# 服务端需求

服务端主要实现以下几个方面的功能模块：

1. 后台数据库查询模块；
2. 网络通信模块，主动发送彩票销售、开奖等动态消息；
3. 监测客户端是否在线；
4. 权限验证模块（需求文档没有提出，但是最后应该实现，以保证信息安全）

# 框架设计

客户端主要由资源管理和加载、后台指令通信模块、指令动画模块等三个部分组成。

# 接口设计

# 关键技术

## 渲染数据预加载技术

将渲染数据进行预加载可以避免程序运行过程中的卡顿和黑屏问题。Unity默认在切换场景的过程中，如果需要加载大场景数据就会需要等待较长时间，给用户不好体验。根据目前的项目需求，本系统是重复销售、止售、开奖、报表的循环过程，因此将共用的并且循环使用的资源进行预加载，可以有效避免运行过程中卡顿、黑屏等问题。

## 渲染数据后台加载技术

对于非预加载的渲染数据，为了避免影响主线程的运行，需要实现渲染数据的后台加载技术。将渲染数据通过后台资源加载线程，将资源从磁盘加载到内存，然后再加载到显存中。这种技术通常也称为资源异步加载技术。

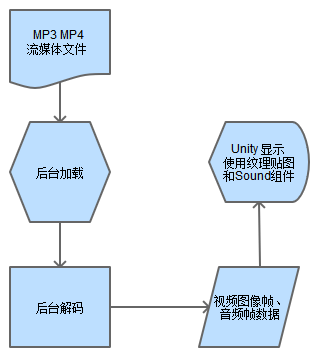
## 动态资源加载技术

动态更新数据比如从网络下载的数据，需要加载到场景中进行显示，那么就需要使用使用动态资源加载技术。比如销售阶段的宣传广告资料，这类资料经常需要更新，从服务器下载完宣传广告资料，就需要使用动态资源加载技术，同时结合渲染数据后台加载技术，可以实现平滑的数据加载效果。

## 流媒体渲染技术

采用流媒体渲染技术支持视频、音频渲染和播放，从而实现客户端更好的展示效果。流媒体资源通常是指视频数据流和音频数据流。大部分流媒体的数据都比较大，如果全部加载到内存中，将会消耗比较大的内存，资源加载等待的时间也会比较长，用户体验和程序性能都不好。但是流媒体资源的有着显著的特征，流媒体数据在时间上连续的，其在单位时间内传输的数据，可以进行解码播放。

该技术关键在于结合后台渐进加载、音频和视频解码后，将解码后每帧的图像和声音数据，加载到纹理贴图和声音缓冲SoundBuffer中，并不断更新贴图和SoundBuffer，从而实现流媒体播放。



表格 1 流媒体播放

## 基于WebSocket通信技术

基于WebSocket通信技术利用HTTP和Socket连接的特点，利用HTTP发起构建连接通信的请求，请求成功完成后，客户端和服务端建立起Socket长连接。基于WebSocket通信技术可以避免HTTP轮询技术的一些明显缺点，也可以避免自己基于Socket自己开发通信接口和通信协议。

# 开发环境

# 参考资料