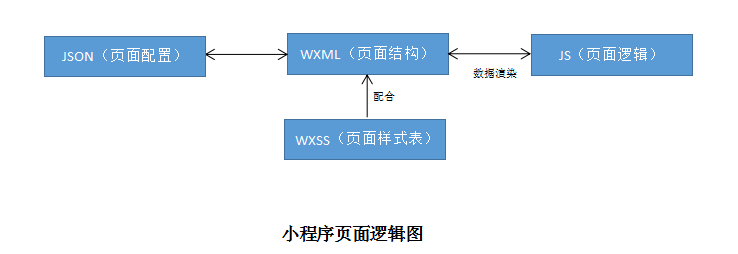
5.1.2小程序端

微信小程序端向服务器发送HTTPS请求，传送相关的数据，并根据服务器反馈的数据进行小程序端的页面渲染。每个页面包括有：页面逻辑js，页面结构wxml，页面样式表wxss和页面配置json。

其中，wxml用来编写页面的标签和骨架，主要使用了一些小程序自己的组件，使用wxss提供的样式和同时与js的事件进行响应，为了方便数据渲染在历史记录和消息提醒中采用了列表的形式；wxss中放了一些样式容器，从而对布局进行控制，主要采用的是flex布局，本产品除了使用自己构建的样式以外，还使用了微信官方设计团队提供的WeUI基础样式库；该小程序的json比较简单，主要是返回键的设置；js主要是点击事件的相应方法，同时使用了官方提供的API，实现对小程序页面的准确渲染。

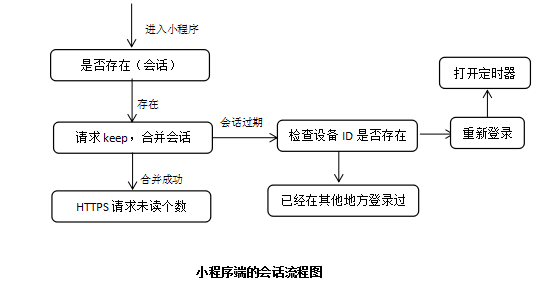
具体框架如下图：



5.2.1

1. 会话机制

微信小程序的会话包括：会话未过期并向服务器发送请求刷新会话，和会话已经过期并提示用户重新登录。小程序登录时系统会首先检测缓存中是否存在sessionId（会话ID），如果已经存在，那么合并会话可开始使用，如果在这个过程中抛出了1002异常，说明该会话已经过期，那么用户此时可以选择重新登录或者离线使用，如果选择了重新登录，为了避免不同系统中同时使用，首先会在缓存中检测设备的唯一ID是否存在，如果存在直接登录，反之则获取设备再进行登录。如果小程序登录时没有监测到sessionId，说明会话不存在，那么这个时候用户可以选择离线使用或者同意登陆。具体的流程如下图所示：



1. 缓存机制

该产品为了在离线的情况下也可以查看历史记录和消息提醒中的信息，小程序端设置了多个缓存，把服务器传送过来的数据以二维数组的形式放在缓存中，需要的时候再进行页面渲染。此外，小程序中还使用了多处的系统数据缓存，功能相当于全局变量，但优于全局变量。

1. 定时器机制

定时器的作用是监测用户使用该小程序的活跃程度，从而进行会话的刷新。小程序进入之后定时器设置一分钟，在一分钟里面，如果有出现点击事件，那么就进行会话合并，直到小程序结束。

1. 消息提醒HTPPS请求

消息提醒采用了官方提供的接口setTabBarBadge，需要时刻对邀请人发来的信息进行刷新，因此需要在每个页面都进行一次HTTPS的请求，获取最新的未读提醒个数，从而显示在界面上。由于该请求放在onShow里面，为了避免与会话发生冲突，所以，需要确定sessionId存在的前提下才发送获取未读提醒的请求，因此在onShow页面进来的时候，向发一次定时器，使会话合并，要是会话过期则可以选择重新登录。