软件工程实验八实验报告

> 姓名： 张烜
>
> 班级： 2班
>
> 学号： 201705130112

软件工程实验八实验报告

实验目的:

实验内容:

1.对比书上各种软件体系结构风格和视图特点，思考自己项目属于哪种设计风格?

1.1 MVVM的优缺点

2.分工协作

2.1 系统层次划分

2.2 按领域划分

实验目的:

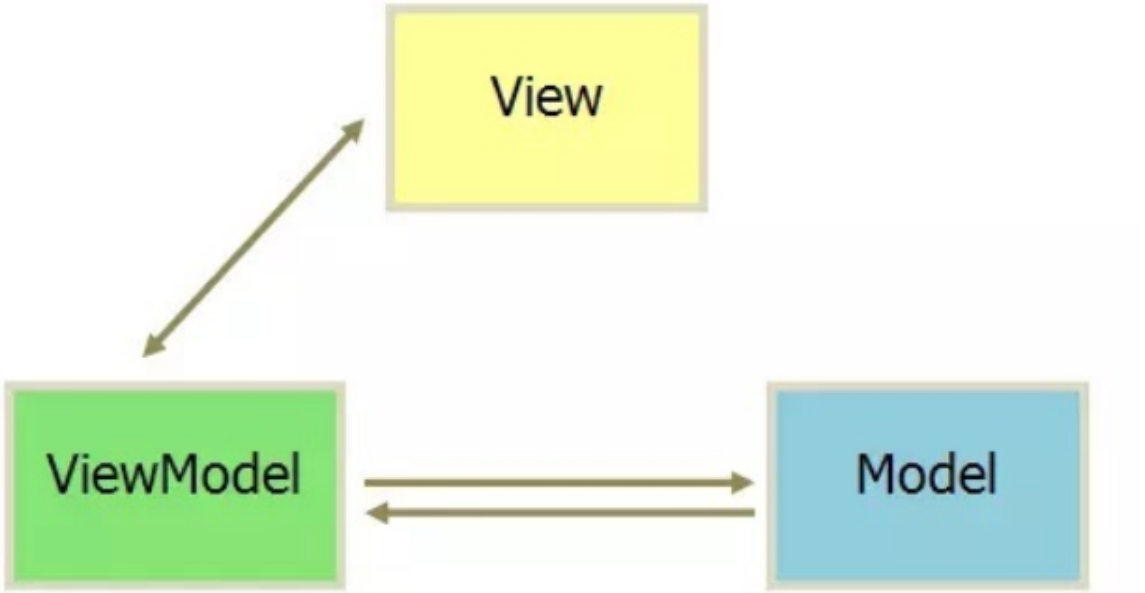
1. 学习对比软件体系结构设计GB和IEEE最新SAD(Software Architecture Document)的标准
2. 遵循体系结构设计的规范，针对自己的项目设计SAD初稿。
3. 记录项目及小组的工作进度。

实验内容:

1.对比书上各种软件体系结构风格和视图特点，思考自己项目属于哪种设计风格?

我负责的部分是前端的工作，采用的工作栈是react。而对于react来说，是经典的mvvm结构，也就是近两年比较火热的一个结构。

MVVM是Model-View-ViewModel的简写。微软的WPF带来了新的技术体验，如Silverlight、音频、视频、3D、动画……，这导致了软件UI层更加细节化、可定制化。同时，在技术层面，WPF也带来了 诸如Binding、Dependency Property、Routed Events、Command、DataTemplate、ControlTemplate等新特性。MVVM（Model-View-ViewModel）框架的由来便是MVP（Model-View-Presenter）模式与WPF结合的应用方式时发展演变过来的一种新型架构框架。它立足于原有MVP框架并且把WPF的新特性糅合进去，以应对客户日益复杂的需求变化。



我们都知道MVP是从经典的模式MVC演变而来，它们的基本思想有相通的地方：Controller/Presenter负责逻辑的处理，Model提供数 据，View负责显示。作为一种新的模式，MVP与MVC有着一个重大的区别：在MVP中View并不直接使用Model，它们之间的通信是通过 Presenter (MVC中的Controller)来进行的，所有的交互都发生在Presenter内部，而在MVC中View会直接从Model中读取数据而不是通过 Controller。

1.1 MVVM的优缺点

优点：

> 1、提高可维护性。解决了MVP大量的手动View和Model同步的问题，提供双向绑定机制。提高了代码的可维护性。
>
> 2、简化测试。因为同步逻辑是交由Binder做的，View跟着Model同时变更，所以只需要保证Model的正确性，View就正确。大大减少了对View同步更新的测试。

缺点：

> 1、过于简单的图形界面不适用，或说牛刀杀鸡。
>
> 2、对于大型的图形应用程序，视图状态较多，ViewModel的构建和维护的成本都会比较高。
>
> 3、数据绑定的声明是指令式地写在View的模版当中的，这些内容是没办法去打断点debug的。

2.分工协作

在SDD中，我负责的主要部分是架构风格实例的编写。

2.1 系统层次划分

M.Shaw 等人根据此框架给出了管道与过滤器、数据抽象和面向对象组织、基于事件的隐式调用、分层系统、仓库系统及知识库和表格驱动的解释器等一些常见的软件体系结构风格。

– 客户端-服务器：将系统分为两个应用，其中客户端向服务器发送服务请求。

– 基于组件的架构：把应用设计分解为可重用的功能、逻辑组件，这些组件的位置相互透明，只暴露明确定义的通信接口。

– 分层架构：把应用的关注点分割为堆栈组（层）。

– 消息总线：指接收、发送消息的软件系统，消息基于一组已知格式，以便系统无需知道实际接收者就能互相通信。

– N 层/三层架构：用与分层风格差不多一样的方式将功能划分为独立的部分，每个部分是一个层，处于完全独立的计算机上。

– 面向对象：该架构风格是将应用或系统任务分割成单独、可重用、可自给的对象，每个对象包含数据，以及与对象相关的行为。

– 分离表现层：将处理用户界面的逻辑从用户界面（UI）视图和用户操作的数据中分离出来。

– 面向服务架构（SOA）：是指那些利用契约和消息将功能暴露为服务、消费功能服务的应用。

2.2 按领域划分

– 通信：SOA，消息总线，管道和过滤器

– 部署：客户端/服务器，三层架构，N 层架构

– 领域：领域模型，网关

– 交互：分离表现层

– 结构：基于组件的架构，面向对象，分层架构