

Joywe Design Model 文档

文件状态： [] 草稿 [√] 正式发布 [] 正在修改	当前版本：	V3.0
	作 者：	吴嘉怡 郑彧 吴茵茵
	完成日期：	2013/06/08

修订历史

序号	变更内容	日期	变更者
1	选择模式架构和原因	2013 年 5 月 1 日	吴茵茵
2	增加各用例交互图	2013 年 5 月 20 日	吴嘉怡
3	修改各用例交互图	2013 年 6 月 8 日	郑彧

工作分工

学号	组员	负责工作	用时统计
10389395	吴嘉怡	UML 包图，新建活动交互图	2 天
10389394	吴茵茵	个人设置交互图	2 天
10389392	何梓荣	文档整理，选择架构模式原因	2 天
10389429	郑彧	添加/删除好友交互图	2 天

一. 逻辑架构

1.1 架构模式

我们小组采用的架构模式是 MVC 模式，原因如下：

耦合性低

视图层和业务层分离，这样就允许更改视图层代码而不用重新编译模型和控制器代码，同样，一个应用的业务流程或者业务规则的改变只需要改动 MVC 的模型层即可。因为模型与控制器和视图相分离，所以很容易改变应用程序的数据层和业务规则。

模型是自包含的，并且与控制器和视图相分离，所以很容易改变应用程序的数据层和业务规则。由于运用 MVC 的应用程序的三个部件是相互独立，改变其中一个不会影响其它两个，所以依据这种设计思想能构造良好的松耦合的构件。

重用性高

MVC 模式允许使用各种不同样式的视图来访问同一个服务器端的代码，因为多个视图能共享一个模型，它包括任何 WEB 浏览器或者无线浏览器。由于模型返回的数据没有进行格式化，所以同样的构件能被不同的界面使用。例如，很多数据可能用 HTML 来表示，但是也有可能用 WAP 来表示，而这些表示所需要的命令是改变视图层的实现方式，而控制层和模型层无需做任何改变。由于已经将数据和业务规则从表示层分开，所以可以最大化的重用代码了。模型也有状态管理和数据持久性处理的功能，例如，基于会话的购物车和电子商务过程也能被 Flash 网站或者无线联网的应用程序所重用。

生命周期成本低

MVC 使开发和维护用户接口的技术含量降低。

部署快

使用 MVC 模式使开发时间得到相当大的缩减，它使程序员集中精力于业务逻辑，界面程序员集中精力于表现形式上。

可维护性高

分离视图层和业务逻辑层也使得 WEB 应用更易于维护和修改。

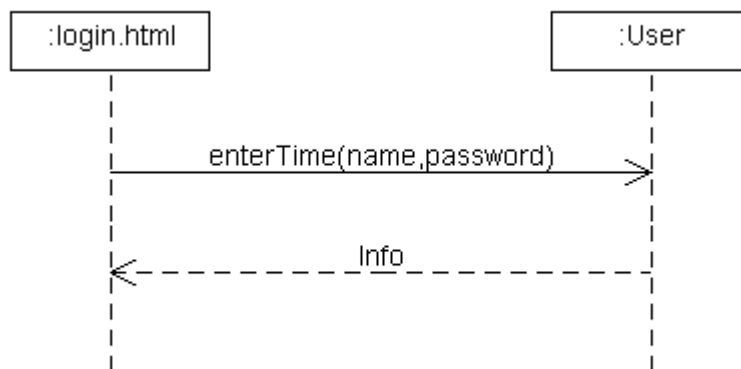
有利软件工程化管理

由于不同的层各司其职，每一层不同的应用具有某些相同的特征，有利于通过工程化、工具化管理程序代码。控制器也提供了一个好处，就是可以使用控制器来联接不同的模型和视图去完成用户的需求，这样控制器可以为构造应用程序提供强有力的手段。给定一些可重用的模型和视图，控制器可以根据用户的需求选择模型进行处理，然后选择视图将处理结果显示给用户。

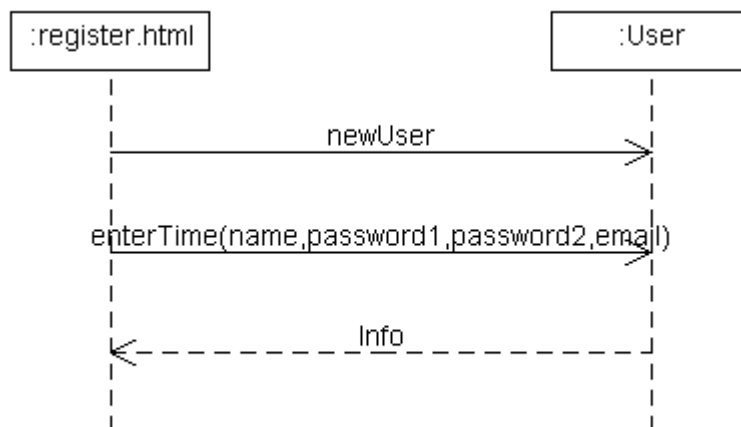
1.2 UML 包图

二. 动态设计

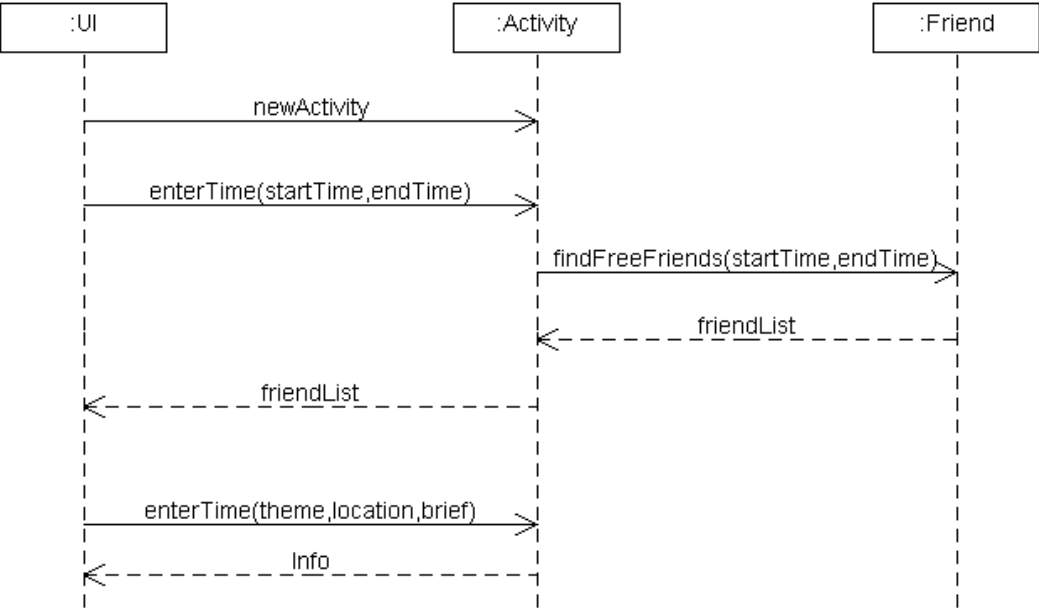
2.1 登陆交互图



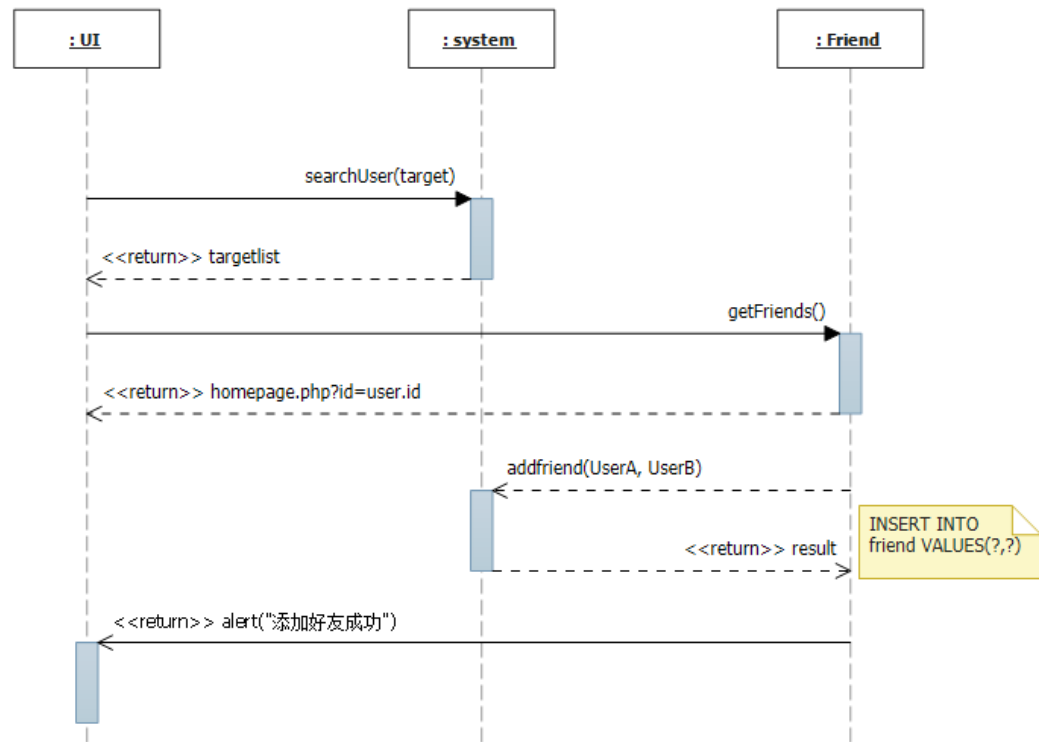
2.2 注册交互图



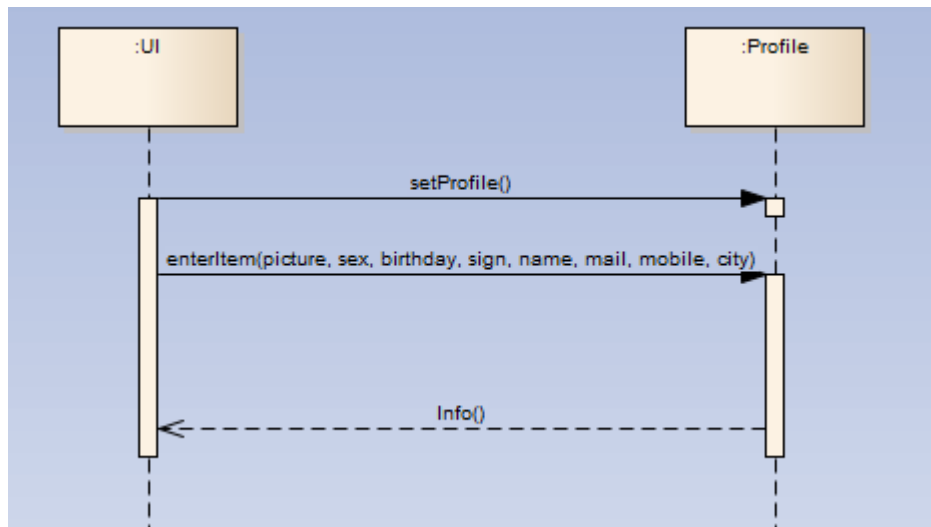
2.3 新建活动交互图



2.4 添加/删除好友



2.5 个人资料设置



2.5 密码修改

