坐公共交通出行的人数不断增长，人均乘坐地铁的时间也愈发延长，人们迫切需要一款能够打发坐地铁途中无聊时光的小游戏。如何开发一款小游戏，能符合在地铁这种信号低、无法进行高精度手机操作的环境是解决问题的关键。

随着HTML5技术的不断发展，制作H5小游戏的成本和门槛越来越低，可内嵌的H5网页小游戏将成为未来各种平台中附加的内容。

结合当代现状因素，本团队设计、搭建了一款适合使用地铁出行主要对象的养成小游戏，为用户提供一个更有趣、更着迷的，用来打发地铁出行时间的利器。本项目搭建的游戏主要应用于移动端，内嵌入平台APP中，基于H5游戏引擎Cocos Creator、SpringBoot框架、AR技术等，致力于为用户提供一个极致的出行体验。

“动物庄园”主要包含了以下内容：

1. **获取、养成宠物**

作为养成小游戏的主要玩法，养成宠物成为了“动物庄园”的主要内容。通过GPS定位技术（有信号时）和出行轨迹记录（无信号时），记录用户出行里程数，用户用里程数获取虚拟货币或宠物孵化值。

将宠物划分等级，等级越高的宠物，购买所需的货币、孵化所需的里程数就越高，以此来鼓励人们多乘地铁，引导绿色出行，达到公司的需求。

1. **聊天社交**

“动物庄园”中提供了好友功能，在养成自己动物的基础上，用户还能够“拜访”好友的庄园，访问他们的宠物并进行互动。由于地铁运行中网络信号不佳，“动物庄园”提前在有网络的情况下缓存好友信息，若访问时网络无信号或信号不佳，则调用缓存中的好友信息，并在有网络后与数据库中原有的数据进行比对、同步。

1. **AR展示宠物**

利用先进的AR增强现实技术，在核心玩法的基础上，结合宠物的3D模型以及外部环境，“动物庄园”增加了AR展示宠物的功能。用户不仅可以在游戏中与自己的宠物互动，也可以在现实中，通过将宠物放置到特定的位置中解锁AR成就，如将宠物放在地铁的扶手上、座位上等。

1. **互动分享出行信息**

在基本的功能以外，“宠物庄园”还设有分享功能，用户可以将自己的出行信息分享到游戏内置的公告频道中或其他的社交平台中。这不仅方便用户之间的沟通，甚至还能遇到坐同一辆地铁的用户，更是方便了我们对用户体验及需求的了解。

除了以上，“动物庄园”的更多内容将在下面一一描述。

本项目充分考虑了中国公共交通出行形势、公司需求，为用户提供一个能够集游戏与社交为一体的H5小游戏。主题绿色健康、卡通形象生动可爱，旨在将游戏与出行APP相结合，提供一个良好的出行体验。

# 目录

[目录 3](#_Toc26676)

[第1章 概述 2](#_Toc4179)

[1.1 项目背景 2](#_Toc3125)

[1.1.1 整体背景 2](#_Toc578)

[1.1.2 社会背景 2](#_Toc20400)

[1.1.3 业务背景 5](#_Toc21140)

[1.1.4 技术背景 5](#_Toc23182)

[1.2 产品概述 6](#_Toc23077)

[1.2.1 产品特点 7](#_Toc30847)

[1.2.2 应用环境 7](#_Toc23295)

[1.2.3主要功能介绍 8](#_Toc31747)

[1.2.4 产品成本分析 9](#_Toc31707)

[第2章 项目需求分析 11](#_Toc24154)

[2.1项目范围 11](#_Toc12471)

[2.2项目风险 11](#_Toc11330)

[2.3业务需求 11](#_Toc301)

[2.3.1功能性需求 11](#_Toc3446)

[2.3.2非功能性需求 13](#_Toc25196)

[2.3.3设计约束 13](#_Toc25924)

[2.4外部接口需求 13](#_Toc18943)

[2.4.1用户界面 13](#_Toc15542)

[2.4.2硬件接口 14](#_Toc1991)

[2.4.3软件接口 14](#_Toc27426)

[2.5内部接口需求 14](#_Toc10454)

[2.5.1储存用户个人信息（add） 14](#_Toc22559)

[2.5.2储存用户账户信息（addAccount） 15](#_Toc29193)

[2.5.3储存道具信息（addProp） 15](#_Toc18987)

[2.5.4创建游戏活动（createAct） 15](#_Toc14487)

[第3章 生产运营 18](#_Toc3980)

[3.1基础理论与技术 18](#_Toc20463)

[3.1.1游戏整体所用的关键技术 18](#_Toc31539)

[3.1.2系统架构 18](#_Toc16274)

[3.1.3系统开发环境 19](#_Toc15779)

[3.1.4系统运行环境 20](#_Toc18648)

[3.2游戏设计与开发 20](#_Toc8749)

[3.2.1游戏设计分析 20](#_Toc17342)

[3.2.2游戏场景分析 25](#_Toc15111)

[3.2.3游戏内容详解 30](#_Toc21289)

[3.2.4功能实现 44](#_Toc6909)

[3.2.5运营设计 46](#_Toc13382)

[3.3数据库设计 47](#_Toc29135)

[3.3.1数据字典 47](#_Toc18929)

[3.3.2数据库设计文字说明 48](#_Toc18567)

[3.3.3数据库建立代码 49](#_Toc19525)

[3.3.4数据库ER图设计 51](#_Toc17720)

[3.4后端设计 52](#_Toc1481)

[3.4.1项目结构 52](#_Toc23052)

[3.4.2接口说明 53](#_Toc4814)

[3.4.3实体类设计 55](#_Toc8598)

[3.4.4Jwt验证 56](#_Toc31999)

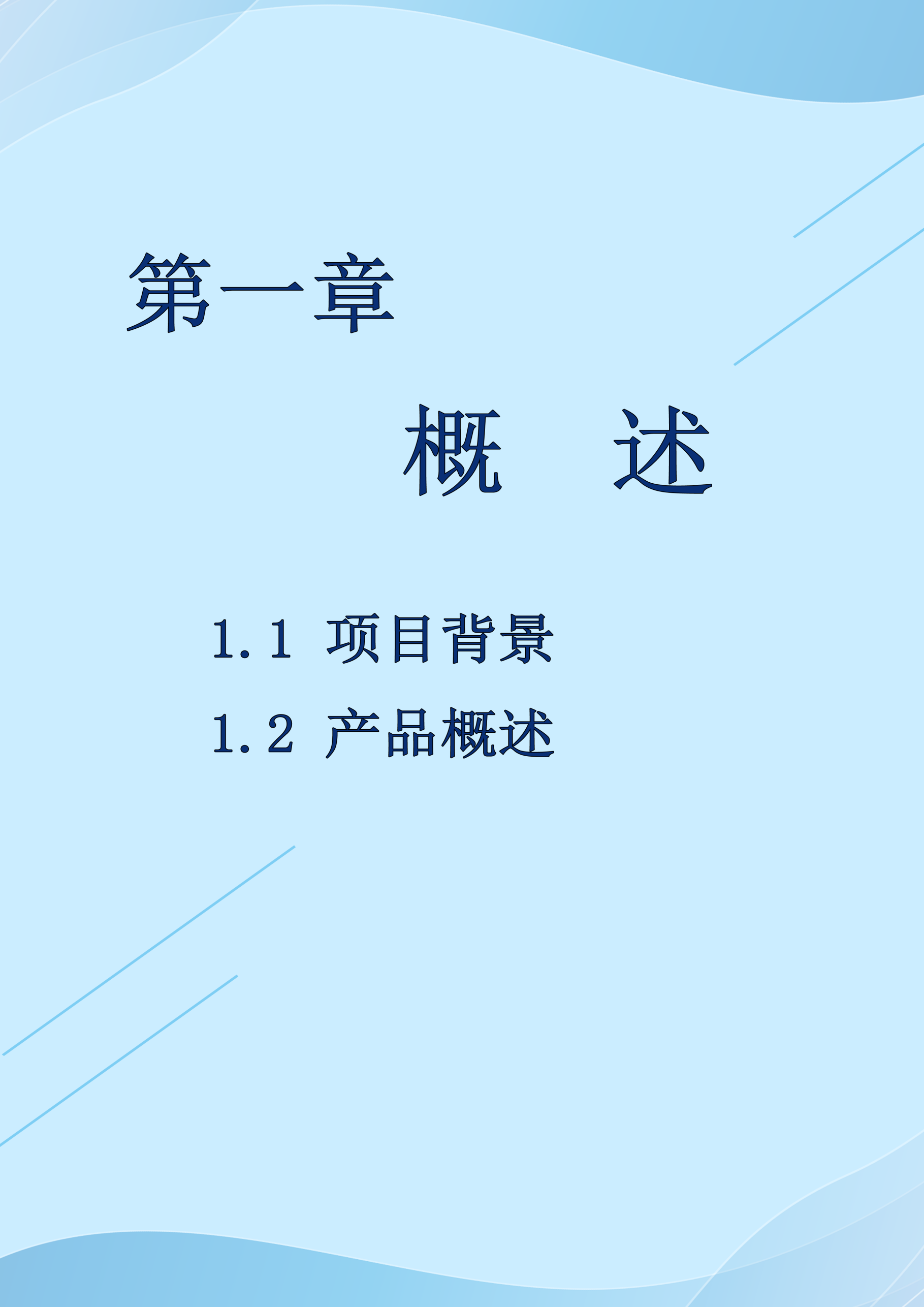
[3.4.5用户登陆 57](#_Toc31414)

[3.4.6信息同步 57](#_Toc7216)

[3.4.7道具更新 58](#_Toc18577)

[第4章 附录 60](#_Toc12019)

[4.1附件一 参赛承诺书 60](#_Toc8049)

****

# 概述

## 项目背景

### 1.1.1 整体背景

近年来，结合游戏元素提升互联网产品的使用体验与用户粘性是一个被广泛

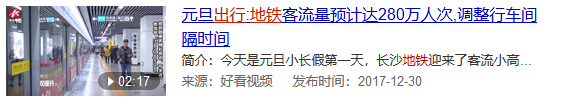
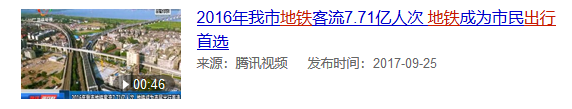
应用的手段。游戏化设计（GamificatiON）或直接嵌入应用内的小游戏越来越常

见，如支付宝蚂蚁森林等，能够给产品运营带来可观的提升。如何结合具体的使用场景与平台特点，设计最合适的游戏元素，达到最佳运营效果，是一个有趣的问题。

### 1.1.2 社会背景

**不断增长的地铁出行量**

随着科技不断发展，时代发展速度的提升，人们对地铁出行的依赖不断增加。

图1-1-2.1 大量市民选择地铁出行

以北京为例，在第三届“首都国企开放日”北京市轨道交通指挥中心活动中，轨指中心展出1973-2017年北京地铁客运量。数据显示，自2008年后，北京地铁客运量基本呈逐年上涨趋势。去年，北京地铁年客运量达到37.78亿人次，轨道交通占北京公共交通出行比例约为50%。地铁出行人数的增长，为相关app的盈利带来极大机会。

**H5小游戏广泛受到大众青睐**

随着移动互联网的普及和HTML5技术的更新，近几年出现了基于H5的小游戏，且广泛收到大众玩家的青睐。

由于H5小游戏能为大家带来轻便的用户体验。用户们想玩的游戏，只需直接通过社交平台的页面就可以打开，无需下载。并且游戏还支持在电脑、手机、平板等多终端运行，能够给用户带来更轻松、更快捷的娱乐体验，这相比于传统APP，H5小游戏的优势是非常明显的

根据游咨网发布的H5小游戏市场研究报告，可以看出2016年H5小游戏增长率到达顶峰，一度到达了203%的高峰。虽然2016年-2018年出现了短暂的下滑，但在近几年又出现了上升的趋势。根据图表数据，在最低谷时H5小游戏的市场规模的增长率仍有50%以上。



图1-1-2.2 H5小游戏市场规模及增长率

在2017年，微信上线小游戏，市场容量一触即发，H5游戏用户数达到3.5亿。H5小游戏的前景也逐渐明朗，无数游戏开发商纷至沓来，尤其是微信首页下拉显示小程序，为小游戏提供了方便的入口，使H5游戏的次日留存率提升到65%，7日留存率达到52%。



图1-1-2.3 H5小游戏用户规模及增长率

**休闲益智游戏占据大量市场**

同样根据游资网发布报告中的数据显示，2017年，H5小游戏类型占比中，角色扮演和休闲益智类型的游戏约占据全部类型的一半之多。其中，角色扮演类型的市场份额约占总游戏类型的26.4%；休闲益智类型的市场份额约占总游戏类型的25.7%，并且这一比例在不断扩大。考虑到地铁出行的场景，本项目选择了后者。

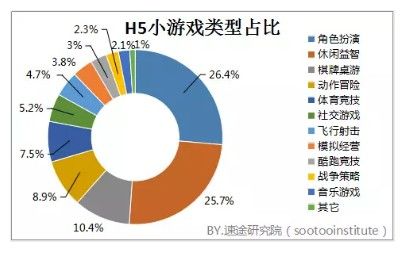


图1-1-2.4 H5小游戏类型占比

以前几年突然爆火的“旅行青蛙”为例。截止2018年1月30日，这款简单的H5小游戏在AppStore的下载量已突破1000万，位居榜首，而中国玩家则占据了其95%。在“佛系”这个词在互联网炸开以后，《旅行青蛙》也从爆冷开始走红，至数据统计那天它依旧保持在AppStore免费榜第一的位置。据统计，该款游戏的玩家在25-40岁左右，因此可以认为卡通画风的游戏对这个年龄段的用户具有一定的吸引力。

**内嵌式H5小游戏的快速发展**

自从H5小游戏爆火后，许多平台推出了内嵌于平台内部的小游戏，其中最具有代表意义的就是支付宝的蚂蚁森林。从蚂蚁森林开服至2019年4月22日，蚂蚁森林玩家已突破5亿，所有玩家一共种下了共计1亿棵树。虽然这款内嵌式小游戏的爆火与支付宝的影响力具有一定的挂钩，但这与此类型小游戏便捷、易操作的特点是不可分割的。

### 1.1.3 业务背景

八维通科技有限公司是中国中车与中国银联合资成立的轨道交通+互联网行

业领先企业。在全国 20 多个城市落地了地铁、公交出行互联网平台产品。然而，

如地铁扫码乘车 APP 等产品具有较强的工具属性，用户停留时间较短，因此需

要探索各种运营手段来提升用户活跃与粘性。

### 1.1.4 技术背景

本项目应用于内嵌式app的技术主要有：

**AR（Augmented Reality）技术**

AR增强现实是一种实时地计算摄影机影像的位置及角度并加上相应图像的技术，是一种将真实世界信息和虚拟世界信息“无缝”集成的新技术，这种技术的目标是在屏幕上把虚拟世界套在现实世界并进行互动。这种技术最早于1990年提出。随着随身电子产品运算能力的提升，增强现实的用途越来越广。由于项目主题为休闲的养成小游戏，正好契合了AR技术的应用点，我们将在未来规划中，把该技术加入到app中。

使用AR技术的优点有：

其一，相比于VR（Virtual Reality）技术，AR技术成本较低、开发技术的需求门槛低。开发一款VR游戏的技术难度和资金需求都要高于同等级别的AR游戏。

其二，AR技术能为用户带来更加新颖的体验，能使原本在手机中的物体影射到显示场景中，为用户枯燥的等车、乘车时间带来更多的乐趣。符合了本次项目的整体背景。

**HTML5游戏开发引擎**

由于近几年微信小游戏的爆火，一系列用于开发内嵌式小游戏的开源引擎涌入市场。本项目选择了Cocos Creator作为游戏开发的引擎。Cocos Creator是以内容创作为核心，实现了脚本化、组件化和数据驱动的游戏开发工具。内部用JS作为支持语言，可用于实现游戏逻辑和高性能游戏效果。

使用开源引擎的优点有：

其一，避免了原生开发中的底层代码，大大降低了前端工程师的工作量，能把注意力全部集中在UI设计和功能开发中。

其二，Cocos Creator引擎相比于其他开源游戏引擎来讲，更易上手。开发人员只需掌握JS一项语言，熟悉一下脚本操作就可以进行小游戏开发。且该引擎中的UI编辑器及其强大，适用于插件化开发，最后发布为Native APP后与原生APP交互能力较强。

**SpringBoot框架**

在后端层面，本项目使用SpringBoot框架作为后台开发的核心框架，该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。如同使用游戏引擎开发游戏，使用框架进行开发已经成为了当下前后端开发必然的趋势。使用集成框架开发，降低了后端工程师对于底层代码的要求，且SpringBoot中的自动化配置Spring容器也为后端开发节省了不少时间。

**用户定位技术**

作为一款在地铁中运行的游戏，获取用户的实时位置必不可少，本项目使用了GPS定位技术实时获取用户的位置。

## 产品概述

本产品基于地铁出行场景，开发了一款不完全依赖网络，可内嵌的H5小游戏。考虑到适用人群以20-40岁为主，本产品设计风格为卡通风不花哨，满足了当代青年、中年用户的审美需求。由于地铁出行网络的限制，以及在车厢内操作手机难度的提升，本产品采用简洁易操的游戏形式——养成小游戏，运用检测网络环境，延时上传数据的技术，降低了用户对网络和操作的依赖；游戏形式即开启即使用，随时可停，防止到站后游戏还未结束的尴尬状况。

### 1.2.1 产品特点

本产品作为一款适用于地铁出行场景，能够独立运行的demo，具有以下特点：

1. 养成类游戏，离开交通用具用户仍可继续游戏，增加用户和八维通的用户黏性
2. 细腻精美UI，针对用户群体特征设计
3. 与周边场景结合紧密，用户行程、POI作为游戏内容加入
4. 收集、成就驱动用户，增加用户使用次数，增加用户和八维通的用户黏性
5. 打通现实八维通和虚拟游戏内容，虚拟货币可作为现实奖励使用，既可增加游戏性，又可作为八维通活动的宣传、推广手段。
6. 游戏虚拟内容考虑到用户实际使用场景时间，例如果实在下班路途中成熟，促进用户打开使用八维通
7. 游戏过程中，网络需求低，无网络连接也可以正常游戏。

### 1.2.2 应用环境

“动物庄园”中使用了数据本地缓存技术，主要适用于高延迟、高并发的地铁出行场景。游戏中的休闲玩法，主要为养成的形式，适合地铁中操作不便的特点。同样，不仅是地铁出行，若企业将来开发出其他公共交通的出行平台，本项目同样能适用。

### 1.2.3主要功能介绍

**与动物角色互动**

在用户主界面中，可通过点击游戏内的动物角色与其互动，虚拟角色会随机冒出对话框和指定的动作。增加了游戏的互动性，用户可以通过出行次数、出行频率解锁不同的动作和对话，已达到鼓励用户绿色出行的目的。

**通过活动获得角色**

在活动页面中，策划商可发布一系列的“活动任务”供用户达成，如使用地铁出行xx次，获得活动宠物一只等。在此页面中，用户还可以使用道具卡直接跳过活动，获取奖励（需要使用八维通积分换取）

**通过八维通平台获得角色**

本项目可与八维通平台对接，通过平台中的活动获取积分，积分可转换成“动物庄园”中的货币换取虚拟角色；也可通过完成八维通平台中的活动、完成地铁行程，获得游戏内限定角色或皮肤，如五一劳动节限定活动、春节限定活动等。

**到达POI获得虚拟明信片**

借助GPS定位技术，游戏可以通过用户手机中的定位功能获取用户所在的实时位置。若出现用户未打开定位功能或在地铁中信号不好等情况，本项目可以通过企业给出的接口获取用户出行的路线来判断用户的出行轨迹。

当用户到达一个新的POI时，游戏会弹出通知告知用户获得了游戏中的虚拟明信片，该明信片设计的样式与解锁匹配的地点相关联。同一地点设有不同种明信片，不同用户获得的虚拟明信片不一定相同，有随机色彩。

**达成成就获取虚拟货币**

在游戏的成就页面中，可看到游戏中的不同成就。当用户达成一定成就时，能够获得成就点，用于升级成就里程碑，换取游戏虚拟货币。如，用户每出行一次，就会增加游戏成就中的“出行次数”条目，若当该条目到达一定次数后，就可获得相应的成就点奖励，升级成就。

**通过虚拟货币换取道具或出行券**

用户在游戏中的获得的虚拟货币可用来换取游戏内部道具，如任务卡（跳过任务直接获取奖励）、加速卡（加速孵化宠物蛋）、经验卡（加快宠物成长）等。在宠物商城中，用户也可以用虚拟货币购买自己喜欢的宠物蛋进行孵化；或为自己已有的宠物购买心仪的皮肤。

虚拟货币也可以与八维通平台对接，攒到一定数量时可以换取平台的出行优惠券。

**好友互动功能**

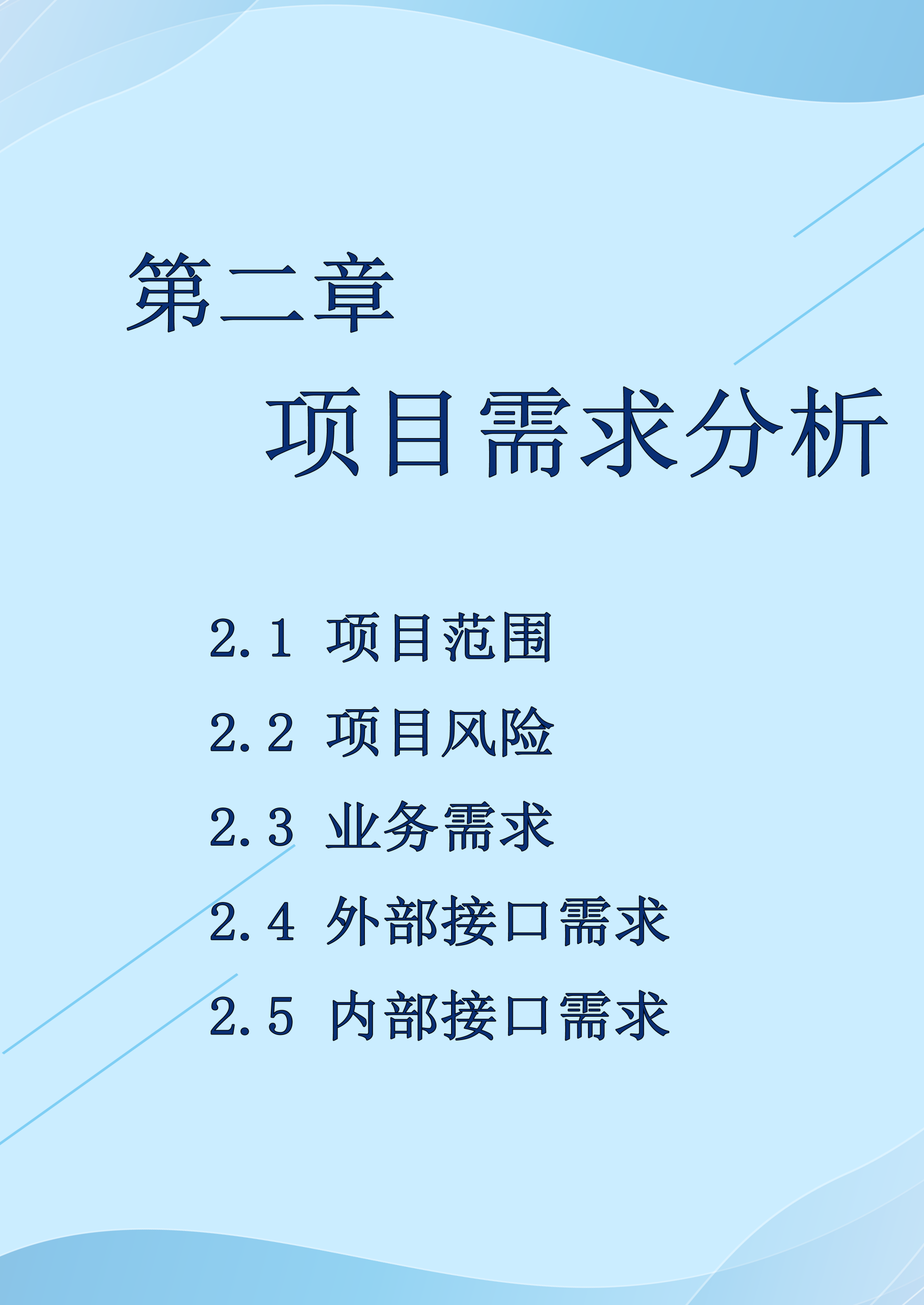
本游戏自带好友功能，玩家可通过输入ID相互添加好友一起游戏。在好友模块中，玩家可访问好友的庄园，为他的庄园点赞或与他的宠物们互动，如 喂食、游戏等，好友亲密度也作为一项成就可供玩家达成。玩家一天可点赞其他玩家指定次数，每次点赞都能获得亲密度，除了解锁成就外还可以在商城中兑换相应商品。

**分享动物角色、成就**

当玩家获得新角色或成就时，会跳出分享页面，鼓励玩家将角色、成就图片分享到社交平台。每天的第一次分享能获得一定的虚拟货币奖励。

### 1.2.4 产品成本分析

本产品属于小型H5游戏，主要的成本在于推广开销、开发开销和后期维护成本。



# 项目需求分析

## 2.1项目范围

本团队开发“动物庄园”的目的不仅是为用户枯燥的出行途中增添乐趣，更是鼓励大众进行绿色出行。项目中使用的本地缓存、路线计算技术适用于信号不佳的地铁场景。发布的Native版本可内嵌入平台，若平台开发出其他公共交通方式，本项目也可以适应。

## 2.2项目风险

由于“动物庄园”为内嵌在八维通平台的游戏，与八维通打通，因此可能会产生一定的恶劣用户，将这个游戏原有的初衷歪曲化，在信息分享中会产生不法分子非法利用游戏获利。如，篡改用户信息、破解游戏帐户、用虚拟金币刷现实代金券等。

## 2.3业务需求

需要实现紧密结合场景特点，具有场景特色、趣味性的 APP 内嵌小游戏。

### 2.3.1功能性需求

该产品主要是一个可内嵌的H5小游戏，需要实现小游戏的各种基本功能：

1. 用户注册登录功能。需要有用户信息管理系统和后台数据库储存用户信息，登录后需要显示主界面和按钮供其操作。
2. 网络检测功能。由于该项目的运行环境在地铁中，网络信号差或无，所以需要实时检测网络环境。若网络信号差或没有，则缓存游戏数据至本地，到有信号时上传数据与后台数据库比对，若有差异，则更新数据库信息。

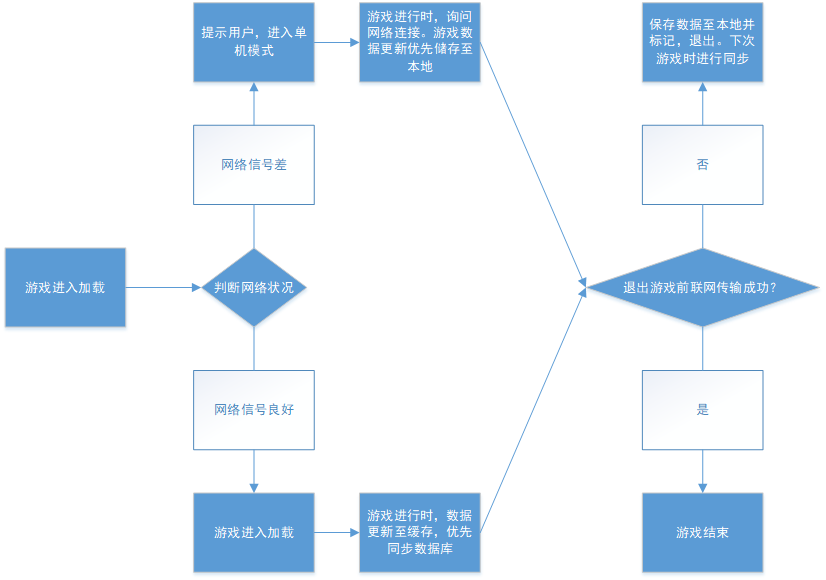


图2-1-1.1 网络检测功能流程图

1. 定位/判断用户出行路线功能。该游戏内设有基于周边POI的成就系统，所以需要实时获取用户的地理位置。考虑到地铁中定位信号也很差，或用户不愿开启手机定位，在用户出行结束后，游戏会通过八维通公司提供API获取用户的出行信息，并判断用户是否经过了预设定的POI。

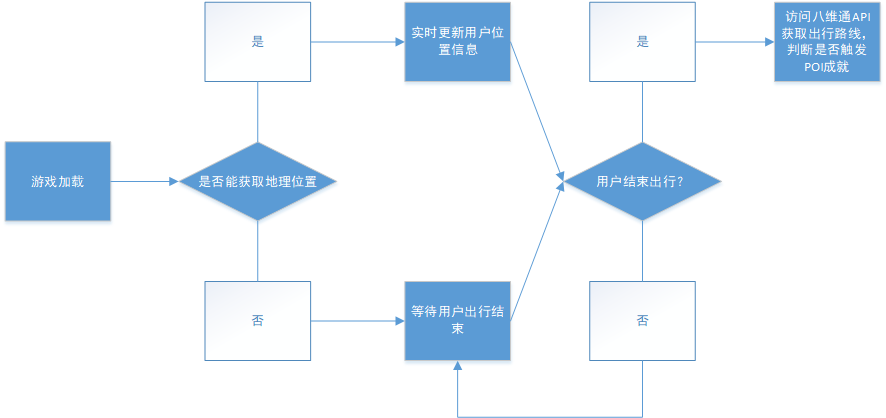


图2-1-1.2 定位功能流程图

### 2.3.2非功能性需求

1. 考虑地铁出行场景中的高并发、高延迟、定位信息不准确等条件
2. 考虑后台参数配置，数据统计需求
3. 考虑场景特点：绿色出行，低碳环保；上下班高峰人流量大、时间固定；站点与周边 POI（PoINT of INTerest）的固定联系等。
4. 考虑用户特征：通勤族、游客等，以 20-40 岁年龄段为主。
5. 考虑公共交通的行业特点，游戏主题应健康、积极。

### 2.3.3设计约束

需设计基于H5可内嵌的小游戏

## 2.4外部接口需求

### 2.4.1用户界面

以下是用户界面包含的一些特征内容：

1. 采用的GUI标准或产品的风格
2. 用户界面采用的屏幕布局
3. 使用在每一个GUI中的组件，包括但不限于：

* 菜单
* 标准按钮
* 各种宠物模型
* 各种场景
* 聊天频道

1. 数据显示格式的规定，包括但不限于：

* 文字对齐方式
* 数字显示格式和对齐方式
* 日期的格式
* 计时方式与时间的显示格式

1. 错误信息的显示标准，如错误代码的含义

### 2.4.2硬件接口

* 支持的硬件类型
* 硬件、软件数据传输格式
* 使用的网络通讯协议等

### 2.4.3软件接口

* 支持的操作系统。如“动物庄园”目前只支持安卓操作系统
* 数据库所在地址。“动物庄园”在demo阶段暂时将数据库保存在个人服务器中
* 需要用到的函数库等。

## 2.5内部接口需求

### 2.5.1储存用户个人信息（add）

请求方式：POST

请求参数： int id，用户ID

String user，用户名

String password，密码

返回值：User user，储存的用户对象

功能：

用户注册时将个人信息储存至数据库。以注册账号时生成uid作为个人的用户身份证明（唯一且无法改变），并根据设定的默认值，初始化用户个人信息。在之后的操作中，也可以通过设置来修改个人信息内容。

### 2.5.2储存用户账户信息（addAccount）

请求方式：POST

请求参数： int id，用户ID

String name，道具名称

int number，道具数量

返回值：Account account，储存的账户对象

功能：

储存用户拥有的道具种类、个数等。以uid作为外键，将个人信息和账户信息连接起来。数据库中存储了各类道具的数量。用户在小游戏中道具的使用和获得均会改变账户信息中道具的数量。如果用户处于网络无连接状态，道具数量发生变化，则再次连接到网络时，会对库存和数据库之前的内容进行比较，如果数量增加明显超过阈值 ，则对该账户进行冻结处理，防止恶意刷道具的现象出现。

### 2.5.3储存道具信息（addProp）

请求方式：POST

请求参数： int id，道具ID

String name，道具名称

String in，道具描述

返回值：Sent sent，储存的道具对象

功能：

道具ID、描述、功能等。以道具id作为主键，游戏每次加载都会检索一遍道具信息数据库版本号，及时更新道具信息，将道具库内容缓存在本地中，如果不是道具的功能更新，就可以在无网络状态下进行游戏。

### 2.5.4创建游戏活动（createAct）

请求方式：POST

请求参数：int id，活动ID

String name，活动名称

Date startTime，开始时间

Date endTime，结束时间

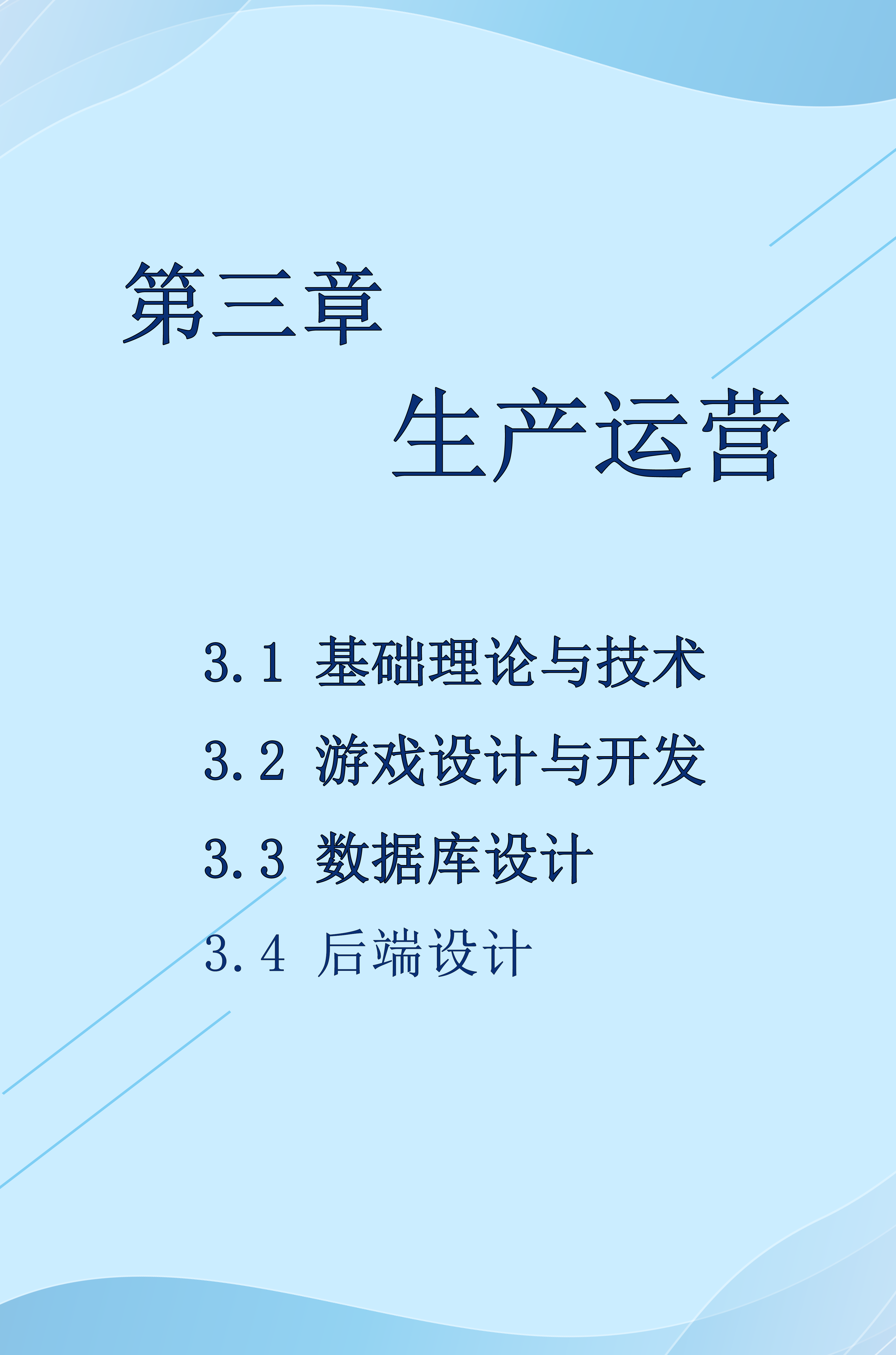
Sent sent，奖励道具

int sentNum，奖励道具数量

返回值：Activity activity，活动对象

功能：

储存游戏历史活动和当前活动。每一条活动都有一个数据段记录活动时间，来区别游戏历史活动和当前活动，并且会生成一个与之对应的数据库来记录玩家的活动状况。



# 生产运营

## 3.1基础理论与技术

### 3.1.1游戏整体所用的关键技术

在该项目中，我们在前端和用户UI界面设计中使用了现在非常火爆的Cocos Creator开源引擎来制作，通过JavaScript语言进行动作逻辑的编写和数据发送和请求。与此同时，我们使用SpringBoot框架作为项目的后端框架，在此基础上开发数据接口，连接数据库，通过JSON传输数据等功能。数据经处理比对后返回到移动端。

### 3.1.2系统架构

对于登陆游戏的用户，用户注册账号，提交的账号、密码等信息进入mysql数据库储存。用户登录后进入游戏主界面，显示相关的“看板”宠物、仓库、活动、商城、个人等界面。在个人界面中可修改头像、昵称、密码等信息，可处理当前的信息或退出登录。在商城界面，用户可以根据自己喜好使用虚拟货币购买宠物，或卖出自己的宠物。在活动界面中，用户可以查看游戏当前的活动，或完成活动时领取奖励。在仓库界面中，用户可以看见自己拥有的所有宠物，并为这些宠物修改姓名、设置皮肤、设置在主界面显示等操作。在首页中，用户能够看到自己设置在主界面显示的宠物，并可以与它互动。

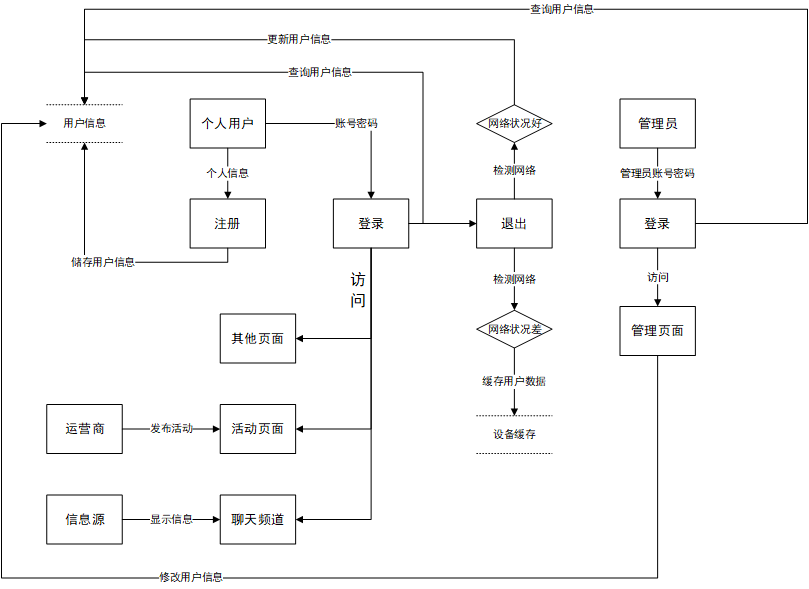


图3-1-2.1 系统架构图

### 3.1.3系统开发环境

前端开发环境：

语言：Javascript

开发工具：Cocos Creator

操作系统：Windows 10

后端开发环境：

语言：Java

开发工具：IntelliJ IDEA

操作系统：Windows 10

框架：SpringBoot、FreeMarker

### 3.1.4系统运行环境

服务器运行环境：CentOS 7.2 64bit

移动端环境：Android 6.0以上

## 3.2游戏设计与开发

### 3.2.1游戏设计分析

3.2.1.1设计目标

设计一个增加八位通平台用户黏性的游戏，从根本上来说，此小游戏为八维通服务，游戏的设计并非为了让用户多游玩游戏本身，而是起到对用户引流增加对八维通平台的目的。由于此游戏内嵌于八维通平台，用户使用游戏的同时，也增加了用户对八维通平台的使用。增加用户黏性的主要方法为吸引用户多使用游戏，从而增加用户对八维通的用户黏性。

3.2.1.2使用分析

**逻辑场景**

游戏内嵌于八维通平台，与八维通平台呈相符相承的关系，游戏目标是鼓励用户在必须使用八维通平台这情况之外使用本游戏。但是最基础的游戏使用情况是用户打开八维通，顺便使用此游戏，所以游戏设计时重点考虑到为使用八维通平台时使用游戏。为了对八维通平台起辅助增加用户黏性的作用，游戏的游戏性设计主要目标是吸引用户在正常使用八维通这情况之外使用游戏。所以游戏使用场景以八维通的使用场景出发，通过游戏的设计实现增加日常生活场景下的使用。

**使用环境**

主要使用场景为地铁等公共交通场景，主要特点为高人流、使用时站立。高人流和地铁处于地下的运行环境，导致使用环境高并发、网络质量波动大，经常出现网络差甚至没有网络的情况，游戏后台设计时就必须针对这一点做出优化。同时由于公共交通的性质，UI设计就必须考虑到使用者面临着站立、拥挤、扶持的环境，帮助用户在这种情况下还能够良好的进行游戏。

**目标用户**

目标用户为八维通平台的主要使用者，由社会背景调查得出，为20-40岁的年轻用户群体，游戏性设计和UI设计就主要针对20-40岁的年轻用户群体进行优化，吸引八维通主要用户群体。

3.2.1.3游戏形式设计

从设计目标可得出，游戏目标为了增加用户使用游戏的次数而非游戏时长，游戏的体裁设计就是实现这一目标的主要出发点。曾经非常风靡的跳一跳、和例如你画我猜、棋牌类游戏这些H5游戏的形式虽有游戏性，但是对增加用户使用次数，增加用户的黏性并没有显著的帮助，并且这些游戏形式的使用场景都不符合需求游戏的使用场景。而之前爆火的《旅行青蛙》这一类型的养成类游戏，游戏内容休闲，单次游戏的强度较小，但是在游戏内收集、成就系统等游戏内容的吸引驱动下，用户的使用次数要显著的大于一般的H5游戏，并且参与游戏的跨度较大，符合游戏的设计目标，能够增加游戏用户和八维通平台的用户黏性。所以我们选择了休闲养成类游戏作为我们所要设计的游戏形式。

3.2.1.4游戏内容设计

**UI设计**

* **画面设计**

选择了游戏形式为养成类游戏之后，决定了游戏是以休闲为主要基调的，为了符合这一游戏基调，游戏画面必须要休闲，并且由于八维通平台面向全年龄段，要起到增加用户黏性的游戏也必须适合全年龄段。卡通画面对于用户有较高的接受度，适合从儿童到老年人全类型人群。并且有《旅行青蛙》和《动物森友会》这类以卡通画面风靡于年轻群体的成功先例，我们选择了卡通画面作为游戏的主要画面。

* **布局设计**

游戏设计运行于移动终端，主流机型高宽比最大的机型为iPhONeX全面屏1242\*2688像素，所以设计分辨率为1242\*2688像素，并且画面设置为符合移动终端的宽，为不同移动终端机型的用户都能提供良好的画面体验。

从使用环境主要为公共交通，游戏的布局针对单手操作做出了优化。如图所示，功能按钮和主要互动的小宠物，布局位置都是针对单手操作做出特殊布局，在保证游戏性和布局美观的前提下，方便用户在拥挤的大人流环境、站立扶持扶手的情况下也能正常的游戏。

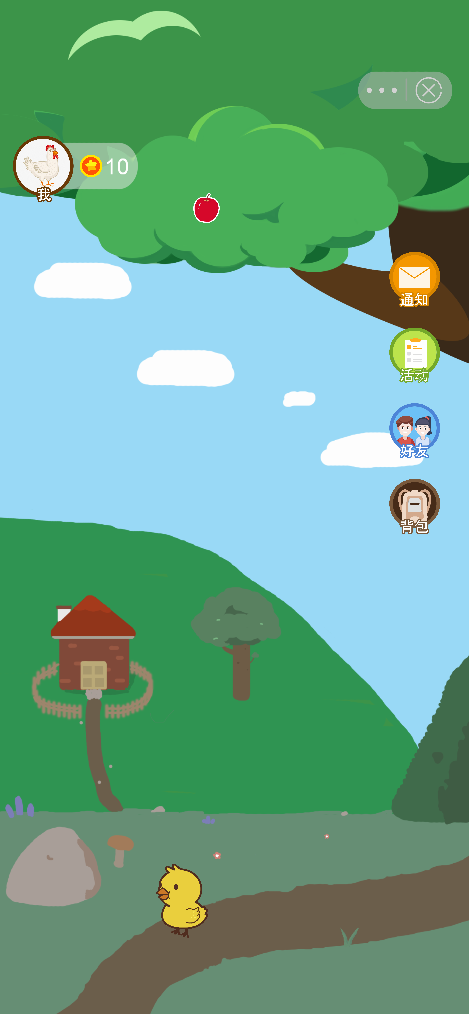
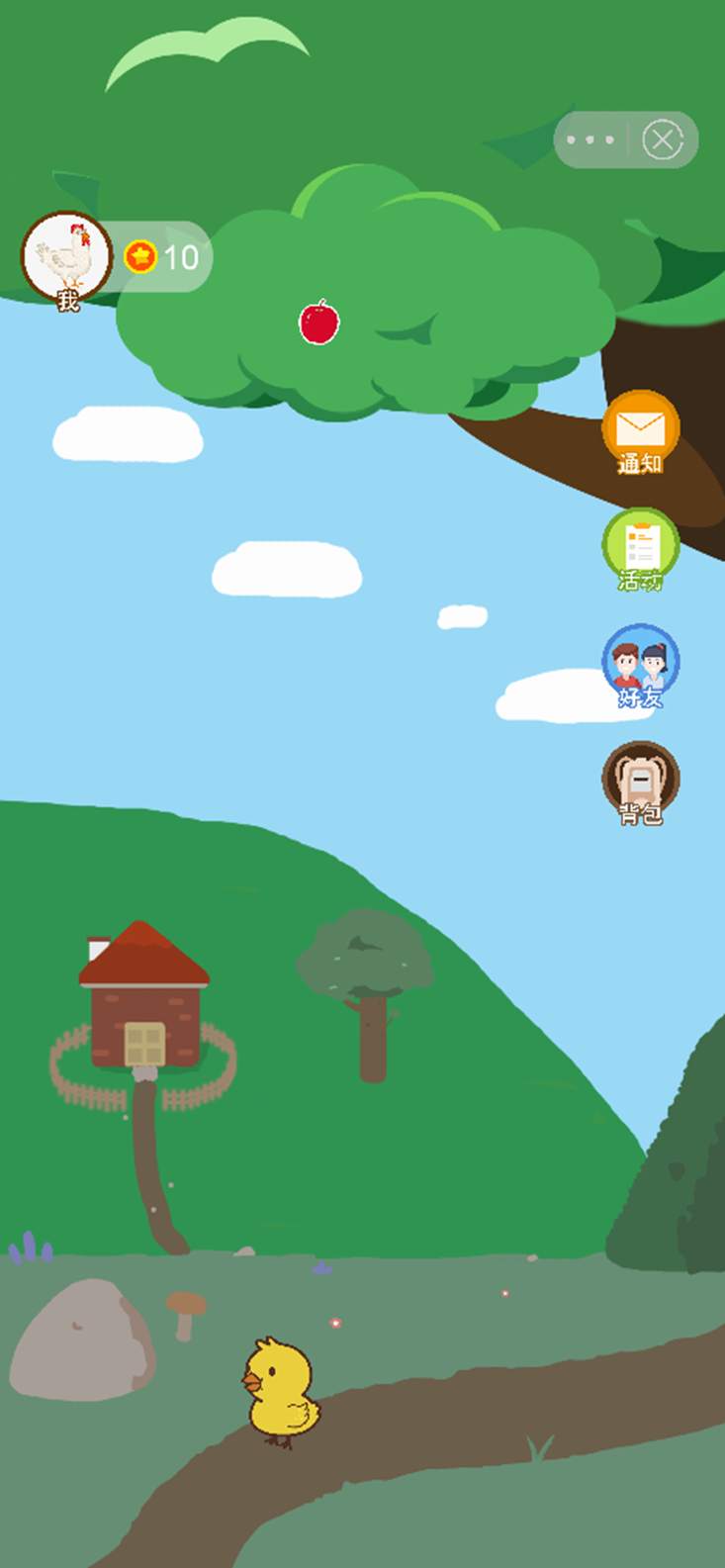
****

图3-2-1-4.1 游戏画面

****

2688px

1242px

图3-2-1-4.1 游戏设计

**游戏性设计**

游戏性的设计是为了增加游戏的吸引力，吸引用户在正常使用八维通的时间之外也使用游戏。在养成类的游戏形式下，我们设计出了收集、成就、分享这样三位一体的用户驱动导向的游戏模式。通过参与游戏内活动、现实在八维通完成出行，获得游戏内的虚拟道具、解锁新的宠物，在对宠物养成的过程中，获得宠物的成长奖励——游戏内的虚拟金币道具，金币可以用于解锁新的道具，同时游戏和八维通平台打通，可以用金币获取类似于出行代金券的现实奖励。并且用户可以把自己的成就分享到朋友圈、微博等社交媒体上，满足用户的成就感。这样的一个三位一体的养成类游戏，能够增加用户参与游戏、使用八维通平台的次数，从而实现增加用户黏性的目标。

### 3.2.2游戏场景分析

游戏场景主要有4个，用于不同功能的游戏内容显示

3.2.2.1主界面（main）

游戏的主界面，在这个界面有用户选择展示的小宠物，用户可以在这个界面看到自己的虚拟金币的数量、和自己的宠物互动。并且通过点击侧面的功能模块，会有不同的功能模块出现。还可以通过点击图中的小房子，进入房间，方便用户深入进行其他游戏内容。

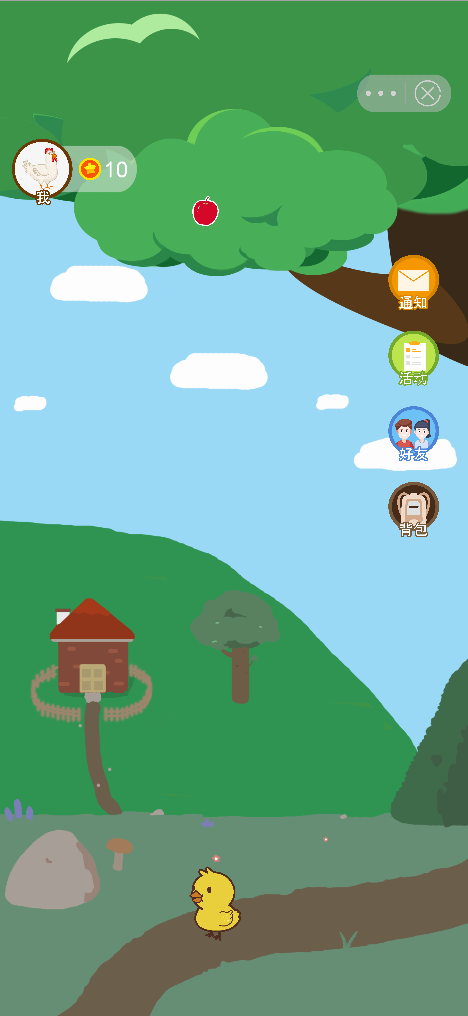
****

图3-2-2-2.1 游戏主界面

3.2.2.2房间界面（house）

此界面为游戏内容的展示收集功能界面，作为深度游戏内容、管理用户道具，通过邮件系统和其他游戏玩家进行互动的界面。此页面可作为展厅展示用户收集的虚拟道具，用户可以在此界面添加放置自己收藏的虚拟道具，个性化定制自己的小屋。此界面在功能模块上，也做为前往收藏界面、明信片界面的集合界面。

****

图3-2-2-2.2 房间界面

3.2.2.3收藏界面（archive）

此界面用于显示游戏内解锁和未解锁的虚拟道具，用户可以在这个界面看到自己所获得收藏的虚拟道具，还可以看到自己还未解锁的道具，可以在这个界面使用游戏内的虚拟金币购买解锁游戏内虚拟道具，也作为一个虚拟和现实的接口，在这用户可以用虚拟金币够买八维通的现实道具。

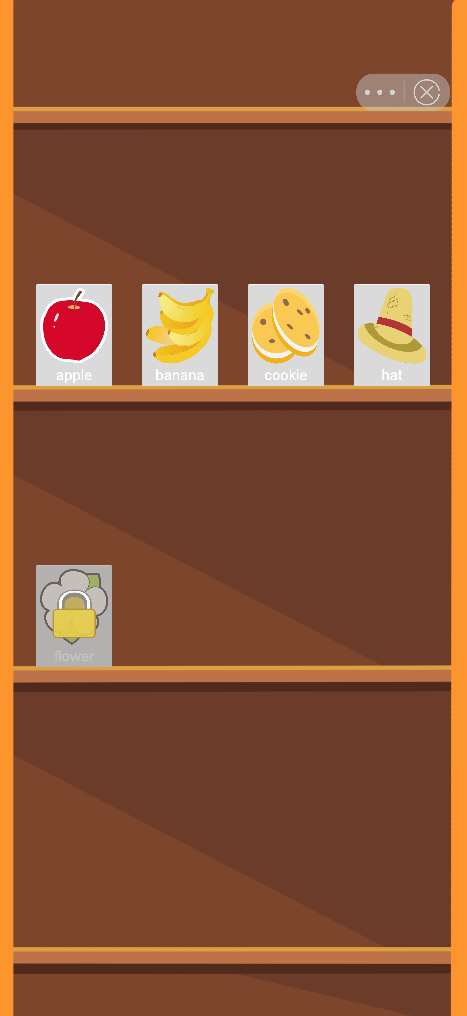
****

图3-2-2-2.3 收藏界面

3.2.2.4明信片界面（pic）

这个界面用于显示用户在达成成就，例如到达新的特殊地铁站、游戏内宠物成长值达到某个里程碑、完成平台活动获得的奖励纪念性质的电子明信片。



### 3.2.3游戏内容详解

3.2.3.1主界面

**通用按钮栏（btn\_general）**

此按钮为全局组件，用户可以通过此按钮快速使用基础功能，如分享、关闭应用。按下分享键，用户可以选择将游戏分享到微博、微信朋友圈，微信好友等主流社交平台。在主界面按下退出键，调用脚本中预先写好的脚本，根据网络状况做对应的响应，上传本地文件到服务器或储存本地文件等到下次游戏时读取上传。

****

图3-2-3-1.1 通用按钮栏

**侧标(btn\_bag,btn\_friend,btn\_acivity,btn\_inform)**

这个组件为全局界面，用于方便用户快速访问通知、活动、好友和背包界面，同。点击按钮，屏幕下方会动态加载出对应的显示内容。

****

图3-2-3-1.2 侧标

此显示框为点击背包按钮后，读取用户的背包道具文件，动态显示的背包内部界面。用户在此界面点击使用虚拟道具，使用虚拟道具后，后台加载使用道具后的成长值、显示变化，并更新背包内容。

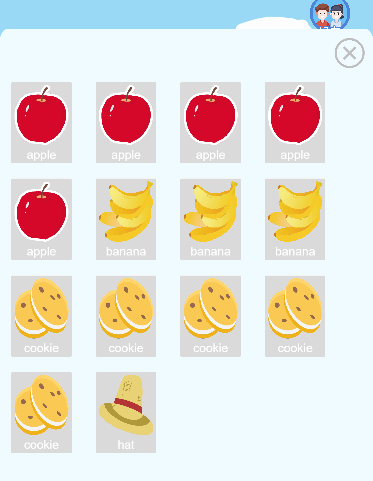
****

图3-2-3-1.3 背包界面

此界面为点击好友按钮后加载出的显示框，后台读取好友文件，显示好友的头像、名称、称号等信息。在此界面点击信件按钮，可以给好友写信，点击右侧的按钮访问好友的农场。

****

图3-2-3-1.4 好友界面

下图为点击前往好友界面后显示的好友农场界面

****

图3-2-3-1.5 好友农场界面

此界面为点击活动按钮后的显示框，加载显示了游戏的活动内容，显示内容包括活动的缩略图、活动概要和前往活动的按钮。用户浏览游戏活动内容，点击前往活动内容，并且获得虚拟道具的奖励。此功能设置用于软件上线后的运营，例如每日活动、八维通活动等，吸引用户上线游戏。

****

图3-2-3-1.6 活动界面

此界面为点击通知按钮后的显示框，加载显示了系统通知、第三方通知等信息，对通知的类型进行了分类显示，通过缩略图了解通知的类型。并显示了通知的标题内容，点击右侧的按钮，进入通知的详细界面进一步了解关于通知的更多内容。

****

图3-2-3-1.7 通知界面

**个人信息栏（account）**

此组件用于显示玩家的个人信息，如金币、昵称、头像等。用户还可以通过点击自己的头像，跳转到个人信息界面。在头像的右侧显示了游戏虚拟货币金币的个数。

****

图3-2-3-1.8 个人信息栏

**小宠物（pet）**

此组件为在主界面显示的宠物，用户可以通过宠物界面中的设置主页宠物。小宠物在用户没有与用户互动时，在屏幕上随机的前进，给用户更加动态、良好的游戏体验。当点击小宠物时，上方会跳出该宠物的成长值和随机与玩家互动的文字，并且使用加载条显示当前小宠物的成长值，当用户使用增加成长值的道具时，小宠物的成长值显示随之改变。当完成阶段性成长之后，成长值清零，小宠物进入下一阶段，并且用户获得虚拟金币。

** **

图3-2-3-1.9 小宠物

**房间入口（btn\_house）**

此组件用于进入个人房间，用户可以通过点击该图标进入自己的房间，即场景改变为房间场景（house）。

****

图3-2-3-1.10 房间入口

**云（cloud）**

作为动态背景，云层缓缓移动，让游戏画面更加生动

****

图3-2-3-1.11 云

3.2.3.2房间界面

**通用按钮栏（btn\_general）**

此按钮为全局组件，分布于游戏的所有界面。用户可以通过此按钮快速使用基础功能，在此界面的“更多”按钮功能同主界面的“更多按钮”，但是在房间界面点击退出键，加载返回到主界面。

****

图3-2-3-2.1 通用按钮栏

**侧标(btn\_bag,btn\_friend,btn\_acivity,btn\_inform)**

这个组件用于方便用户快速访问通知、活动、好友和背包界面，同样为全局组件。功能同主界面。

****

图3-2-3-2.2 侧标

**书籍（btn\_book）**

作为前往博物馆界面的加载接口，点击前往博物馆界面

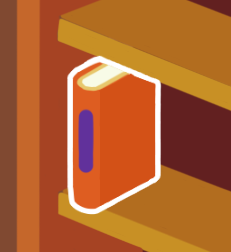
****

图3-2-3-2.3 书籍

**相框（btn\_pic）**

此组件用于查看已获得的明信片，用户可以通过点击相框，进入明信片界面，浏览自己已获得和未获得的明信片。

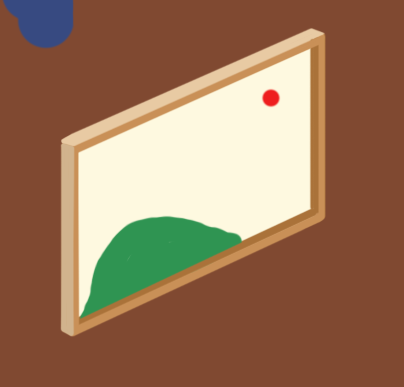
****

图3-2-3-2.4 相框

**宠物角色（character）**

玩家在房间内展示自己的宠物角色，实现了用户对自己宠物的管理和展示。同时宠物角色作为动态内容装饰房间界面，让房间界面更美观，给了用户更多定制游戏内容的空间，为养成游戏增加游戏性。

****

图3-2-3-2.5 宠物角色

**邮件（btn\_mail）**

此组件用于查看用户收到的邮件，玩家可以通过点击信封打开邮件界面进行邮件内容的阅读。

****

图3-2-3-2.6 邮件

此界面为点击了邮件按钮后，加载显示出的邮件内容，用户在这可以看到给自己写信的好友的头像和好友的名字，以及好友给自己写的邮件的标题，点击右侧的按钮可以进入浏览邮件的内容

****

图3-2-3-2.7 邮件界面

**桌（desk）**

用户可以在桌子上选择放置收集到的虚拟道具，作为用户虚拟道具的展示功能点，为用户提供更多可定制的游戏内容，增加了游戏性

****

图3-2-3-2.8 桌

3.2.3.3博物馆界面（archive）

**通用按钮栏（btn\_general）**

此按钮为全局组件，分布于游戏的所有界面。用户可以通过此按钮快速使用基础功能，“更多”按钮功能同主界面和房间界面，点击“退出”按钮后返回房间界面

****

图3-2-3-3.1 通用按纽栏

**货架（shelf）**

此组件作为博物馆界面的背景用于展示用户拥有的道具。

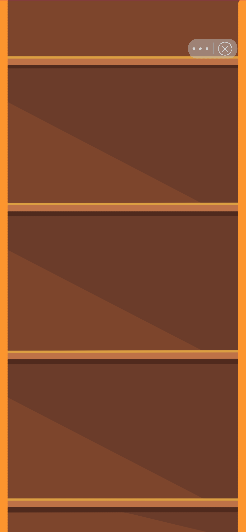
****

图3-2-3-3.2 货架

**虚拟物件**

此界面用于显示用户已拥有的道具和未解锁的道具，道具作为个体显示在货架上。未解锁的道具会以上锁的形式展示，用户通过点击未解锁的道具可以使用游戏内的虚拟金币购买解锁道具，解锁成功的道具会以解锁的形式重新显示，并且添加到背包中供用户使用。

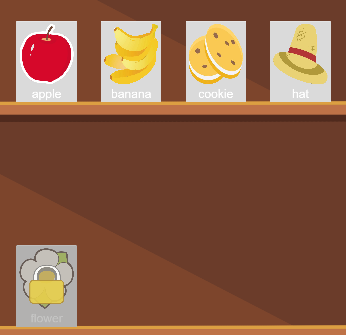
****

图3-2-3-3.3 虚拟物件

**明信片界面**

**通用按钮栏（btn\_general）**

此按钮为全局组件，分布于游戏的所有界面。用户可以通过此按钮快速使用基础功能，在此界面的“更多”按钮功能同主界面的“更多按钮”，但是在房间界面点击退出键，加载返回到房间界面。

****

图3-2-3-3.4 通用按纽栏

**明信片**

用户通过达成游戏里程杯、参加活动获得的纪念意义的明信片展示在这。显示明信片的内容，供用户收集观赏。

****

图3-2-3-3.5 明信片界面

### 3.2.4功能实现

3.2.4.1游戏功能实现

使用Cocos Creator引擎开发游戏

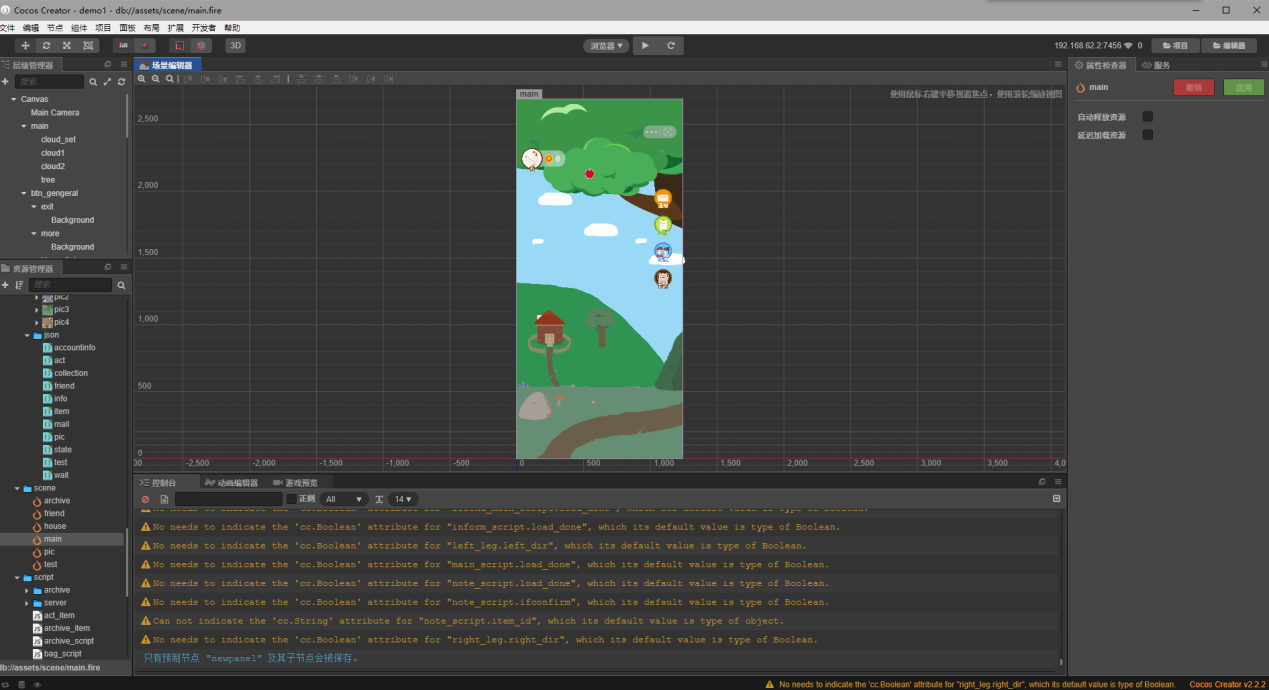


图3-2-4-1.1 游戏开发界面

实现了主要的四个游戏场景，通过给场景内增加编辑节点，实现游戏内容的添加。通过编写JavaScript文件，并挂载JavaScript文件到节点上，调用JavaScript文件实现了节点之间的相互作用，从而实现了设计的游戏内容。

3.2.4.2无网络游戏实现

游戏主要对网络的使用为加载用户资源。游戏刚开始时需要通过网络加载游戏资源，将资源下载到本地并且实现用户信息同步，如果在游戏开始时资源加载和信息同步失败，游戏将无法正常启动知道获得网络连接完成加载。

在游戏过程中，遇到网络状况不佳甚至断网的情况时，发出提示并且游戏进入单机状态，不需要网络连接，但是无法进行联网的游戏内容，比如访问好友的庄园和给好友发送邮件。

单机游戏时，用户仍可以使用虚拟道具，查看明信片等本地已下载好的资源，并且对账户信息的修改将储存到本地文件中，保存用户的修改等待上传服务器。

在用户退出游戏时，调用JavaScript中的main文件里的Update\_user方法请求上传用户文件，如果还是没有网络连接，将在下一次游戏开始时上传到服务器，实现用户信息的同步。

3.2.4.3分享游戏实现

在后续游戏运营过程中，开发出相应的微信小程序，实现分享的接口，完成游戏内容的分享至主流社交平台

### 3.2.5运营设计

游戏上线正式运营后，需要运营团队进行程序维护、游戏更新等维持、发展游戏的工作，主要分为美工、前后端程序员、策划等四部分

3.2.5.1美工

负责游戏内容更新时，提供游戏内容更新所需的美工原画素材。

3.2.5.2前后端程序员

后端负责维护储存有用户数据的服务器，并且在需求改变、业务拓展时实现数据的更新换代

前端程序员负责实现由游戏策划提出的内容需求，完成前端UI和游戏逻辑功能的改变

3.2.5.3策划

为运营团队的主要职位。

负责设计游戏内容的更新设计，给游戏添加新的内容，例如新的道具、新的游戏方式等

负责线上线下的游戏运营推广，增加游戏的知名度，吸引更多的游戏用户

负责游戏活动的设计，和游戏相关的游戏内虚拟活动、线下的实体活动的方案策划。

## 3.3数据库设计

### 3.3.1数据字典

表 3-3-1.1 **User**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许重复 | 默认值 | 允许非空 | 自动递增 | 备注 |
| UserId | INTeger | NO |  | NO | Yes | 主键 |
| Uid | VARCHAR(50) | NO |  | NO | NO | 玩家身份证明 |
| NickName | VARCHAR(50) | No |  | NO | No | 昵称 |
| User\_Level | INTeger | YES | 1 | NO | NO | 等级 |
| User\_INTroductiON | VARCHAR(255) | YES | 这个人很懒，不写介绍 | YES | No | 个人介绍 |
| User\_Tel | VARCHAR(11) | NO |  | NO | NO | 联系电话 |

表 3-3-1.2 **Backpack**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许重复 | 默认值 | 允许非空 | 自动递增 | 备注 |
| BPId | INTeger | NO |  | NO | Yes | 主键 |
| Uid | VARCHAR(50) | NO |  | NO | NO | 玩家身份证明 |
| Apple\_255 | INTeger | YES | 0 | NO | No | 255类型苹果的数量 |
| Banana\_222 | INTeger | YES | 0 | NO | NO | 222类型香蕉的数量 |

表 3-3-1.3 **Materials**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许重复 | 默认值 | 允许非空 | 自动递增 | 备注 |
| MaterialsId | INTeger | NO |  | NO | Yes | 主键 |
| Ma\_name | VARCHAR(50) | NO |  | NO | NO | 道具名 |
| Ma\_INTroductiON | VARCHAR(255) | No | 无 | NO | No | 道具介绍 |
| Ma\_icON | VARCHAR255) | YES |  | NO | NO | 道具图标链接 |

表 3-3-1.4 **Activity**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许重复 | 默认值 | 允许非空 | 自动递增 | 备注 |
| AcID | INTeger | NO |  | NO | Yes | 主键 |
| Ac\_name | VARCHAR(50) | Yes |  | NO | NO | 活动名称 |
| Start\_time | date | Yes | 1999-01-01 | NO | No | 活动开始时间 |
| End\_time | date | YES | 1999-01-01 | NO | NO | 活动结束时间 |

### 3.3.2数据库设计文字说明

在数据库的设计过程中，考虑了游戏道具的储存、游戏活动的储存和当网络状况不佳时，数据的备份库。

在设计User表中，我们考虑了用户的用户名、身份证明（是否为管理员）、昵称、等级和个人简介。在登录时，自动检验Uid是否为管理员，从而做出不同的页面跳转。在Materials和Activity表中，我们考虑了道具的名称、介绍和图标；活动的名称、开始时间和结束时间。

由于当网络状况不佳时，游戏需要将用户信息备份到本地缓存中，因此我们设计了Backpack表，用于备份数据。由于游戏处于demo阶段，道具种类不多，因此设计了用于测试道具的字段，用于保存用户道具的拥有量。

### 3.3.3数据库建立代码

**表1**

**CREATE TABLE** User  
(  
 UserId **INT AUTO\_INCREMENT**,  
 Uid **VARCHAR**(50) **NOT NULL**,  
 NickName **VARCHAR**(50) **NOT NULL**,  
 User\_Level **INT DEFAULT** 1 **NOT NULL**,  
 User\_INTroductiON **VARCHAR**(255) **NULL** **COMMENT** '这个人很懒，不写介绍',  
 User\_Tel **VARCHAR**(11) **NOT NULL**,  
 **CONSTRAINT** User\_pk  
  **PRIMARY KEY** (UserId)  
);  
**CREATE UNIQUE INDEX** User\_NickName\_uindex  
 **ON** User (NickName);  
**CREATE UNIQUE INDEX** User\_Uid\_uindex  
 **ON** User (Uid);  
**CREATE UNIQUE INDEX** User\_User\_Level\_uindex  
 **ON** User (User\_Level);  
**CREATE UNIQUE INDEX** User\_User\_Tel\_uindex  
 **ON** User (User\_Tel);

**表2**

**CREATE TABLE** Backpack  
(  
 BPId **INT AUTO\_INCREMENT**,  
 Uid **VARCHAR**(50) **NOT NULL**,  
 Apple\_255 **INT DEFAULT** 0 **NOT NULL**,  
 Banana\_222 **INT DEFAULT** 0 **NOT NULL**,  
 **CONSTRAINT** Backpack\_pk  
  **PRIMARY KEY** (BPId),  
 **CONSTRAINT** Backpack\_User\_Uid\_fk  
 **FOREIGN KEY** (Uid) **REFERENCE** User (Uid)  
);  
**CREATE UNIQUE INDEX** Backpack\_Uid\_uINDEX  
 **ON** Backpack (Uid);

**表3**

**CREATE TABLE** Materials  
(  
 MaterialsId **INT AUTO\_INCREMENT**,  
 Ma\_name **VARCHAR**(50) **NOT NULL**,  
 Ma\_INTroductiON **VARCHAR**(255) **NOT NULL**,  
 Ma\_icon **VARCHAR**(255) **NOT NUL**L,  
 **CONSTRAINT** Materials\_pk  
 **PRIMARY KEY** (MaterialsId)  
);  
**CREATE UNIQUE INDEX** Materials\_Ma\_name\_uindex  
 **ON** Materials (Ma\_name);

**表4**

**CREATE TABLE** Activity  
(  
 AcID **INT AUTO\_INCREMENT**,  
 Ac\_name **VARCHAR**(50) **NOT NULL**,  
 Start\_time **DATE NOT NULL**,  
 End\_time **DATE** **NOT NULL**,  
 **CONSTRAINT** Activity\_pk  
 **PRIMARY KEY**(AcID)  
);

### 3.3.4数据库ER图设计

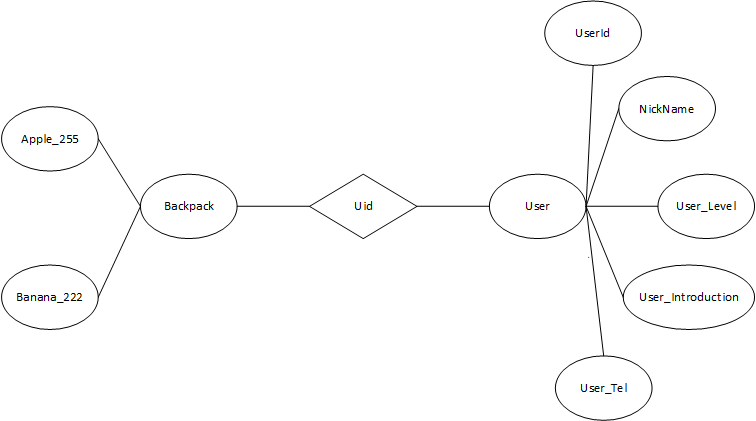
****

图3-3-4.1 BackPack表和User表ER图

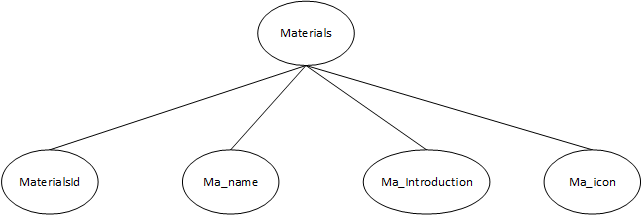
****

图3-3-4.2 Materials表ER图

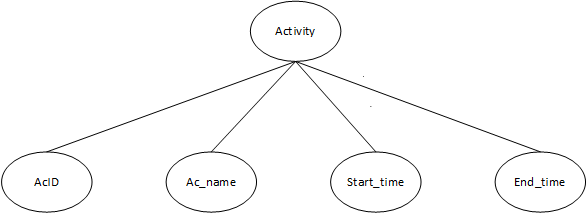
****

图3-3-4.3 Activity表ER图

## 3.4后端设计

### 3.4.1项目结构

本项目采用MVC架构，使用Springboot作为基础框架编写。由于后端业务功能并不复杂，因此省略了Service层，直接使用Controller和Dao交互。其中，Controller层实现前后端交互，Dao层进行与数据库的数据读取存储；Entity负责声明各个实体类，对应数据库中的各张表。

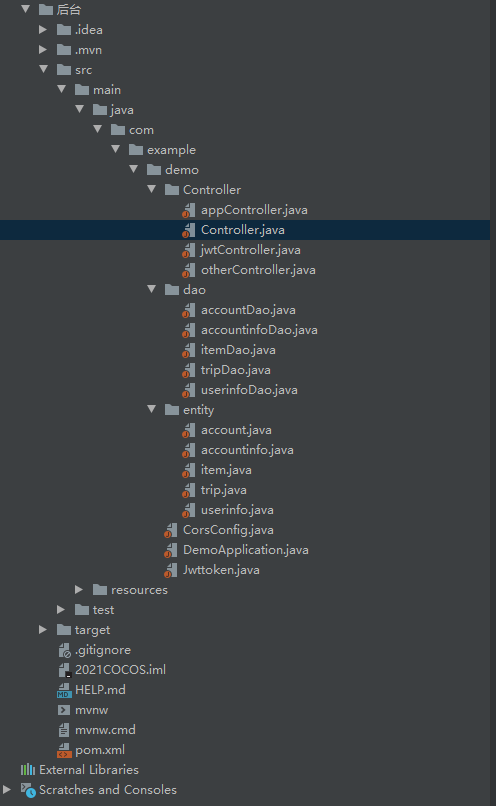


图3-4-1.1 项目结构图

### 3.4.2接口说明

|  |  |
| --- | --- |
| URI | /user/accountinfojson |
| 功能描述 | Jwt验证 |

* 请求内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| userid | String | 用户id |  |
| authHeader | String | 包含token的  请求头 |  |

* 应答内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| list | List | 用户列表 |  |

|  |  |
| --- | --- |
| URI | /updatestate |
| 功能描述 | 更新用户状态 |

* 请求内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| userid | Integer | 用户id |  |
| authHeader | String | 包含token的  请求头 |  |

* 应答内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| CITY\_ID | Integer | 城市编号 |  |
| SERVICE\_ID | Integer | 服务号 |  |
| Status | Integer | 状态 |  |

* 返回码：

|  |  |
| --- | --- |
| Code | Msg |
| 200 | 成功 |
| 500 | 失败 |

|  |  |
| --- | --- |
| URI | /validate |
| 功能描述 | 登陆验证 |

* 请求内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| account | String | 用户名 |  |
| userid | Integer | 用户编号 |  |
| password | String | 密码 |  |

* 应答内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| token | String | Jwt验证token |  |

* 返回码：

|  |  |
| --- | --- |
| Code | Msg |
| 200 | 成功 |
| 500 | 密码错误 |

|  |  |
| --- | --- |
| URI | /updateuserinfo |
| 功能描述 | 更新用户信息 |

* 请求内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| userid | String | 用户编号 |  |
| coins | Integer | 金币 |  |
| growth | Integer | 成长值 |  |

|  |  |
| --- | --- |
| URI | /additem |
| 功能描述 | 获取道具 |

* 请求内容：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 参数描述 | 备注 |
| userid | String | 用户编号 |  |
| amount | Integer | 道具数量 |  |
| id | String | 道具编号 |  |
| Name | String | 道具名称 |  |
| Type | String | 道具类型 |  |

### 3.4.3实体类设计

#### 3.4.3.1用户账号实体——account

用于表示用户账号，含有密码编号（passwordid）、用户id（userid）、账号（account）、密码（password）属性。

#### 3.4.3.2账号信息实体——accountinfo

用于表示用户账号的信息，含有用户id（userid）、金币（coins）、总行程（completed\_travel）、成长值（growth）属性，其中userid为指向account实体的外键。

#### 3.4.3.3物品实体——item

用于表示用户的道具，含有物品id（itemid）、用户id（userid）、名称（name）、成长值（growth）、类型（type）、数量（amount）属性，其中userid为指向account实体的外键。

#### 3.4.3.4行程实体——trip

用于表示用户的行程，含有行程id（tripno）、用户id（userid）、账号（account）、行程状态（status）、城市编号（city\_id）、服务编号（service\_id）、进站口名称（in\_station\_name）、进站口编号（in\_station\_id）、出站口名称（out\_station\_name）、出站口编号（out\_station\_id）、进站时间（in\_time）、行程实际花费（trip\_total\_actual\_fee）属性，其中userid、account均为指向account实体的外键。

#### 3.4.3.5用户信息实体——userinfo

用于表示用户个人信息，含有用户id（userid）、用户姓名（user\_name）、性别（sex）、手机号（mobile\_phone）、职业（profession）、id编号（id\_no）、id类型（id\_type）、生日（birthday）、头像路径（user\_image\_path）、昵称（nickname）、城市编号（city\_id）、城市名（city\_name）、实名登记名称（real\_name\_reg）、实名认证标志（real\_name\_auth）、实名校验标志（real\_name\_chk）、是否开户标记（real\_name\_open），其中userid为指向account实体的外键。

### **3.4.4Jwt验证**

出于对大量用户涌入时，服务器处理并发能力有限的考虑，本项目采用了Jwt验证代替传统的Session认证。基于token的鉴权机制类似于http协议也是无状态的，它不需要在服务端去保留用户的认证信息或者会话信息。这就意味着基于token认证机制的应用不需要去考虑用户在哪一台服务器登录了，这就为应用的扩展提供了便利。

大致流程为：

* 用户使用用户名和密码请求服务器
* 服务器对用户信息进行验证
* 服务器通过验证，并发送给用户一个token
* 客户端储存token，并在请求时附送上这个token
* 服务端验证token值，并返回数据

### 3.4.5用户登陆

大致流程为，用户在前端页面输入用户名、密码，点击提交后，页面发送请求至后端controller层处理。Controller层调用validate方法，调用Dao层获取账号（account）对应的密码，并与用户输入的相比对，若相同，则生成jwtToken加入至响应头中，并返回“成功”信息，状态码为200；若比对失败，则返回“失败”信息，状态码为500。

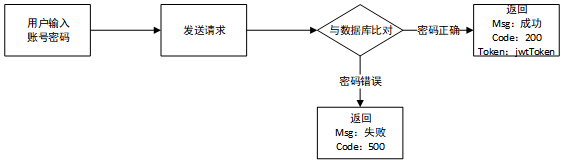


图3-4-5.1 登陆操作流程图

### 3.4.6信息同步

由于该项目运行环境为地铁场景，在地铁中可能会由于信号不好导致无法与服务器良好的连接，无法进行数据交换。因此本项目采用本地缓存，将不良网络状况下用户更新的数据先储存至本地缓存，等到用户转移到网络良好的环境下（如：用户出站）再在后台对用户数据进行服务器端同步。

具体流程为：检测到用户网络状况良好时，前端自动发送/updateuserinfo请求，调用Dao层的findByuserid方法，将临时储存至本地数据库的用户信息调用，并更新当前用户信息，最后保存，即上传至服务器数据库。

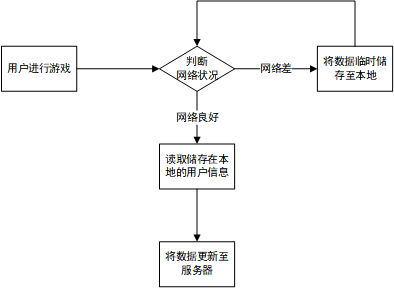


图3-4-6.1 信息同步流程图

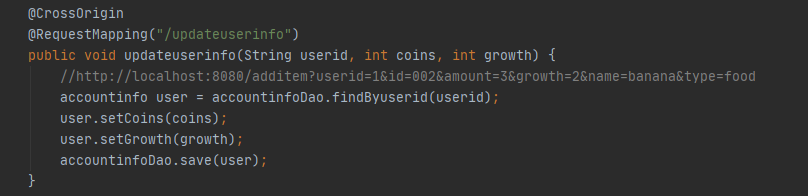


图3-4-6.2 /updateuserinfo请求具体实现代码

### 3.4.7道具更新

每次当用户登陆时，角色包裹里的道具都会从后端进行更新。

当用户登陆成功，并通过jwt的验证时，前端会发送/getitem请求，从服务器数据库中获取用户拥有的道具，并返回给前端，由此加载出各项道具的图片以及数量。

物品更新流程图

图3-4-7.1 物品更新流程图

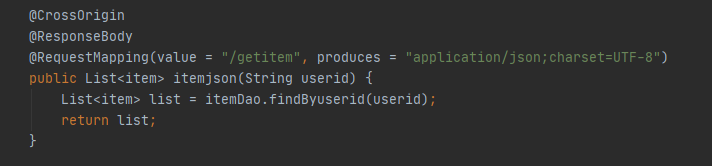


图3-4-7.2 /getitem请求具体实现代码

# 附录

## 4.1附件一 参赛承诺书

