**软件体系结构大作业**

**17130120116-李云水**

**一、软件系统的需求**

某公司想创建一个类似bilibili的软件系统，系统运行于Internet，用户可以在系统上发布视频内容，其他用户可以观看发布到系统上的视频内容，用户之间可以相互关注，也可以随时取消关注。一旦某用户关注的博主发布视频或者消息，会立刻通知到该用户。一旦取消关注，以后便不再进行通知。用户可以在系统上修改自身的资料，包括个人信息、密码等，具体需求如下：

1.用户可以任何时间发布视频内容到该系统，但视频内容需要经过系统的自动审核。

2.用户可以在系统上修改、删除自身发布的内容。

3.用户之间可以相互私信。

4.每个用户可以在任何时间关注一个或多个up主，可以在任意时间对该up主取关。

5.博主发布内容将在一秒内通知到他的“粉丝”们。

6.系统界面与哔哩哔哩动画类似，所以很容易使用。

7.每次系统小的更新由三名开发人员在一天内完成

8.系统要防止非法分子的流量攻击 ，防止系统卡顿崩溃，受攻击率小于0.001%。

9.进入系统的时间应该小于0.1s

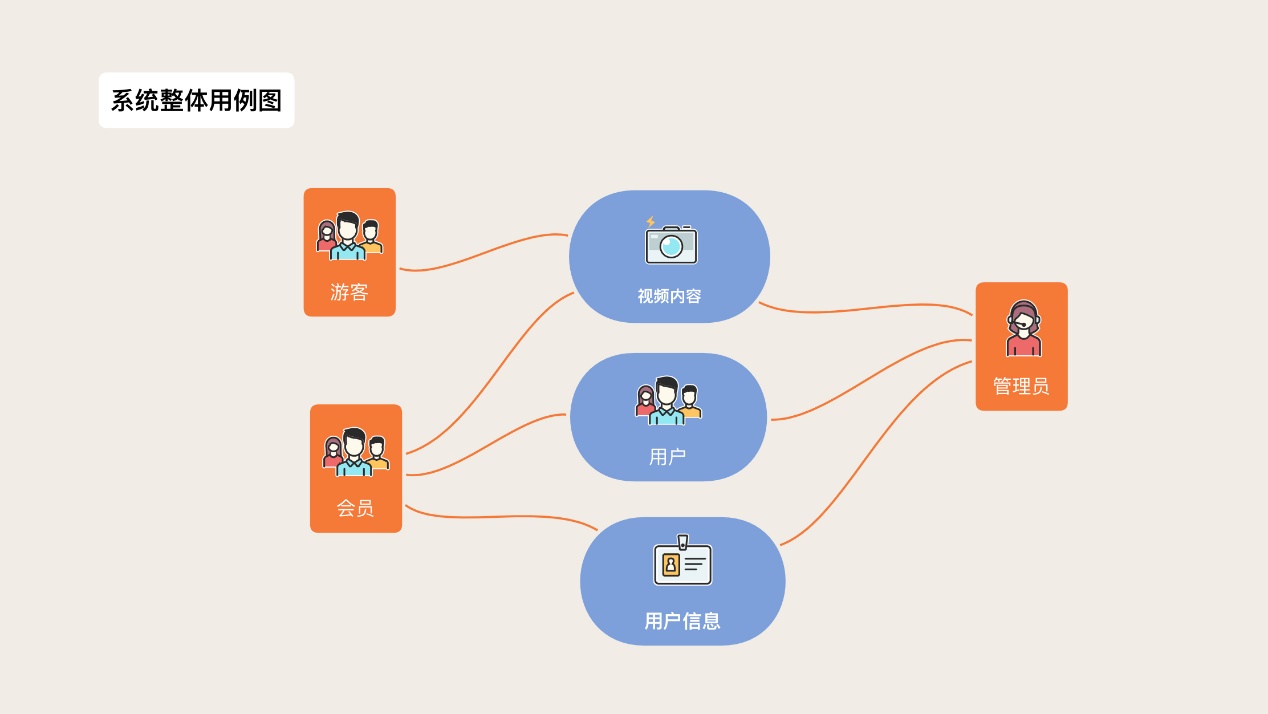
10.系统提供特殊的页面来进行自动测试。

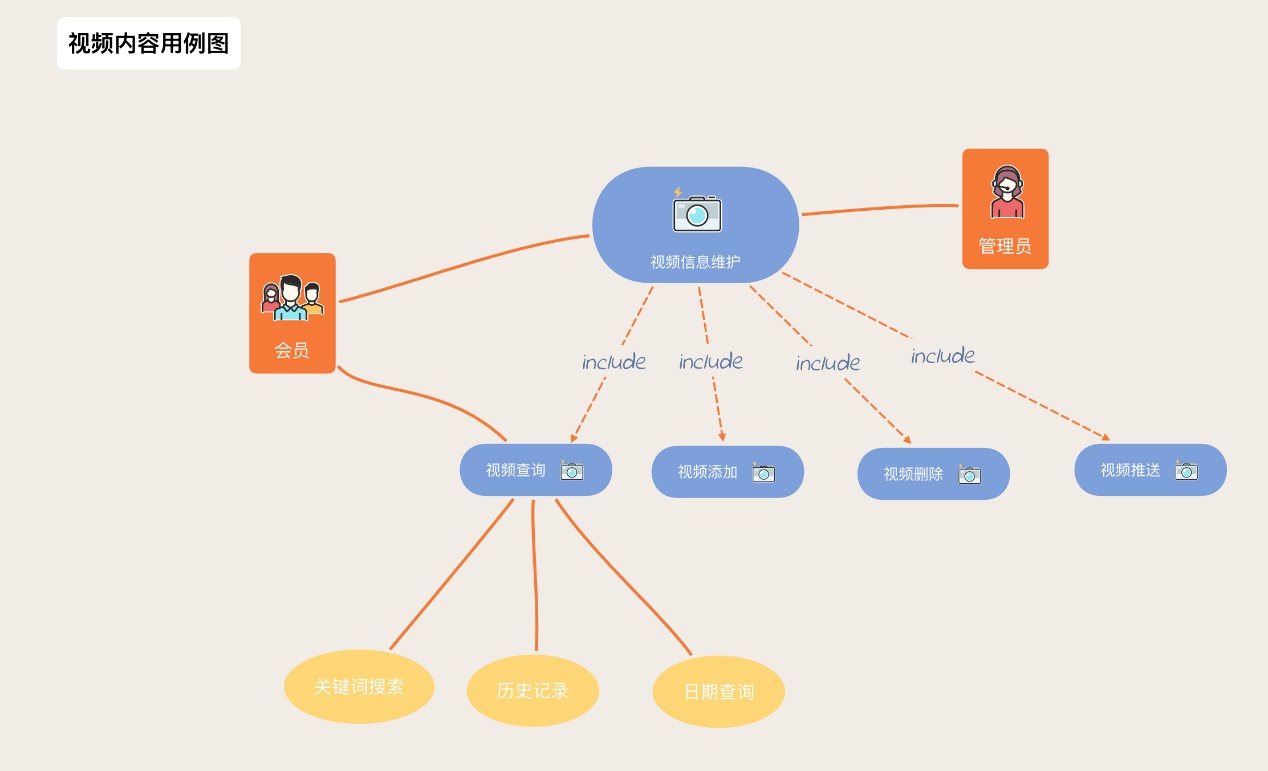
11.每年该系统的不可用时间应小于24小时，每次故障应在2小时内修复。

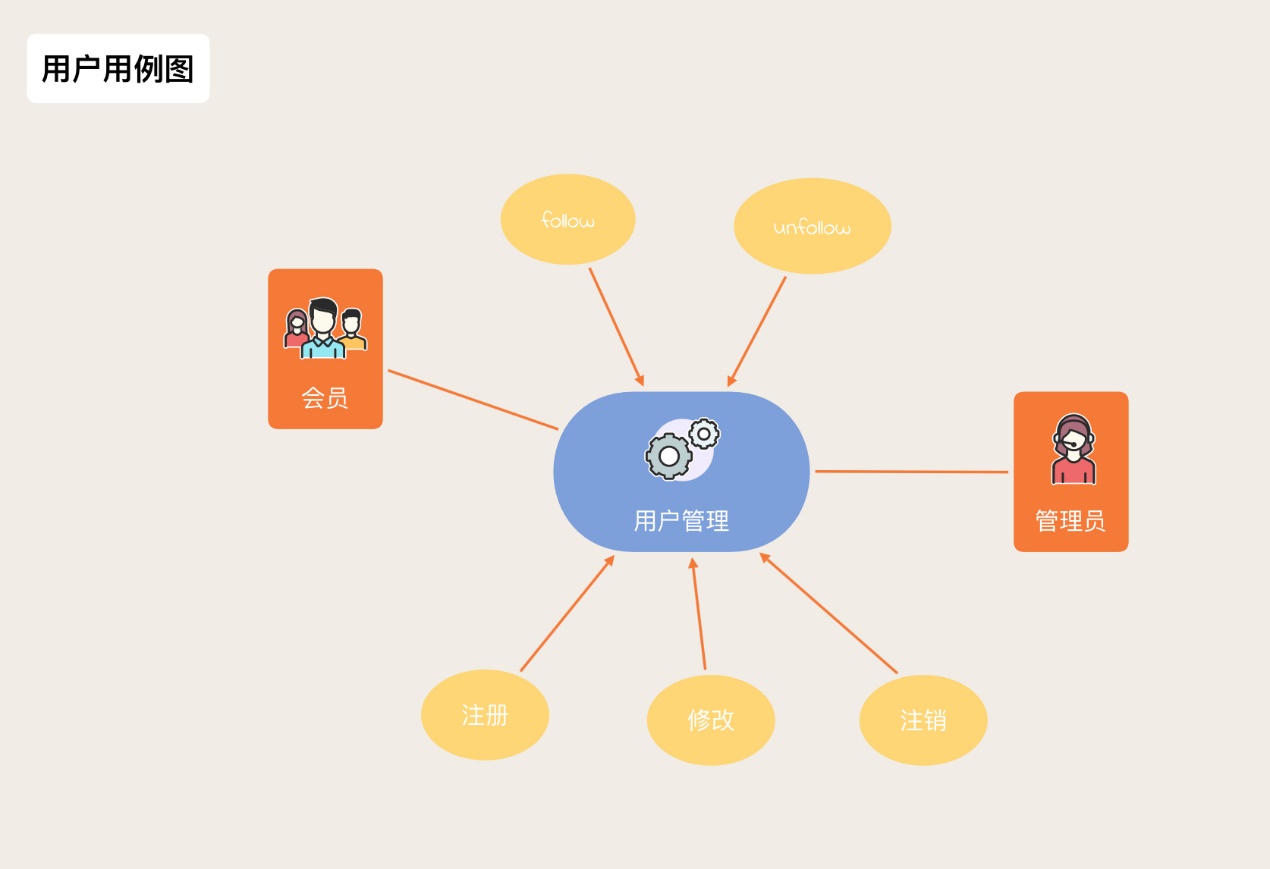
其中蓝色部分为功能需求，绿色部分为非功能需求。

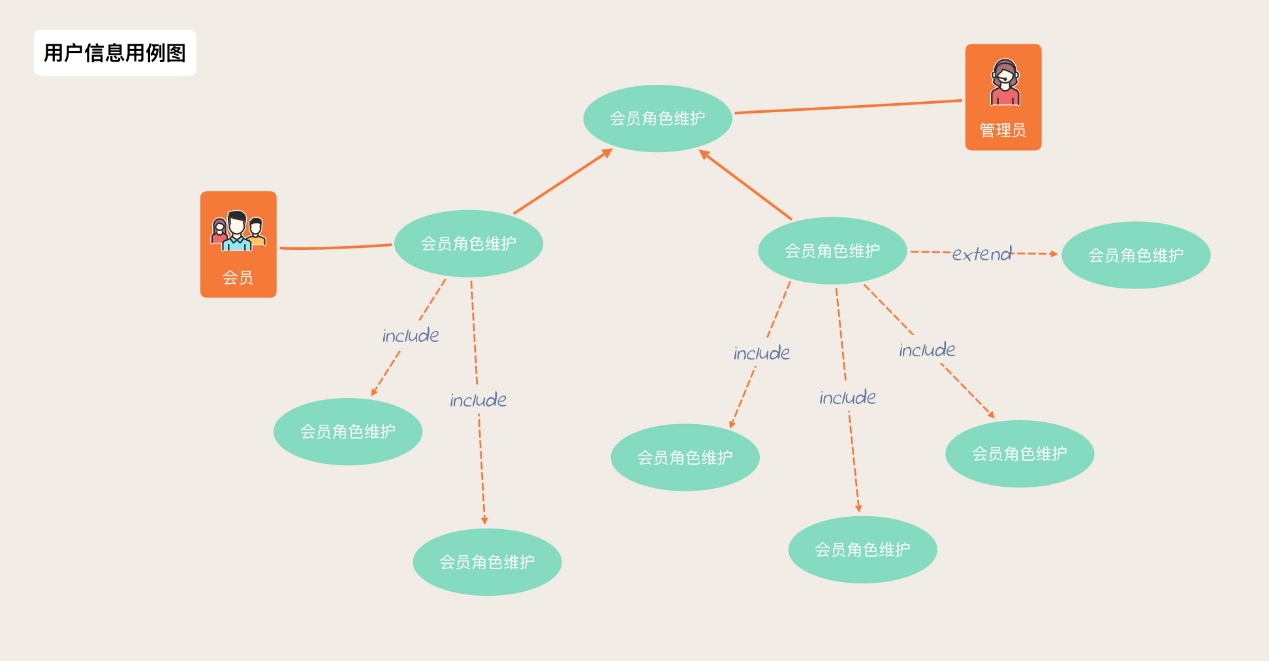
**二、需求分析**

功能性需求的用例分析：









非功能性需求分析：

1.系统界面与哔哩哔哩动画类似，所以很容易使用。（易用性）

刺激源：终端用户

刺激：终端用户希望学习系统的使用，提升系统的使用效率，减少出错。

制品：整个系统

环境：运行时

响应：系统响应用户的需求

响应衡量指标：每个功能用户出错的次数小于两次。

7.每次系统小的更新由三名开发人员在一天内完成（可修改性）

刺激源：开发人员

刺激：对系统进行小规模的更新

制品：系统的功能/交互界面

环境：开发时/运行时

响应：具体修改的操作、部署

响应衡量指标：人力成本不大于三人，修改完成的时间应该在一天之内。

8.系统要防止非法分子的攻击，防止系统卡顿崩溃，受攻击率小于0.001%。（安全性）

刺激源：攻击由人或系统发起

刺激：对系统的攻击

制品：系统的服务或数据

环境：联网时

响应：对于非法的用户或ip地址进行屏蔽，放置robots协议起威慑作用

响应衡量指标：疑似攻击会经过自动系统和管理员的审查，总受攻击率小于0.001%，若遭受攻击，系统应在两个小时内恢复。

9.进入系统的时间应该小于0.1s（性能）

刺激源：用户

刺激：进入系统

制品：系统

环境：系统运行前

响应：用户进入系统

响应衡量指标：进入系统的时间应该小于0.1s。

10.系统提供特殊的页面来进行自动测试。（可测试性）

刺激源：测试人员

刺激：系统开发到达里程碑

制品：系统

环境：开发阶段/运行阶段

响应：通过测试给出观察结果

响应衡量指标：白盒测试中的功能测试应该达到100%，能够给出当前系统的负载，反应系统是否卡顿。

11.每年该系统的不可用时间应小于24小时，每次故障应在2小时内修复。

刺激源：故障的迹象

刺激：系统错误/崩溃

制品：系统

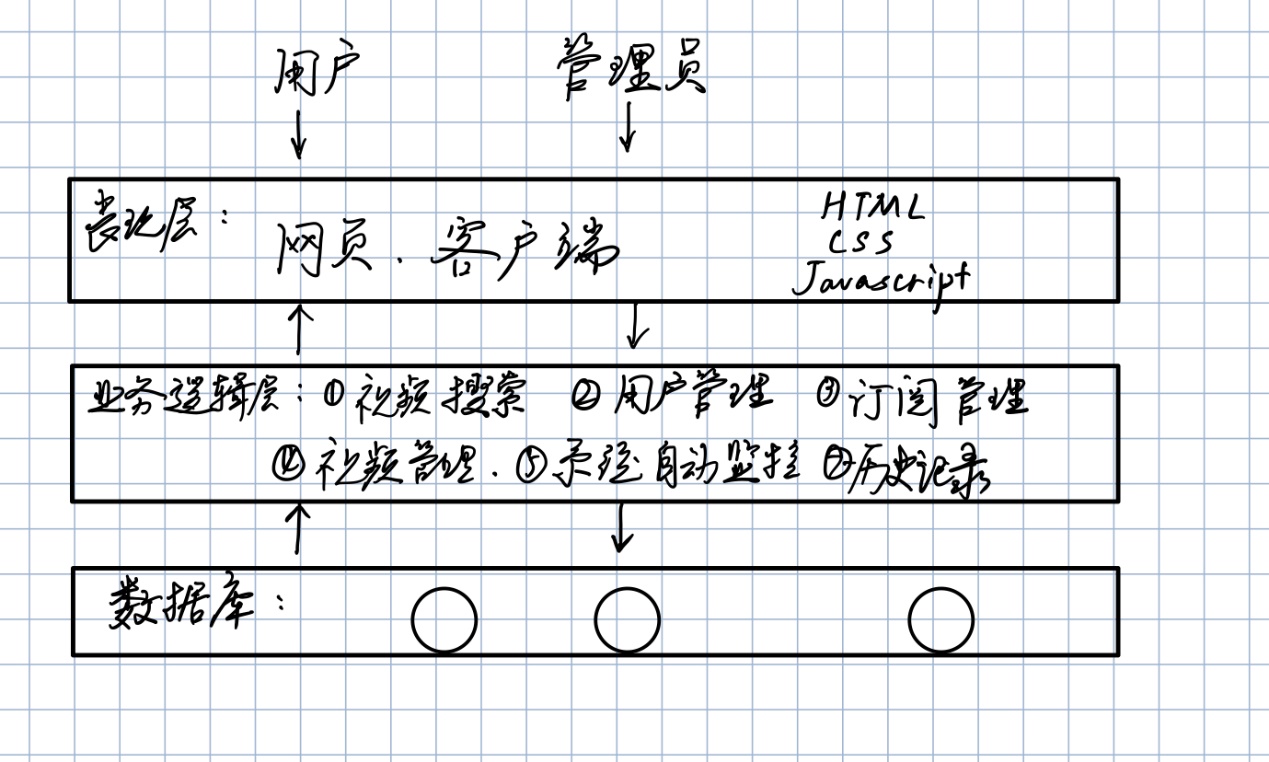
环境：运行阶段健康/不健康

响应：通知管理员或系统

响应衡量指标：每年不可用小于24小时，每次故障应在2小时内修复。

**三、软件系统的架构设计**

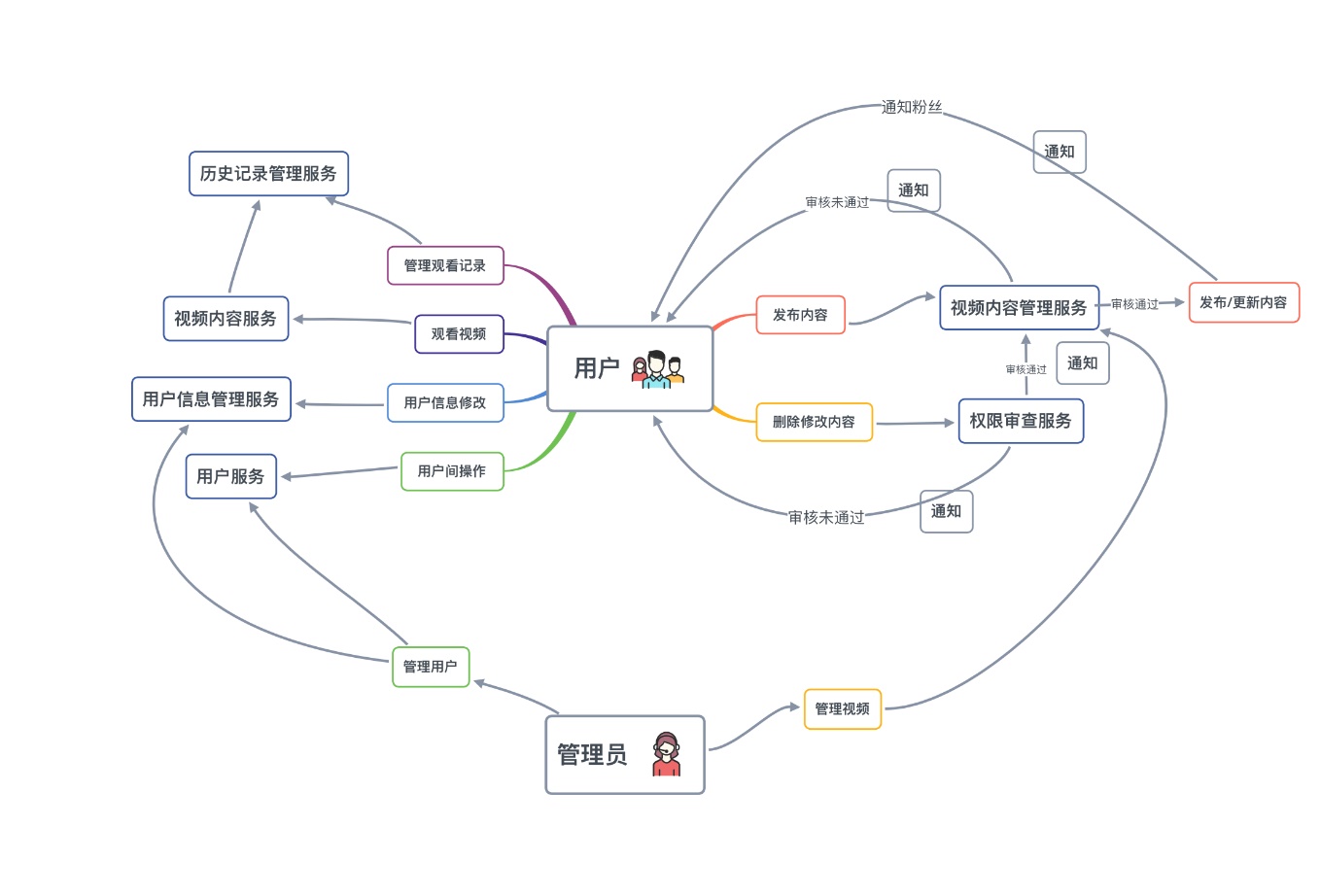
1.总体架构采用三层设计



分为表现层、业务逻辑层及数据库层，通过对用户界面与系统其他部分分离，提高了系统的易用性和可修改性。

2.系统架构设计

使用了面向对象和事件系统的设计思路。



面向对象的特性：高内聚，模块通用性好，隐藏信息，接口不变提升了系统的可修改性和安全性，同时模块化的功能也为测试带来了便利，提高了可测试性。经常性的模块通用也为开发降低了成本。

事件系统的设计：符合系统给出的功能需求，是系统中重要的部分。

**四、体系结构评估**

运用ATAM方法进行评估：

**ATAM-Phase 0（准备阶段）：**

团队交流交流评估的方法以及系统的结构；

给出完成评估的标准，协议；

**ATAM-Phase 1（初始评估）：**

给出ATAM的总概：

1.技术层面：生成效用树；场景的头脑风暴；

2.输出：效用树和场景；敏感点、权衡点、有风险决策、无风险决策；

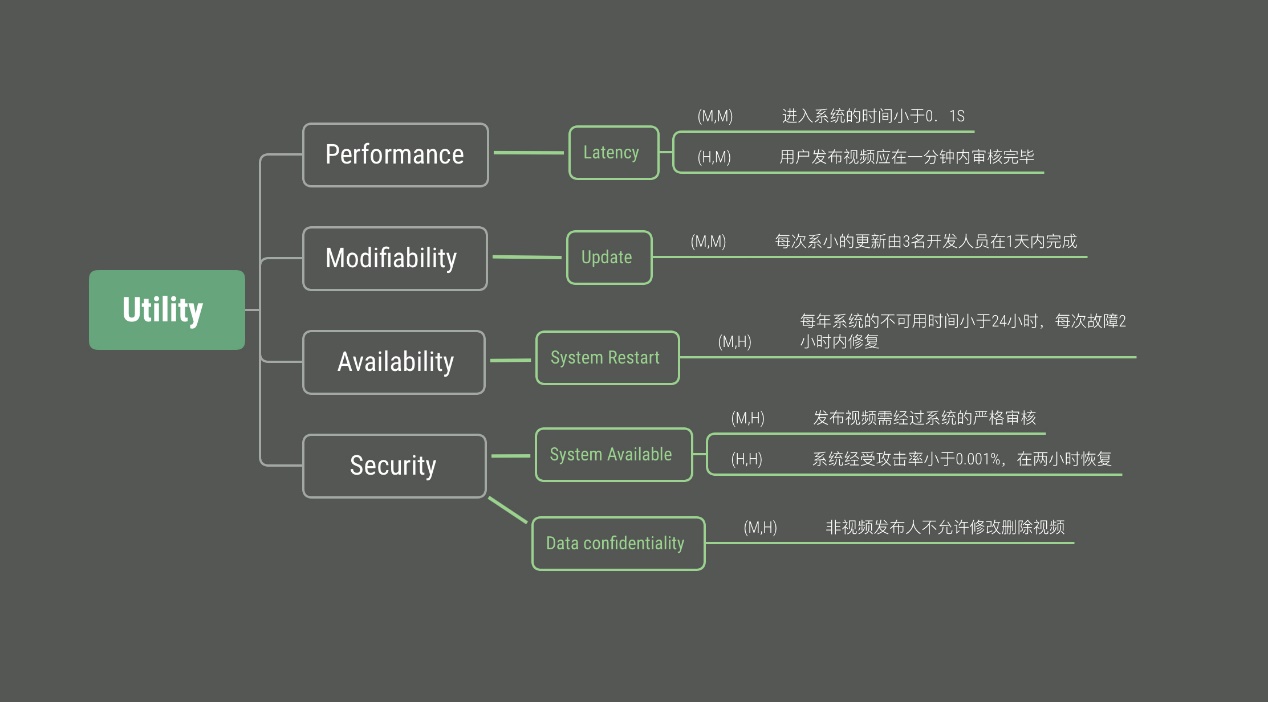
关于系统的商业驱动

具体见一：需求及二：需求分析

系统架构总概和架构方法

具体见三：软件系统的架构设计

构建效用树：分析重要的质量属性目标



关键的质量属性及场景如图所示。

从各质量属性中辨别风险

1.敏感点：对于进入系统的速度，数据的初始化速度是它的敏感点；对于数据的私密性，用户密码的复杂度是其敏感点；视频审核的精细程度是防止视频滥发，维护系统可用性的敏感点；对于系统破坏后的重新修复或者系统的更新的时限，技术人员的能力和数量是其敏感点；审核的精细程度是用户发布视频速度的敏感点。

2.权衡点：改变视频审核的精细程度会同时影响系统的安全性和性能，当我们提高系统的审核复杂度，势必会带来时间的延迟从而影响性能，反之则降低安全性提高性能。由于系统安全性的优先级较高，我们更侧重安全性的表现。

3. 有风险决策:视频审核的精细程度可能对系统具有一定的风险，若审查的复杂度较低，容易导致系统被大量风险视频涌入，可能导致系统中毒崩溃。

4.无风险决策：对于进入系统的时间，一般情况为0.05s进入系统，小于0.1s，此系统可行；对于系统小的更新，由三名开发人员在一天内完成，此系统可行；每年该系统的不可用时间应小于24小时，每次故障应在2小时内修复，此系统可行。

Recap Phase1 and turn to Phase2；

**ATAM-Phase 2（检验phase1的成果）：**

项目干系人用头脑风暴的形式构造出一系列软件系统场景。

用先前的场景识别系统所用的架构方法。

总结ATAM的步骤并给出输出文件。