基于Android的实时共享白板的设计与实现

目录

[摘 要： 3](#_Toc482658823)

[Abstract: 4](#_Toc482658824)

[1. 绪论 5](#_Toc482658825)

[1.1. 背景 5](#_Toc482658826)

[1.2. 意义 5](#_Toc482658827)

[1.3. 目标 5](#_Toc482658828)

[1.4. 可行性分析 6](#_Toc482658829)

[1.5. 相关技术说明 6](#_Toc482658830)

[1.6. 软硬件环境 9](#_Toc482658831)

[2. 需求分析 11](#_Toc482658832)

[2.1. 功能需求 11](#_Toc482658833)

[2.2. 性能需求 15](#_Toc482658834)

[2.3. 质量属性 15](#_Toc482658835)

[2.4. 对外接口 16](#_Toc482658836)

[2.5. 约束 16](#_Toc482658837)

[3. 概要设计 17](#_Toc482658838)

[3.1. 网络结构设计 17](#_Toc482658839)

[3.2. 功能结构设计 18](#_Toc482658840)

[3.3. 数据库设计 19](#_Toc482658841)

[3.4. 综合考虑 23](#_Toc482658842)

[4. 详细设计 24](#_Toc482658843)

[4.1. 模块设计 24](#_Toc482658844)

[4.2. 接口设计 24](#_Toc482658845)

[结论 25](#_Toc482658846)

[致谢 26](#_Toc482658847)

[【参考文献】 27](#_Toc482658848)

[附录： 27](#_Toc482658849)

摘 要：随着互联网的飞速发展，各种出行类APP和网站也层出不穷，“晥美出行”则是一个主要提供网上车次查询、网上订票、网上购票、网上退票等客运汽车票的服务的网站平台。也由此“晥美出行”需要和客运站系统进行对接，获取客运站数据并且进行相关操作。由于晥美出行平台不能直接对客运站系统进行任何更改，只能在允许的范围进行相关数据操作。所以基于此，本文研究了国内外WebService和SOA技术的发展情况，发现SOA这种粗粒度、松耦合的服务架构正好适合这个应用场景，设计并据此设计实现了一套基于WebService和SOA技术的Web接口系统。

关键词：WebService、接口系统、SOA

Abstract: English English English English English English English English English English English English English English English English English English English English English English English English

Keywords: English english English

1. 绪论
   1. 背景

随着我国普遍进入互联网时代，“互联网+”已经深入到国民工作生活的方方面面，人们逐渐习惯使用智能手机上的APP解决问题，譬如购物软件、外卖订餐软件、同城租房软件、打车软件等。随后，基于互联网，出现了另一个概念“共享经济”，指的是一种共用人力与资源的社会运作方式，它包括不同个人与组织对商品和服务的创造、生成、分配、交易、和消费的共享，源于“共享经济”的应用也比比皆是，譬如Uber、摩拜单车等。互联网以其“连接万物”的能力，高效地传递信息，同时结合分布式计算，可以带来超大的计算能力；“共享”很好地解决了社会资源不足的问题，提高了社会的整体资源利用率。这两点在某种程度上揭示了人们的需求，更简单快捷地解决以往需要现场沟通并支付资源独享费用的问题。

* 1. 意义

随着社会的飞速发展，人们的要求和期望越来越高，越来越多的行业参与到 ”互联网+” 的大家庭中，公司规模在变大，经常有多个公司并分布在不同的地方，项目的复杂度也在不断提升，项目组成员也随之不断增多。如此一来，沟通变得越来越重要，但也越来越难，尤其是跨地区沟通的效率低而成本高。如何解决这个问题? 网络给出了答案，因为高速网络联通了你我，使得我们可以跨越地理相隔实现交互，而传统的即时通讯工具一般局限于富文本聊天，文本相对于图像，缺少直观性，不能保证沟通的简单便捷高效。而基于Android的实时共享白板是通过创建一块虚拟会议空间，多人连入并共享这块空间，在里面可以通过白板绘画、发文字、发图片、发语音等方式实现信息的交互共享，从而简单快捷地解决团队沟通问题。

* 1. 目标

按照软件工程的方法，设计并实现一个基于Android的实时共享白板APP，要求它有较为完善的基础功能（例如注册、登录、重置密码）和具有特色的专职功能（例如共享白板、会议管理、联系人管理），同时保证软件的可靠性、可维护性，努力提升软件的易用性、兼容性、效率。

* 1. 可行性分析
     1. **难点及解决方案**
        1. **绘图控件的渲染**

难点：目标实现的白板，包括多种绘画形式，并且考虑到需求变更的可能，需要尽可能让各个绘画工具相互独立，同时为了提高用户体验，需要尽量提高画板的性能，从而让用户更流畅的操作。

解决：经过查资料，了解到使用Java的Canvas或HTML5的Canvas都可以实现，如果使用HTML5的Canvas，可以结合当下流行的React，而React的内部机制使得它相比原生JavaScript拥有更高的效率，可以满足高性能的需要，同时其组件化的思想可以满足易扩展易重构的需要。因此HTML5的Canvas与 React相结合可以解决问题。

* + - 1. **多人实时同步通信**

难点：目标实现的共享白板，需要实时同步信息到多个用户，这就需要服务器可以主动向客户端发送消息。

解决：经过查资料，HTTP协议具有被动性，虽然通过轮询的方式也可以实现实时通信，但是性能不佳，相比而言，HTML5的WebSocket协议可以更好地满足需求，建立WebSocket连接后，服务器可以主动向客户端发送消息。

* + 1. **分析结果**

综上，两个难点均可以得到有效解决，结合课题的预期难度、工作量以及自身情况得出，课题是可行的。

* 1. 相关技术说明
     1. **Java**

Java是一种面向对象的编程语言，它和c以及c++比较接近，但是在某些特性上，对前面两种语言进行了简化，譬如不再使用指针、多继承等等；它支持网络应用开发，提供了网络编程接口和相关的类库；它支持多线程，可以通过实现Runnable接口或者继承Thread类并重写run方法来创建一个线程对象；它支持动态载入，可以通过网络远程载入需要的类；它是解释执行的，先通过与平台无关的编译器把源程序.java编译成与平台无关中间码.class，然后根据目标运行系统选择不同平台的解释器，也就是说解释器是平台相关的，用相应的解释器把中间码.class解释成和平台相关的机器码，然后运行，从而实现一次编译到处运行的目标，也就是跨平台，如图1.1。

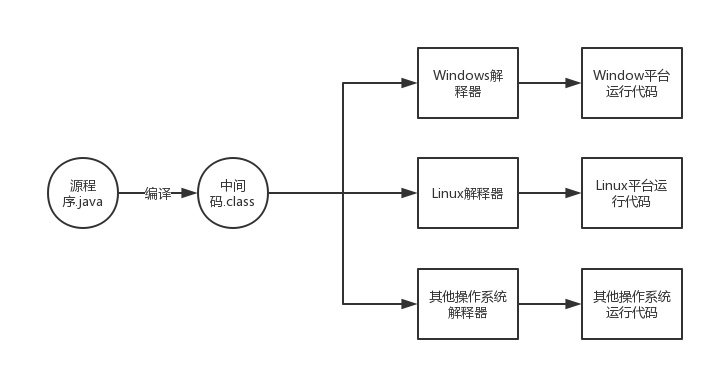


图1.1 Java编译解释过程

* + 1. **PHP**

PHP是一种通用脚本语言，主要用于生成动态网页内容、操作服务器上的额文件和数据库、数据加密、用户授权访问等。PHP凭借其开源、免费、易学、服务器兼容性好、跨平台等特性，被称为“世界上最好的语言”。PHP拥有超过150个扩展库，可供开发人员调用，很大程度上提高了开发效率和代码的性能，例如PDO扩展，它提供了PDO类来对数据库进行访问，开发人员使用不同数据库时候调用的函数名是相同的，使得应用层不用去关心具体要连接的数据库服务器的类型，除此之外，相比原生MYSQL，可以有效防御SQL注入攻击。

* + 1. **JavaScript**

JavaScript是一种解释型脚本语言，它是弱类型、函数优先的，一般作为开发网页的脚本语言，在其他环境也有使用。随着当前网站开发前后端更严格的分离，JavaScript负责越来越多的业务逻辑检查、处理等工作，而不仅仅是单纯的DOM操作和实现网站特效。

* + 1. **React**

React是Facebook研发的JavaScript库，具有易入门、代码易重构、更好支持响应式网页、高性能等特点。React的主要原理包括Virtual DOM、Components、State和Render。其中Virtual DOM就是在真实DOM上面抽象出一个对象，用来表示DOM应该怎么呈现，当需要更新页面的时候，不是直接更新真实DOM，而是更新Virtual DOM，React会等到当前事件循环结束，通过diff算法计算Virual DOM和真实DOM的差别，并计算出最小步数来进行更新，从而提高了性能；Virual DOM的每一个节点就是一个commponent，它的存在使得diff算法更高效；State包含该定义commponents的数据，当数据改变时，它会调用Render进行重新渲染。

* + 1. **WebSocket**

WebSocket是HTML5中的一个新的协议，它和HTTP协议基本没关系，HTTP中每个response都对应一个request，也就是只有客户端主动请求服务器才会回复，而不能做到服务器主动推送消息给客户端，基于这个原理的AJAX轮询和polling技术，可以变向实现实时交互，但是性能不佳且代价比较大；而WebSocket只需要客户端一次request，建立持久连接后，服务器便可以主动向该客户端推送任意多次消息，同时较AJAX轮询和polling技术具有更高的效率和性能。WebSocket.org 对传统的轮询方式和 WebSocket 调用方式作过一个详细的测试和比较，将一个简单的 Web 应用分别用轮询方式和 WebSocket 方式来实现，并把测试结果用柱状图进行表示，如图1.2。



图1.2 轮询和WebSocket的性能对比

* + 1. **Gateway Worker**

Gateway Worker是一个开源的Socket服务器框架，用PHP编写，该框架采用的是Gateway和Worker进程模型，它支持分布式部署、支持高并发、支持多种应用层、支持代码热更新、支持HVVM协议、提供心跳检测、在Linux服务器上可以做到守护进程化。使用该框架，开发者只需要简单配置并通过实现框架内部Events类的函数，就可以轻松监听WebSocket连接的建立、消息请求、连接断开等事件，并实现消息的单发、群发、广播、定时推送等功能。

* 1. 软硬件环境
     1. **硬件环境**

联想Y500笔记本:

CPU Intel Core i5-3230M 2.6GHz

RAM 8G

腾讯云服务器：

CPU 1核

RAM 2G

带宽 2M

小米4手机：

CPU: 高通 骁龙801（MSM8974AC）

RAM 3G

* + 1. **软件环境**

Windows8.1系统、Ubuntu14.04系统、Android 6.0系统、AndroidStudio2.2.3、 PhpStorm 2016.12、JDK 1.8、PHP 5.5.9、Apache 2.4.7、MySQL 5.5、MySQLWorkbench 6.3、SQLiteStudio 3.1.1、FiliZilla 3.25

1. 需求分析
   1. 功能需求
      1. **列举功能需求**
2. 用户可以注册
3. 用户可以登录
4. 用户可以注销
5. 用户可以重置密码
6. 用户可以更改头像
7. 用户可以更改姓名
8. 用户可以更改密码
9. 用户可以更改会议偏好
10. 用户可以检查更新软件
11. 用户可以反馈建议
12. 用户可以添加联系人
13. 用户可以接受添加联系人申请
14. 用户可以拒绝添加联系人申请
15. 用户可以删除联系人
16. 用户可以以主持人身份安排会议
17. 用户可以以主持人身份查看已安排的会议
18. 用户可以以主持人身份删除已安排的会议
19. 用户可以以主持人身份进入已安排的会议
20. 用户可以以主持人身份直接召开会议
21. 用户可以以主持人身份共享（图片或网页）
22. 用户可以以主持人身份结束共享（图片或网页）
23. 用户可以以主持人身份锁定会议
24. 用户可以以主持人身份取消锁定会议
25. 用户可以以主持人身份查看会议成员列表
26. 用户可以以主持人身份禁止与会者讲话
27. 用户可以以主持人身份取消禁止与会者讲话
28. 用户可以以主持人身份禁止与会者使用白板
29. 用户可以以主持人身份取消禁止与使用白板
30. 用户可以以主持人身份把与会者踢出会议
31. 用户可以以与会者身份加入会议
32. 用户可以以与会者身份观看主持人的共享
33. 用户可以以与会者身份查看会议成员列表
34. 用户可以邀请别人加会
35. 用户在会议中可以使用白板绘画
36. 用户在会议中可以给白板录屏
37. 用户在会议中可以进行群聊
38. 用户可以离开会议
    * 1. **建立用例模型**
         1. **识别参与者**

系统的参与者就是软件的注册用户，而注册用户可能扮演两种角色：主持人和与会者。故系统有三个参与者，分别是Registrant（注册用户）、Host（主持人）、Participant（与会者），其中注册用户和主持人之间是类属关系，注册用户和与会者之间也是类属关系。

* + - 1. **识别用例**

1. Register(注册)：提供用户新建个人账号的功能
2. Login(登录)：提供用户进入个人账号的功能
3. Logout（注销）：提供用户退出当前账号的功能
4. ResetPassword（重置密码）：提供用户忘记密码后重置密码的功能。
5. SetMyInfo（设置个人信息）：提供用户更改姓名、头像、密码的功能。
6. SetMeetingPrefs（设置会议偏好）：提供用户更改会议默认偏好设置的功能。
7. CheckUpgrade（检查更新）：提供用户获取新版本软件更新的功能。
8. FeedBack（反馈）：提供用户反馈使用情况、建议的功能。
9. ContactsManage（联系人管理）：提供添加、同意、拒绝、查询、删除联系人的功能。
10. ArrangeMeetingManage（会议安排管理）：提供会议安排、查看、删除。
11. HostMeeting（召开会议）：提供主持人召开会议的功能。
12. InviteOutter（会外邀请）：提供主持人会外邀请别人加会的功能。
13. InviteInner（会内邀请）：提供用户会内邀请别人加会的功能。
14. JoinMeeting（加入会议）：提供与会者加入会议的功能。
15. LeaveMeeting（离开会议）：提供用户离开会议的功能。
16. ShareRes（共享资源）：提供主持人会内共享（图片或网页）给与会者以及取消共享的功能。
17. LockMeeting（锁定会议）：提供主持人锁定会议、解锁会议的功能。
18. MeetingPrivilegeManage（会内权限管理）：提供主持人对与会者禁言、取消禁言、禁画、取消禁画、踢出会议的功能。
19. Board（白板）：提供用户会内使用白板的功能。
20. Chat（群聊）：提供会内群聊的功能。
21. Member（参与者）：提供查看会议参与者的功能。
    * + 1. **用例图**

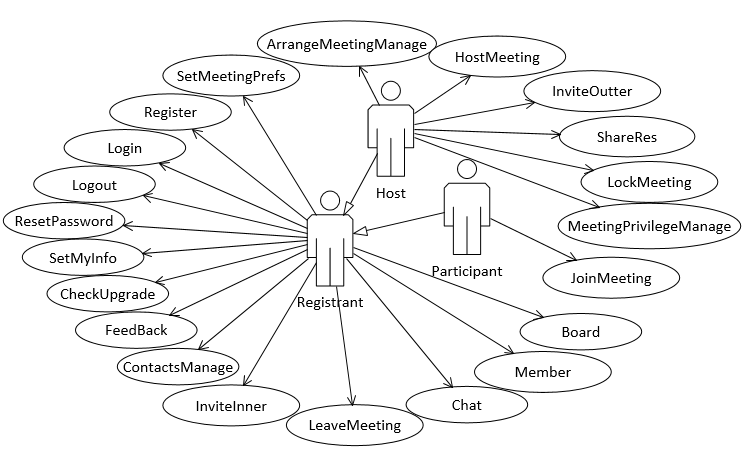


图2.1 系统用例图

* + 1. **业务流程分析**



图2.2 系统业务流程图

* 1. 性能需求
     1. **速度**

1. 注册验证码必须在15秒内发送到用户注册所用邮箱。
2. 登录必须在5秒内完成。
3. 所有用户查询必须在5秒内完成。
4. 开会、加会必须在10秒内完成。
   * 1. **负载**
5. 系统至少可以维持300个WebSocket并发连接。
6. 系统至少可以维持20个容量为20人的会议。
   * 1. **实时性**
7. 白板绘画、群聊信息必须在1.5秒内同步到会议的其他参与者。
8. 联系人的添加、接受、拒绝、删除、加会邀请必须在2秒内推送到目标客户端。
   1. 质量属性
      1. **安全性**
9. 需要保障用户账户安全。
10. 需要保障服务器的数据安全,包括数据库和用户上传文件；
11. 需要保障通信安全，通信数据需要加密，防止传递的信息被他人监听并轻松得到信息的内容；
    * 1. **可靠性**
12. 软件在用户输入极端数据或恶意操作时候，有充分的合法性检查，可以避免崩溃或至少不会殃及其他用户。
13. 程序报错时有效提醒用户错误产生的可能原因以及可以采取的措施，生成错误报告，上传到服务器，以便对报错进行重现、分析、纠错。
    * 1. **易用性**
14. 图标醒目易懂，尽量辅助以文字说明。
15. 流程尽量简化。
    * 1. **可维护性**
16. 代码命名规范。
17. 代码注释详略得当。
18. 对代码进行重构。
    1. 对外接口
    2. 约束

# 概要设计

* 1. 网络结构设计

软件采用C/S结构，其中客户端部分为安卓智能手机，通过互联网和服务器进行通信，服务器主要涉及到四个：Web服务器、Socket服务器、QQ邮件服务器、极光推送服务器，如图3.1。

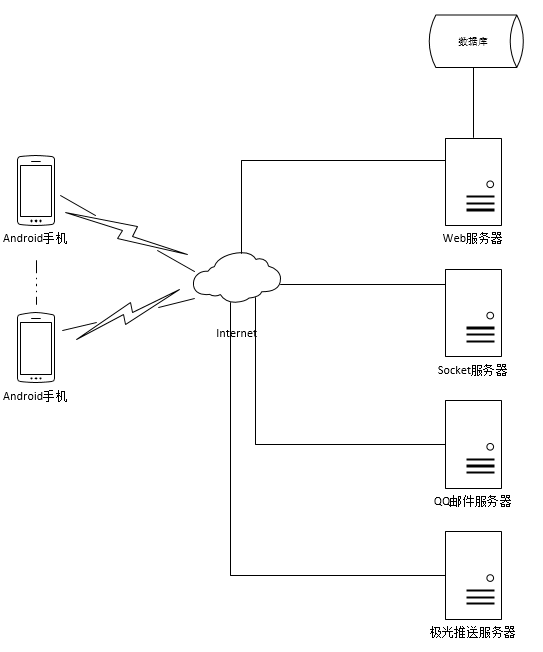


图3.1 系统网络拓扑图

四个服务器的作用如下：

1. Web服务器：处理客户端发送的HTTP请求，包括授权访问检查、业务逻辑处理、文件上传下载、数据库操作等。
2. Socket服务器：用户进行白板绘画、会内群聊的时候，每个客户端都和Socket服务器之间维持着一个WebSocket连接，用来进行数据和消息的实时同步。Socket服务器就是负责处理Socket请求，单发、群发、广播消息的。
3. QQ邮件服务器：用来给用户发送邮箱验证码。
4. 极光推送服务器：用于推送联系人添加、接受、拒绝、删除、邀请加会请求到客户端。
   1. 功能结构设计

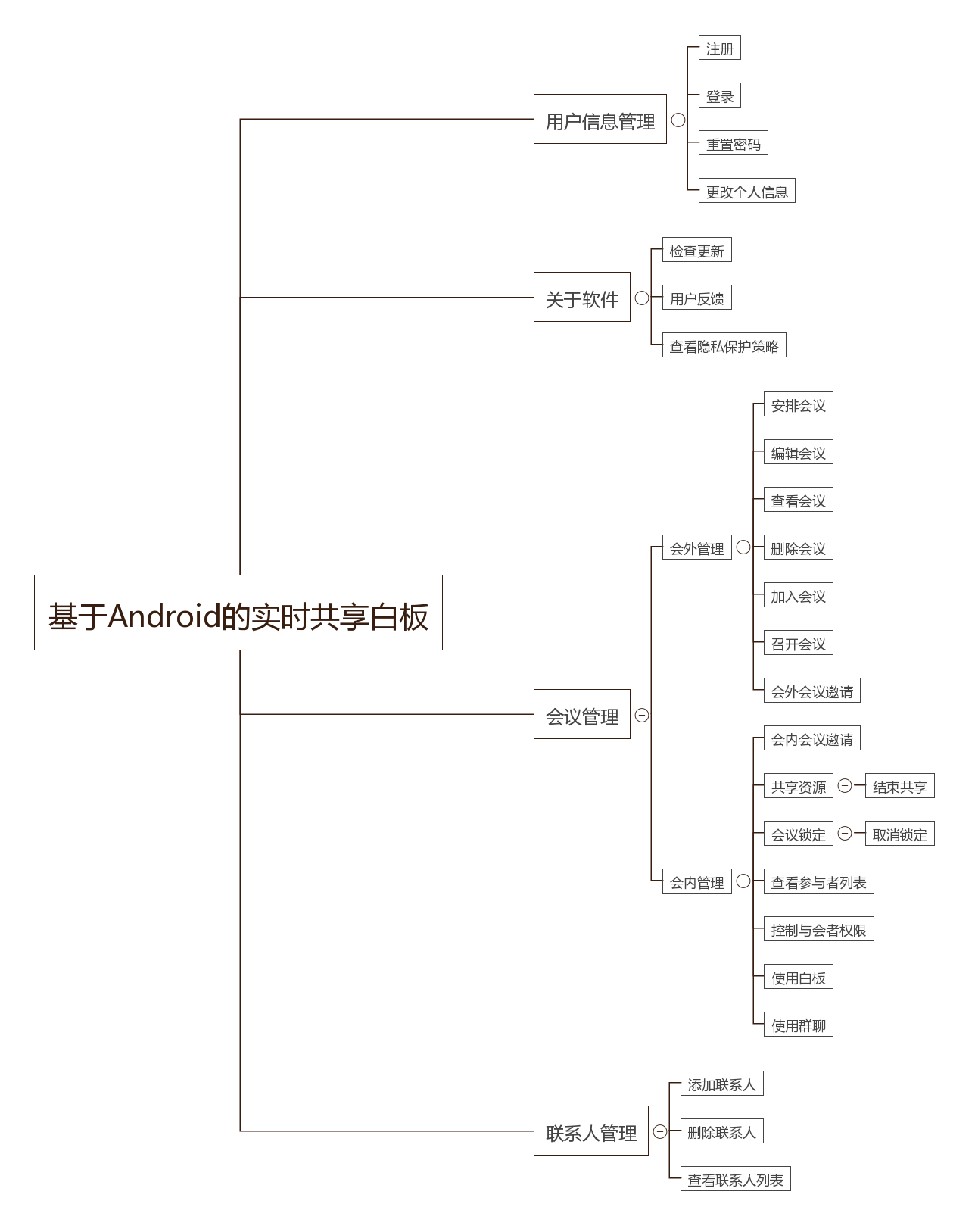


图3.2 系统功能结构图

* 1. 数据库设计

服务器端数据库采用MySql，客户端数据库采用SQLite。

MySql 中数据库包括5张表，见表3.1到表3.5。

表3.1 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 索引 | 注释 |
| user\_id | INT | 10 | 主键 | 用户ID |
| user\_email | VARCHAR | 30 | 主键 | 邮箱 |
| user\_family\_name | VARCHAR | 15 |  | 姓 |
| user\_given\_name | VARCHAR | 15 |  | 名 |
| user\_register\_time | DATETIME | 32 |  | 注册时间 |
| user\_login\_recent\_time | DATETIME | 32 |  | 最近一次登陆时间 |
| user\_password | VARCHAR | 32 |  | 登录密码 |
| user\_token | VARCHAR | 32 |  | Token |
| user\_avatar | VARCHAR | 60 |  | 用户头像地址 |

表3.2 会议信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 索引 | 注释 |
| meeting\_id | INT | 10 | 主键 | 会议ID |
| meeting\_url | BIGINT | 20 | 主键 | 会议号 |
| meeting\_theme | VARCHAR | 20 |  | 会议主题 |
| meeting\_host\_user\_id | INT | 10 | 外键 | 主持人用户ID |
| meeting\_is\_drawable | TINYINT | 4 |  | 与会者默认能否使用白板绘画： 1：不能 2： 可以 |
| meeting\_is\_talkable | TINYINT | 4 |  | 与会者默认能否聊天： 1：不能 2： 可以 |
| meeting\_is\_add\_to\_calendar | TINYINT | 4 |  | 是否添加到日历提醒：1：不添加 2：添加 |
| meeting\_password | VARCHAR | 32 |  | 登录密码 |
| meeting\_start\_time | DATETIME | 32 |  | 会议预期开始时间 |
| meeting\_end\_time | DATETIME | 32 |  | 会议预期结束时间 |
| meeting\_status | TINYINT | 4 |  | 会议状态：  1：未开始并且未到期 2：未开始并且过期了 3：正在进行 4：开会结束  5：锁定 |
| event\_id | BIGINT | 20 |  | 日历事件ID |
| meeting\_desc | TEXT |  |  | 会议描述 |

表3.3 用户参加会议表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 索引 | 注释 |
| user\_and\_meeting\_id | INT | 10 | 主键 | 用户参加会议ID |
| check\_in\_type | TINYINT | 4 |  | 用户参会类型：1 ：与会 2：主持会议 |
| check\_in\_time | DATETIME | 32 |  | 最新入会时间 |
| check\_out\_time | DATETIME | 32 |  | 最新离会时间 |
| bd\_user\_user\_id | INT | 10 | 外键 | 参会者的用户ID |
| bd\_meeting\_meeting\_id | INT | 10 | 外键 | 参会者进入会议的会议ID |

表3.4 联系人信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 索引 | 注释 |
| friend\_id | INT | 10 | 主键 | 联系人ID |
| response\_status | TINYINT | 4 |  | 回复状态：1：未回复 2：拒绝 3：同意 4.好友关系已经删除 |
| message\_time | DATETIME | 32 |  | 消息时间 |
| bd\_user\_user\_id | INT | 10 |  | 用户1的ID |
| bd\_user\_user\_id1 | INT | 10 |  | 用户2的ID |

表3.5 版本更新表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 索引 | 注释 |
| version\_id | INT | 10 | 主键 | 用户ID |
| app\_name | VARCHAR | 32 |  | 应用名称 |
| server\_version | VARCHAR | 32 |  | 服务器端APP版本号 |
| last\_force | TINYINT | 4 |  | 是否强制更新：0：不强制 1：强制 |
| server\_flag | TINYINT | 4 |  | 标记 |
| update\_url | TEXT |  |  | 新版本APP下载地址 |
| upgrade\_info | TEXT |  |  | 更新提示信息 |
| update\_time | DATETIME | 32 |  | 版本发布时间 |

SQLite中数据库包括一张表：如表3.3.6 。

表3.3.6 系统消息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 索引 | 注释 |
| \_id | INTEGER | 根据值的大小存储在 1、2、3、4、6 或 8 字节中 | 主键 | 联系人ID |
| EMAIL | TEXT |  |  | 用户邮箱 |
| TITLE | TEXT |  |  | 消息标题 |
| CONTENT | TEXT |  |  | 消息内容 |
| FAMILY\_NAME | TEXT |  |  | 用户的姓 |
| GIVEN\_NAME | TEXT |  |  | 用户的名 |
| FEATURE | TEXT |  |  | 消息类别 |
| AVATAR | TEXT |  |  | 用户头像地址 |
| STATUS | INTEGER | 根据值的大小存储在 1、2、3、4、6 或 8 字节中 |  | 消息状态：0：初始态 1：已同意 2：已拒绝 |
| MSG\_TIME | INTEGER | 根据值的大小存储在 1、2、3、4、6 或 8 字节中 |  | 消息到达时间 |
| TAG | TEXT |  |  | 消息全局唯一标识：用户邮箱+时间戳毫秒值 |

* 1. 综合考虑
     1. **稳定性和可扩展性**
     2. **性能分析**
     3. **安全性**
     4. **命名规范**

# 详细设计

* 1. 模块设计
     1. **模块概览**

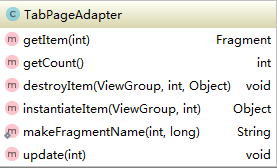
表4.1 模块一览表

|  |  |  |
| --- | --- | --- |
| 序号 | 模块名称 | 功能描述 |
| 1 | 注册 | 用户首次使用软件，需要注册个人账户，需要使用有效的邮箱进行注册，系统会发送验证码到该邮箱，用户查看后，填写正确的验证码，即可开始注册，设置头像、姓名、密码后，注册完成。 |
| 2 | 登录 | 1. 用户注册完成后，就会直接登录进去 2. 用户在同一设备上一次登录后，下次使用会自动使用TOKEN登录 3. 用户在注销后或在新设备上使用本软件时，需要使用用户邮箱、密码进行登录 |
| 3 | 重置密码 | 如果用户忘记了自己的登录密码，用户可以通过邮箱重置密码，和注册过程相似，也是通过邮箱接受验证码，填写正确的验证码后可以设置新的登录密码，重置之后会跳到登录界面。 |
| 4 | 更改个人信息 | 用户登陆后，可以更改自己的头像，用户名，密码。更改头像可以通过拍照和在相册中选择图片的方式；更改密码需要填写旧的密码、新密码、确认密码。 |
| 5 | 检查更新 | 用户点击版本更新栏，获取服务器最新版本信息，如果比当前版本新，就会提醒用户更新，更新分为强制性更新和选择性更新，用户点击更新，开始下载，下载完毕跳安装界面，确认后进行安装。 |
| 6 | 用户反馈 | 用户可以反馈自己的使用感受、建议等，不超过150字，反馈内容会通过服务器以邮件的形式发送给系统管理员。 |
| 7 | 查看隐私保护策略 | 用户可以查看软件开发者的隐私保护策略、免责条款等。 |
| 8 | 安排会议 | 用户可以制定会议计划，需要设置会议的主题、入会密码、预期开始时间、预期结束时间、与会者默认能否使用白板绘画、与会者默认能否使用会内群聊、是否添加到日历事件提醒，会议计划会保存到服务器数据库。 |
| 9 | 编辑会议 | 用户可以编辑安排好的会议，重新设置安排会议时候设置的那几项内容，数据会更新到服务器，日历事件提醒相应也会更新。 |
| 10 | 查看会议 | 用户可以查看自己安排的会议列表，点击列表项，进入会议详细信息；点击列表项中的开始按钮，可以开始会议。 |
| 11 | 删除会议 | 用户可以在查看会议详细信息时删除该会议安排。 |
| 12 | 加入会议 | 1. 用户可以通过会议号、密码加入会议 2. 用户可以点击联系人的加会邀请通知栏通知，直接加入会议 |
| 13 | 召开会议 | 用户可以召开会议而不经过安排那一步，这样召开的会议的设置和用户默认会议设置一致。 |
| 14 | 会外会议邀请 | 用户在查看自己安排好的会议时候，在详细会议信息页面，可以邀请别人加会，邀请方式包括邮箱邀请、短信邀请、复制邀请内容到剪贴板。邮箱邀请是通过用户手机上安装的邮箱客户端进行发送，邮件内容是系统生成的，收件人需要用户自己填写；短信邀请是通过用户手机上的短信应用发短信进行邀请，内容是系统生成的，收信人需要用户自己填写。 |
| 15 | 会内会议邀请 | 包括会外会议邀请的三种邀请方式，增加“联系人邀请”：用户可以通过浏览、查询联系人列表，选择需要邀请的联系人（多选），点击“邀请”，即可通过服务器推送邀请信息给他们，他们收到后点击即可加入会议。 |
| 16 | 共享资源 | 用户作为主持人，可以共享资源，可以共享图片或者网页，共享图片需要选择一张相册图片，之后可以进行移动、缩放等，使要显示的内容出现在理想位置；共享网页需要在输入框输入网址，之后可以像使用浏览器一样进行浏览，直到到了目标分享的网页。点击进入白板，分享的内容就会作为白板的背景出现在屏幕上，后面就可以在白板上绘画了。分享完毕，退出白板，点击结束分享，分享结束，白板背景清空。可以任意次共享、结束共享。 |
| 17 | 会议锁定 | 用户作为主持人，在会议中计划参与者到齐时，可以点击锁定会议，这样任何人不能在进入会议，保证了会议的安静。需要再加人时候，可以取消锁定。 |
| 18 | 查看参与者列表 | 用户在会议中可以查看参与者列表，能看到参与者总人数，参与者头像、姓名，“主持人”和“我”要特别标出。 |
| 19 | 控制与会者权限 | 用户作为主持人，在会议中，在参与者列表中，可以点击与会者列表项的“话筒”按钮来控制他能否发送群聊信息，但是与会者一直可以收到看到群聊信息；点击“画笔”按钮，控制他能否使用白板进行绘画，但是用户一直可以收到看到白板的内容；点击列表项，弹出“踢人”确认框，确认则把他踢出会议。 |
| 20 | 使用白板 | 功能包括画笔、简单几何图形（圆、方框、箭头、直线等）、添加文字、橡皮、撤销、清空等，还可以对白板进行屏幕录制，用户在白板上的绘画等操作会实时同步到其他人。 |
| 21 | 使用群聊 | 可以发送文字、表情、图片、语音，可以看到消息发送者的头像、姓名、消息时间 |
| 22 | 添加联系人 | 用户a可以通过邮箱申请添加联系人用户b，用户b可以在系统消息列表中选择接受申请或拒绝申请，系统会把结果反馈给用户a，如果是接受，则双方的联系人列表中会增加对方。 |
| 23 | 删除联系人 | 用户可以在联系人列表中长按想要删除的联系人所在的列表项，弹出删除联系人确认框，点击确认即可删除联系人，对方会收到被删除的提醒，同时双方的联系人列表中会删除对方。 |
| 24 | 查看联系人列表 | 用户可以查看联系人列表，列表按照姓名首字母排列，用户可以通过姓名进行联系人查找。 |

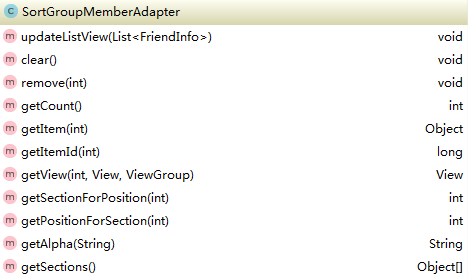
* + 1. **模块具体设计**
    2. **建立静态结构模型**

1. 识别系统中的类
2. 定义系统界面类
   * WelcomeActivity：系统的欢迎界面，进行用户是否在此设备上已登录检查，如果登陆过直接跳到主页面，否则跳到登录界面。
   * LoginActivity：登录页面，可以输入邮箱、密码进行登录，可以点击注册、重置密码进行相关操作。
   * RegisterActivity：注册页面，输入邮箱，获取验证码，填写验证码，点击下一步进入设置个人信息。
   * MainActivity：系统的主页面，包括会议页、联系人页、设置页。
   * MainActivity：主页面，左右滑动，包括会议页、联系人页、设置页。
   * MeetingFragment：主页中的会议页部分，可以进入加入会议、召开会议、安排或主持会议。
   * ContactFragment：主页中的联系人页部分，可以进入查看联系人，删除联系人，通过姓名查询联系人
   * SettingsFragment：主页中的设置页部分，可以进入我的账户、会议设置、关于软件。
   * JoinMeetingActivity：加会页面，可以通过会议号、入会密码加入会议。
   * ArrangeOrHostMeetingActivity：包括主持会议、进入安排会议、进入查看我的会议列表。
   * ArrangeMeetingActivity：安排新增会议、编辑会议。
   * SetUserInfoActivity：注册的时候保存用户个人信息（头像、姓名、密码）。
   * SetUserInfoActivity：查看我安排的会议列表，可以点击进入会议详细信息、点击开始来开始会议。
   * MeetingInfoActivity：查看我安排的会议详细信息，可以邀请别人、删除会议、添加日历提醒、编辑会议、开始会议。
   * MessageListActivity：查看系统信息列表，可以删除消息、同意或拒绝添加联系人。
   * MeetingActivity：会议界面，可以进行会内管理。
   * InviteActivity：选择联系人推送加会邀请页面。
   * ChatActivity：群聊页面。
   * SettingsMyInfoActivity：我的账户页面。可以进入设置头像、姓名、密码。
   * SettingsMeetingActivity：设置默认会议设置。
   * SettingsAboutActivity：关于软件页面。
   * FeedbackActivity：用户反馈页面。
   * AlterPasswordActivity：修改密码、重置密码。
3. 定义系统适配器类

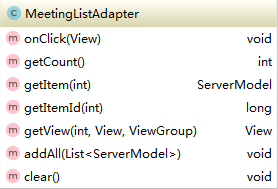
* TabPageAdapter：MainActivity的适配器类。

****

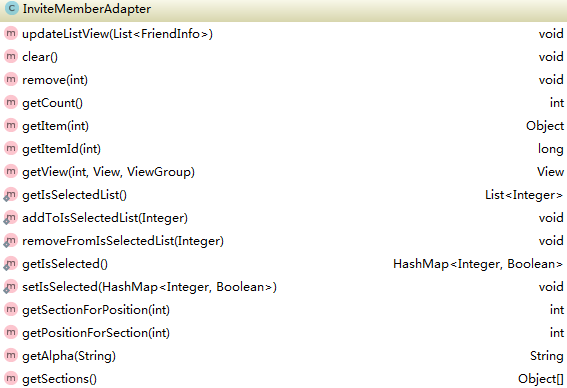
* SortGroupMemberAdapter：联系人列表的适配器类。

****

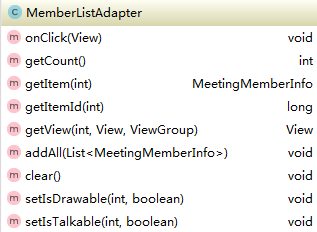
* MeetingListAdapter：我的会议列表的适配器类。

****

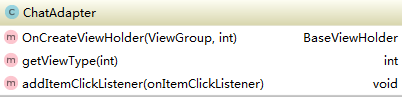
* InviteMemberAdapter：选择联系人推送加会邀请页面的适配器类。

****

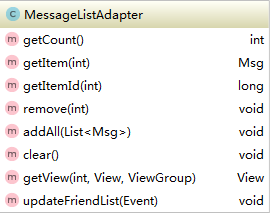
* MemberListAdapter：会议参与者列表的适配器类。

****

* ChatAdapter：聊天页面的适配器类。

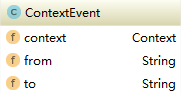
****

* MessageListAdapter：系统消息列表的适配器类。

****

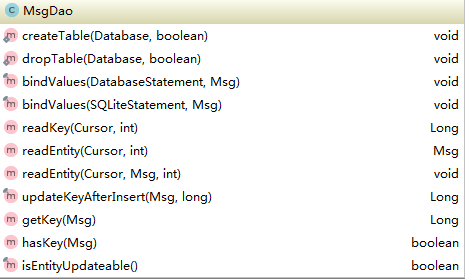
1. 定义数据实体类

* User：用户信息实体类。
* Meeting：会议信息实体类。
* FriendInfo：联系人信息实体类。
* MessageInfo：系统消息实体类。
* MeetingMemberInfo：会议参与者状态信息实体类。
* MeetingMessage：群聊消息实体类。
* ContextEvent：页面间事件通知实体类。

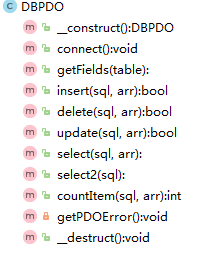
****

1. 定义数据库操作类

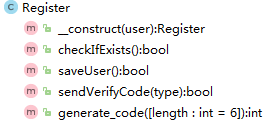
* Msg：客户端数据库系统消息数据库操作类。

****

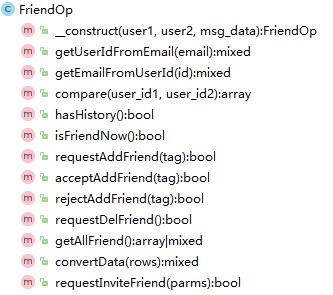
* DBPDO：服务器数据库底层操作类。

****

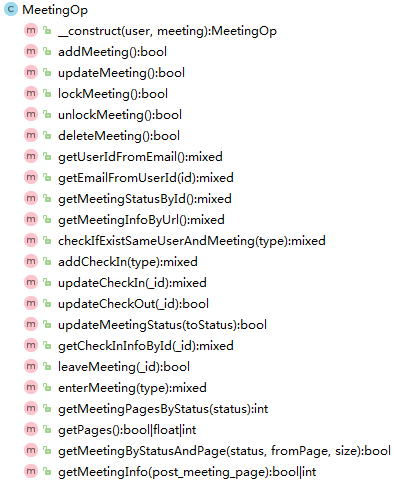
* Register：服务器数据库注册操作类。

****

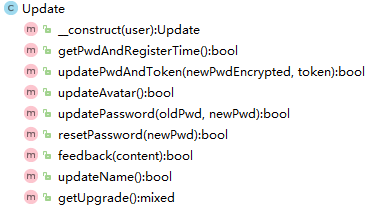
* FriendOp：服务器数据库联系人管理操作类。

****

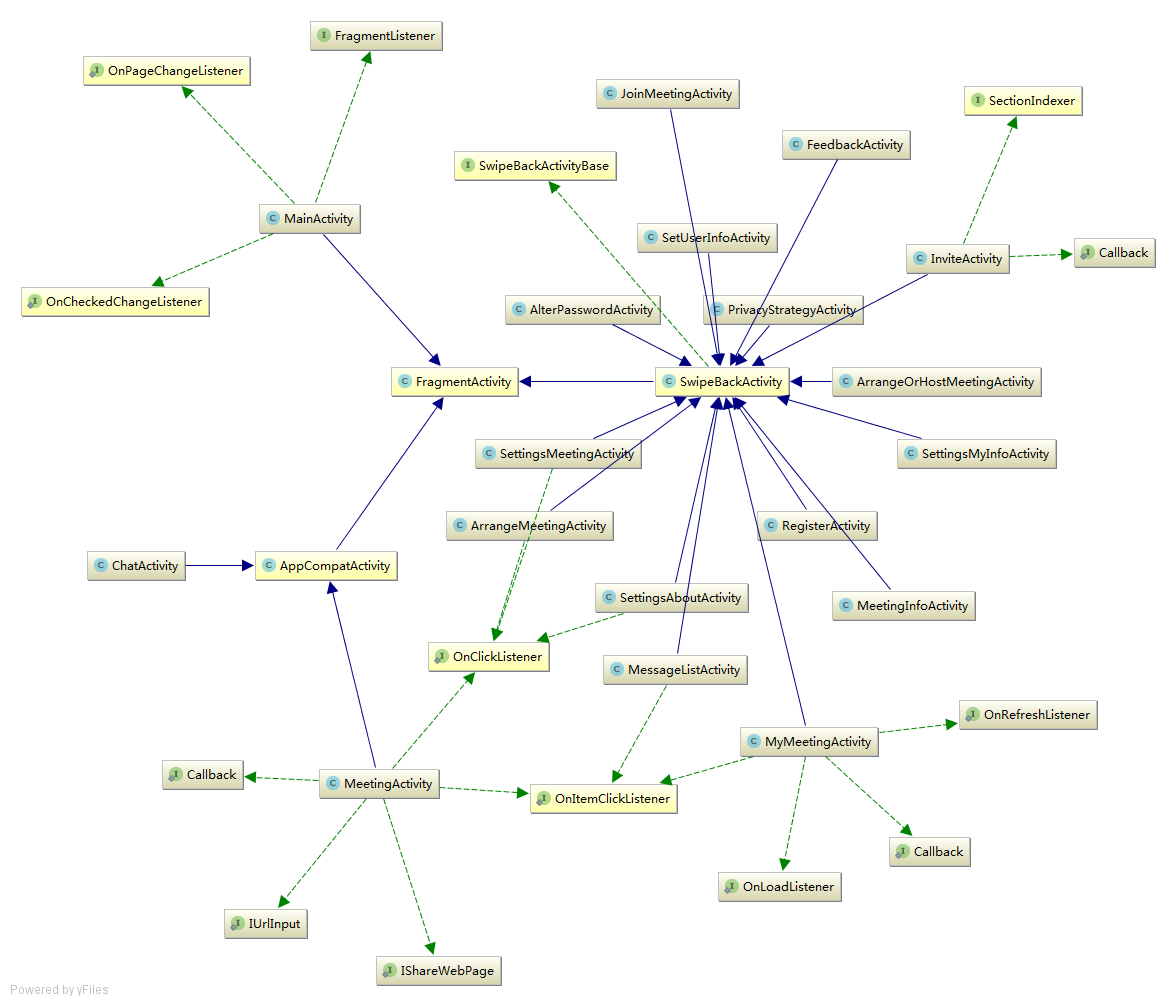
* MeetingOp：服务器数据库会议管理操作类。

****

* Update：服务器数据库更改个人信息操作类。

****

1. 建立类图
2. 用户界面类的类图



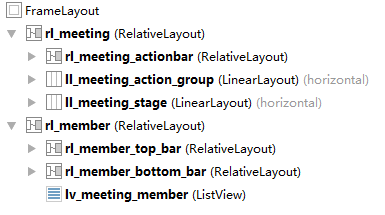
1. 实体类和数据操作的类图（服务器端）



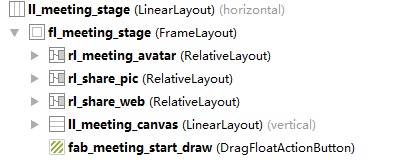
* + 1. **关键界面设计**
       1. **设计**

1. 会议主界面

会议页面的客户端界面设计比较复杂。根布局采用FrameLayout，内部分两层，每一层是一个RelativeLayout，上面一层是会议主页面，下面一层是参与者列表页面，这里主要讲会议主页面。



会议主页面由三部分组成，其中rl\_meeting\_actionbar是页面最上面的标题栏，显示会议号和离会按钮；ll\_meeting\_action\_group是页面下面的按钮栏，如果是主持人进会后会看到三个按钮，依次是“参与者”、“共享”、“锁定会议”，如果是与会者只能看到“参与者”按钮；页面中央是ll\_meeting\_stage，这部分分为四层。

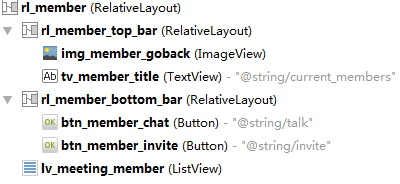


其中rl\_meeting\_avatar是用户进入会议主页面看到的样子，页面正中央有一个圆形的图片，显示的是用户的头像，默认显示；rl\_share\_pic是用户共享图片时候的舞台的样子，默认隐藏； rl\_share\_web是用户共享网页时候的舞台的样子，默认隐藏；ll\_meeting\_canvas是用户进入白板页面看到的样子，默认隐藏。除此之外，舞台还有一个浮动按钮fab\_meeting\_start\_draw，是通过自定义控件实现的，该控件的继承自FloatActionButton，可以通过setTitleBarSize和setBottomBarSize设置它活动范围的上界和下界，之后用户就只能在该范围内移动这个按钮，并且通过监听TouchEvent，实现了按钮移动后会自动贴屏幕最右边，从而提高了舞台空间的利用率。进入会议主页面，可以通过单击悬浮按钮，之后单击PopMenu “进入白板”，即可进入白板页面，进入白板后，把之前会议主页面的顶部标题栏、底部按钮栏、中间头像会隐藏（通过setVisibility(View.GONE)），把ll\_meeting\_canvas设为可见（通过setVisibility(View.VISIBLE)）），舞台便扩大为全屏，白板页面是一个WebView控件，显示的是从服务器获取到的白板网页，因为需要进行JavaScript交互等，需要对WebSettings 进行如下设置。

|  |  |  |
| --- | --- | --- |
| 方法名 | 作用 | 设置值 |
| setSaveFormData | 设置是否保存表单数据 | false |
| setJavaScriptEnabled | 设置是否启用JavaScript | true |
| setSupportZoom | 设置是否支持用户手动缩放 | false |
| setUseWideViewPort | 设置是否页将图片调整到适合webview的大小 | true |
| setLoadWithOverviewMode | 设置是否把页面缩放至屏幕的大小 | true |
| setCacheMode | 设置webview缓存类型 | false |
| setAppCacheEnabled | 设置是否使用APP缓存 | false |
| setDefaultTextEncodingName | 设置编码格式 | "utf-8" |

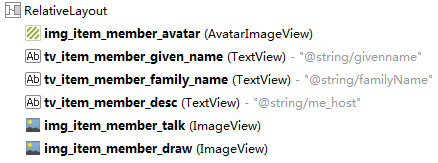
在白板页面，单击悬浮按钮，会弹出PopMenu，包括两个菜单项：“退出白板”、“开始录屏”，单击“退出白板”，即可暂时退出白板，回到会议主页面，白板内容并不会清空，只是隐藏了而已，用户还可以再次进入白板；单击“开始录屏”，则会开始屏幕录制。

1. 参与者列表界面



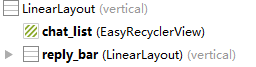
结构包括三部分，顶部的标题和返回按钮，顶部标题会显示参与者的总人数，中间的联系人列表，可以用来浏览参会者的头像、姓名，“主持人和“我”会特殊标出，主持人还可以在这里控制与会者的权限。”底部的按钮栏，包括“聊天”和“邀请”，分别用户进入聊天页面和会内邀请别人加会。

中间的列表是一个ListView，其中每一行是一个Item，结构如下图：



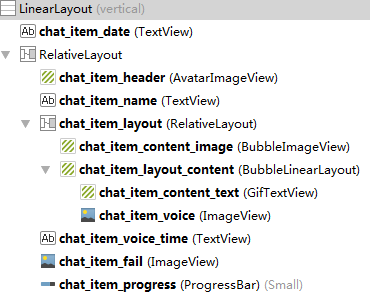
最左侧是显示头像的AvatarImageView，紧接着是显示姓、名的TextView，最右侧是权限控制按钮，依次是显示当前是否可使用白板绘画的图片和当前是否可发送聊天的图片。

1. 聊天界面



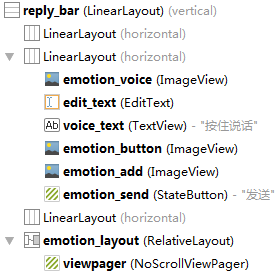
根布局是一个LinearLayout，里面分为两部分，上面是聊天内容，用EasyRecyclerView实现，底部是工具栏（输入框，语音输入按钮，表情选择，本地图片选择等）。

因为自己发送和接受别人的消息在界面体现上是不一样，所以聊天列表的Item也分为两种，一种是send,一种是accept，它们两个的组成结构是一样的，只不过相对位置不同，它们相对聊天内容界面纵向中轴对称。所以这里只对send Item进行分析。



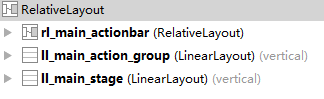
从上往下，最上边是显示消息时间的TextView，它下面是显示消息发送者信息和消息内容的RelativeLayout，其中chat\_item\_header显示发消息用户的头像，chat\_item\_name显示发消息用户的姓名，chat\_item\_content\_image显示发送的图片消息，chat\_item\_content\_text显示聊天文字消息（GifTextView控件支持gif图片表情），chat\_item\_voice显示语音消息的图片，chat\_item\_voice\_time显示语音消息的时长，chat\_item\_fail显示消息发送失败的提示图片，chat\_item\_progress显示消息发送进度。

底部的reply\_bar结构如下：



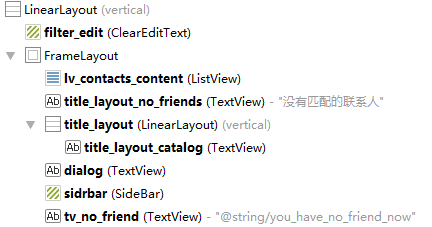
从上到下分为四块，第一块是一个水平分割线，第二块是按钮栏，其中emotion\_voice是语音消息按钮，edit\_text是文本（包括gif表情）消息输入框，voice\_text是用户发送语音时候需要长按的按钮，emotion\_button是打开表情选择面包的按钮，emotion\_add是打开其他操作（如 选择本地图片）选择面板的按钮，emotion\_send是消息发送按钮，第三块是一个水平分割线，第四块是选择面板（表情选择、其他操作选择），每个选择面板是一个fragment，然后把它们放到一个NoScrollViewPager中。

1. 软件主页面



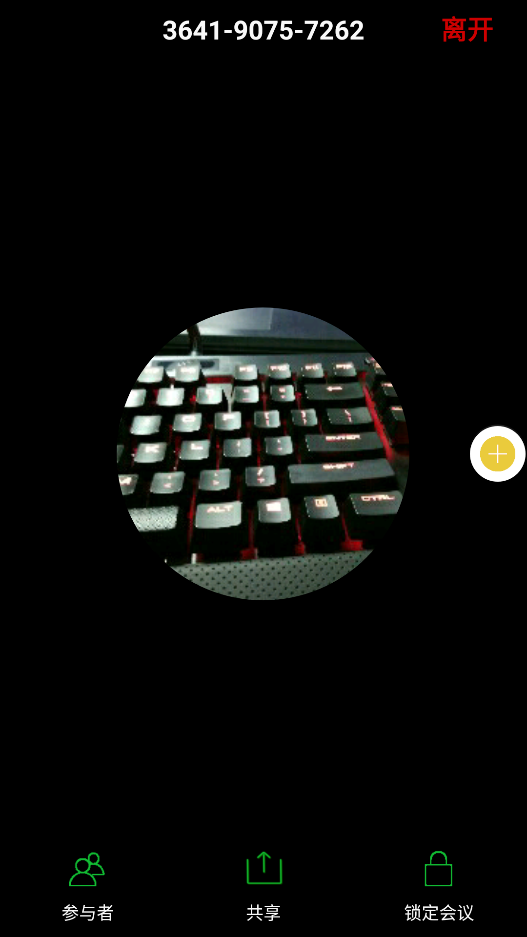
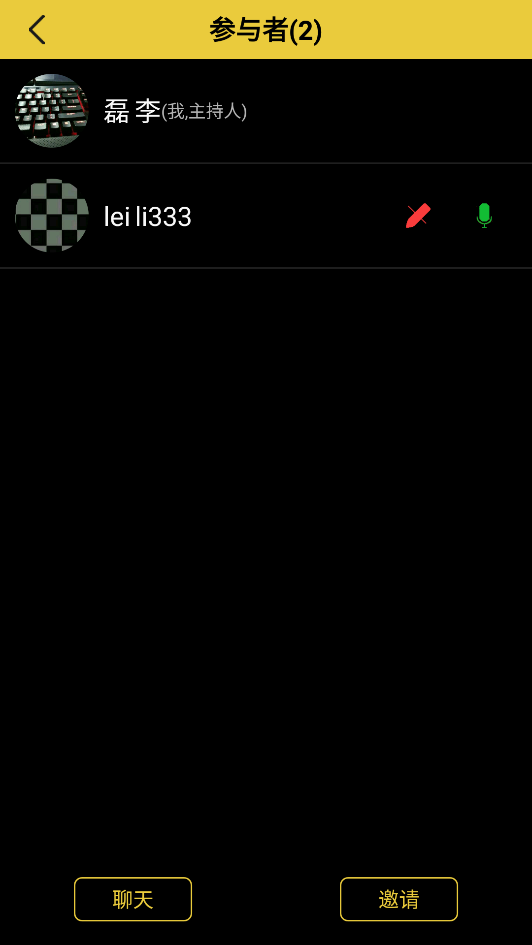
软件主页面分为三个页面，可以左右滑动切换，也可以点击按钮切换，所以中间ll\_main\_stage部分采用的是ViewPager来实现，每个页面是一个fragment，上面是标题栏（如果是联系人页面，还有一个“添加联系人”按钮），下面是按钮栏，用一个RadioGroup实现，包含三个RadioButton，分别是“会议”、“联系人”、“设置”。

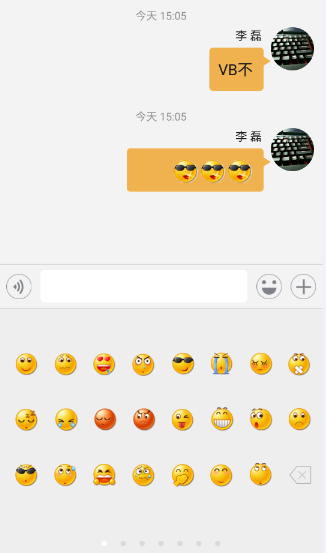
* “会议”fragment是一个ListView，它的Item的左边为图标和功能文字描述，最右边有一个TextView，用来显示未读消息数。
* “设置”fragment自上而下分为四块，每一块是一个LinearLayout布局，第一块是“账户设置”，显示用户的头像、姓名、邮箱，点击进入“我的设置”；第二块是“会议”设置；第三块是“关于”；第四块是版权信息。
* “联系人”fragment，按照联系人姓名首字母分组，可以通过滑动最右侧字母列表快速导航到想找的联系人，也可以直接输入联系人姓名关键字进行搜索。

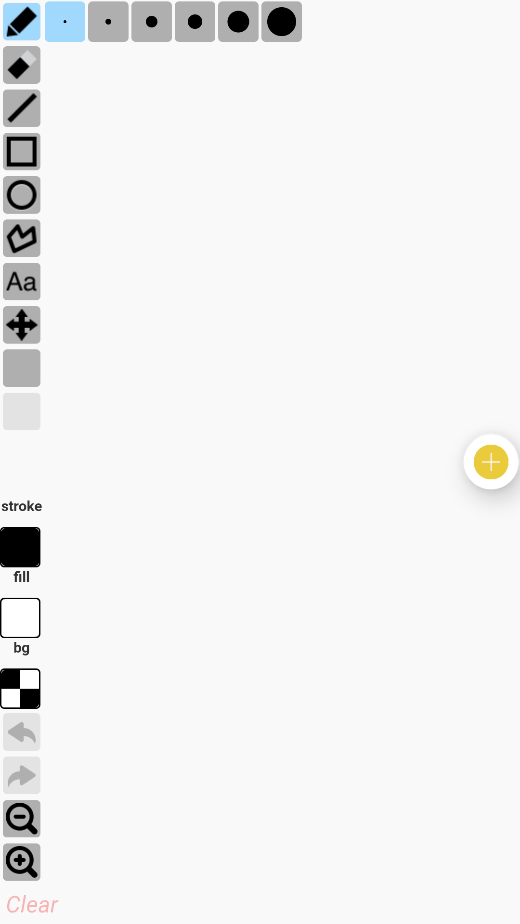


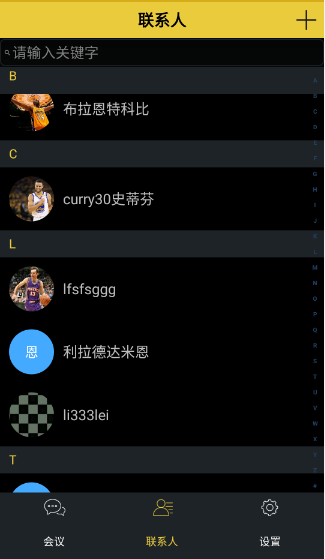
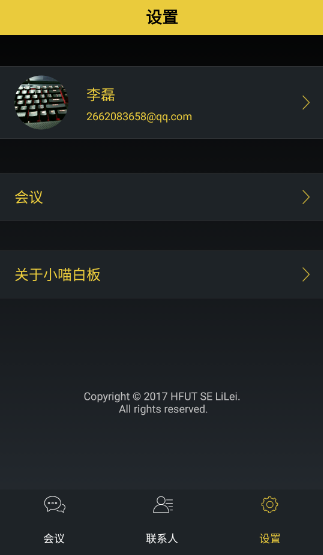
最上面的ClearEditText是联系人搜索框，下面的lv\_contacts\_content是联系人列表，title\_layout\_no\_friends是在搜索没有结果时候提示用户的，title\_layout\_catalog显示分类字母，dialog显示滑动右侧字母导航列表时候的当前字母，sidrbar就是右侧的字母快速导航栏，tv\_no\_friend是在用户没有联系人时候显示提示信息的。

* + - 1. **效果**

* + 1. **关键过程设计**

1. 白板
2. 绘画部分（关键函数）

* init(element, options={})：初始化白板

element:与白板绑定的html节点；

options:设置选项，包括：imageURLPrefix（图片文件的路径，最后不加斜杠）、imageSize（图片大小，例如{width: 500, height: null}）、primaryColor（开始是画笔的颜色，默认黑色）、secondaryColor（开始时的填充色，默认白色）、backgroundColor（开始时的背景色，默认透明）、snapshot（初始时的白板数据）、toolbarPosition（工具栏的位置）、tools（工具列表）、strokeWidths（画笔粗细值选择列表，默认是[1, 2, 5, 10, 20, 30]）、defaultStrokeWidth（开始时的画笔粗细，默认5）。

* on('drawingChange', callback)：设置白板内容变化监听器

“drawingChange”：监听的事件名，指的是白板内容的改变，不包括工具中画笔颜色和填充色的变化；

Callback：处理事件的回调函数。

* setPan(x, y)：把白板的内容的移动到指定位置（x,y）。
* setZoom(zoom)：把白板的内容缩放到指定值。
* setColor(colorName, colorValue)：设置某种颜色的CSS色值

colorName ：'background', 'primary', 或'secondary'；

colorValue ：CSS色值。

* getColor(colorName)：获取指定名称颜色的CSS色值

colorName ：'background', 'primary', 或'secondary'。

* getPixel(x, y)：获取指定位置（x,y）的像素的CSS色值。
* clear()：清空白板内容。
* undo()：撤销上一步。
* redo()：重做上一步。
* getSnapshot(keys=['shapes', 'imageSize', 'colors', 'position', 'scale', 'backgroundShapes'])：获取当前白板的Json对象，默认包括对象全部内容，如果不需要这么多，可以通过keys来指定需要获取的信息类别。
* loadSnapshot(snapshot)：加载白板Json对象snapshot到白板。
* teardown()：销毁白板。

1. 通信部分

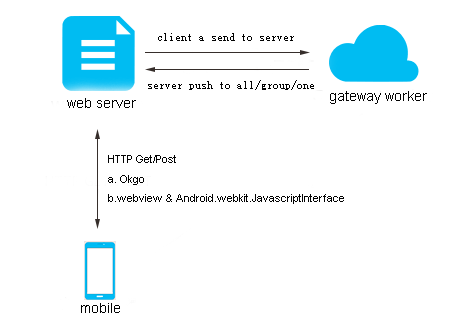


图4.22 白板的消息同步机制

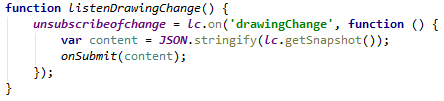
1. 首先用户通过与会者加会或者主持人的方式，完成入会信息的登记与入会凭证的验证。

* 主持人进会需要携带的信息包括：TOKEN、用户邮箱、会议ID、会议号、入会类型（2代表主持人），通过HTTP发送到服务器，服务器首先要检查这些参数是否齐全，其中TOKEN参数的检验，是系统合法访问的检验，如果用户携带的TOKEN在服务器的SESSION中可以找到，或者SESSION中找不到但是在数据库中可以找到，即证明是合法访问，方可进行下一步操作。首先通过用户邮箱获取用户ID，如果失败，返回错误信息，如果成功，需要查询Meeting表，是否有一个会议的会议ID为这个会议ID并且会议主持人为该用户ID并且会议状态为”未开始且未到期”（meeting\_status=1）并且用户参加会议表不存在该用户主持该会议的记录,如果不能满足这些条件，返回“无效的会议”错误信息给用户，如果满足，进行该用户的入会登记（在用户参与会议表中增加一条记录，记录用户ID,会议ID,主持人入会时间，check\_in\_type=2）,然后更新该会议的meeting\_status为3，代表会议正在进行中，到此，主持人进会验证和登记结束，返回“主持人进会成功”给用户。
* 加会者进会需要携带的信息包括：TOKEN、用户邮箱、会议号、入会密码、入会类型（1代表与会者）。TOKEN检查和参数完整性检查成功之后，首先通过用户邮箱获取用户ID，如果失败，返回错误信息，如果成功，通过会议号查询会议表获取会议详细信息，通过其中的主持人用户ID获取主持人用户邮箱，之后根据该会议的状态进行不同操作：如果meeting\_status=1，也就是会议未开始未过期，回复用户“会议还没开始，须等待主持人进入”；如果meeting\_status=2，会议过期，回复用户“无效的会议”；如果meeting\_status=4，回复用户“会议已结束”；如果meeting\_status=3，查询用户参与会议表中有没有该用户加入该会议的记录，如果没有，进行该用户的入会登记（在用户参与会议表中增加一条记录，记录用户ID,会议ID,加会者入会时间，check\_in\_type=1），如果有，则更新入会时间为当前时间。

1. 入会登记之后，进入会议页面。首先初始化各种布局、控件，设置监听器，设置wv\_meeting\_canvas的WebSettings，构造URL，通过拼接URL的方法，把需要的参数都拼接在“http://118.89.102.238/view/index.php”后面，需要的参数如下： post\_token、post\_user\_email、post\_user\_family\_name、post\_user\_given\_name、post\_user\_avatar、post\_meeting\_id（会议ID,主持人必须带的参数）、post\_meeting\_is\_drawable、post\_meeting\_is\_talkable、post\_meeting\_check\_in\_type、post\_meeting\_url、post\_meeting\_host\_email（主持人邮箱，与会者必须带的参数）。然后调用WebView的loadUrl方法，显示网页，同时给wv\_meeting\_canvas添加一个JavascriptInterface。

C:\Users\Bryant\Desktop\QQ截图20170520162139.png

这样，在加载网页的时候，首先会进行TOKEN检查和参数完整性检查，如果通过，把参数保存到SESSION中，然后开始执行index.php代码中的JavaScript，根据check\_in\_type和isDrawable的值，创建白板lc（如果check\_in\_type=2创建可画白板，如果check\_in\_type=1且isDrawable为True，创建可画白板，如果check\_in\_type=1且isDrawable为False，创建不可画白板），然后设置白板内容变化监听器的回调函数。



然后和Socket Server建立WebSocket连接，并和Gateway Worker 的Event.php 进行通信，在Event.php中onConnect($client\_id)和onClose($client\_id)函数对WebSocket的建立和关闭进行监听和处理，而onMessage($client\_id, $message)中首先判断$message是否为合法Json格式，合法则通过判断功能参数进行不同的消息转发任务。

通信详细过程如下：

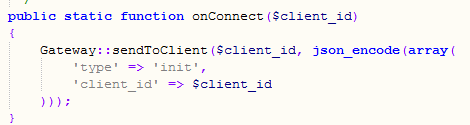
* 首先在html的body加载的时候建立WebSocket连接



在connect()中，设置Url、连接打开、有消息、关闭、出错的回调函数。



随着连接的建立首先调用的是WebSocket服务器的event.php中的onConnect函数



这里面 的$client\_id是Gateway Worker自动生成的连接的UID，可以通过它给该用户发消息，这里发送“init”消息给index.php，消息的内容是“client\_id”,index.php的onmessage函数接收到服务器发送的消息，并且把“client\_id”赋值给变量myid。接着index.php的onopen函数被调用，发送”login”消息（Json格式，包括会议的详细信息）给服务器,Event.php的onMessage函数接收“login”消息，把消息的数据依次保存到SESSION（这里需要搞清楚，WebServer和SocketServer的SESSION是完全不先关的两个SESSION，所以需要的信息在两个SESSION中要各自保存），并且把用户邮箱作为用户另一个标识，通过Gateway::bindUid($client\_id, $client\_email)把用户邮箱和$client\_id绑定，把会议号作为房间号，通过Gateway::joinGroup($client\_id, $room\_id)把$client\_id和房间号绑定，这样后面可以给指定会议号或指定用户邮箱的用户发消息。之后通过Gateway::sendToGroup($room\_id, $message)群发自己的“login”消息给房间内的所有人，用于同步参与者列表，index.php收到“login”消息，判断如果check\_in\_type=1并且email == data['client\_email']，也就是用户本身是与会者，并且收到了自己的“login”消息时，需要向主持人请求白板目前的数据，通过getInitCanvasData函数发送"getInitCanvasData"消息（包括"from\_client\_email":myemail、"to\_client\_email":host\_email）给服务器，再由服务器的onMessage函数调用sendToUid转发给主持人,主持人的index.php接收到消息，首先调用JSON.stringify(lc.getSnapshot())函数获得白板数据datajson（Json格式），然后发送"CanvasData"消息（包括"from\_client\_email"、“to\_client\_email"、"content"：datajson）给服务器，服务器根据“to\_client\_email"转发给指定用户，该与会者接收到"CanvasData"消息，通过lc.loadSnapshot(JSON.parse(newdata))加载白板数据，因为主持人可能在共享资源（无论是共享图片还是共享网页，都是当主持人进入白板的时候同步一个当前共享舞台的截屏给所有与会者，然后把该截屏图片作为白板的背景图片），当用户刚进入会议时，还需要主动向主持人请求共享数据，所以请求完白板数据后通过getInitShareData函数发送"getInitShareData"消息(包括"from\_client\_email":myemail、"to\_client\_email":host\_email）给服务器并转发给主持人，主持人的index.php接收到"getInitShareData"消息，通过之前实例化过的JavascriptInterface 直接调用Android的MeetingActivity的getSharePic函数。



为了可以实现这种调用，getSharePic函数的声明需要满足如下形式，后面其它需要类似调用的函数的声明也要满足一条件（函数需要为public且有下图的注解）：



主持人的getSharePic函数判断当前主持人的是否在共享，共享的类型是哪种，如果在共享，通过



或



获取相应的截屏，并转换成Base64编码的字符串；如果没有在共享，Base64字符串赋值为“nothing”。然后调用index.php的syncPicToNewer(to\_client\_email, base64Str)函数，要实现Android对WebView中javaScript函数的调用，需要以如下形式调用：



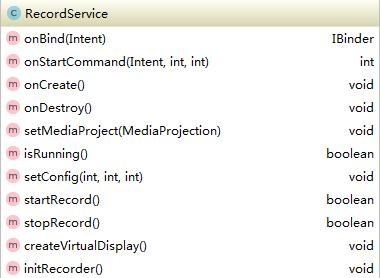
也就是拼接一个内容为javascript:函数名（参数）的字符串，然后通过WebView的loadUrl进行调用。

通过这个调用，把Base64字符串传给了主持人的index.php，然后通过服务器转发"ShareData"消息（包括"from\_client\_email"、"to\_client\_email"、"content":base64Str）给相应的与会者，该与会者的index.php接收到"ShareData"消息，调用Android的initShareContent(final String str)函数，隐藏会议主页面，设置白板页面可见，设置白板的背景图片，这样用户加会的白板同步完成，新入会的与会者看到了白板当前的样子，接下来就可以进行白板操作了，白板变化会触发之前给画板改变监听器设置的回调，可以在得到白板改变后的数据（Json格式），并调用onSubmit函数发送"say"消息（包括'from\_client\_id'、'from\_client\_name'、'to\_client\_id'：‘all’、 'content'：白板数据，“time”:发送时间）给会议房间内的所有人，该会议房间内所有用户的index.php都收到"say"消息，首先获取到白板数据，然后移除白板变化监听的回调，判断消息的from\_client\_id和自己的myid是否相同，如果相同，代表这个"say"消息就是自己发出去的，不作操作；如果不同，则代表此消息是别人发来的，需要加载新的白板数据到白板。然后，重新为白板设置内容变化监听的回调。这样就完成了白板的通信部分。

1. 屏幕录制

目前只支持Android5.0 以上系统，因为在Android4.4之前，录屏的解决方案一般是读取 /dev/graphics/fb0 中的framebuffer，然后通过编码器转换成视频，但是需要root，且一些机型root后仍不成功；到了Android 4.4，谷歌开放了通过 ADB 命令录屏的方法，需要打开USB调试并连接PC，输入命令才行，且有时长最长三分钟的限制；Android5.0谷歌开放了屏幕采集接口，即通过 MediaProjection 和 MediaProjectionManager 进行录屏。具体操作包括：

* 申请权限：android.permission.RECORD\_AUDIO（录屏）和android.permission.WRITE\_EXTERNAL\_STORAGE（保存录像），且如果targetSdkVersion如果是23，还要进行动态权限申请。
* 创建一个service负责创建屏幕镜像、保存录像、停止录像、提示保存路径等操作。



* 设置服务连接与断开连接的回调函数，绑定RecordService和MeetingActivity。



* 获取 MediaProjectionManager。



* 构造捕获屏幕Intent。



* 取得 MediaProjection，并使用RecordService开始录屏，保存在本地
* 记录当前正在录制状态，如果用户再次点击悬浮按钮，PopMenu会有停止录制菜单项，点击，停止录制，并提醒用户保存的路径。

1. 会议内参与者同步

在白板的通信部分中提到过，新的与会用户会发送“login”消息给服务器，服务器转发“login”消息给会议房间内的所有用户，该消息

1. 会内共享资源
2. 会议内权限管理
3. 加会邀请
4. 群聊
5. 好友管理
6. 版本更新
   1. 接口设计
7. 发送验证码
8. URL：http://118.89.102.238/controller/register/send\_verify\_code.php
9. 请求方式：POST
10. 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数字段 | 类型 | 限制 | 说明 |
| post\_user\_email | String | 满足邮箱合法性检查的正则 | 用户邮箱 |
| post\_need\_feature | String |  | 功能参数："Register": 注册,"resetPassword":重置密码 |

1. 返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数字段 | 类型 | 说明 |
| Code | String | 100：成功  4004：验证码发送失败  4001：参数不完整 |
| msg | String | 错误信息 |
| data | Int | 数据 |

1. 注册
2. URL：http://118.89.102.238/controller/register/save\_user\_info.php
3. 请求方式：POST
4. 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数字段 | 类型 | 限制 | 说明 |
| post\_user\_email | String | 验证码，6位数字 | 用户收到的验证码 |
| post\_user\_family\_name | String | 最长15，可以包括汉字，英文字母，数字（不能数字开头） | 姓 |
| post\_user\_given\_name | String | 最长15，可以包括汉字，英文字母，数字（不能数字开头） | 名 |
| post\_user\_login\_password | String | 12位，可以字母开头，数字和下划线 | 登录密码 |

1. 返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数字段 | 类型 | 说明 |
| Code | String | 100:成功  4001：参数不完整  4007 :注册失败 |
| msg | String | 错误信息 |
| data | Json | {'token':, 'avatar':} |

1. 登录
2. URL：http://118.89.102.238/controller/login/user\_login.php
3. 请求方式：POST
4. 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数字段 | 类型 | 限制 | 说明 |
| post\_user\_email | String | 合法邮箱 |  |
| post\_user\_login\_password | String | 12位，可以字母开头，数字和下划线 | 登录密码 |

1. 返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数字段 | 类型 | 说明 |
| Code | String | 100:成功  4001：参数不完整  4008:用户名密码错误 |
| msg | String | 错误信息 |
| data | Json | {"token":,"familyName":, "givenName":,"avatar":  } |

1. 安排会议
2. URL：http://118.89.102.238/controller/meeting/host\_meeting.php
3. 请求方式：POST
4. 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数字段 | 类型 | 限制 | 说明 |
| post\_user\_email | String | 合法邮箱 |  |
| post\_token | String | 32位UID |  |
| post\_need\_feature | String |  | "add" ：安排会议 |
| post\_meeting\_theme | String |  | 会议主题 |
| post\_meeting\_is\_drawable | Int | 0或1 | 与会者默认是否可用白板绘画 |
| post\_meeting\_is\_talkable | Int | 0或1 | 与会者默认是否可发送群聊 |
| post\_meeting\_is\_add\_to\_calendar | Int | 0或1 | 是否添加会议提醒到日历 |
| post\_meeting\_start\_time | long |  | 会议预计开始时间 |
| post\_meeting\_end\_time | long |  | 会议预计结束时间 |
| post\_meeting\_password | String | 12位数字 | 入会密码, |
| post\_meeting\_event\_id | long | -1或正整数 | 日历提醒事件ID |
| post\_meeting\_desc | String |  | 会议描述 |

1. 返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数字段 | 类型 | 说明 |
| Code | String | 100:成功  4001：参数不完整  4017:安排会议失败 |
| msg | String | 错误信息 |
| data | Json | {'meeting\_id':,'meeting\_url':} |

1. 进入会议
2. URL：http://118.89.102.238/controller/meeting/enter\_meeting.php
3. 请求方式：POST
4. 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数字段 | 类型 | 限制 | 说明 |
| post\_user\_email | String | 合法邮箱 | 用户邮箱 |
| post\_token | String | 32位UID |  |
| post\_meeting\_check\_in\_type | Int | 1或2 | 1：与会者进会 2：主持人进会 |
| post\_meeting\_id（主持人必须） | Int |  | 会议ID |
| post\_meeting\_url | String |  | 会议号 |
| post\_meeting\_password（与会者必须） | String | 12位数字 | 入会密码 |

1. 返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数字段 | 类型 | 说明 |
| Code | String | 100:成功  4001：参数不完整  4015:主持人进会失败 4016：加会失败 |
| msg | String | 错误信息 |
| data | Int | 数据 |

1. 分页获取我的会议
2. URL：http://118.89.102.238/controller/meeting/host\_meeting.php
3. 请求方式：POST
4. 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数字段 | 类型 | 限制 | 说明 |
| post\_user\_email | String | 合法邮箱 | 用户邮箱 |
| post\_token | String | 32位UID |  |
| post\_need\_feature | String |  | "get"： 获取会议 |
| post\_meeting\_id（主持人必须） | Int |  | 会议号 |
| post\_meeting\_page | Int |  | 需要获取会议的起始页 |

1. 返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数字段 | 类型 | 说明 |
| Code | String | 100:成功  4001：参数不完整  4020:获取会议列表失败 |
| msg | String | 错误信息 |
| data | Json | [  {"meeting\_id":,"meeting\_url":,"meeting\_theme":,"meeting\_is\_drawable":,"meeting\_is\_talkable":,"meeting\_is\_add\_to\_calendar":,"meeting\_password":,"meeting\_start\_time":,"meeting\_end\_time":,"event\_id":,"meeting\_desc":},  ......  ] |

1. 发送聊天文件
2. URL：http://118.89.102.238/controller/meeting/send\_chat\_data.php
3. 请求方式：POST
4. 请求参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| 参数字段 | 类型 | 限制 | 说明 |
| post\_user\_email | String | 合法邮箱 | 用户邮箱 |
| post\_token | String | 32位UID |  |
| post\_meeting\_url | String |  | 会议号 |
| post\_need\_feature | String |  | 功能参数，'image'：图片，'voice':语音 |
| post\_chat\_data | File | 大小不超过10M | 聊天文件，语音或图片 |

1. 返回参数说明

|  |  |  |
| --- | --- | --- |
| 参数字段 | 类型 | 说明 |
| Code | String | 100:成功  4001：参数不完整  4025:发送文件失败 |
| msg | String | 错误信息 |
| data | Int | 数据 |

# 测试与调试

结论

结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论结论。

cjrg创建日期

ddzt订单状态

ddjg订单价格

ffcrq发车日期

fpzs票张数

fpmm取票密码

fplx票类型

fqph取票号

ckxm乘客姓名

sfzhm身份证号码

lxdh联系电话

mdddm到达站代码

致谢

致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢致谢

【参考文献】

1. 参考文献1
2. 参考文献2
3. 参考文献3
4. 参考文献4 ……

附录：