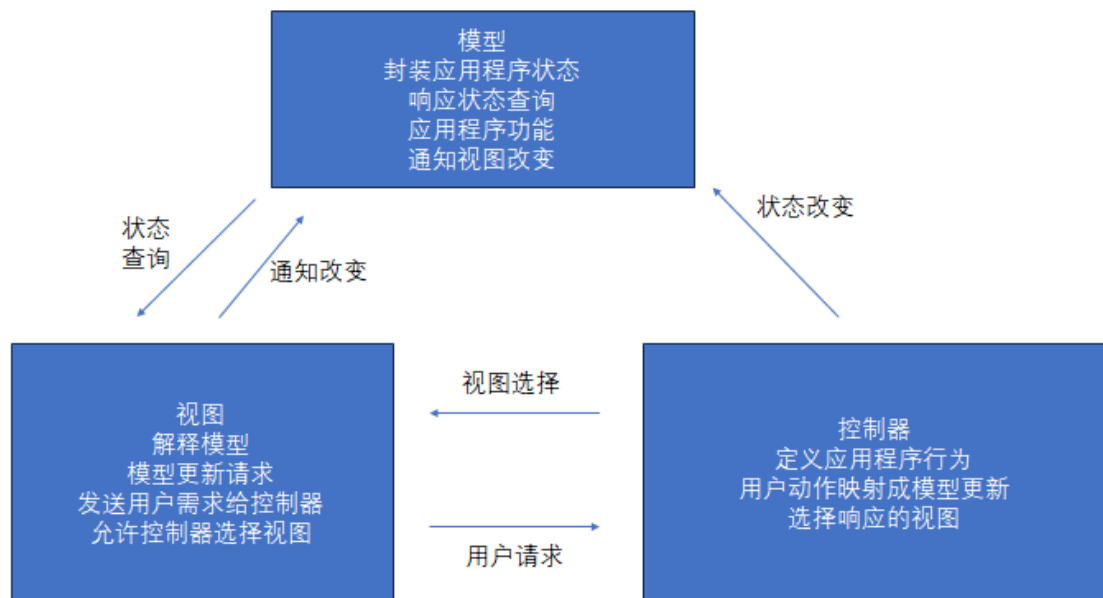


MVC结构由**Model (模型)**、**View (视图)**、**Controller (控制器)** 组成。MVC将人机交互从核心功能分离出来，模型对于用户来说是透明的，用户只需要观察视图，用户和模型的交互通过控制其提供的安全方法。

- **模型 (Model)**：包含核心功能和数据以及后台的业务逻辑。模型封装了访问数据的函数，例如公文信息、用户信息等。用户看不到后台对数据库的操作，模型对用户来说是透明的。
- **视图 (View)**：负责向用户显示信息。不同的角色可以看到不同的视图。用户在视图上与系统进行交互，一些用户的行为（例如发件、收件）会触发模型的功能，从而向模型传递数据或者获取模型更新后的数据。
- **控制器 (Controller)**：与视图一一对应，每个视图都有一个相关的控制器。控制器组件接收事件，并将事件翻译成对模型或视图的请求。如果控制器的行为依赖于模型的状态，那么控制器也需要向模型登记请求变更通知。

例如，当用户在视图中点击发件按钮时，控制器接收到此事件后，调用模型的发件方法。模型会导入数据库中收发双方用户对应的信息，并将更新后的数据显示在视图中。



用户角度功能架构

用户管理

- **注册**: 用户可以在系统中注册自己的账户，输入账户名、密码、个人信息等。
- **登录**: 用户可以使用已注册的账户通过登录页面登录系统。
- **注销**: 用户可以在任何时候注销自己的账户，退出系统。
- **更改密码**: 用户可以在登录后进入个人账户设置界面，更改自己的账户密码。

文件处理

- **发件**: 用户在系统中可以发送公文，输入公文的标题、内容、收件人信息等。

- 收件：用户可以在自己的收件箱中查看已经收到的公文，根据需求对公文进行分类和归档。
- 阅件：用户可以在收到公文后阅读公文，系统会提醒公文的时限和密级等信息。
- 归档：用户可以将已经阅读或处理的公文进行归档，方便后续查阅和管理。
- 删除：用户可以将不需要保留的公文进行删除，释放存储空间。

公文分块处理

对于公文中的各项内容，系统需要提供分块处理的功能，将公文的标题、单位、时限、密级、时间等核心信息提取出来，方便用户进行分类和检索。

具体设计如下：

- 标题：公文的标题必须清晰明确，简要描述公文的主题和内容。
- 单位：公文中需要标明发文或收文的相关单位信息，包括收件单位和发件单位等。
- 时限：公文中需要标明办理等需要的时限，方便用户及时处理和办理公文。
- 密级：公文的密级需要清晰标明，保证公文的安全性和保密性。
- 时间：公文需要标明具体的发文或收文时间，方便用户进行归档和检索。

用户管理

注册

登录

注销

记住密码

找回密码

菜单栏

新建公文

未读公文

已读公文

收藏

垃圾站

新建公文

上传公文

设置内容

收件机关

公文标题

处理时限

密级

草稿操作

保存

发送

删除

未读公文

翻页浏览

提取内容

收件机关

公文标题

处理时限

密级

收文日期

待办管理

处理时限

已读公文

翻页浏览

提取内容

收件机关

公文标题

处理时限

密级

收文日期

日志记录

处理时间

收藏

翻页浏览

提取内容

收件机关

公文标题

处理时限

密级

收文日期

日志记录

处理时间

垃圾站

提取内容

公文标题

处理时限

密级

日志记录

删除时间

使用openssl作为加解密签名验证签名的工具，使用SQL数据库