（问候）

我们大作业做的是《校园系统》，设想是集帐户与权限、通知与公示、课程等等子系统于一体的，一个完整的在线校园平台。（翻页）下我们自顶向下的设计思路，分三个小节（模块设计、项目结构、代码结构）进行。

（模块设计）

（翻页）首先是模块设计。我们的成果是一个平台，要与用户交互，进而读写数据。作为一个有实际意义的平台，应该分为服务端与客户端，也就是要有两个程序，一个跑在服务器上，另一个跑在用户的设备上；他们互相通信。（翻页）其中，客户端的逻辑是一个状态机，在不同页面间切换，所以其核心是页面模块，提供不同的页面类；是页面通过ui模块与用户交互，也是页面通过传输模块与服务端通信。（翻页）服务端的逻辑是一个无限循环，不断检查客户端请求，并调用其核心——系统模块——提供的接口并作出应；是系统通过data\_base模块读写数据，也是系统间接通过传输模块与客户端通信。（翻页）为了代码复用，在两端我们还各立出来另外两个模块：string\_integral和file\_system。（翻页）两端模块有重叠部分；最后可以用该示意图表示。

值得注意的是，ui模块使用了第三方多媒体库sfml。然而我们只是使用这个库绘制图形字体和图片而已，这个库并没有提供按钮、输入框等ui组件。我们基于这个库自行构建了按钮类、输入框类等轮子（可能是因为我们比较闲吧）。

（翻页）进一步地，在代码层次上，页面模块在的页面基类派生出了很多页面类，因此根据这些页面类是谁写的分别放在了三个子模块中：eea、lab、vio\_vv；系统模块中的单例类多继承自若干子系统类，因此也分出若干子模块：account\_and\_access、nolify、course等等。为了代码复用，两个模块还分别各有一个header子模块。

（项目结构）

（翻页）然后是项目结构。关于项目文件夹中各目录和文件的说明如图。

（翻页）其中include目录下的头文件与上一节设计的模块、子模块高度对应。它们的依赖关系如图。

（翻页）其中对应源代码有工作量的几个头文件及其分工如图。

（代码结构）

（翻页）最后是代码结构。在代码层次上，每个模块、子模块都有其层次性。限于时间，此处以点带面，只总结在类体系的设计层次上的情况。

项目中定义的主要的类及其关系如图所示。

（翻页）值得注意的是，类体系中，有两处继承体系使用了虚特性。

一处是ui模块，（翻页）Control类是其中最重要的抽象类之一。Control——一切组件的基类，应当对组件位置和尺寸的计算、事件的处理以及绘制作出抽象。为此我们分别设计了抽象方法Update、Process和Draw，它们的意义和调用时机如图。（翻页）其中，Update处理我们设计的如图几个与位置和尺寸有关的属性。这几个属性的意义和关系如图。（翻页）（等一秒）（翻页）为了提升性能，我们还设计了UpdateInQueue、FreshUp、FreshFown等方法。一个组件在一帧中可能被多次Update，但实际上没有每次都立即执行的必要。所以，当一个组件需要Update时，我们先标记一下，在帧这一帧结束时再统一执行。

（翻页）另一处是页面模块，PageBase类也是一个抽象类。PageBase——一切页面类的基类，应当对页面的布局、逻辑和与事件循环有关的操作作出抽象。为此我们设计了如图几个抽象类，它们的作用如图；（翻页）页面运行时它们所在的流程位置如图。（翻页）其中，与事件循环有关的这几个抽象方法在这次项目中没有被重写过。因为要利用事件循环的完成的操作无外乎与服务端通信、跳转页面和定时器功能。我们为它们分别封装了WaitServer和Listen、SwitchTo和Timer等方法和ui组件。

（翻页）我们的介绍结束，至于更多的细节，请查看我们的文档。

（感谢和请求批评指正）