三层架构，分为UI显示层、BLL业务逻辑层、DAL数据访问层，UI层负责关于UI的搭建、UI逻辑处理和UI对于业务逻辑的相互调用。BLL层处理非UI的业务逻辑。DAL层创建数据模型，跟数据库交互，提供对数据模型的增删改查等接口给BLL层：

形状

中度可信度描述已自动生成

下图是MVC、MVP和MVVM的区别。MVC的图在不同的人的地方有不同的定法，但它和其他架构最大的区别就在于它的View和Model之间存在交互（双向或者单项）。MVP模式解除了这层交互，把这类交互转移到了P里面，但是导致了P的内容过大。MVVM利用了WPF等的数据绑定功能，减少了V和VM层的数据绑定类的改动操作：

图示

描述已自动生成

图示

描述已自动生成

图示

描述已自动生成

下图是MVC和三层架构的结合，其中，MVC是前端架构，三层架构为UI、BLL和DAL层，三层为软件整体架构，MVC是UI层的内容。

图示

描述已自动生成

一般意义上，UI层中MVVM/MVC的Model和DAL里面的Model并不是同一个东西，UI层的Model只提供界面展示所需要的内容，而DAL里面的Model其实是一个Entity，他跟数据库紧密对应（[entity、model和domain三者区别\_domain和model-CSDN博客](https://blog.csdn.net/qq_39071599/article/details/118897637)）：

文本

描述已自动生成

希望实现的架构模式如下图，在UI层，采用MVVM，在整体上采用UI、BLL、DAL三层架构模式，在架构中要特别注意存在的多个模型（Model）之间的关系，DAL中的模型称为Entity，是用于使用ORM来构建数据库用的；三层架构的模型是贯穿整个项目的模型，它在业务层和UI层贯穿始终；UI层中的模型是PageModel，大部分情况下是用于管理改页面/组件所用到的全部对象模型的，这些被管理的对象模型就是三层架构的Model：

图表, 箱线图

描述已自动生成