第三次作业说明

一、作业概述

目的：

考察学生基于分布式组件技术（CORBA, JMI, DCOM, Web Service, and Message Queue）实现异构系统集成的能力。

场景：

随着互联网技和电子商务的发展，越来越多的互联网公司通过商业模式的创新在努力争夺着电子商务的市场，各式各样的 B2B、B2C、C2C 网站争先恐后地出现， 这些系统应用了各式各样的技术，通过异构系统的集成来支撑业务的运转。在本次作业中，我们将基于团购网站的简单模型来实现异构系统的应用集成，作业中包含了团购管理系统 A、团购网站系统 B、短信系统 C、零售商系统 D 以及银行系统 E 的实现原型。 现要求通过上述的五种分布式组件技术来实现这 5 个系统之间的业务交互，具体的接口和流程将在后续章节中详细描述。

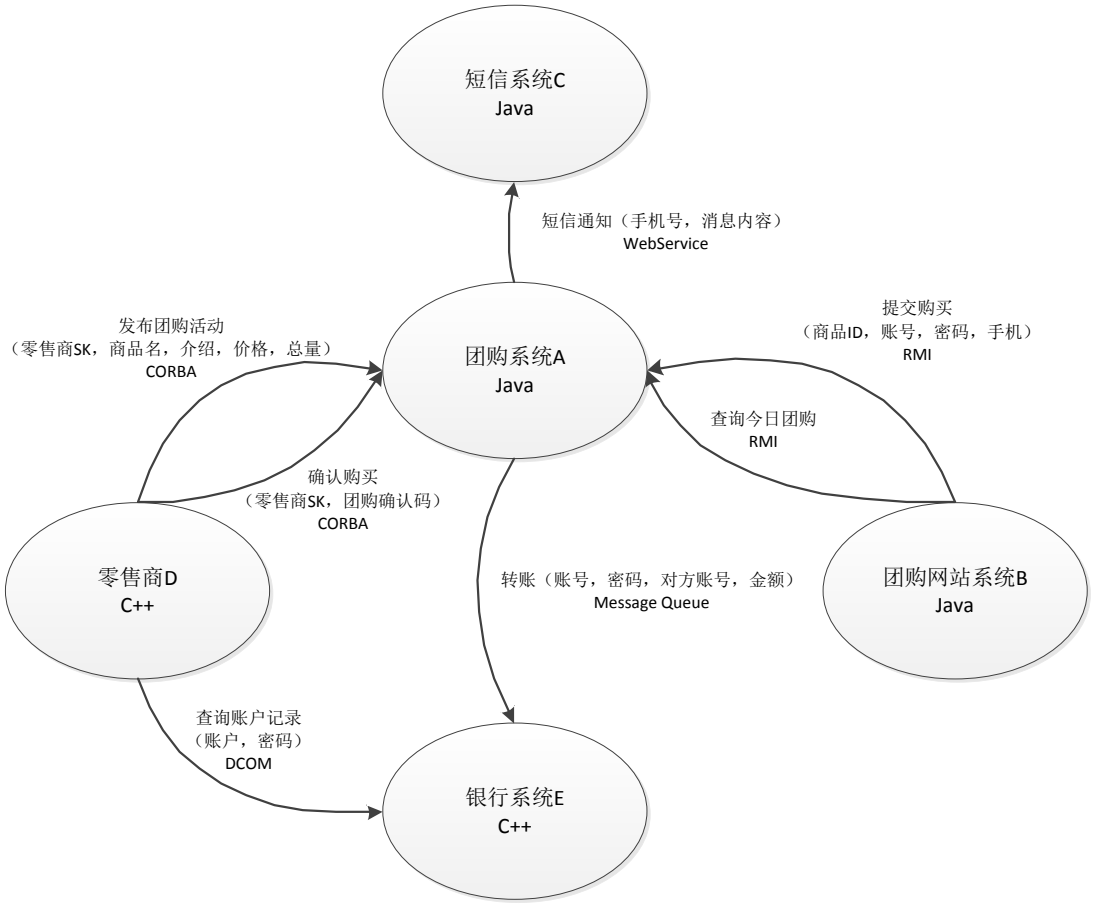
要求：

1. 上述 5 个系统的实现原型已经给出，只要求实现异构系统之间的交互，不要求对系统的实现细节进行了解或修改 ；

2. 根据下述的接口定义以及用例，分别按照要求使用相应的组件技术来实现不同系统之间的调用。

二、系统结构和接口定义

五个系统的结构可以用下图来描述：



1. 发布团购活动

   参数：

   零售商 **SK**（**Secret Key**）：分配给零售商的密匙代表当前零售商

   商品名：要发布的团购商品的名称

   介绍：关于该商品的介绍

   价格：团购的价格，不小于零

总量：最大允许团购的数量，不小于零，零表示无限制

返回值：

该商品的 ID，如果返回为空表示发布不成功

2. 确认购买

参数：

零售商 SK（Secret Key）：分配给零售商的密匙代表当前零售商

团购确认码：消费者在购买团购后得到的确认码

返回值：

True 表示成功，False 表示失败

3. 短信通知

参数：

手机号码：短信通知的目标手机号码

消息内容：短信通知的消息具体内容

返回值：

True 表示成功，False 表示失败

4. 提交购买

参数：

商品 ID：商品的 ID

账户：消费者的银行账号

密码：消费者银行账号的密码

手机：用户接收团购确认码的手机号

返回值：

True 表示购买成功，False 表示失败

5. 查询今日团购

   返回值：

   今日团购商品的列表

6. 转账

   参数：

   账号：转账的账号

   密码：转账账号的密码

   对方账号：接收转账的账号

   金额：转账金额

   返回值：

   True 表示转账成功，False 表示失败

7. 查询账户记录

   参数：

   账号：要查询的账号

   密码：要查询的账号对应的密码

   返回值：

   被查询账号的收支记录列表

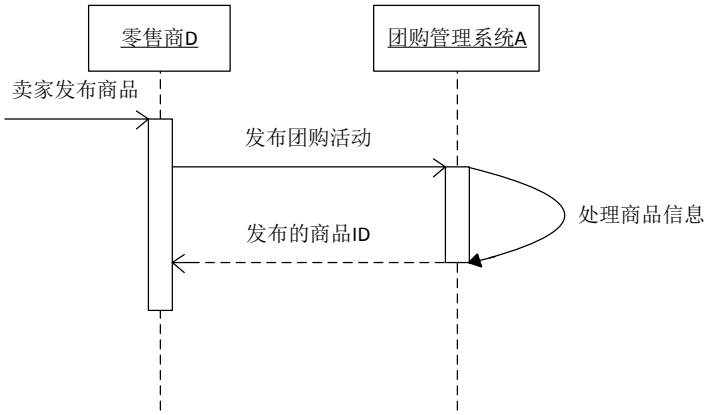
在实现时，请使用MOM或者WS来完成系统之间的交互。

三、业务流程定义

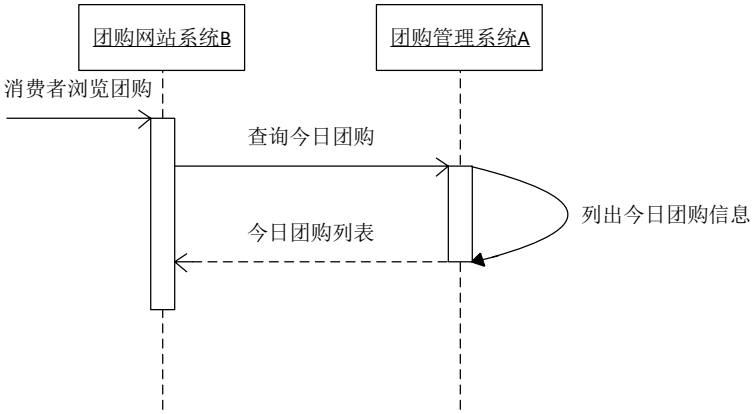
在本次作业中要实现的团购网站的场景非常简单，为了简化整个系统的业务逻辑，将主要考察重点放在异构系统之间通过分布式组件技术实现系统集成上，因此在五个系统的原型中只实现基本业务流程，而不考虑性能、安全性、容错性等非功能性需求。

因此，通过定义如下的四种基本业务流程，我们可以简单实现一次团购商品从发布、购买到领取的整个业务流程：

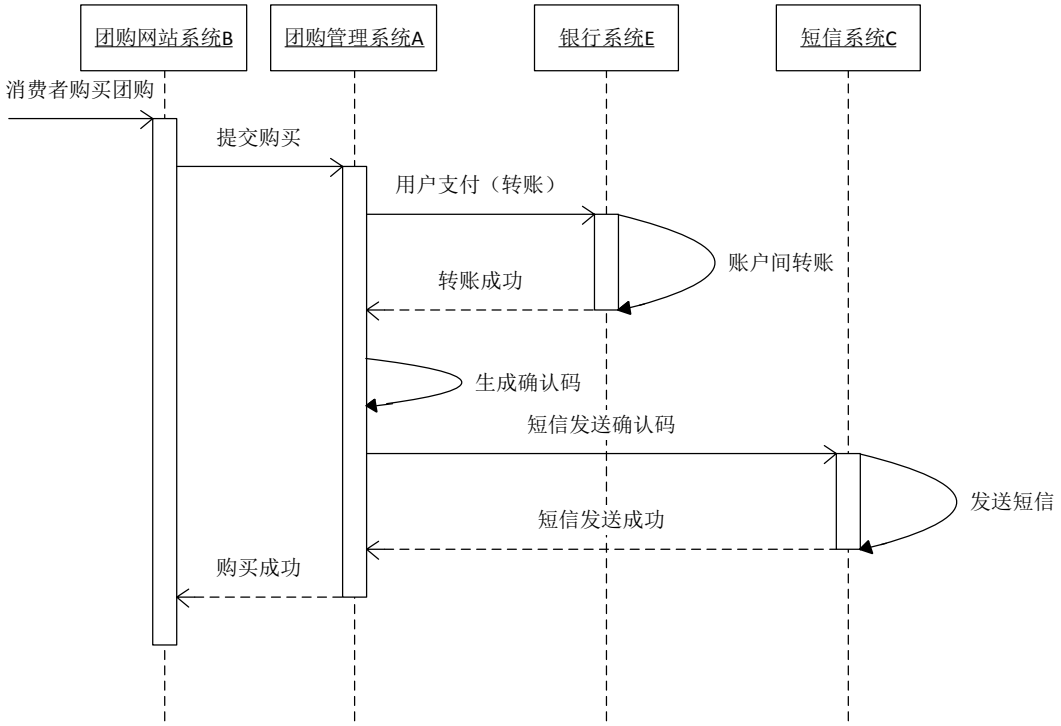
1. 发布团购



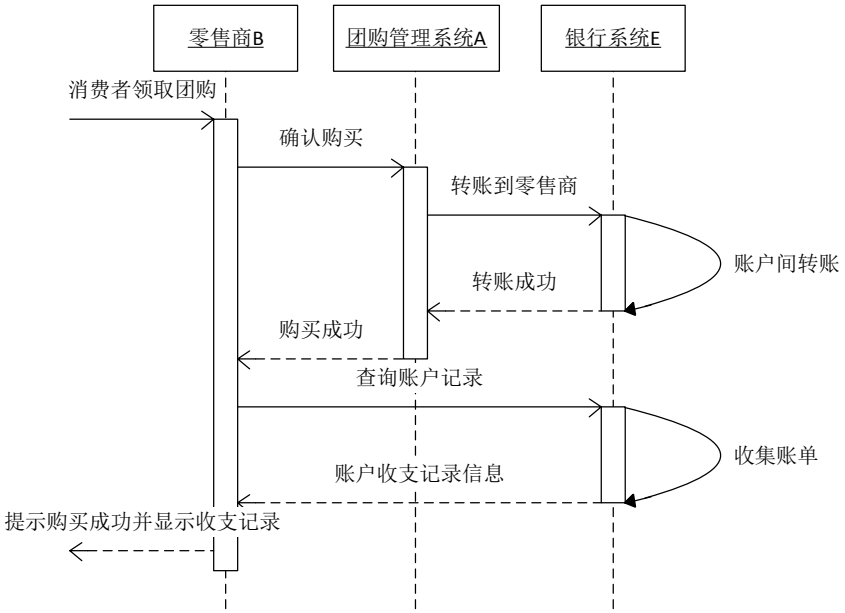
2. 查询团购



3. 购买团购



4. 领取团购商品



四、任务范围及实施

本次作业的主要目的是考察学生使用分布式组件技术实现不同系统之间集成的能力，因此为了降低实现系统的额外复杂度对考察目的的影响，本次作业已经给出了上述五个系统的简单实现，分别使用 C++和 Java 实现，已经编译成静态链接库或者 jar 包，伴随本文档给出。

下面的章节将主要描述每个系统具体的实现步骤。

银行系统

银行系统使用 C++实现，对应的静态链接库前缀为 BankSystem[1]，头文件为 BankSystem.h。需要实现的是分别利用 Message Queue 和 DCOM 技术分别暴露银行系统的两个方法，参考的 Main.cpp 如下（创建项目时使用 C++空项目即可）：

```
#include "BankSystem.h"
#pragma comment(lib, "BankSystem_vc10_mtd.lib")

int main(int args, char** argv)
{
    // TODO: use Message Queue to expose the transfer method as a service
    // TODO: use DCOM to expose the history method as a service
    return 0;
}
```

至于如何将两个方法分别使用 MQ 和 DCOM 暴露可以自由发挥，可以自由组织源代码结构，可以使用开源项目。

---

[1] 静态链接库为 lib 文件，后缀中包括了 Visual C++的版本（例如 vc10）以及编译时用的运行时环境（Runtime Environment：包括 Multi-Threaded 和 Multi-Threaded DLL 以及对应的 Debug 和非 Debug 版本，例如 Multi-Threaded 的 Debug 版本后缀为 mtd），在实现时请选择你当前使用的运行时环境所对应的版本。

零售商系统

零售商系统也是使用 C++实现，对应的动态链接库前缀为 RetailSystem，头文件为 RetailSystem.h，需要使用 DCOM 技术连接银行系统，使用 CORBA 连接团购管理系统。

实现时创建 C++空项目即可，并首先实现连个系统的接口，例如银行系统：

```
#include "RetailSystem.h"
class RemoteBankSystem : public BankSystem
{
public:
    list<record> listHistory(string account, string password)
    {
        // TODO: adapt to a remote procedure call to the bank system
    }
};
```

又如团购管理系统：

```
#include "RetailSystem.h"
class RemoteGroupPurchaseManagementSystem : public GroupPurchaseManagementSystem
{
public:
    bool publishGroupPurchaseItem(string sellerSecretKey, string productName, string
introduction, double price, int limit)
    {
        // TODO: adapt to a remote procedure call to the group purchase management system
    }
    bool confirmPurchase(string sellerSecretKey, string confirm)
    {
        // TODO: adapt to a remote procedure call to the group purchase management system
    }
};
```

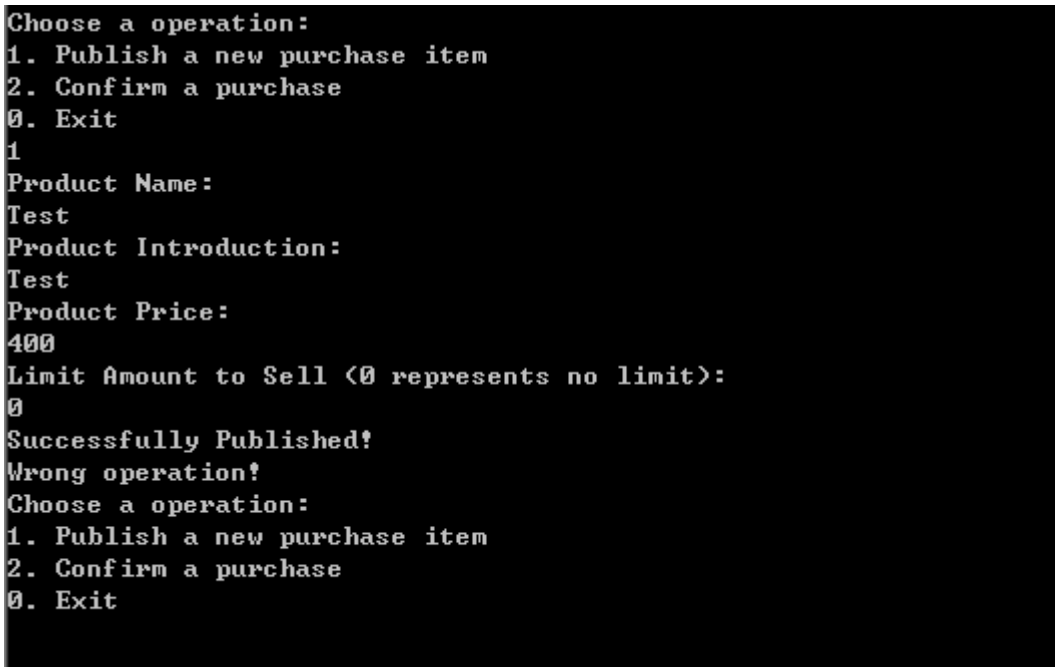
最后使用如下的 Main 函数即可启动零售系统：

```
#include "RetailSystem.h"
#pragma comment(lib, "RetailSystem_vc10_mtd.lib")
int main(int args, char** argv)
{
    RemoteBankSystem bank;
    RemoteGroupPurchaseManagementSystem gpms;
```

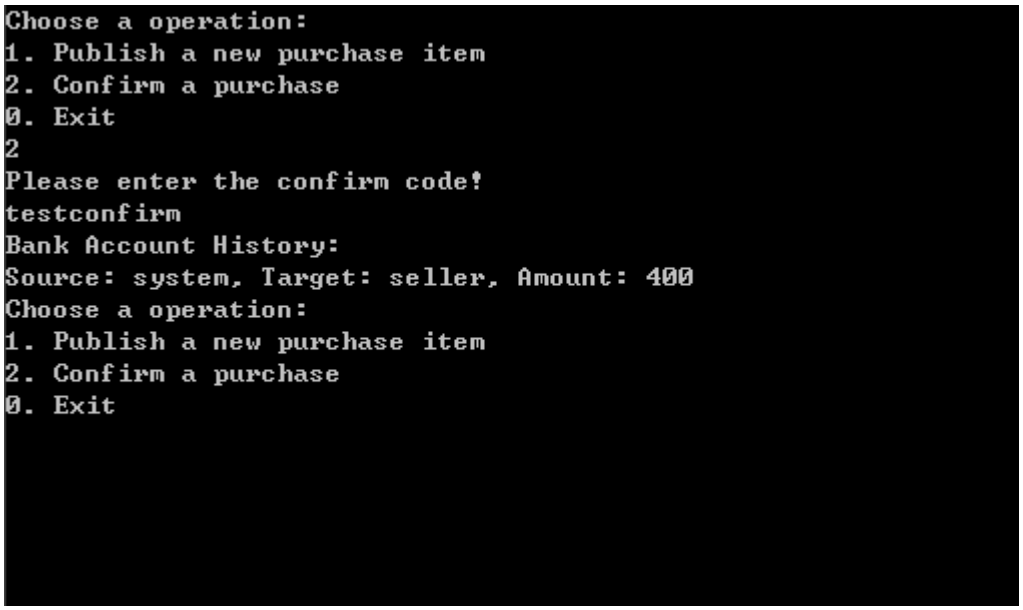
```
    return launchRetailSystem(&gpms, &bank);
}
```

启动零售系统后得到控制台界面如下：

并可以按照操作创建团购项目：



或者通过团购确认码完成一次消费：



可以自由组织源代码结构，可以使用开源项目。

注：以上图片来自测试环境截图

短信系统

短信系统是使用 Java 实现的，对应的 Java 包为 ShortMessageNotificationSystem.jar，

需要使用 WebService 技术将该系统接口暴露给团购管理系统，需要暴露的接口如下：

```
package assignment3;

public interface ShortMessageSender {

    boolean sendMessage(String receiver, String msg);
}
```

创建这个接口实例的方法为：

```
ShortMessageSender sms =
ShortMessageSenderFactory.createShortMessageSender();
```

可以自由组织源代码结构，可以使用开源项目。

团购管理系统

团购管理系统同样使用 Java 实现，对应的 Java 包为 GroupPurchaseManagementSystem.jar，需要使用 CORBA 和 RMI 技术分别将他的方法暴露给零售商系统和团购网站系统，并且需要使用 WebService 方法调用短信系统，使用 Message Queue 的方法连接银行系统。需要暴露的接口如下：

```
package assingment3;

import java.util.List;

public interface GroupPurchaseManagementSystem {

    boolean confirmPurchase(String sellerSecretKey, String confirm);

    List<GroupPurchaseItem> listGroupPurchase();

    boolean publishGroupPurchaseItem(String sellerSecretKey, String
productName, String introduction, double price, int limit);

    boolean submitPurchase(String itemId, String bankAccount, String
password, String phone);
}
```

在获得团购管理系统的实例之前，需要先实现银行系统的远程调用代理，接口如下：

```
package assingment3;

public interface BankSystem {

    boolean transfer(String account, String password, String target,
double amount);
}
```

以及短信系统的远程调用代理，接口见上一子章节中的描述。

最后，创建两个系统的远程调用代理，并且获得团购管理系统的实例：

```
GroupPurchaseManagementSystem instance =
GroupPurchaseManagementSystemFactory
.createGroupPurchaseManagementSystem(messageSender, bankSystem);
```

并且分别通过 CORBA 和 RMI 的方法来分别暴露团购管理系统的几个方法。

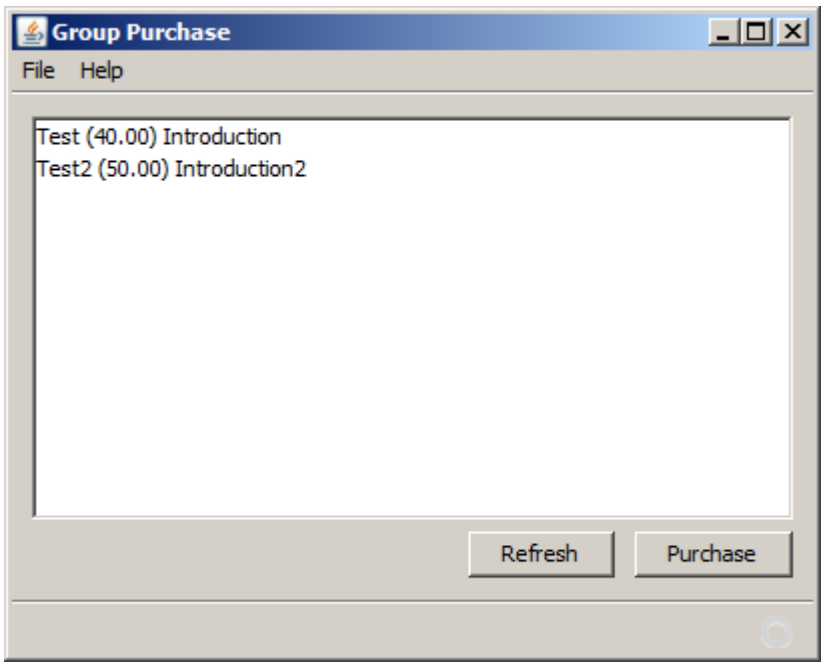
具体实现时可以自由组织源代码结构，可以使用开源项目。

团购网站系统

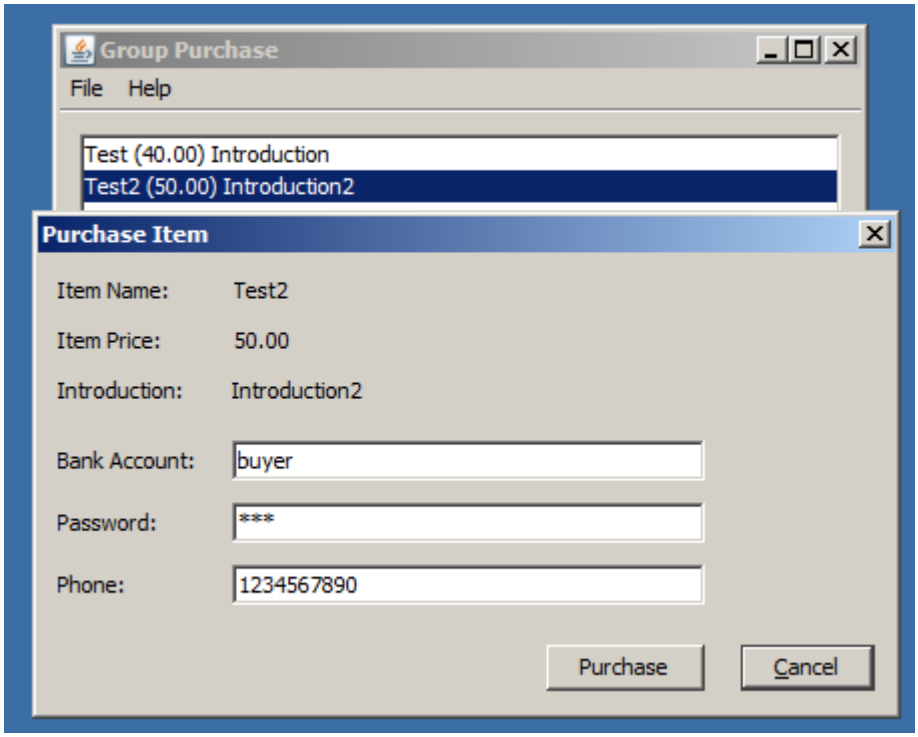
团购网站系统也是使用 Java 实现，在实现其原型系统时，为了更方便作业的开发，使用了 swing 桌面程序代替应有的网站，对应的 Java 包为 GroupPurchaseWeb.jar，并且使用到了 external-libs 中的 appframework-1.0.3.jar 以及 swing-worker-1.1.jar。学生需要使用 RMI 技术连接管理系统，在启动团购网站系统之前，需要先实现团购管理系统的远程调用代理类，其对应接口如下：

```
package assignment3;

import java.util.List;

public interface GroupPurchaseManagementSystem {

    List<GroupPurchaseItem> listGroupPurchase();

    boolean submitPurchase(String itemId, String bankAccount, String
password, String phone);
}
```

最后通过`GroupPurchaseWeb.launch(gpms);`即可启动团购网站系统，启动后界面如下：



并且可以进行购买：



具体实现时可以自由组织源代码结构，可以使用开源项目。

其它注意事项以及说明

- 以上原型系统的开发软件环境为：Windows 2008 R2 SP1 x64，Visual Studio 2010 Ultimate SP1，JDK 7 update 1，NetBeans 7.0.1。
- 原型系统中不包含数据库持久层，购买者的银行账号为 buyer 密码为 123
- 作业任务自由分工，但在检查时小组中每个同学必须至少能解释其中一项技术的实现细节
- 检查作业时，五个系统至少部署在两台机器上即可，其中 BCD 系统可以共用一台，AE 系统可以共用一台，两台机器通过以太网建立通讯连接