INFO323
Distributed Information Systems
Course Project

1 Deliverables

Phase 1 – 5:00 pm, Friday 12 April. The design document is worth 5%, and the service and client implementations are worth 5%.

Phase 2 – 5:00 pm, Friday 10 May.

2 Introduction

The purpose of this project is to get you working on a realistic problem where an organisation has some existing services for performing some of their day to day business operations, and wish to combine these services to create a new service that automates an additional business operation.

In phase 1 of the project you will be describing the system that you will be working on, creating the services that represent the existing services, and creating clients that test the features of those services.

In phase 2 you will be creating the composite service that makes use of the existing services, and an AJAX client for interacting with that service.

3 Phase 1

Think of a domain where a distributed system could be used to help manage information used by a business or other entity in day-to-day operations.

You will need to come up with three operations that this entity needs to perform that can be automated via services.

Two of the services can be unrelated, but the third (which is the composite service) should make use of the other two services.

As an example, a retailing business may have existing services for:

- Monitoring stock levels of products.
- Producing invoices for billing customers.

The composite service could be an ordering service which allows an order to be placed. As part of its operation this service would need to check the stock levels of the products being ordered to ensure there is stock available for the order, and then produce an invoice to be sent to the customer. It would make use of the existing services to perform these tasks.

Remember the principles of SOA services. Each service should perform a single task (although may use other services to help it perform this task).

3.1 Design Document

Create a document that describes the system you are creating.

You document should include:

- A brief description of the domain/business operations.
- A description of any domain classes that would be used by the services. List any fields that the domain classes would need to contain.
- A description of what each service does. Ensure that you specify which of the services represent the "existing" services, and which one represents the "composite" service.

  Describe the interface that the services provides. Describe any inputs or outputs that the services need.
- For each of the existing services state which middleware (SOAP/RPC, RMI, or REST) you are using to implement the service. Also explain why the chosen middleware is suitable for implementing the service.

  You do not need to discuss the middleware for the composite service. You will be using an integration middleware that you haven't seen yet for implementing this service.
- A description of how the composite service would use the existing services. Does the order in which the existing services are invoked matter? Does the output of one of the services feed into the input of the other service?

3.2 The Services

Implement the two "existing" services.

One of the services must be either RMI or SOAP/RPC, and the other must be RESTful.

The implementations should be simple. You can hard code any data that the service depends on, although you should try to extract this data into a DAO class if you can. The implementation of the operations can just print messages to the console where other business operations would be involved. For example if you have an operation that would email an invoice to a customer, then displaying a message containing ("emailing invoice '1234' to customer 'fred@yahoo.com'") is sufficient.

3.3 Clients

Implement a client for each of your services. You can either create a console based client, or a Swing GUI client. The client should make use of all of the features that the service provides.

Again your implementation should be simple. Hard code any data that needs to be sent to the service (in the case of a console client). Display any messages that the service returns to the client.

4 Phase 2

This section will be made available at a later date.

5 Submission

We will be using the Wiki to submit the project.

Click the ‘Project Submission’ link in the navigation menu, and then click ‘Phase 1’. You should see a page with your user name as the title. This is your personal project submission page. Click your user name to open the page.

Add your design documentation directly to the page, by clicking the ‘Edit Page’ button and adding your content via the editor. Leave your user name as the title of the page.

Zip up your NetBeans projects, and attach them to the page by clicking the ‘attach file or image’ button.