“重大约吗”技术梳理

**一 产品的背景&核心竞争力**

“重大约吗”是一款面向重大学生的基于位置，社交性质的即时"约会"APP；该产品可以为用户提供一种方便快捷、高效、专业的"找约","起约"的服务，通过使用本APP可以让用户做到随时随地只要拿着手机或者移动设备就可以找到想要的人一起进行学习、生活、娱乐等活动，这些活动包括但不限于如下所列举的几种：

“约人组队愉快参加比赛”、“约人一起听讲座”、“约人参加舞会”、“约人一块出去旅游”、“约人进行运动”、“约其他人或者团体进行联谊”、“约人一块自习”、“约吃饭约逛街约看电影”等等，总之只要你能够想到的，大学生喜欢的活动，想要的活动，我们都可以为之提供一个专业而高效的平台，让他快速找到相关的同伴愉快的进行活动。

我们产品的核心竞争力：

（1）热情、专业、高效、奉献、优秀的团队，团队成员具备较强的技术背景，可以为用户约人过程提供强有力的技术保障。

（2）

a）一款专业而又专注的产品，该产品基于学校附近位置的人群，该产品不仅是一款娱乐社交产品，更是一款用户生活中得力的助手，我们专注于"约"这么一个核心功能，保证我们的用户能够快捷方便的找到自己要的“约会”。

b）用户群明确，重大CQU师生，这能够保证我们的使用用户都拥有共同的背景和高素质，让他们在活动过程中能够打成一片。

（3）一款拥有积极、正面、正能量的校园社区文化的APP，我们将尽最大努力为我们的用户营造一种让他感受到正能量的文化，让他可以享受到美好而又愉快的校园生活。

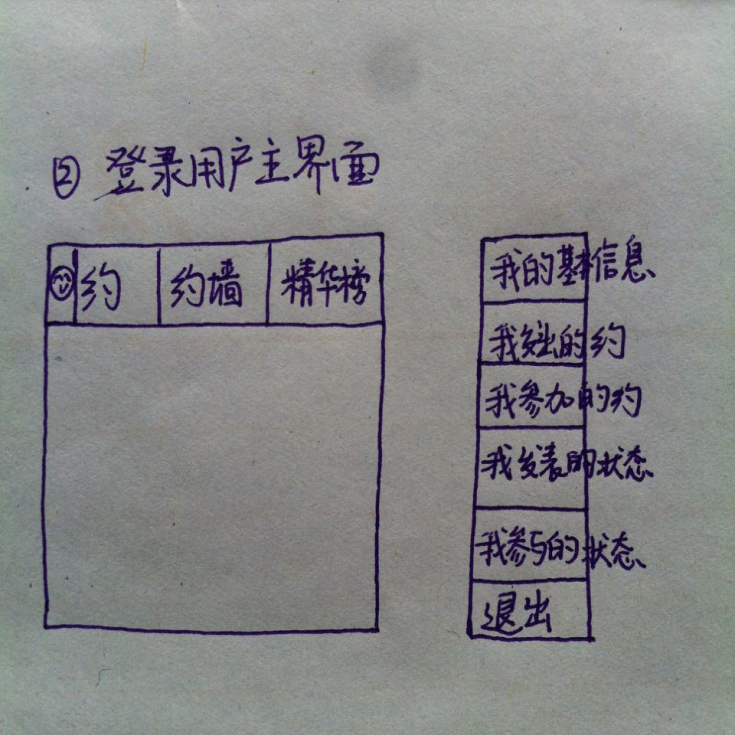
（4）专注而不单调的产品功能,既有学术，比赛，互帮互助，旧物互换等用户需要的活动，同时也有生活娱乐这些放松身心的功能，真正为用户提供一种能够享受在学校生活的服务。

**二 产品的流程涉及到技术梳理**

**登陆**



1）未登录用户



2）登录用户主界面

用户开启产品的一瞬间就会与我们的后台服务器进行连接和信息的获取交互，其中涉及到的主要技术有如下几种：

（1）**界面设计**：主要为每一个对话框设计一些功能按钮，做到功能清晰明确的呈现；

**Level--中等偏容易（）**

（2）**Android**：该技术为用户提供一个前端操作界面，用户可以操作界面可以操作的功能；

**Level--中等偏容易（主要在于界面功能的逻辑实现和与后台的交互，不存在大的技术难点）**

（3）**C/S通信技术**

1）**SOCKET编程**：本产品是一款C/S产品，对用户对对服务器（初版用主、备两个服务器支持，留出扩展接口，能够实现用户量增大时候的多服务器并行支持的扩展），C/S之间主要通过SOCKET建立连接通信，所以需要保证在每一个会话独立而稳定的通信，其中会涉及定时的探测网络状况，并在掉线进行重连或者提示断开连接；

**Level--中等（主要在于连接的维护机制，通信的稳定可靠性保障）**

2）**通信协议的设计**：由于每一个client点击每一个功能都可能涉到与后台数据库或者文件系统的交互，所以需要事先约定好一套通信的协议格式

**Level--中等（主要在于区分开每一个通信指令，并能够通过协议包的校验来保证协议包的正确性）**

3）**互斥消息队列的设计和编解码**：对每一个用户通过功能按钮的点击产生的通信协议包，我们必须预先存放到消息队列中，以等待后续线程的处理（解码并进行后续的后台数据库交互操作，可以使用责任链（chain）设计模式来设计消息的生命期的维护）；

**Level--中等（主要在于保证协议包的正确编码和解码以及校验；并且保证线程之间访问消息队列时候的互斥（不能同时多个线程操作同一个消息队列））**

（4）**多线程或者多进程技术**：前端可以用主线程就可以满足用户的交互操作，多线程的考虑主要在于后台的运行保障，相应指令的解析，处理，数据库交互，文件系统的交互。

1）

a）对于每一个用户通信，每一个用户需要一个独立的线程进行交互，这个线程主要接收或者回送执行动作后的对应response消息包；（其中可以用线程池来实现，要求用户线程是轻量级的，即不用经常去创建线程，只有线程池中没有限制的线程才进行线程创建）；

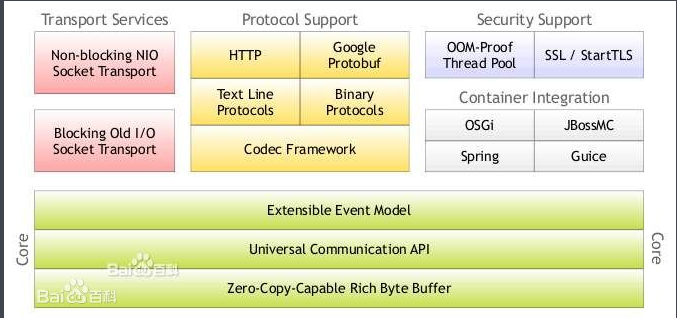
b）必须起若干解析消息协议的线程用来对消息包进行解码，并把解码后的操作或者内容进行处理。

2）多线程之间的同步机制：通过Event,Critical Section,Mutex,Semaphore实现。

**Level--中等偏难（主要在于多线程的维护机制）**

（5）**后台服务器主框架的设计**：后台服务器主要有同步和异步区分，本框架倾向于采用异步的多路复用机制；其中关键的挑战是如何保证服务器7天\*24小时不掉线，不宕机，稳定的可靠的保障支撑用户的服务，所以必须考虑服务器故障时候主备服务器的切换，以及各种异常情况的处理和异常的恢复；通过使用前面的通信技术与前端进行通信，并通过多线程进行并发的支持，通过消息队列实现消息的线程对接处理；

**Level--困难（因为服务器是本APP项目的核心和大脑，所以服务器的处理能力决定了我们产品的服务性能表现，所以要设计一个高可靠，高并发，高稳定，异常恢复，容灾，负载均衡这么一个服务器来，需要较强的后台服务器开发能力，不过我们可以借鉴一些已有的开源服务器框架来做开发，比如Netty，Mina等开源服务器框架）**



图一 Netty框架图

（6）**数据库的设计**：由于用户的个人信息，用户创建的活动相关信息，历史活动信息，约墙信息，精华信息都需要设计相关的数据库表来进行存储，所以如何设计一个满足用户需要，产品需要的数据库将是本产品成功与否的关键；

**Level--中等偏难（关键在于如何正确的设计数据库表，以及表相关的关系，并能够做到快速高效的响应用户的指令，由于相关的信息较多，所以如何设计这些数据库将是一个不小的挑战，可以采用MySQL作为后台数据库系统）**

（7）**文件系统**：由于每一次活动都可能上传照片，如何把这些照片存储起来，并跟每个活动关联起来，在用户点击查看时候如何压缩并显示出来将是一个不小的挑战；（自己想到的是设计一个后台的进程用来专门处理这个东西，为每个照片独一无二的进行编号，并把这些编号存储在数据库中与活动关联起来，每次用户点击活动时候前端会关联活动和相关的照片去显示出来）

**Level--困难（关键在于有效的存储，并响应用户的指令去显示出来，如何压缩，这个估计都需要相关领域知识的支持，所以将是一个不小的挑战，可以作为关键技术来研究）**

（8）**搜索排序系统（核心算法&关键算法）&推荐系统**：

1）由于需要为每一个“约”，“约墙”，“精华”等版块的活动进行相关的排序，所以需要相应的算法支持，比如数据库查询到了相关的活动之后，如何按照要求显示这些活动将需要这些搜索，排序等算法的支持；

2）为用户推荐一些活动或者精华版块的实现都需要这个系统进行处理

**Level--中等偏难（算法要求高效及时的响应用户需求）**

（9）**注册、登陆、校验、权限系统**：每一个用户的登陆、注册、校验、游客和会员用户的权限控制都需要这个系统进行相关的功能提供；

**Level--中等（关键理清流程，难度中等）**

（10）**后台存储&维护系统**：由于随着时间和用户量的增大如何维护以前的历史信息表，历史活动图片都将需要我们去考虑，所以我们需要设计一个这么系统定期来更新维护这些信息；

**Level--中等偏难（主要因为没有相关经验，所以需要进行相应的技术学习和储备，挑战不小）**

（11）**服务器日志系统**：由于交互量比较大，当服务器运行过程中我们需要定期的进行相应信息的记录和观测，以便出现问题或者想查看之前的记录时候有一个凭证，所以需要一套日志系统的支持；

**Level--容易（已有的LOG4CXX,LOG4J可以满足要求）**

（12）**缓存系统**：由于用户量大的时候如果每次都需要去跟后台数据交互，估计会降低后台的处理效率，所以对热门的活动进行一个缓存，并在用户查看时候直接通过缓存提供将会提高查询效率，有助于服务性能；

**Level--中等偏难（可以借鉴MEMCACHED这个系统来做）**

**三 产品的关键技术点分析**

**综上来看，其实前端的东西都不是什么难点，作为一款线上产品，技术里面的最有挑战性和困难的是后台服务器系统以及相关技术；还有后台交互相关的缓存、文件、搜索排序、推荐系统、数据库等系统技术，这些技术的解决将会决定我们的开发进度，是开发流程图中的关键路径，所以也是我们需要着力进行关注的地方；**

**四 其他**

由于自己并未进行实际的开发，对整个产品的技术搜集还未全面的进行，所以还需要进行相应的补充收集，这个将在后续产品的探讨，与相关专业人士的交流过程中完善。