讲讲做了什么，使用了哪些设计理念，实现了哪些功能

设计理念：

**设计风格：styles**

Simple curves and geometries 简单界面和图形，

Multi-color and font contrast for better readability

多色彩 鲜明对比 易读

Functional interactions and animations 互动性动画与碎片切换

Information data visualization 信息数据可视化

**设计原则：principles**

fragments：

SRP: “Single Responsibility Principle”

Meeting listview：

Dependency Injection

**设计模式：patterns**

登录：工厂，

数据库： 单例，

登陆注册 feedback： 观察者，

Meeting model： 策略，

Meeting listview：适配器，

**设计模型：models**

Feebback： MVVM

Firebase+sqlite 数据库

**创新技术设计：Innovative technical design**

登陆注册: eventbus+ life circle+viewmodel+ livedata

Eventbus: 不依赖于 Context，调度时无需注入 Context明显提高调度灵活 ，解除了Handler带来的耦合，对于登录状态这类高优先级的事件，使得创建结构良好组织紧密的通知系统成为可能。

Lifecircle: 降低代码耦合，可以观察act和fgt的生命周期，实现自动监听，不需要手动在act和fgt中去调用，简化了代码增加了可维护的程度.

Viewmodel: Activity和Fragment就可以不用承担保留数据的责任，保留数据的任务交给ViewModel，极大降低数据和视图的解耦性，同时ViewModel会在configuration 更改时自动保留数据。

Livedata：可以做到在组件处于激活状态的时候才会回调相应的方法，从而刷新相应的 UI。

Feed back MVVM + recycleview

Model veiw viewmodel,

Recycleview:提供并真正操作ViewHolder，提供操作动画，高性能与插件化的高扩展性