

ICS 52: Introduction to Software Engineering

Fall 2001

Instructor: Dr. Richard Taylor
TA: Girish Suryanarayana
Readers: Liang Jin, Volkan Aginler, Xiao Zhang

Assignment 3: Implementation

Version 3

Issued: Monday, 5th November 2001

Due: Monday, 19th November 2001(beginning of Discussion)

SUMMARY

For this assignment, you will implement part of the design of the Order Fulfillment Process as described in the official design document. Do not base your implementation on the design document that you created for assignment 2, only implementations based on the official design document will be graded. **As with the previous assignment, only a partial "official design document" is provided. It should be sufficient for your needs, though perhaps not "model".**

Item Likely to Change: The deliverable for this assignment is twofold: a hardcopy of the source code of your implementation and a floppy with the Java source files of your implementation. Remember the implementation needs to be in Java 1.3, since this is the version of Java with which your source code will be tested. What's likely to change? We will probably require electronic submission to a drop box. Details to follow.

Your implementation should follow, to the letter, the modules and interfaces of the design document. Each module should be implemented as a separate class. Your implementation should also be well-documented.

You will implement the following modules: Workflow Controller, File Reader and the Computation Module. You should also implement the abstract data types that you need. An implementation of the SMTP component, meeting the specs below, will be provided for you.

GRADING

100% for implementation and documentation

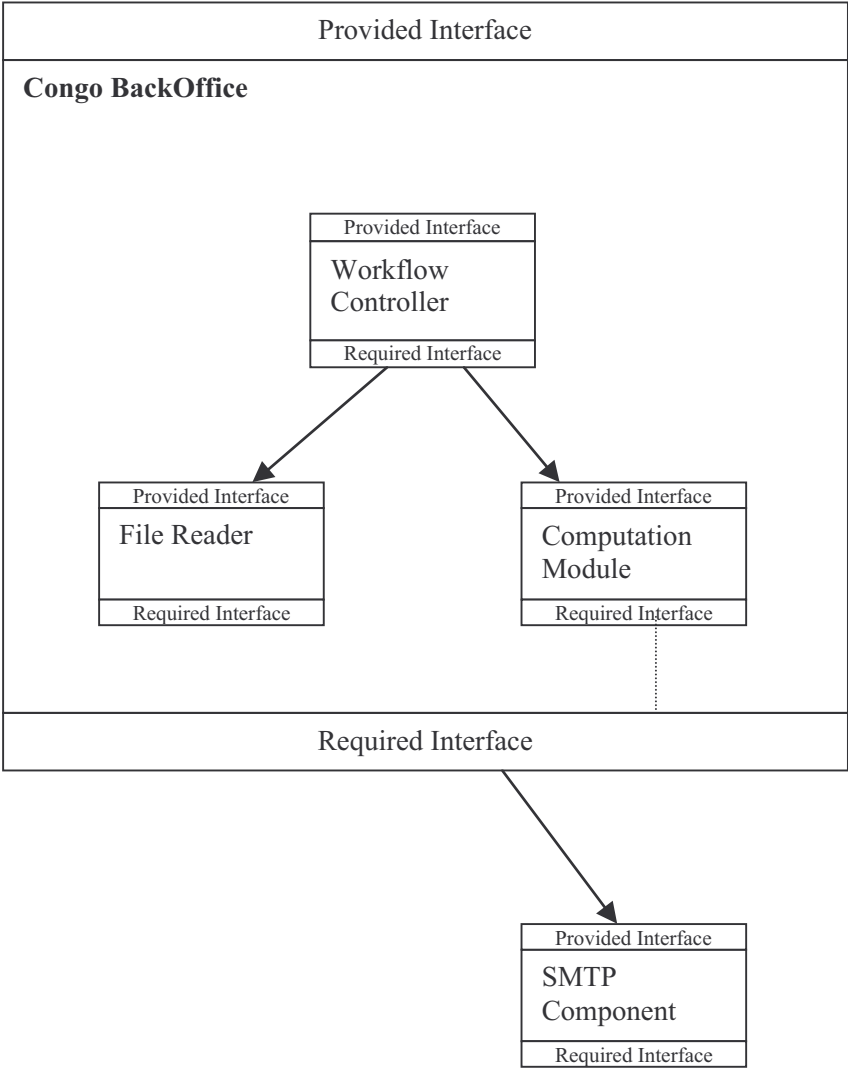
The assignment counts 7% toward your final grade for the course.

Note:

- **Do not work in teams to complete this assignment**
- **No late assignments will be accepted**

DESIGN DOCUMENT

Figure 1-1



Congo BackOffice

Purpose

The purpose of this module is to support numerous functions like reading the input files, interacting with the publishers and shippers, sending relevant information to the customer by emails and accounting. *For the purpose of this assignment*, Congo BackOffice deals only with reading the Order Input File and sending emails to the customer.

High-Level Module Design

The Congo BackOffice module is broken down into the relevant sub-modules, as shown in Figure 1-1.

Required Interface

The required interface of this module is the required interface of the Computation Module.

```
boolean sendEmail(String sender, String recipient,  
String subject, String message);
```

Provided Interface

This module has no provided interface.

Module ADTs NOTE: The requisite accessor functions (get/set) are not shown. Your implementation should provide them, however.

```
OrderRecord {  
    String orderRecordNumber;  
    String customerName;  
    Address customerAddress;  
    String customerEmailAddress;  
    String customerPhoneNumber;  
    int numberOfBooksPurchased;  
    BookInformation bookInformation[];  
    ShipmentInformation shipmentInformation;  
    CreditCardInformation creditCardInformation;  
}
```

```
Address {  
    String aptAndStreet;  
    String cityName;  
    String stateName;  
    String zipCode;  
}
```

```
BookInformation {  
    String bookTitle;  
    String bookAuthor;  
    Float bookPrice;  
}
```

```
ShipmentInformation {  
    String nameOfShippingCompany;  
    String modeOfShipment;  
    Float costOfShipping;  
}
```

```
}  
  
CreditCardInformation {  
    String typeOfCreditCard;  
    String creditCardNumber;  
}
```

Workflow Controller Module

Purpose

The purpose of this module is to manage the execution of the Order Fulfillment Process.

Required Interface

```
void readOrderInputFile() throws IOException;  
OrderRecord getNextOrderRecord();  
boolean sendConfirmEmail(OrderRecord orderRecord);
```

Provided Interface

This module has no provided interface.

File Reader Module

Purpose

The purpose of this module is to read the Order Input File (*see Assignment 2: SampleOrderRecordFile*), insert all the extracted order data obtained into an internal data structure and allow the Workflow Controller to get the order records.

Required Interface

This module has no required interface.

Provided Interface

```
void readOrderInputFile() throws IOException;
```

Description:

Reads the Order Input File, extracts all the order records and stores them in an internal data structure.

Exceptions:

IOException: If there is an error in reading the file.

```
OrderRecord getNextOrderRecord();
```

Description:

Gets the next order record from the internal data structure containing all the order records.

Returns:

Returns the next order record or null if there is none.

Computation Module

Purpose

The purpose of this module is to process the order record and compute some new values, and provide access to an external SMTP component that sends email with the processed order record data.

Required Interface

```
boolean sendEmail(String sender, String recipient,  
String subject, String message);
```

Provided Interface

```
boolean sendConfirmEmail(OrderRecord orderRecord);
```

Description:

Computes the total bill (Total Cost) and extracts the last 4 digits of the credit card number from the order record. Requests the SMTP component to send an initial email to the customer with the above computed data (total bill and last 4 digits of the credit card) and the rest of the order information [*see Assignment 2: Order Fulfillment Process for what information is sent to the customer*] confirming his order with Congo.com.

Returns:

Returns if the confirmation email was successfully sent.

Parameters:

orderRecord: It contains all the information about the particular order. It includes the customer's email address.

SMTP (Simple Mail Transfer Protocol) Component

Purpose

The purpose of this module is to send email with order record information to the address specified by the Congo BackOffice.

Required Interface

This module has no required interface.

Provided Interface

```
boolean sendEmail(String sender, String recipient,  
String subject, String message);
```

Description:

Sends an email to the recipient from the sender with the relevant subject and message.

Returns:

Returns if the email was successfully sent.

Parameters:

sender: The sender's email address.

recipient: The recipient's email address.

subject: The subject of the email.

message: The message in the email.