

文件编号：触及云端-SWC2020-20200135

受控状态： 受控 非受控

保密级别： 公司级 部门级 项目级 普通级

采纳标准： CMMI DEV V1.2



面向物联网的“云触”交互新架构

New Human-Computer Interaction Architecture for the internet of Things

项目开发文档

Version 1.0.0

2019.12.28

Written by 触及云端



All Rights Reserved

目录

1	项目概述.....	1
1.1	项目背景.....	1
1.2	项目定位.....	1
1.2.1	应用场景.....	1
1.2.2	目标人群.....	1
1.3	项目方案.....	1
1.4	项目目标.....	2
1.5	项目价值.....	2
2	开发计划.....	2
2.1	最终呈现形式.....	2
2.2	主要功能描述.....	3
2.3	运行环境.....	3
2.4	验收标准.....	3
2.5	关键问题.....	3
2.6	进度安排.....	3
2.7	开发预算.....	3
3	可行性分析.....	4
3.1	技术可行性分析.....	4
3.2	资源可行性分析.....	4
3.3	市场可行性分析.....	4
4	需求分析.....	4
4.1	数据需求.....	4
4.1.1	静态数据.....	4
4.1.2	动态数据.....	5
4.1.3	数据词典.....	5
4.1.4	数据采集.....	6
4.2	功能需求.....	6
4.2.1	NFC 处理功能模块.....	6
4.2.2	共享单车功能模块.....	9
4.2.3	导游机功能模块.....	12
4.2.4	医疗功能模块.....	12
4.3	性能需求.....	15
4.3.1	时间特性.....	15
4.3.2	适应性.....	16
4.4	界面需求.....	16
4.5	接口需求.....	26
4.5.1	硬件接口.....	26
4.5.2	软件接口.....	27
4.6	其他需求.....	27
5	概要设计.....	27

5.1	处理流程.....	27
5.2	总体结构设计.....	29
5.3	功能设计.....	29
5.4	用户界面设计.....	30
5.5	数据结构设计.....	41
5.6	接口设计.....	42
5.6.1	外部接口.....	42
5.6.2	内部接口.....	42
5.7	错误/异常处理设计.....	44
5.7.1	错误/异常输出信息.....	44
5.7.2	错误/异常处理对策.....	44
5.8	系统配置策略.....	44
5.9	系统部署方案.....	44
5.10	其他相关技术与方案.....	44
6	数据库设计.....	44
7	详细设计.....	49
7.1	NFC 处理功能模块.....	49
7.1.1	功能描述.....	49
7.1.2	性能描述.....	49
7.1.3	输入.....	49
7.1.4	输出.....	49
7.1.5	程序逻辑.....	49
7.1.6	限制条件.....	50
7.2	共享单车功能模块.....	50
7.2.1	功能描述.....	50
7.2.2	性能描述.....	50
7.2.3	输入.....	50
7.2.4	输出.....	50
7.2.5	程序逻辑.....	50
7.2.6	限制条件.....	51

1 项目概述

1.1 项目背景

5G 即将普及的今天，物联网迸发出了新的活力，万物互联的未来仿佛就要呼之欲出，但是现有的应用生态还是基于移动互联网来构造，并没有一个明确的方向可以让人们看到万物互联时代的应用场景。面向线下交互场景丰富的物联网时代，现有的基于移动互联网的应用生态，存在这样一些问题，比如服务依托于传统 app，需要用户使用前下载安装；各个服务之间相互隔离，用户需要独立的注册登录各个账号；线下应用的交互方式太过复杂，需要用户主动进行许多步的操作；各个设备之间配对非常麻烦，需要复杂的步骤完成配对；各个厂商的产品之间兼容性极差等等。

面对上述的这些人与物之间交互的问题，一直没有一个非常好的解决方案，所以也就导致了现有人们很难想象在物联网时代人物交互方式。在 2018 年的国家级本科生双创项目——“基于 NFC 技术的小程序新交互方式”中，我们提出了一种新的交互方式，使用 NFC (Near Field Communication, 近距离无线通信) 技术拉起小程序来完成与线下设备的交互动作，在该项目中我们依托这种方式，构建了智慧医疗，智慧交通，智慧旅游等若干项目。都取得了非常不错的效果，用户可以非常轻松的获取各种服务，而不需要反复打开界面，其次由于小程序的优越性，这种交互方式具有天生的兼容性，可以实现一个交互动作获取各种线上服务。我们总结这个项目得出结论，物联网生态下的交互方式不应该是用户主动去使用某种服务，而应该是服务在用户进入某个场景中或者使用某种设备时自动出现在用户的设备上。

并且在现阶段，5G 还尚未普及的今天，线下设备已经越来越多，各种线下设备要求用户完成各种各样的交互动作。这些动作各不相同且非常复杂，用户需要进行的前置操作非常多，现有交互生态的弊端已经逐渐显露出来，急需一种更加成熟的，兼容性更好，操作更加简便的，使用成本更加低的，面向物联网生态的交互架构。这种交互架构应当可以充分体现 5G 交互的快速便捷，并且能非常好的关联各种线下设备，兼容各种品牌的设备。

1.2 项目定位

1.2.1 应用场景

传统的服务需要下载安装 app，且各个服务都需要各自的用户登录，造成交互方式复杂，用户需要进行的操作太多等问题。

1.2.2 目标人群

各个年龄段需要使用这些服务的人群。

1.3 项目方案

针对 app 数量多且需要下载注册的问题，采用轻应用（如各大手机厂商的快应用和各大 app

的小程序等)来代替,以实现无需安装即可使用。

针对轻应用入口深且复杂、不便于用户浏览的问题,采用 NFC 触碰跳转的方式,以实现快速、便捷的调用服务。

1.4 项目目标

1. 完成多个轻应用使用场景的构建。
2. 建立一个将多个服务串联在一起的平台。

1.5 项目价值

(1)变主动为被动,用户体验更好。在线下用户在某一个场景中或者某一智能设备边上时,用户急需某种服务,但是还是需要主动去打开手机,经过多步操作才能打开需要的功能或者服务。用户需要主动的去调用所需要的服务,这对用户来说体验并不好,所以在特定场景中应将用户当前最需要的服务直接推送给用户,不需要用户进行过多的筛选与调用,进而提升用户体验。

(2)封装服务,用户无需担心下载问题。用户的需求是多种多样的,在各个细分领域中有着各种服务的提供商,但是他们之间往往是相对隔离的。这会给用户带来许多逻辑上的冗余。我们希望统一用户的操作逻辑,以这种新的交互模式为基础,构造一个大的服务生态,用户在线下使用标签与设备交互的时候,不用主动去打开应用,程序会主动拉起应用或者云端的小程序。

(3)操作直观缩短学习曲线。将服务信息封装在 NFC 标签中之后,用户只需要将手机使用与标签碰触,就能调用服务,操作简单,老人孩子也能轻松上手,可以让更多的人享受到互联网的便捷。

(4)更低的使用门槛,更大的市场。

目前市场上有许多针对线下的应用服务,他们往往针对的是某个用户基数大,但是使用频率较低的场景。对于用户来说,如果操作逻辑复杂,使用繁琐,会在用户心中设立一个心理门槛,从而阻碍用户使用服务。使用新的交互方式后,这个心理门槛的高度将会大大压缩,从而给线下服务带来更大的用户市场。

2 开发计划

2.1 最终呈现形式

主要在安卓手机上安装一个用来读取 NFC 标签信息和跳转的 app。然后有若干轻应用作为服务,通过手机触碰 NFC 标签来跳转到轻应用打开服务。以共享单车轻应用为例,可以通过触碰相应的 NFC 标签直接打开车锁。

2.2 主要功能描述

安卓 app 可以读写 NFC 标签中的信息和跳转到相应的轻应用。轻应用提供多样服务，比如共享单车开锁等。

2.3 运行环境

安卓手机需支持 NFC 功能，安卓系统版本在 10 以上，同时性能足够支持轻应用运行，网络情况良好。服务端运行在 ubuntu 16 环境下。车锁目前使用树莓派控制。

2.4 验收标准

1. 2~3 个新的应用场景：小程序的形式
2. 完善的交互平台：以网页端以及 app 端的形式
3. 各个应用场景间完善的交互逻辑

2.5 关键问题

因为是自动进行跳转，所以需要针对安全进行一定设计。在正式环境中，NFC 标签采用不可篡改的类型，同时 app 支持自行设置是否跳转服务，未设置的服务无法跳转。然后针对所有的服务都是经过审核之后再加入。

2.6 进度安排

2020/01/9~2020/01/22

讨论确定下一部分需要实现的应用场景，以及交互平台的具体细节。

2020/01/22~2020/02/25

开始构建新的应用场景，编写相应的小程序，并反复讨论交互流程；于此同时，开始着手构建交互平台的后端架构

2020/02/25~2020/3/10

开始正式构建交互平台的开发工作，同时从网页端以及安卓端进行编写。

2020/03/11~2020/03/30

完成平台搭建，测试其数据收集与梳理功能，并不断完善

2.7 开发预算

总额： 8900 元。

交通费：2000 元，到相关高校，研究所，公司进行沟通交流所需交通费用

图书费：800 元，购买相关专业书籍

耗材：4000 元，NFC 芯片标签、NFC 开发套件、共享单车车锁、闸机开发套件等相关套件。

资料费：600 元 复印、扫描相关文件

服务器：2000 元 相关服务器以及线上服务费用

相关设计素材费用：500 元

3 可行性分析

3.1 技术可行性分析

技术上 NFC 技术已经较为成熟，而且安卓系统对其支持很好，可以简单的完成对 NFC 标签的读写。同时可以再触碰的时候自动发出相关事件，app 获取相应的事件进行处理就行了。同时轻应用再厂商方面有提供 sdk，并且有相应的跳转接口，这样跳转轻应用的功能就没有什么问题。车锁方面目前使用树莓派控制，树莓派有 GPIO 接口，可以轻松的通过 Python 代码控制相应的硬件设备，同时自带 wifi 模块，方便进行联网调试。

3.2 资源可行性分析

开发难度不大，我们四位开发人员在计划时间内可以完成。

3.3 市场可行性分析

一张 NFC 标签的价格仅几毛钱，量大的话成本更低，该方案使用的场景很多，市场大。

4 需求分析

4.1 数据需求

4.1.1 静态数据

序号	对象	类型	来源表	说明
1	author	文本	travel	文章的作者
2	content	文本	travel	文章的内容
3	name	文本	travel	文章的名字
4	guideMachineId	文本	travel	文章的 id
5	createDate	日期/时间	travel	文章的创建时间

6	poster	文本	travel	图片的 url 链接
7	src	文本	travel	语音解说的链接
8	title	文本	travel	文章的标题
9	department_name	文本	department	科室名字
10	department_addr	文本	department	科室地址
11	hospital_name	文本	department	医院名字
12	doctor_name	文本	doctor	医生名字
13	department_cost	数值	department	挂号费
14	drug_name	文本	drug	药品的名字
15	drug_price	数值	drug	药品单价

4.1.2 动态数据

序号	对象	类型	来源表	说明
1	symptom	文本	medicalRecord	症状描述
2	medicalRecord_content_first	文本	medicalRecord	初步诊断内容
3	medicalRecord_content_finally	文本	medicalRecord	最终诊断内容
4	general	文本	medicalRecord	概括
5	active	文本	medicalRecord	是否活跃状态
6	examination_result	文本	examination_report	检查结果

4.1.3 数据词典

- 名称: 导游机信息表

描述: 记录了导游机内容的基本信息

来源: 管理员从网上公开的旅游信息进行加工

去处: 数据库中存储的导游机信息

组成: 文章的作者+文章的内容+文章的名字+文章的 id+文章的创建时间+文章图片的 url 链接+文章的音频链接
- 名称: 科室信息

描述: 医院的科室信息

来源: 管理员根据医院提供的信息进行加工

去除: 智慧医疗系统的科室信息表

组成: 科室名字+科室地址+医院名字+医生名字
- 名称: 医生信息

描述: 医院的医生信息

来源: 管理员根据医院提供的信息进行加工

去除: 智慧医疗系统的医生信息表

- 组成: 医生名字+医生简介
4. 名称: 药品信息
描述: 医院的药品信息
来源: 管理员根据医院提供的信息进行加工
去除: 智慧医疗系统的药品信息表
组成: 药品名字+药品信息+药品价格+药品库存
5. 名称: 病历信息
描述: 病人的病历信息
来源: 管理员根据医院和病人提供的信息进行加工
去除: 智慧医疗系统的病历信息表
组成: 病人名字+症状描述+初步诊断内容+最终诊断内容

4.1.4 数据采集

对于导游机相关的数据是从网上找的公开的资源，然后手动添加到数据库中。
智慧医疗相关的数据根据医院提供的信息进行加工。

4.2 功能需求

4.2.1 NFC 处理功能模块

表 1 NFC 处理功能模块描述

功能模块	功能	功能描述	优先级
NFC 处理模块	读取标签信息	读取 NFC 标签中的信息	2
	读取二维码信息	调用系统相机扫描二维码并读取信息	1
	写入标签信息	将相应信息写入 NFC 标签	2
	控制服务开关	控制相应的服务是否能够跳转	1
	跳转轻应用	根据相应信息跳转到相应的轻应用	2

表 2 读取 NFC 标签信息用例规约

用例名称	读取 NFC 标签信息
功能简述	利用手机的 NFC 功能读取 NFC 标签中的信息
用例编号	TOUCHEZ_001.1
执行者	用户
前置条件	手机支持 NFC 功能且安装了该 app

后置条件	读取信息成功
涉众利益	用户希望快速准确的进行操作
基本路径	1. 手机触碰到 NFC 标签 2. 系统自动识别调用 app 读取信息 3. App 读取信息成功
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	系统未识别到相应的 app 可能会跳出其它与 NFC 相关的 app 进行选择。
备注	无

表 3 跳转轻应用用例规约

用例名称	跳转轻应用
功能简述	调用厂商提供的接口跳转到相应的轻应用
用例编号	TOUCHEZ_001.2
执行者	手机
前置条件	手机安装了该 app 且成功读取 NFC 信息
后置条件	跳转轻应用成功
涉众利益	用户希望快速准确的进行操作
基本路径	1. 根据信息解析出要跳转的轻应用和路径等信息 2. App 调用 sdk 提供的接口进行跳转 3. 轻应用跳转成功
扩展路径	2a. 检查 app 上相应服务的开关是否打开 2a1. App 反馈该服务尚未打开 2a2. 返回 1
字段列表	1. 解析的消息=轻应用类型+轻应用 id+轻应用路径+轻应用参数
设计规则	无
未解决的问题	无
备注	无

表 4 写入 NFC 信息用例规约

用例名称	写入 NFC 信息
功能简述	利用手机的 NFC 功能往 NFC 标签中写入信息
用例编号	TOUCHEZ_001.3

执行者	用户
前置条件	手机支持 NFC 功能且安装了该 app
后置条件	信息写入成功
涉众利益	用户希望快速准确的进行操作
基本路径	<ol style="list-style-type: none"> 1. 选择相应的信息 2. 手机触碰 NFC 标签, 选择写入 3. 写入信息成功
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

表 5 读取二维码信息用例规约

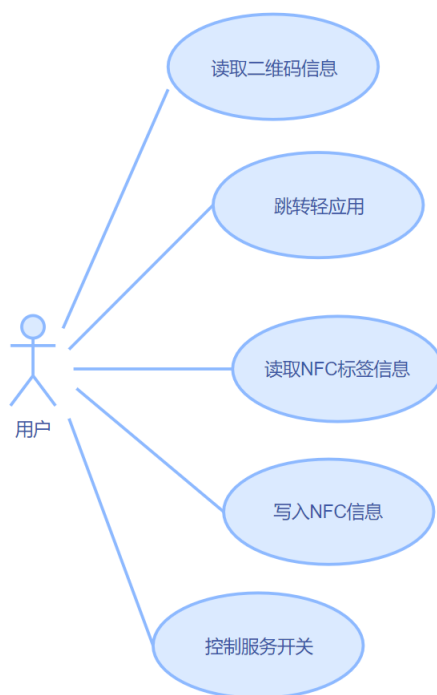
用例名称	读取二维码信息
功能简述	利用手机的摄像头扫描二维码读取二维码中的信息
用例编号	TOUCHEZ_001.4
执行者	用户
前置条件	手机支持摄像头且安装了该 app
后置条件	信息读取成功
涉众利益	用户希望快速准确的进行操作
基本路径	<ol style="list-style-type: none"> 1. 用摄像头对准二维码 2. App 自动识别读取信息 3. 读取信息成功
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

表 6 控制服务开关用例规约

用例名称	控制服务开关
功能简述	在 APP 中选择可以跳转哪些服务
用例编号	TOUCHEZ_001.5
执行者	用户

前置条件	手机安装了该 app
后置条件	设置开关成功
涉众利益	用户希望快速准确的进行操作
基本路径	1. 进入控制服务开关页面 2. 选择服务开启还是关闭
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

图 1 NFC 处理功能用例图



4.2.2 共享单车功能模块

表 7 共享单车功能模块描述

功能模块	功能	功能描述	优先级
共享单车	骑行记录	用户触碰相应的 NFC 标签即可打开车锁，	1

模块		并记录骑行时间、花费等数据，关锁后停止记录	
	定位刷新	在共享单车首页对用户及周边单车定位的刷新	1
	余额充值	在指定共享单车快应用的充值页面对余额进行充值	1

表 8 骑行记录用例规约

用例名称	骑行记录
功能简述	使用快应用对共享单车开锁，并记录骑行时间、花费等数据，关锁后停止记录
用例编号	TOUCHEZ_002.1
执行者	用户
前置条件	手机网络状况良好并且支持快应用开启定位功能
后置条件	记录成功
涉众利益	用户希望快速准确的进行操作
基本路径	<ol style="list-style-type: none"> 1. 打开轻应用之后调用相应的接口打开车锁 2. 记录骑行时间和费用 3. 点击上锁按钮停止记录
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

表 9 定位刷新用例规约

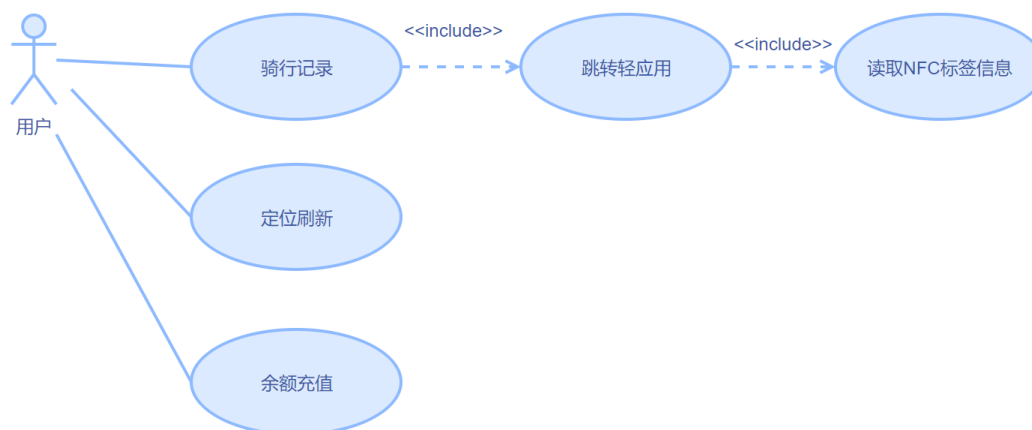
用例名称	定位刷新
功能简述	在共享单车首页对用户及周边单车定位的刷新
用例编号	TOUCHEZ_002.2
执行者	用户
前置条件	手机网络状况良好并且支持轻应用开启定位功能
后置条件	获取位置信息成功
涉众利益	用户希望快速准确的进行操作
基本路径	<ol style="list-style-type: none"> 1. 点击定位按钮

	2. 更新定位信息和共享单车位置信息
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

表 10 余额充值用例规约

用例名称	余额充值
功能简述	在指定共享单车轻应用的充值页面对余额进行充值
用例编号	TOUCHEZ_002.3
执行者	用户
前置条件	手机网络状况良好并且支持轻应用
后置条件	充值成功
涉众利益	用户希望快速准确的进行操作
基本路径	1. 在钱包页面点击充值按钮 2. 选择一个充值金额，点确认充值按钮
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

图 2 共享单车功能用例图



4.2.3 导游机功能模块

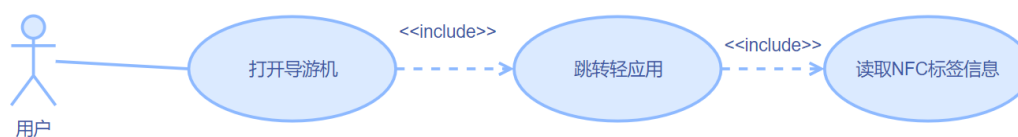
表 11 导游机功能模块描述

功能模块	功能	功能描述	优先级
导游机模块	导游词	用户触碰相应的 NFC 标签跳转后显示相关信息页面并且自动播放导游词	1

表 12 打开导游机用例规约

用例名称	打开导游机
功能简述	打开了指定导游机轻应用的相关页面，显示内容，播放导游词
用例编号	TOUCHEZ_003.1
执行者	手机
前置条件	手机网络状况良好并且有相应的 app
后置条件	打开导游机成功
涉众利益	用户希望快速准确的进行操作
基本路径	1. 根据参数调用接口获取导游词等数据 2. 显示相应的图片和信息，并自动播放导游词
扩展路径	无
字段列表	1. 接口获取的数据=文章的标题+文章的作者+文章的内容+文章的图片url+文章的语音 url
设计规则	无
未解决的问题	无
备注	无

图 3 导游机功能用例图



4.2.4 医疗功能模块

表 13 智慧医疗功能模块描述

功能模块	功能	功能描述	优先级
智慧医疗	挂号	用户触碰相应的 NFC 标签即可在相应的科	1

模块		室挂号，轻应用跳转到挂号界面	
	就诊	用户触碰相应的 NFC 标签即可在医生挂号，轻应用倒转的医生简介界面，pc 上进入诊断界面	1
	检查	用户触碰相应的 NFC 标签即可在检查处排队等待检查	1
	取药	用户触碰相应的 NFC 标签即可在取药处取药	1

表 14 挂号用例规约

用例名称	挂号
功能简述	手机触碰 NFC 标签自动挂号
用例编号	TOUCHEZ_004.1
执行者	用户
前置条件	手机网络状况良好并且有相应的 app
后置条件	挂号成功
涉众利益	用户希望快速准确的进行操作
基本路径	<ol style="list-style-type: none"> 1. 手机触碰相应的 NFC 标签 2. app 读取 NFC 标签信息并跳转到相应的轻应用 3. 轻应用根据跳转信息自动挂号 4. 挂号成功
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

表 15 就诊用例规约

用例名称	就诊
功能简述	手机触碰 NFC 标签自动就诊
用例编号	TOUCHEZ_004.2
执行者	用户
前置条件	<ol style="list-style-type: none"> 1. 手机网络状况良好并且有相应的 app 2. 用户已在相应科室挂号成功

后置条件	就诊进行中
涉众利益	用户希望快速准确的进行操作
基本路径	<ol style="list-style-type: none"> 1. 手机触碰相应的 NFC 标签 2. App 读取 NFC 标签信息并且跳转到相应的轻应用 3. 轻应用根据跳转信息自动就诊 4. 医生 PC 端自动跳转到就诊界面 5. 就诊中
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

表 16 检查用例规约

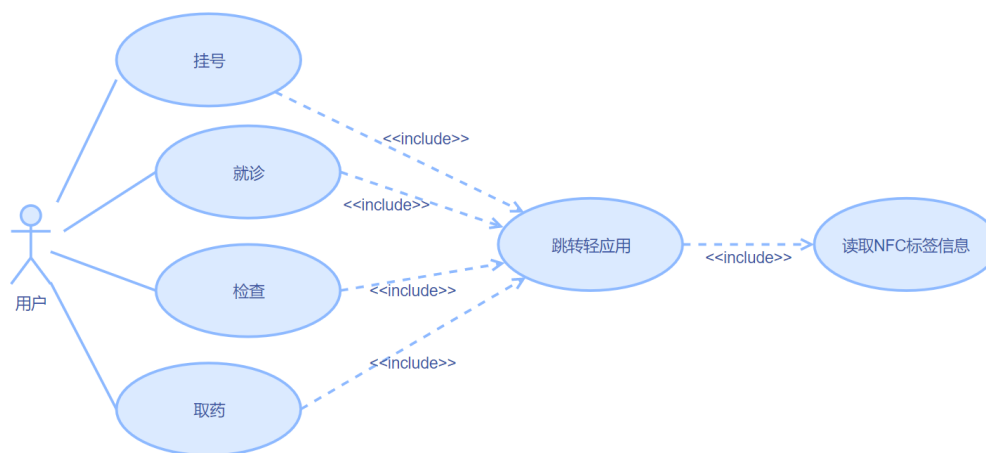
用例名称	检查
功能简述	手机触碰相应的 NFC 标签即可在检查处排队等待检查
用例编号	TOUCHEZ_004.3
执行者	用户
前置条件	<ol style="list-style-type: none"> 1. 手机网络状况良好并且有相应的 app 2. 用户以已经就诊过且医生开取了相关检查
后置条件	进入检查排队状态
涉众利益	用户希望快速准确的进行操作
基本路径	<ol style="list-style-type: none"> 1. 手机触碰相应的 NFC 标签 2. App 读取 NFC 标签信息并且跳转到相应的轻应用 3. 轻应用根据跳转信息自动在相应检查排队 4. 进入检查排队状态
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

表 17 取药用例规约

用例名称	取药
功能简述	用户触碰相应的 NFC 标签即可在取药处取药

用例编号	TOUCHEZ_004.4
执行者	用户
前置条件	1. 手机网络状况良好并且有相应的 app 2. 用户以已经就诊过且医生开取了相关药品
后置条件	取药成功
涉众利益	用户希望快速准确的进行操作
基本路径	1. 手机触碰相应的 NFC 标签 2. App 读取 NFC 标签信息并且跳转到相应的轻应用 3. 轻应用根据跳转信息自动取药 4. 取药成功
扩展路径	无
字段列表	无
设计规则	无
未解决的问题	无
备注	无

图 4 智慧医疗功能用例图



4.3 性能需求

4.3.1 时间特性

从触碰 NFC 标签到打开相应的轻应用应当在 3 秒内完成。

从触碰 NFC 标签到打开车锁应当在 5 秒内完成。

4.3.2 适应性

在不同品牌的安卓手机中都应当适用。

4.4 界面需求

快应用共享单车解锁界面：从 APP 跳转进来自动解锁



请稍等, 正在开锁中



取消开锁

开锁成功, 可以开始骑行

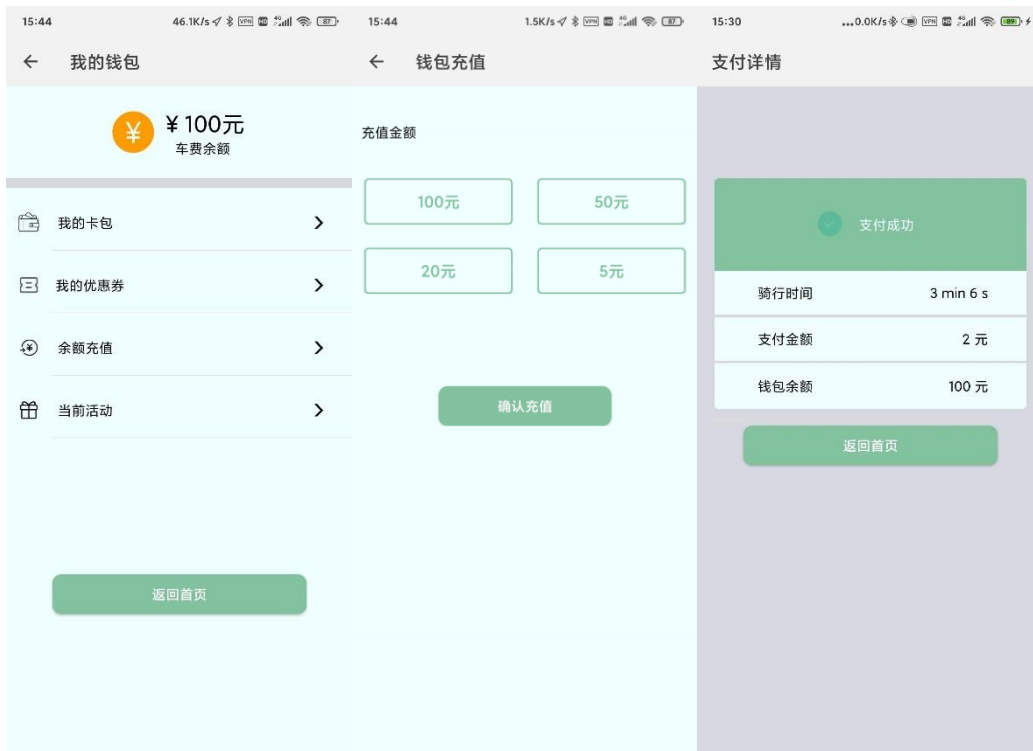


查看骑行详情

快应用共享单车首页, 骑行中骑行详情界面:



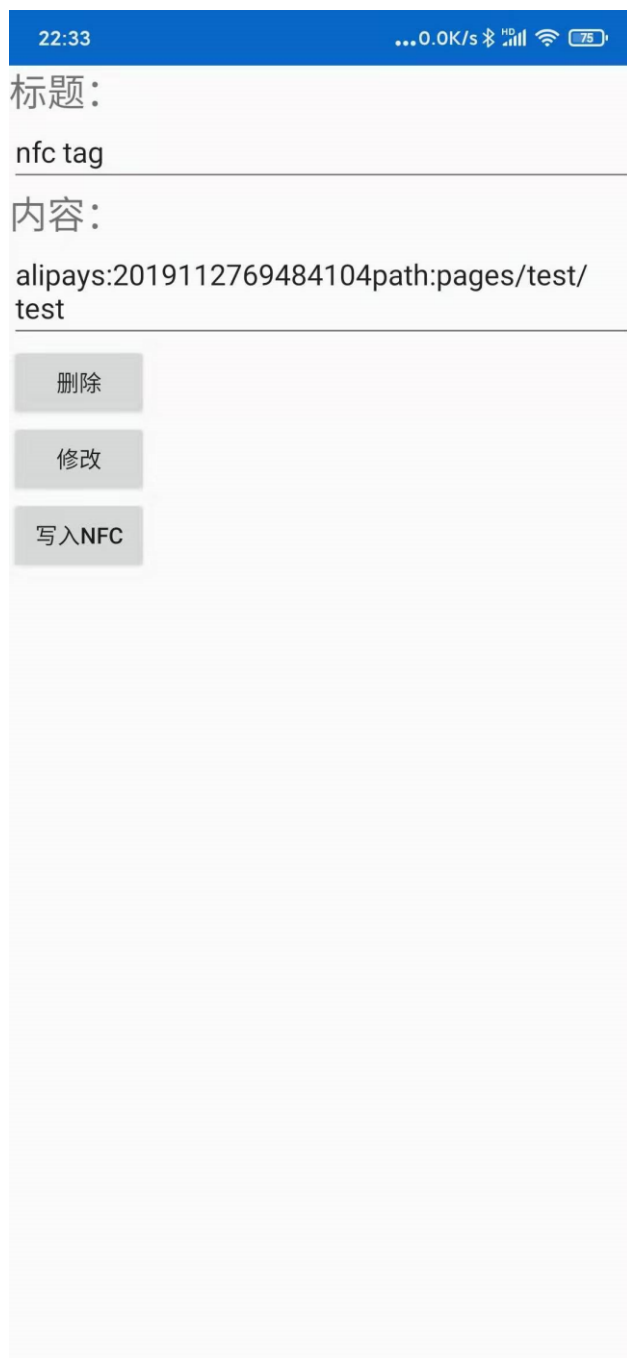
快应用共享单车钱包和充值和支付界面:



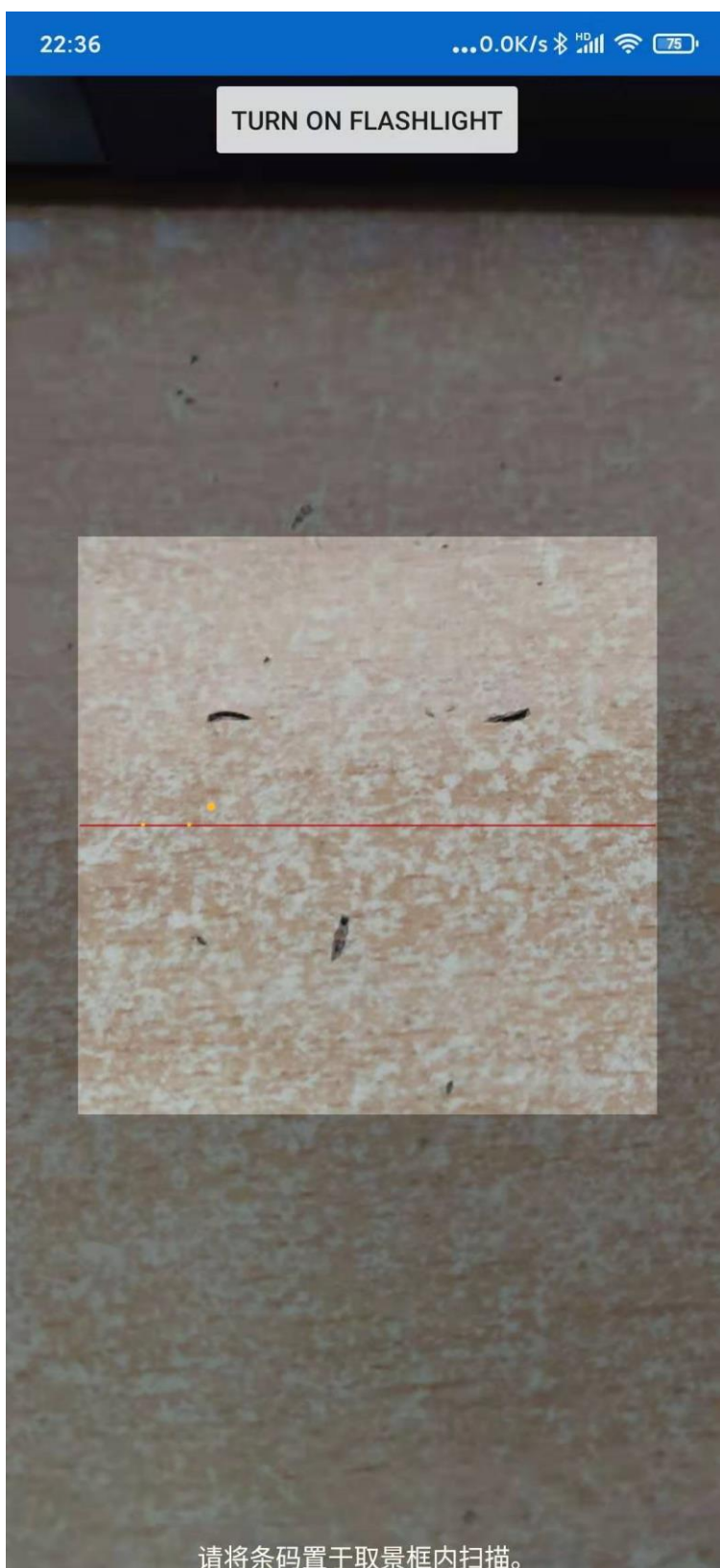
安卓 app 界面：显示服务的名称，旁边的开关用于控制服务是否启用。
右上角的二维码图标用于开启扫描二维码功能。



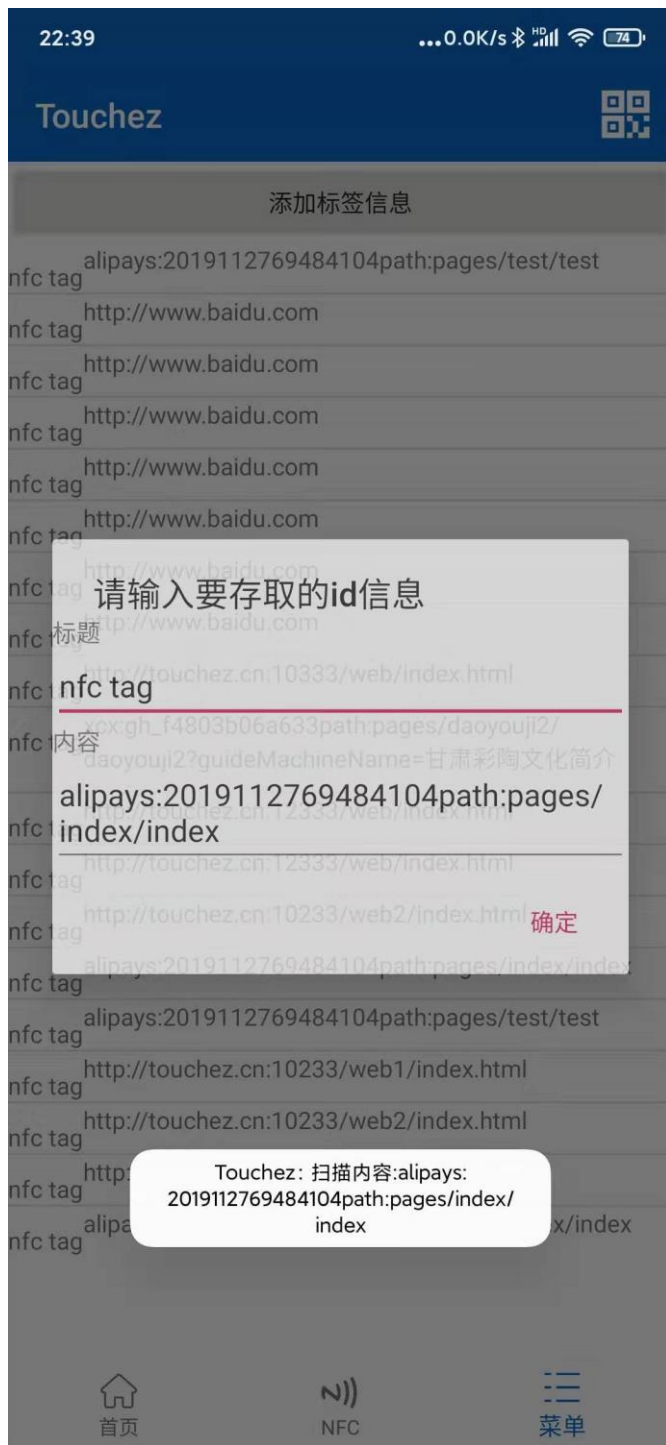
安卓 app 写入 NFC 标签信息界面：修改内容框内的东西，然后点击修改，再点击写入 NFC 按钮即可将信息写入 NFC 标签。



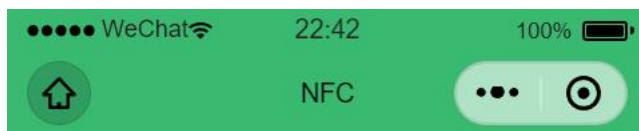
安卓 app 扫描二维码界面:



安卓 app 扫描二维码成功界面：点击确定即可存储在手机内，点击添加标签信息按钮将触发相同页面用于存储相应信息。



导游机界面图：分为标题，图片和内容，同时会自动播放解说词。

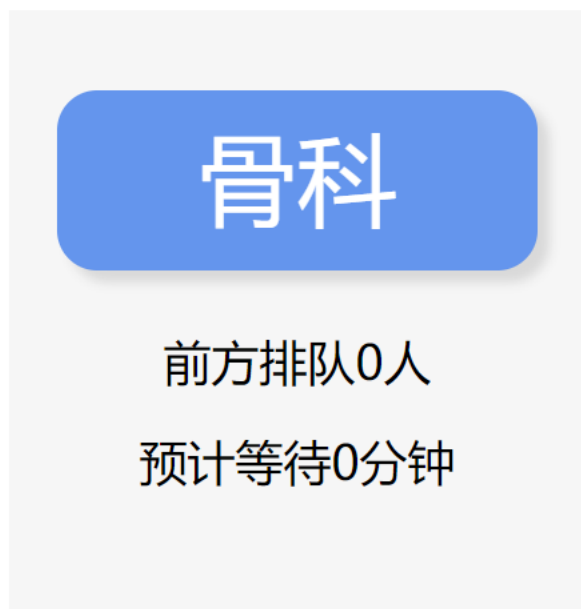
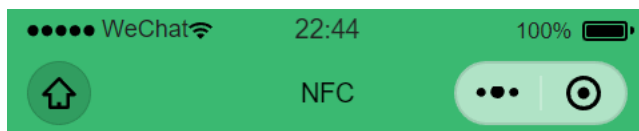


安特生考察路线图

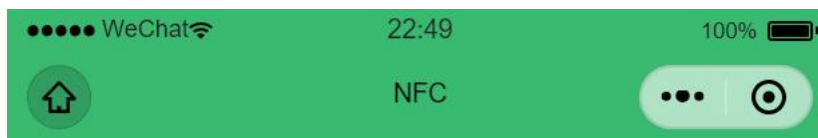


安特生，1874年出生于瑞典的Kinsta。1901年毕业于乌普萨拉大学，取得地质学专业的博士学位，并从此开始了他的学者生涯。他先后两次参加了南极考察活动，从中结识了后来对其考古生涯有着重要影响的人物，同时在这两次南极考察中安特生所表现出来的专业素养和组织能力，更使其成为名噪一时的地质学家，在此期间他所主编和编写的《世界铁矿资源》和《世界煤矿资源》两本调查集更是安特生在地质学研究上硕果累累的结晶。

智慧医疗挂号界面图：显示科室和等待人数与预计等待时间。



智慧医疗就诊界面：显示当前医生的简介。



医生简介



医生A

科室：骨科

主任医师

医生id : 45482648

长期从事骨外科的临床诊疗和研究，擅长骨与软组织肿瘤、复杂的关节周围骨折韧带损伤、老年人骨折、肢体畸形矫正等各类困难的骨与关节疾患。

正在就诊。。。。

智慧医疗检查界面：显示检查的 id，时间，地点，费用，部位，样片，结论等



智慧医疗取药界面: 取药的药品界面, 包括药品名字, 药品数量, 药品价格等。



4.5 接口需求

4.5.1 硬件接口

树莓派的 GPIO 接口，使用 Python 的 GPIO 库进行调用。

4.5.2 软件接口

各家厂商的调用轻应用的接口，如微信小程序的调用接口。根据其厂商提供的安卓平台的 sdk 进行调用。

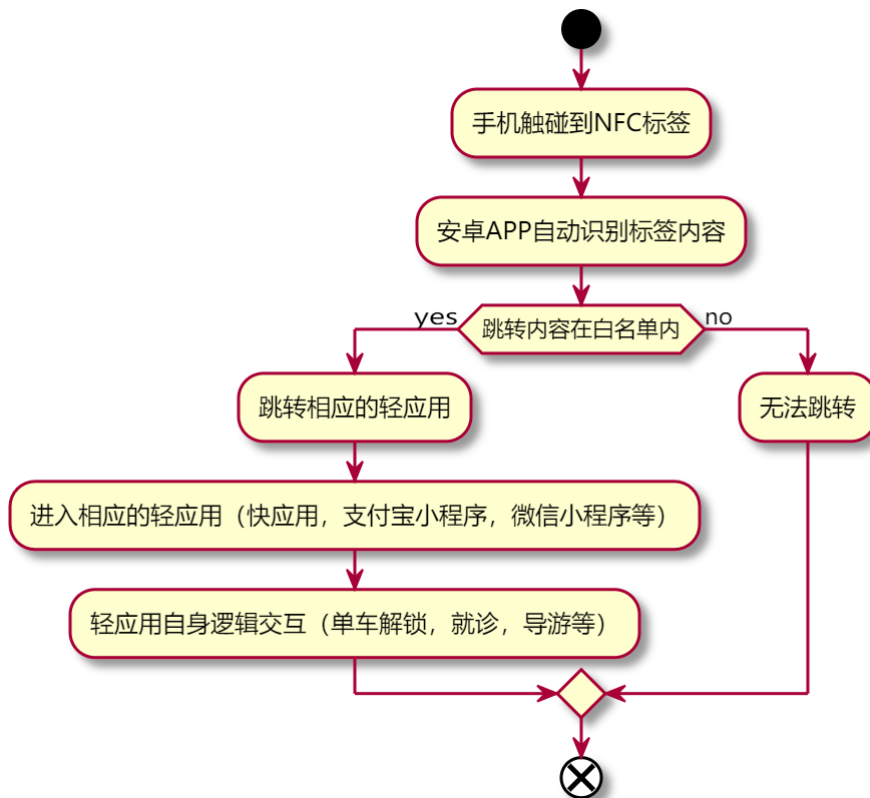
4.6 其他需求

出于安全方面的考虑，在正式环境下应当采用不可修改的 NFC 标签，同时配合 app 上的服务开关以及云端服务审核后接入的机制来保证跳转到的是安全的轻应用。

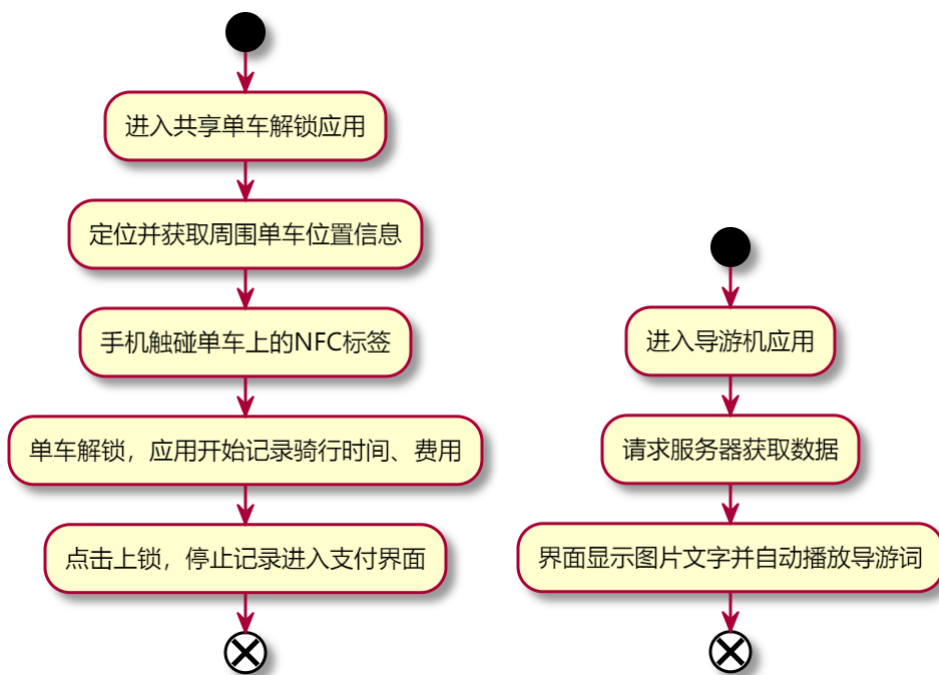
5 概要设计

5.1 处理流程

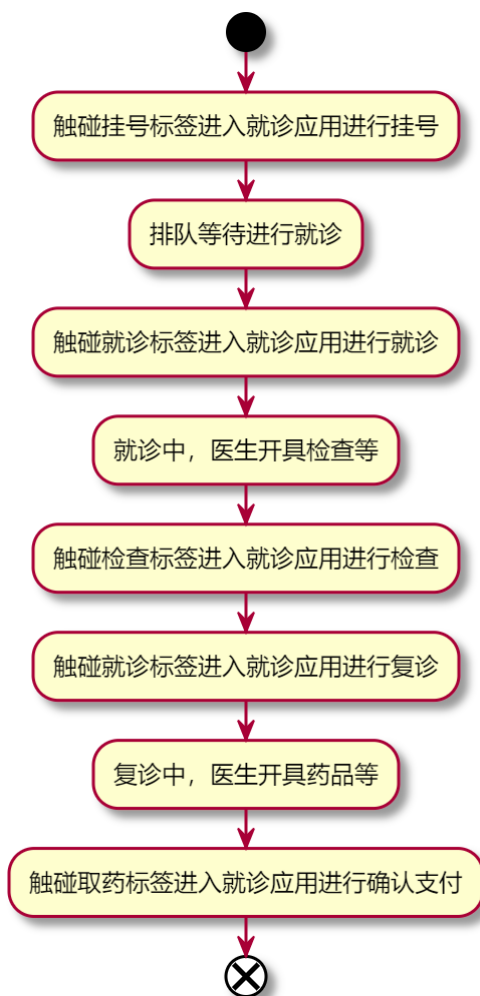
下图为系统整体流程图：



主要就是手机触碰到 NFC 标签后可以自行跳转到定义的轻应用。
以下分别为共享单车解锁，导游机流程图：

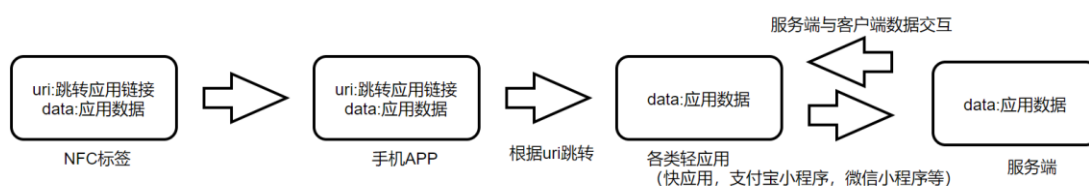


以下为就诊应用的流程图:



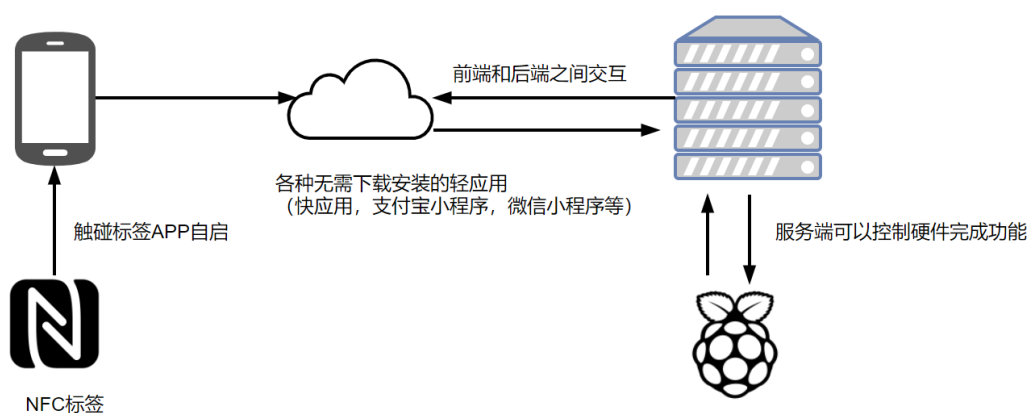
需要注意的是项目不仅限于这几个应用，可以在整体逻辑流程图中扩展其中的应用。

以下为数据流图:



5.2 总体结构设计

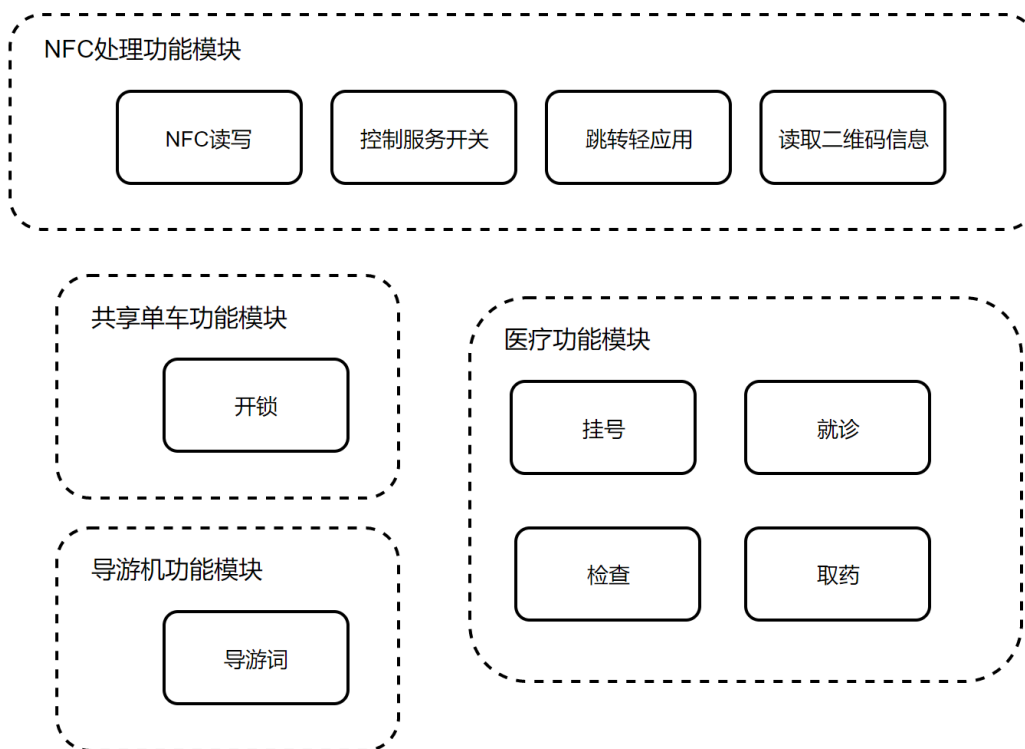
项目架构图如下所示:



主要就是手机触碰 NFC 标签然后自动启动 APP 来读取信息进行跳转, 跳转到相应的轻应用。之后就是轻应用和服务端之间的交互, 同时服务端还可以控制硬件来实现某些功能 (比如打开车锁)。

5.3 功能设计

功能模块结构图如下图所示, 具体的功能可见 4.2 中所示, 其中 NFC 处理功能模块其实就是这个项目的底层模块, 在此基础上搭配各种业务功能模块提供简单快速的使用体验。



5.4 用户界面设计

快应用共享单车解锁界面:



请稍等, 正在开锁中



取消开锁

开锁成功, 可以开始骑行

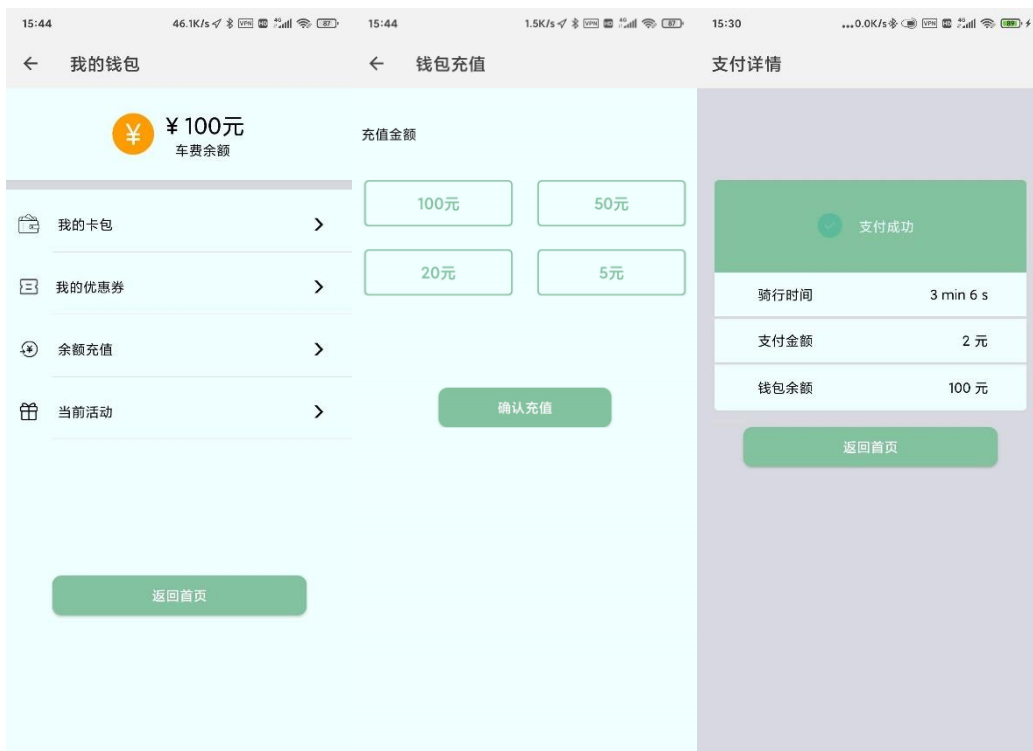


查看骑行详情

快应用共享单车首页, 骑行中界面:



快应用共享单车钱包和充值和支付界面:



安卓 app 界面:

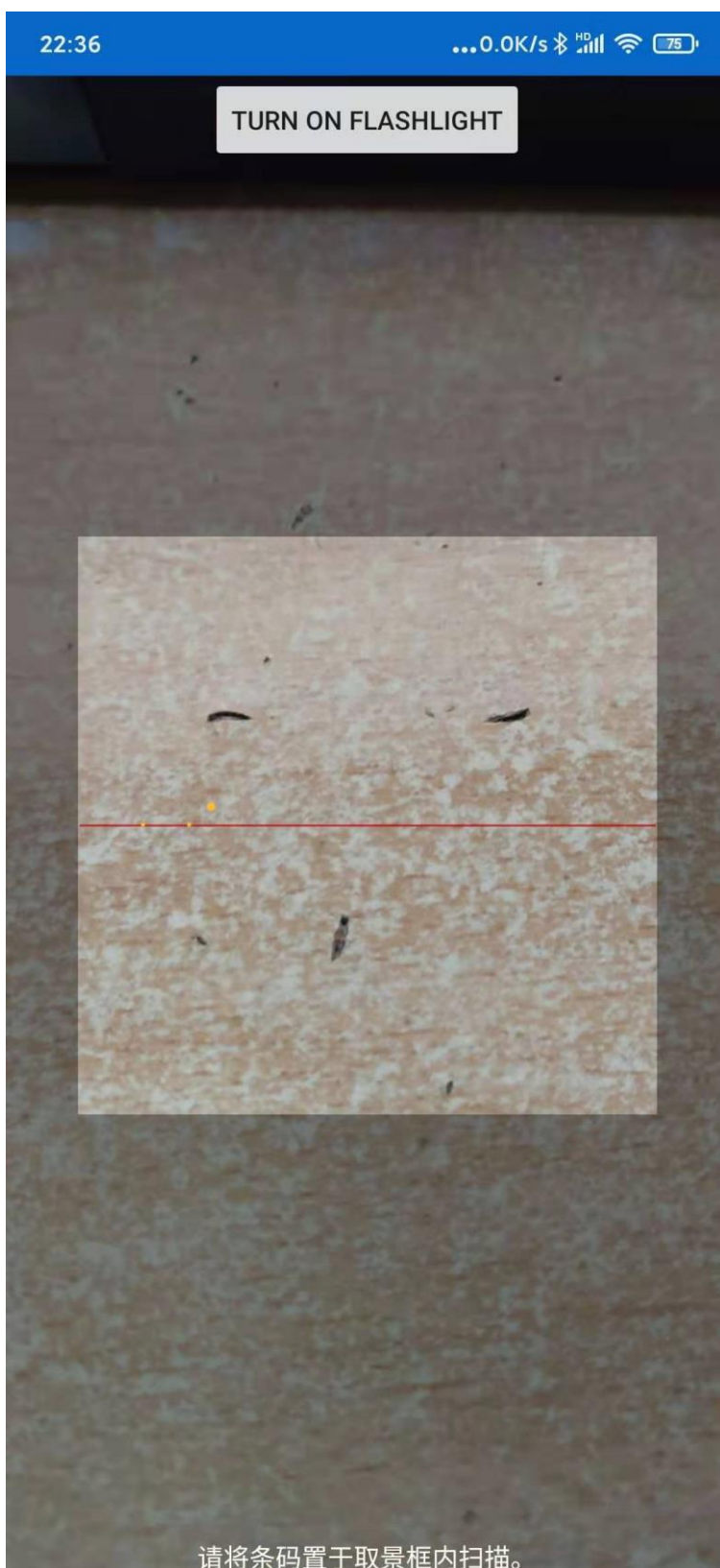
右上角的二维码图标用于开启扫描二维码功能。



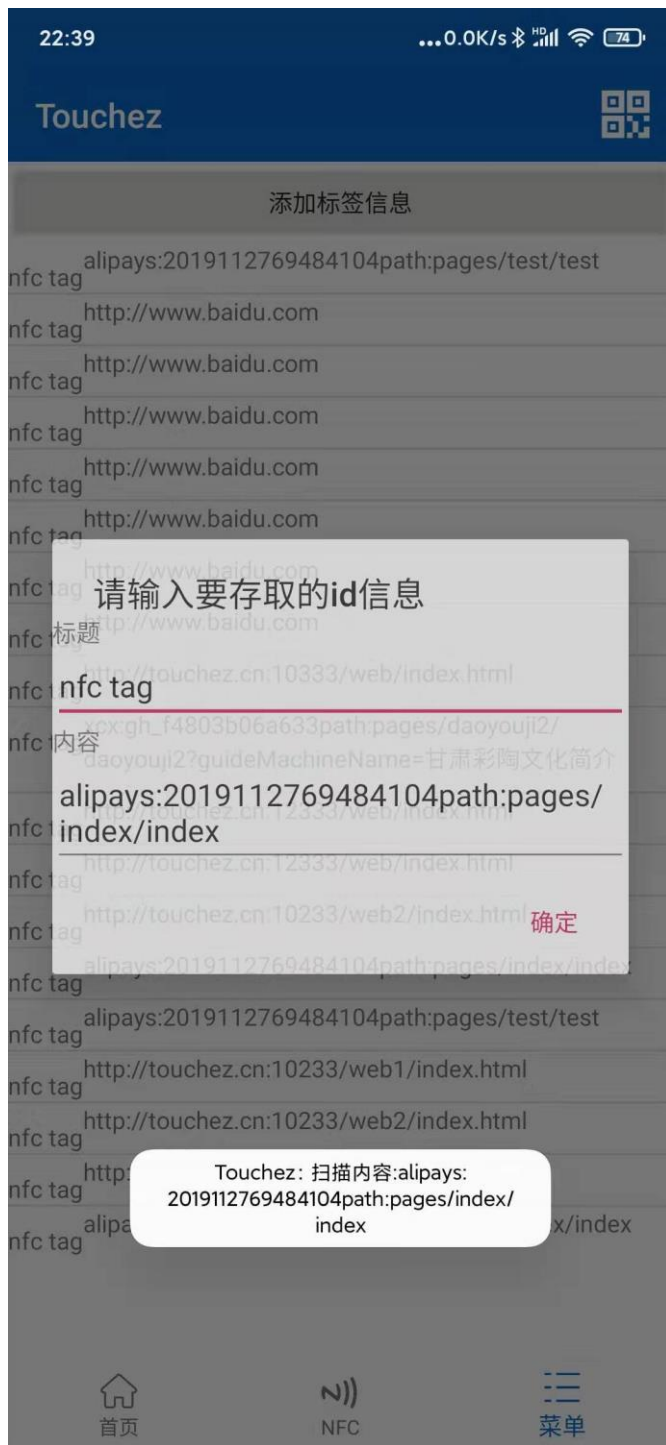
安卓 app 写入 NFC 标签信息界面：修改内容框内的东西，然后点击修改，再点击写入 NFC 按钮即可将信息写入 NFC 标签。



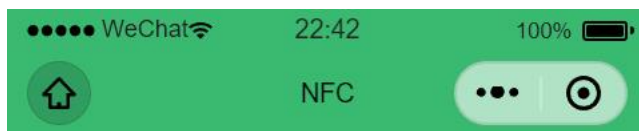
安卓 app 扫描二维码界面:



安卓 app 扫描二维码成功界面：点击确定即可存储在手机内，点击添加标签信息按钮将触发相同页面用于存储相应信息。



导游机界面图：分为标题，图片和内容，同时会自动播放解说词。

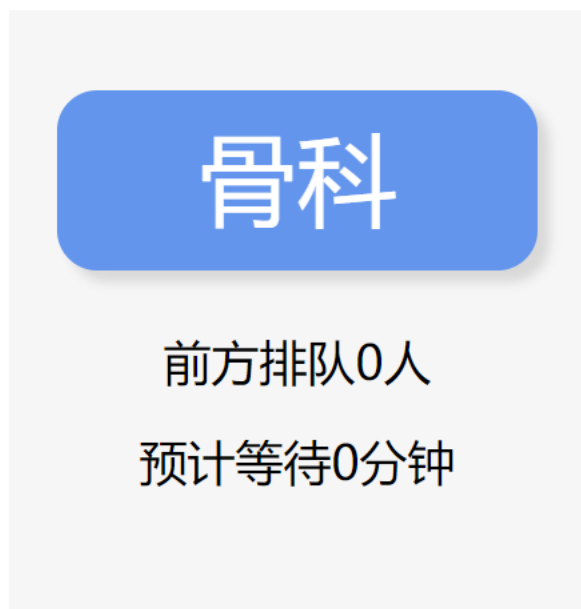
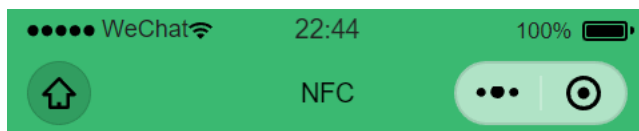


安特生考察路线图

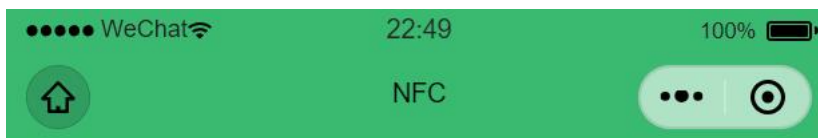


安特生，1874年出生于瑞典的Kinsta。1901年毕业于乌普萨拉大学，取得地质学专业的博士学位，并从此开始了他的学者生涯。他先后两次参加了南极考察活动，从中结识了后来对其考古生涯有着重要影响的人物，同时在这两次南极考察中安特生所表现出来的专业素养和组织能力，更使其成为名噪一时的地质学家，在此期间他所主编和编写的《世界铁矿资源》和《世界煤矿资源》两本调查集更是安特生在地质学研究上硕果累累的结晶。

智慧医疗挂号界面图：显示科室和等待人数与预计等待时间。



智慧医疗就诊界面：显示当前医生的简介。



医生简介



医生A
科室：骨科
主任医师
医生id : 45482648

长期从事骨外科的临床诊疗和研究，擅长骨与软组织肿瘤、复杂的关节周围骨折韧带损伤、老年人骨折、肢体畸形矫正等各类困难的骨与关节疾患。

正在就诊。。。。

智慧医疗检查界面：显示检查的 id，时间，地点，费用，部位，样片，结论等



智慧医疗取药界面: 取药的药品界面, 包括药品名字, 药品数量, 药品价格等。



5.5 数据结构设计

针对存放在 nfc 标签中的数据设计的结构为<type>:<id>path:<path>

其中 type 为应用类型如微信小程序为 xcx, 支付宝小程序为 alipay, 快应用为 hap。Id 为对应的 appId 或者包名, path 为跳转的路径。

5.6 接口设计

5.6.1 外部接口

现在暂无提供给外部的接口。

5.6.2 内部接口

共享单车模块的接口如下所示:

打开共享单车车锁接口: (现在车锁不在身边, 默认返回成功)

url: <https://touchez.cn:8090/pyapi/openlock>

参数: 无

方法: GET

返回值:

```
{
  "code": 0,          //返回信息状态码
  "data": "success", //实际的返回数据
  "msg": "success"   //返回状态信息
}
```

在访问了服务器的接口后, 服务器通过车锁与之相连的 websocket 连接通信, 打开车锁。

导游机模块使用接口如下所示:

获取导游机信息接口:

url: <https://touchez.cn:8090/api/v1/guideMachine?guideMachineName=<name>>

参数: name (如: 蒙娜丽莎的微笑简介)

方法: GET

返回值:

```
{
  "code": 0,
  "data": [
    {
      "author": "", //作者
      "content": "", //内容
      "createDate": "", //创建时间
      "guideMachineId": "", //导游机 id
      "name": "", //数据的名称, 也是查询时用的名称
      "poster": "", //图片链接
      "src": "", //音频链接
      "title": "" //标题
    }
  ],
  "msg": "success"
}
```

医疗模块接口较多，均为常用的 CRUD 接口，下面以获取病例信息接口为例：

获取病例信息接口：

url: `http://127.0.0.1:18090/medicalRecord?medicalRecordId=<id>`

参数: id (如: 1)

方法: GET

返回值:

```
{
  "code": 0,
  "data": {
    "examinationOrder": [ //检查信息
      {
        "examinationOrderId": 1,
        "examinationType": "exsanguinate"
      },
      {
        "examinationOrderId": 2,
        "examinationType": "ct"
      },
      {
        "examinationOrderId": 3,
        "examinationType": "xray"
      }
    ],
    "medicalrecord": { //病历信息
      "createDate": "2019-06-02 21:40:27",
      "departmentId": 1,
      "doctorId": 1,
      "general": "高血压", //用于显示在外部的简要名字
      "medicalrecordContentFinally": "最终诊断高血压", //最终诊断结果
      "medicalrecordContentFirst": "初步诊断高血压", //初步诊断结果
      "medicalrecordId": 1,
      "symptom": "头痛、头晕、注意力不集中", //症状
      "userId": 1
    }
  },
  "msg": "success"
}
```


5.7 错误/异常处理设计

5.7.1 错误/异常输出信息

安卓中会通过 Toast 显示部分错误信息，格式为 Touchez:<error>。
接口部分会返回对应的状态码和错误信息。

5.7.2 错误/异常处理对策

出现错误时，提示错误信息并且进入不可用状态。

5.8 系统配置策略

对于服务器仅开放需要使用的端口。

5.9 系统部署方案

开锁的树莓派上部署 python 环境运行相关代码。
服务器为 Ubuntu，部署 java 环境运行相关代码。

5.10 其他相关技术与方案

无

6 数据库设计

数据库表结构如下所示：

```
CREATE TABLE IF NOT EXISTS `user`(  
    `user_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '用户 id',  
    `user_name` VARCHAR(100) NOT NULL DEFAULT '' COMMENT '用户名字',  
    `uid` varchar(128) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '' COMMENT '用户 openid',  
    `uname` varchar(40) COLLATE utf8mb4_unicode_ci DEFAULT '' COMMENT '用户微信名',  
    `user_gender` ENUM('男','女','保密') DEFAULT '保密' COMMENT '用户性别',  
    `user_age` INT NOT NULL DEFAULT 20 COMMENT '用户年龄',  
    `uavatar` varchar(256) COLLATE utf8mb4_unicode_ci DEFAULT NULL COMMENT '用户头像',  
    `skey` varchar(128) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '' COMMENT '用户登录态标识',
```

```
`sessionkey` varchar(128) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT ''
COMMENT '微信登录态标识',
`user_balance` DECIMAL(10,2) DEFAULT '0.00' COMMENT '用户余额',
PRIMARY KEY ( `user_id` )
)ENGINE=InnoDB CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
CREATE TABLE IF NOT EXISTS `department`(
  `department_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '科室 id',
  `department_name` VARCHAR(100) DEFAULT NULL COMMENT '科室名字',
  `department_addr` VARCHAR(100) DEFAULT NULL COMMENT '科室地址',
  `hospital_id` BIGINT(20) NOT NULL COMMENT '医院 id',
  `hospital_name` VARCHAR(100) DEFAULT NULL COMMENT '医院名字',
  `department_cost` DECIMAL(10,2) DEFAULT '0.00' COMMENT '挂号费',
  PRIMARY KEY ( `department_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `doctor`(
  `doctor_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '医生 id',
  `doctor_name` VARCHAR(100) DEFAULT NULL COMMENT '医生名字',
  `department_id` BIGINT(20) NOT NULL COMMENT '科室 id',
  PRIMARY KEY ( `doctor_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `guahao`(
  `guahao_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '挂号 id',
  `user_id` BIGINT(20) NOT NULL COMMENT '用户 id',
  `department_id` BIGINT(20) NOT NULL COMMENT '科室 id',
  `department_name` VARCHAR(100) DEFAULT NULL COMMENT '科室名字',
  `doctor_id` BIGINT(20) NOT NULL COMMENT '医生 id',
  `create_date` DATETIME DEFAULT NULL COMMENT '挂号的时间',
  `end_date` DATETIME DEFAULT NULL COMMENT '自动结束的时间',
  `state` INT(1) DEFAULT 1 COMMENT '是否有效',
  PRIMARY KEY ( `guahao_id` ),
  UNIQUE KEY `user_department` ( `user_id`,`department_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE IF NOT EXISTS `exsanguinate`(
  `exsanguinate_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '抽血 id',
  `user_id` BIGINT(20) NOT NULL COMMENT '用户 id',
  `exsanguinate_time` DATETIME DEFAULT NULL COMMENT '抽血时间',
  `exsanguinate_addr` VARCHAR(100) DEFAULT NULL COMMENT '抽血地址',
  `exsanguinate_cost` DECIMAL(10,2) DEFAULT '0.00' COMMENT '抽血项目费用',
  `exsanguinate_report` VARCHAR(100) DEFAULT NULL COMMENT '抽血报告',
  `report_time` DATETIME DEFAULT NULL COMMENT '报告生成时间',
```

```
`exsanguinate_attr1` VARCHAR(100) DEFAULT NULL COMMENT '生化指标 1',
`exsanguinate_attr2` VARCHAR(100) DEFAULT NULL COMMENT '生化指标 2',
`exsanguinate_attr3` VARCHAR(100) DEFAULT NULL COMMENT '生化指标 3',
PRIMARY KEY ( `exsanguinate_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `xray`(
  `xray_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT 'X 光 id',
  `user_id` BIGINT(20) NOT NULL COMMENT '用户 id',
  `xray_time` DATETIME DEFAULT NULL COMMENT '检验时间',
  `xray_addr` VARCHAR(100) DEFAULT NULL COMMENT '检验地址',
  `xray_cost` DECIMAL(10,2) DEFAULT '0.00' COMMENT '检验项目费用',
  `xray_report` VARCHAR(100) DEFAULT NULL COMMENT '检验报告',
  `report_time` DATETIME DEFAULT NULL COMMENT '报告生成时间',
  `xray_img` VARCHAR(100) DEFAULT NULL COMMENT '图片 url',
  `xray_part` VARCHAR(100) DEFAULT NULL COMMENT '检验部位',
  PRIMARY KEY ( `xray_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `ct`(
  `ct_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT 'CT 的 id',
  `user_id` BIGINT(20) NOT NULL COMMENT '用户 id',
  `ct_time` DATETIME DEFAULT NULL COMMENT '检验时间',
  `ct_addr` VARCHAR(100) DEFAULT NULL COMMENT '检验地址',
  `ct_cost` DECIMAL(10,2) DEFAULT '0.00' COMMENT '检验项目费用',
  `ct_report` VARCHAR(100) DEFAULT NULL COMMENT '检验报告',
  `report_time` DATETIME DEFAULT NULL COMMENT '报告生成时间',
  `ct_img` VARCHAR(100) DEFAULT NULL COMMENT '图片 url',
  `ct_part` VARCHAR(100) DEFAULT NULL COMMENT '检验部位',
  PRIMARY KEY ( `ct_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `examinationOrder`(
  `examinationOrder_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '检查项目流水 id',
  `user_id` BIGINT(20) NOT NULL COMMENT '用户 id',
  `medicalRecord_id` BIGINT(20) NOT NULL COMMENT '病历 id',
  `examination_id` BIGINT(20) NOT NULL COMMENT '某种检查的流水号',
  `examination_type` VARCHAR(100) DEFAULT NULL COMMENT '某种检查的类别',
  `examination_type_id` BIGINT(20) DEFAULT NULL COMMENT '检查类型的 id',
  `pay_state` INT(1) DEFAULT NULL COMMENT '是否付款, 0 表示未付款, 1 表示已付款',
  `examination_cost` DECIMAL(10,2) DEFAULT '0.00' COMMENT '检验项目费用',
  `time` DATETIME DEFAULT NULL COMMENT '检查时间',
```

```
`active` INT(1) DEFAULT NULL COMMENT '检查是否活跃状态',
PRIMARY KEY ( `examinationOrder_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*检查从 examinationOrder 中查找*/
CREATE TABLE IF NOT EXISTS `medicalRecord`(
  `medicalRecord_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '病历 id',
  `user_id` BIGINT(20) NOT NULL COMMENT '用户 id',
  `department_id` BIGINT(20) NOT NULL COMMENT '科室 id',
  `doctor_id` BIGINT(20) NOT NULL COMMENT '医生 id',
  `symptom` longtext DEFAULT NULL COMMENT '症状描述',
  `medicalRecord_content_first` longtext DEFAULT NULL COMMENT '初步诊断内容',
  `medicalRecord_content_finally` longtext DEFAULT NULL COMMENT '最终诊断内容',
#   `treatment` longtext DEFAULT NULL COMMENT '治疗方式',
  `general` longtext DEFAULT NULL COMMENT '概括',
  `create_date` DATETIME DEFAULT NULL COMMENT '创建时间',
  `active` INT(1) DEFAULT NULL COMMENT '检查是否活跃状态',
  PRIMARY KEY ( `medicalRecord_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `treatment`(
  `treatment_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '治疗 id',
  `user_id` BIGINT(20) NOT NULL COMMENT '患者 id',
  `treatment_cost` DECIMAL(10,2) DEFAULT '0.00' COMMENT '治疗费用',
  `medicalRecord_id` BIGINT(20) NOT NULL COMMENT '病历 id',
  `pay_state` INT(1) DEFAULT NULL COMMENT '是否付款, 0 表示未付款, 1 表示已付款',
  `create_date` DATETIME DEFAULT NULL COMMENT '创建时间',
  PRIMARY KEY ( `treatment_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `drug`(
  `drug_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '药品 id',
  `drug_type` VARCHAR(100) DEFAULT NULL COMMENT '药品的类别',
  `drug_name` VARCHAR(100) DEFAULT NULL COMMENT '药品的名字',
  `production_place` VARCHAR(100) DEFAULT NULL COMMENT '药品的产地',
  `drug_price` DECIMAL(10,2) DEFAULT '0.00' COMMENT '药品单价',
  `drug_stock` INT(10) DEFAULT NULL COMMENT '药品的库存数量',
  `create_date` DATETIME DEFAULT NULL COMMENT '创建时间',
  PRIMARY KEY ( `drug_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `treatment_drug_order`(
  `treatment_drug_order_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '治
```

疗药品流水 id',

```

`drug_id` BIGINT(20) NOT NULL COMMENT '药品 id',
`drug_name` VARCHAR(100) DEFAULT NULL COMMENT '药品的名字',
`treatment_id` BIGINT(20) NOT NULL COMMENT '治疗 id',
`instruct_days` INT(10) DEFAULT NULL COMMENT '服用天数',
`instruct_count_per_day` INT(10) DEFAULT NULL COMMENT '每天服用数量',
`drug_price` DECIMAL(10,2) DEFAULT '0.00' COMMENT '药品单价',
`drug_count` INT(10) DEFAULT NULL COMMENT '药品的数量',
`total_price` DECIMAL(10,2) DEFAULT '0.00' COMMENT '总价',
`create_date` DATETIME DEFAULT NULL COMMENT '创建时间',
PRIMARY KEY ( `treatment_drug_order_id` )

```

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `examination_type`(`

`examination_type_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '检查类型的 id',

`examination_type_name` VARCHAR(100) DEFAULT NULL COMMENT '检查类型的名字',

`examination_type_detail` VARCHAR(100) DEFAULT NULL COMMENT '检查类型的详细内容, 比如部位',

`examination_addr` VARCHAR(100) DEFAULT NULL COMMENT '检验地址',

`examination_cost` DECIMAL(10,2) DEFAULT '0.00' COMMENT '检验项目费用',

PRIMARY KEY (`examination_type_id`)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE IF NOT EXISTS `examination_report`(`

`examination_report_id` BIGINT(20) NOT NULL AUTO_INCREMENT COMMENT '检查报告的 id',

`examinationOrder_id` BIGINT(20) NOT NULL COMMENT '检查项目流水 id',

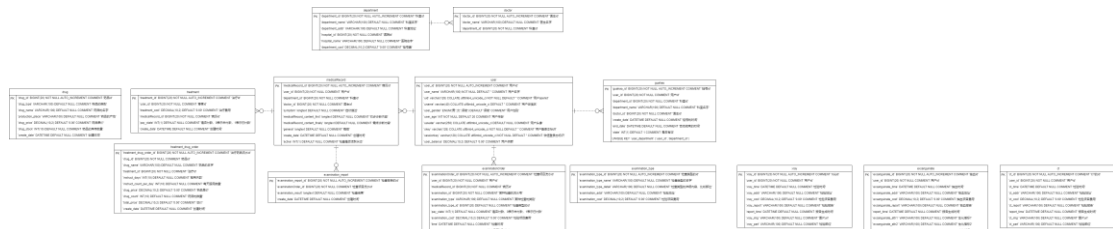
`examination_result` longtext DEFAULT NULL COMMENT '检查结果',

`create_date` DATETIME DEFAULT NULL COMMENT '创建时间',

PRIMARY KEY (`examination_report_id`)

)ENGINE=InnoDB DEFAULT CHARSET=utf8;

ER 图如下:



7 详细设计

7.1 NFC 处理功能模块

7.1.1 功能描述

该功能模块可以读写 NFC 标签信息，读取二维码信息，控制服务开关和跳转轻应用。

7.1.2 性能描述

能在触碰到 NFC 标签 3 秒内自动跳转到对应的轻应用。

7.1.3 输入

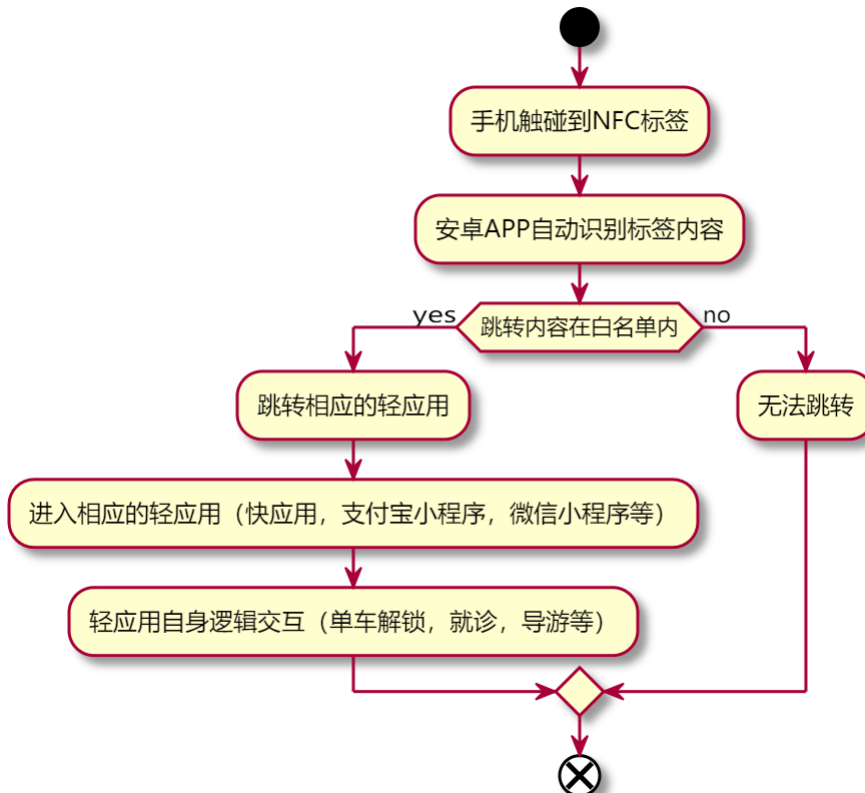
触碰到的 NFC 标签中的信息，用于写入 NFC 标签中的信息，要读取的二维码信息。

7.1.4 输出

跳转到轻应用的数据，写入到 NFC 标签中的信息，解析后的二维码信息

7.1.5 程序逻辑

流程图如下：



7.1.6 限制条件

无

7.2 共享单车功能模块

7.2.1 功能描述

该功能模块有骑行记录、定位刷新和余额充值的功能。

7.2.2 性能描述

能在 3 秒内完成定位。

7.2.3 输入

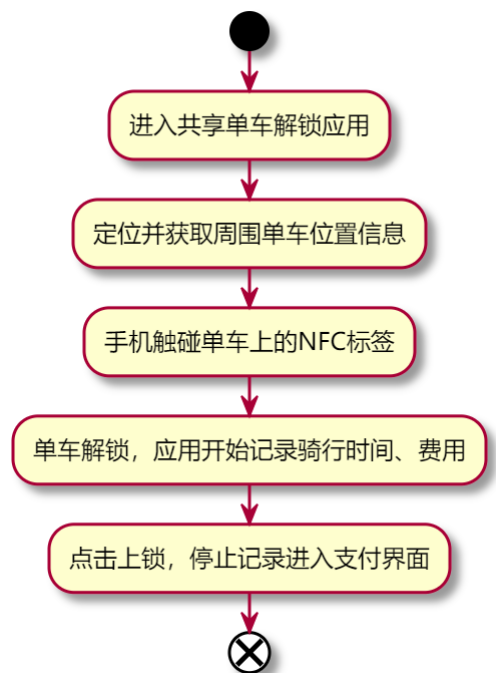
手机的定位信息，共享单车的位置信息，充值金额。

7.2.4 输出

骑行时间和费用

7.2.5 程序逻辑

流程图如下：



7.2.6 限制条件

无