本项目名为个人健康数据存储与流通平台（PHDC），本文档将会详细介绍平台的操作流程、提交的代码结构、部署方法等内容。本项目开发过程全部由参与过开源项目的在职开发人员利用业余时间在网上通过远程协作方式完成，代码不完善之处在所难免，还请评委见谅。

核心点：评测人员可按照提供的操作步骤完成从端到端的场景验证

* 代码结构说明

本项目代码分为五个部分，Chaincode链码、Java后台、H5页面、Android客户端以及fabric-sdk-android，分别在如下目录存储：

chaincode：Chaincode链码，链码基于go语言开发

java： java后台代码，基于springboot开发

pages： H5页面代码，基于Bootstrap+jQuery开发

UOClient：Android客户端代码

fabric-sdk-android：适配安卓之后的最新版fabric-sdk-java代码

接下来详细讲解每个目录下的内容：

1. chaincode：

drcc：DRCC链码，及其包含的组件

hdcc：HDCC链码

uicc：UICC链码

需要注意的是为了方便链码之间的引用，链码的main都放在cmd目录下，打包时需要指定到链码的cmd目录；

2. java：

pom.xml：maven工程文件

Readme.md：接口说明文件

src：源码目录，内容如下：

src/main/resource：配置文件存放目录

src/main/resource/config/mapper：数据库映射

src/main/resource/fabric-config：Baas网络配置

src/main/resource/application.yml：应用配置，包含数据源等配置

src/main/java/io/mycat/fabric/phdc：后台代码

src/main/java/io/mycat/fabric/phdc/baas：链码相关代码目录，主要修改了Baas sdk的ChaincodeExutor.java与FabricUser.java

后台代码遵循MVC模式，ctrl代表controller，其余部分在这里就不再一一赘述；

3. UOClient目录：

app/libs/fabric-sdk-java-1.3.0-SNAPSHOT.jar：修改过后适应安卓系统的最新版fabric-sdk-java

app/src/main/assets：安卓版资源目录，需要注意的是因为安卓版使用Hypbrid App模式开发，界面资源都是使用的H5页面，打包时需要将pages目录下的内容拷贝到该目录下

app/src/main/assets/connection-profile-standard.yaml：Baas配置文件

app/src/main/java/utils：区块链相关代码，在这里主要修改了Baas sdk的ChaincodeExutor.java与FabricUser.java

app/src/main/java/io/mycat/uoclient/MainActivity.java：安卓项目代码，包括所有界面与逻辑

4. pages目录：

bootstrap-4.0.0-dist：bootstrap框架

js：jQuery框架

views/duo：duo部分界面代码

views/duo/userList.html：duo界面入口，邀请用户列表

views/duo/userDetail.html：duo查看用户详情

views/ho：ho部分界面代码

views/ho/index.html：ho界面入口

views/ho/appointmentList.html：ho预约管理界面

views/ho/dataList.html：ho数据管理界面

views/ho/userList.html：ho用户管理界面

views/ho/userItem.html：ho用户检查列表界面

views/uo：uo部分界面代码，注意这部分代码需要在安卓客户端内运行

views/uo/index.html：uo界面入口

views/uo/hoList.html：uo健康服务预约界面

views/uo/duoApplyList：uo数据访问请求界面

views/uo/userDataList.html：uo查看数据界面

5. fabric-sdk-java目录：

fabric-sdk-java因为文件较多，就不一一列举，在这里只列举修改的部分：

src/main/java/org/hyperledger/fabric/sdk/helper/Utils.java：修改了安卓支持的正则表达式

src/main/java/org/hyperledger/fabric/sdk/Endpoint.java：将Netty替换为OkHttp

pom.xml：增加grpc-okhttp库，屏蔽掉部分安卓不支持的库，文件已提交github: https://github.com/ishotoli/fabric-sdk-java

此外，项目使用到的数据库文件存放在项目根目录下，文件名为：phdc\_20180904.sql

* 安装步骤

项目运行环境服务器端要求CentOS 7、Nginx 1.15、MySQL 5.7、JDK 1.8，浏览器需求Chrome 68或者Safari 11.1版本，安卓客户端要求8.0（API v26）及以上版本。

1. 部署数据库：将phdc\_20180904.sql文件导入到MySQL数据库当中，即可完成建表；

2. 部署后端服务：首先使用mvn package命令打包jar，然后上传到服务器目录下，运行如下命令：

nohup java -XX:MetaspaceSize=128m -DCONFIG\_HOME=${basepath}/config -jar PHDC-0.0.1-SNAPSHOT.jar &

注意环境变量basepath指向项目所在的根目录下，配置文件放在config文件夹下，防火墙入网方向的8666端口打开，如果服务器地址有变化，可能需要替换代码中部分回调地址；

3. 部署H5页面：将pages目录下的内容上传到nginx指定的html文件夹下，启动nginx即可，注意防火墙开放入网方向的80端口，如果后台服务器地址有变化，请注意替换代码中的Ajax请求访问的地址；

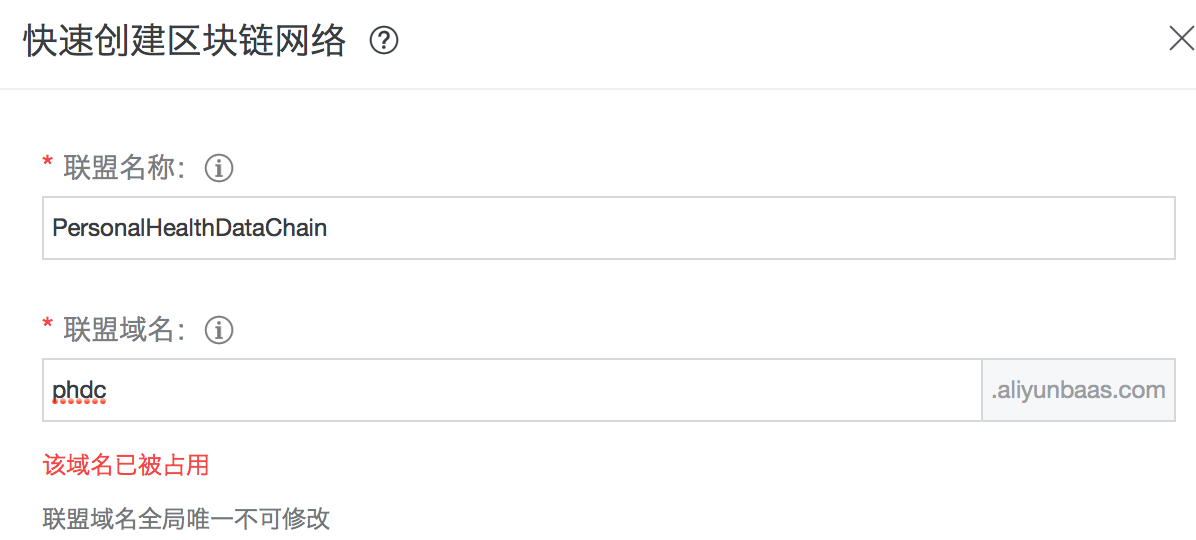
4. 安装安卓客户端：将代码打包apk后，上传安卓手机，安装apk即可，或者直接安装zip代码包中的uoclient.apk。需要注意的是如果后台服务器地址有变化，请注意替换代码中的Ajax请求访问地址；

5. 部署Baas区块链：

a. 登录Baas管理平台后，点击管理概览页的“快速创建区块链”：

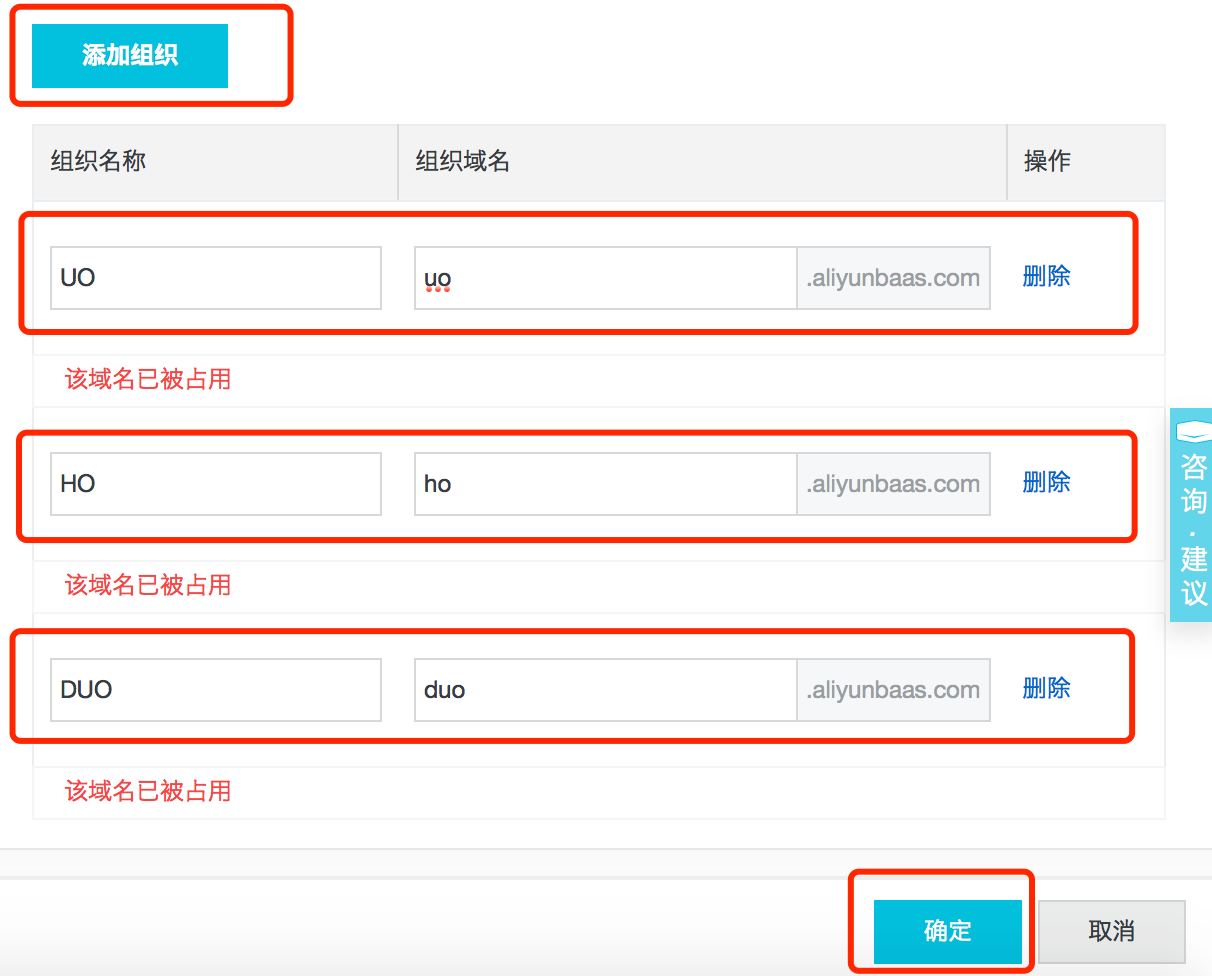


b. 填写联盟信息:



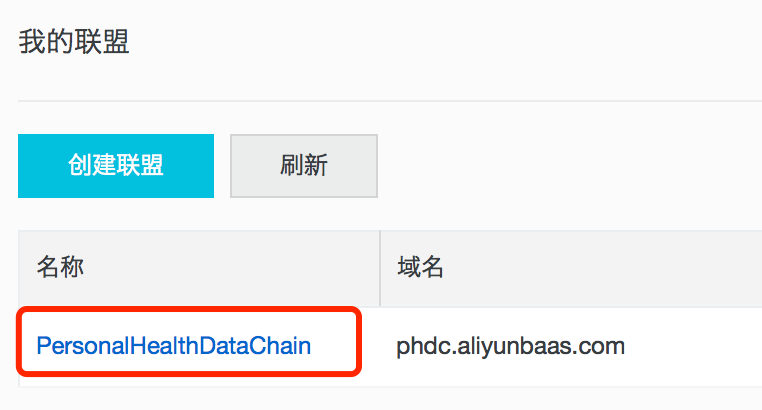
需要注意的是，如果Baas创建时的信息有修改，那么也需要修改代码当中对应的配置文件和用户信息；

c. 填写组织信息，需要新建三个组织，分别名为uo、duo、ho：



添加完信息后，点击确定创建联盟与组织；

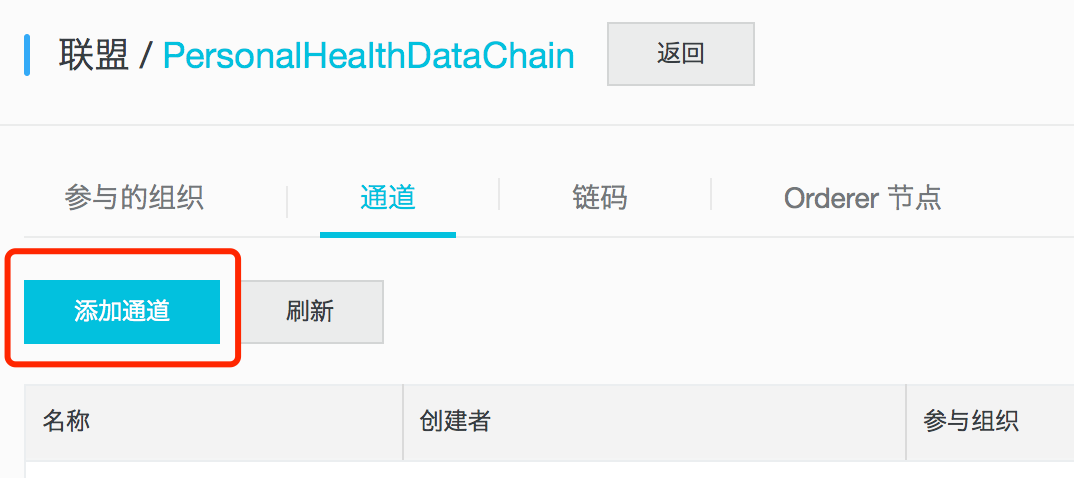
d. 联盟与组织创建完成后，需要创建对应的通道，首先我们点击概览页的联盟名称：



e. 在联盟信息页点击“通道”标签：



f. 在通道信息页，点击“添加通道”：



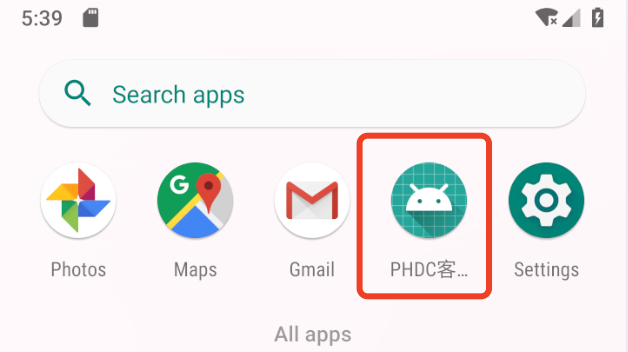
g. 在创建通道页，分别创建三个通道：uic、hdc、drc，其中uic通道只包含UO组织，hdc通道包含HO与UO组织，drc包含HO与UO组织（通道添加组织的时候好像自动把组织的节点也加进去了，这个其实没有太大必要，建议改进）



h. 在通道创建完成后，创建对应组织用户，并对应通道安装链码：uic通道安装uicc链码，hdc通道安装hdcc链码，drc通道安装drcc链码，PHDC网络，网络安装流程就结束了。

* 应用使用方法

1. 启动安卓应用，点击PHDC客户端的APP图标：



2. 进入App后，先开始健康服务预约：



3. 这时会从UIC通道获取机构列表，列表展示出来后，点击机构名称进入下一步：



4. 填写用户信息：



用户姓名先填写kaiz，手机号填写12345678912，出生日期请选择2000年之前的日期，性别选择男，因为演示数据有限，请先按照提供的数据填写；填写完成后点按下一步；

5. 这时会从HDC通道获取可以检查的项目，测试数据暂时只有血糖、血脂、血压三种，请随便选择其中的一种：

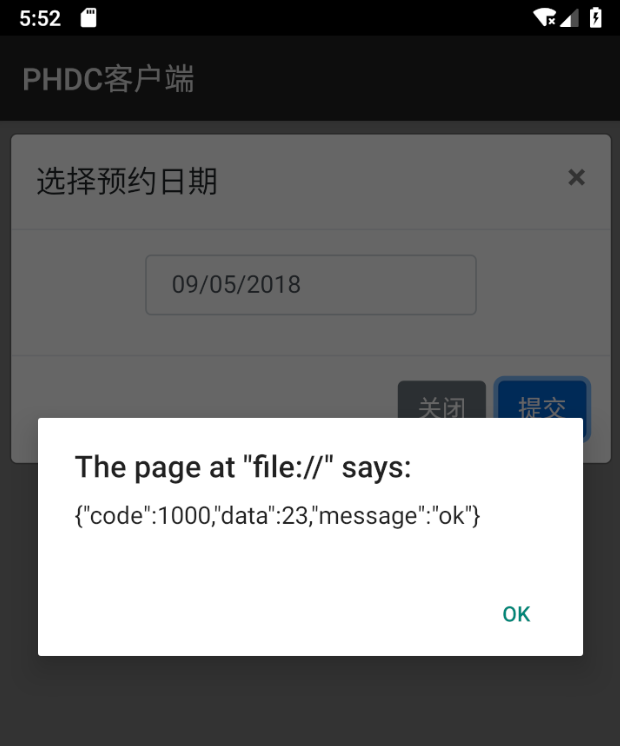


选择之后会跳转到下一个页面；

6. 选择预约的服务日期后，点击提交即可：



提交成功后，会Alert显示后台服务处理结果：



7. 接下来我们进入HO管理页面 http://39.104.99.78/views/ho/index.html：



点击预约管理处理刚刚的预约；

8. 在预约管理界面，我们点击刚刚预约信息的“到检”按钮：



所有列表当中的数据都是按照提交时间排列的，最新一次提交的数据都在最上方，点击预约后页面会刷新加载最新数据，加载完成后我们点击后退，回到HO管理页面；

9. 在HO管理首页点击数据管理，进入数据管理页面：



10. 在数据管理页面，我们可以看到最新一条添加的数据，在这里数据会按照点击到检的时间顺序排列，我们点击最新一条数据的“输入结果”按钮：



11. 输入结果和检查医生，建议使用中文、英文、数字，不要使用标点符号，字数10个字以内，填写完成后点击确定按钮提交数据；



12. 此时数据会由后台保存到HDC通道上，时间可能会稍慢，请耐心等待一会，直到出现如下界面：



点击确定后页面会刷新，我们继续填写结论；

13. 点击最上一条的“输入结论”按钮：



14. 输入结论，规则与输入结果相同：



点击确定后提交；

15. 提交成功后，页面会刷新，我们可以看到最新一条修改的内容：



到此HO端操作基本结束；我们回到安卓客户端继续流程

16. 在安卓客户端首页，点击查看数据按钮：



17. 在查看数据页面，客户端会自动检查最新一次的预约是否已经录入数据，如果已经录入数据，则会自动向UIC通道提交数据索引，在这个过程中会有几次弹窗提示过程中的后台与链码返回值，因为会等待提交Orderer成功，所以时间较长，请耐心等待，成功之后，在界面可以看到最新一条记录：



添加健康数据的流程到此结束；

18. 接下来我们需要开始请求数据流程，首先我们回到安卓客户端首页，点击“数据访问请求”按钮：



19. 在数据访问请求界面会从UIC链上加载数据访问请求列表，偶尔时间较长，请耐心等待：



界面加载完成后，点击“生成邀请ID”；

20. 在生成邀请ID的弹框界面上，记录邀请ID：



记录邀请ID之后，关闭对话框；原理上来说应该生成包含公钥与加密信息的二维码，但这里为了简化流程只写死了一个数字邀请ID；

21. 打开DUO首页 <http://39.104.99.78/views/duo/userList.html> ，点击“邀请新用户”：

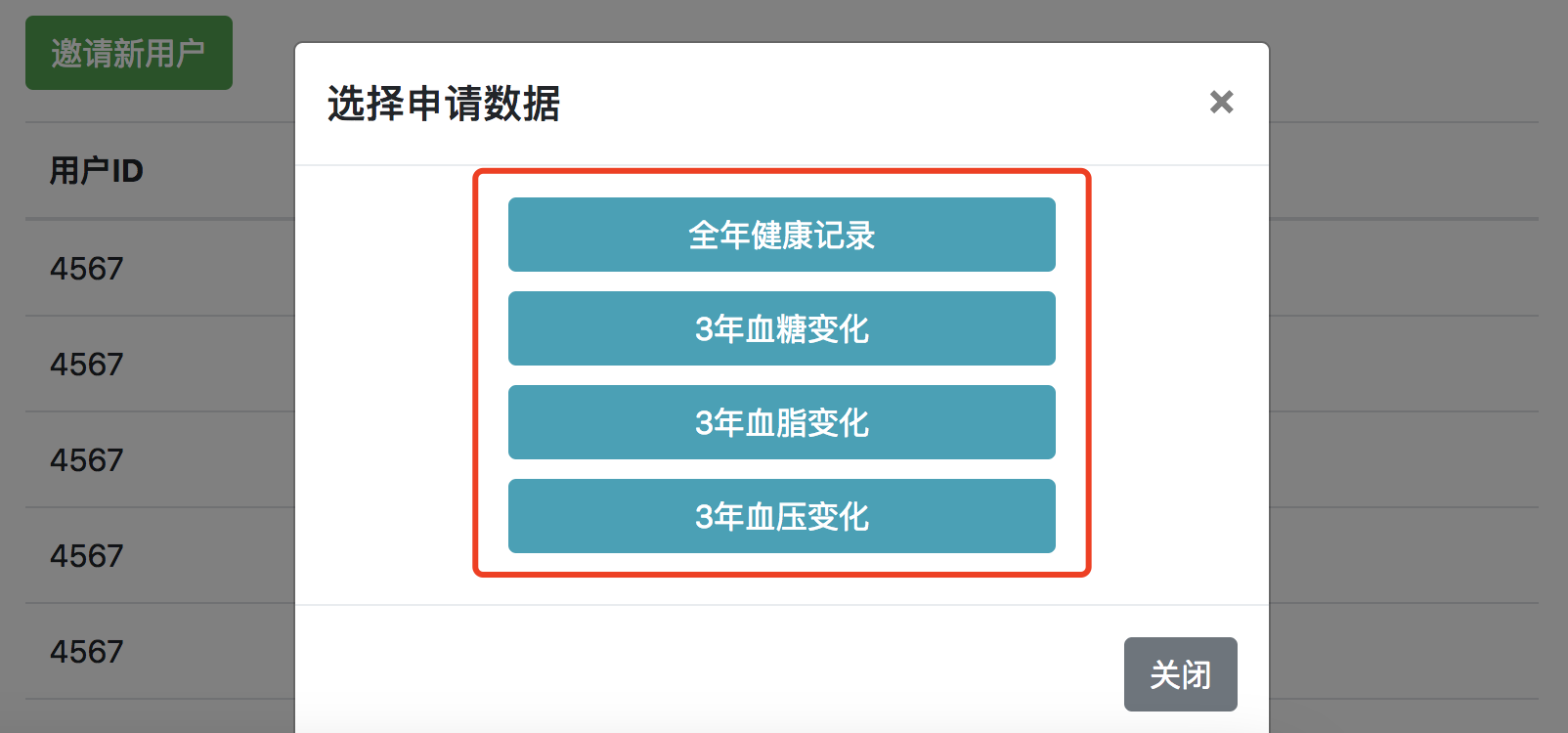


22. 填写邀请用户信息：

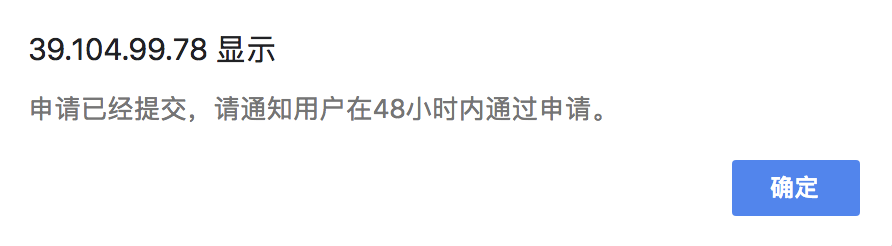


用户姓名先填写kaiz，手机号填写12345678912，邀请ID填写4567，然后点击下一步；原理上来说这里应该录入用户生成的邀请二维码，但为了简化步骤这里只填写了邀请ID；

23. 接下来选择请求数据的模板：



24. 接下来会将邀请信息写入DRC：



DUO的后台服务会等待DRC的链码确认事件，DRC通道出块时间较长（已经发工单问过了，不确定原因是什么，有时需要等待5分钟才能出块）

25. 关闭对话框后手动刷新列表，可以看到最新的数据请求：



状态为“已申请”代表信息已经写入DRC，如果提交了DRC但还没有通过VSCC验证，则会显示“申请中”，此时需要多等待几分钟后刷新页面，直到状态变为“已申请”

26. 在安卓客户端主页面，点击“数据访问请求”按钮，重新进入数据访问请求页面：



27. 将页面滚动到最下，可以看到最新的一条数据：



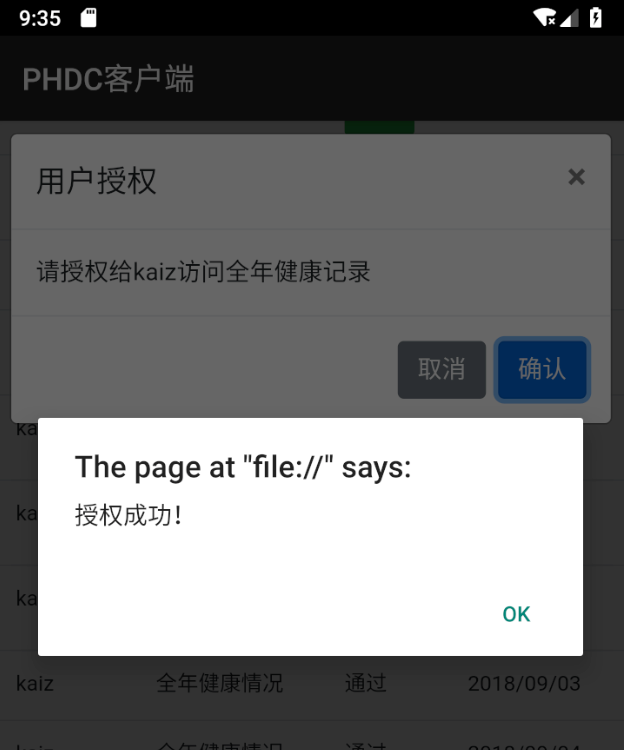
因为链上测试数据较多，还请见谅，点击“允许”按钮；

28. 在弹出的确认界面点击“确定”：



此时DRCC链码会提取对应的数据发送给DUO后台，提交Orderer后出块时间可能会有几分钟（DRC通道偶发问题，不确定原因），请耐心等待；

29. 弹出“授权成功”界面后，授权流程完成，DUO即可在用户列表界面查看用户健康数据：



30. 刷新DUO用户列表界面，可以看到申请状态已经变化：



点击对应的“查看”可以查看对应的数据

31. DUO用户详情页可以查看对应数据：



到此，流程基本上就结束了，因为业务涉及Baas操作较多，而当前Baas网络不是特别稳定，我们在测试过程中经常发生超时等情况，所以使用过程中请耐心一点，如果遇到失败的情况请重试操作，安卓端的链码操作触发后下方会有toast提示，如果没有提示可以尝试重新点击，谢谢合作！

* 示例数据

本项目主要默认使用的示例数据如下：

用户姓名：kaiz

用户手机号：12345678912

邀请ID：4567

生日建议填写2000年之前的日期

预约日期建议填写当天之后的日期

健康数据结果、结论建议填写中文、英文、数字，10个字符之内

* 可在线测试URL地址

HO管理页面：<http://39.104.99.78/views/ho/index.html>  
DUO管理页面：<http://39.104.99.78/views/duo/userList.html>

UOClient请使用zip压缩包中提供的UOClient.apk