Rational Software

Payroll System Subsystem Design Solution

Version 2004
### Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Issue</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/01/2000</td>
<td>2000</td>
<td>Generation for beta</td>
<td>Shawn Siemers</td>
</tr>
<tr>
<td>10/03/2000</td>
<td>2000</td>
<td>Final release</td>
<td>Shawn Siemers</td>
</tr>
<tr>
<td>01/14/2003</td>
<td>2003</td>
<td>Final Release</td>
<td>Alex Kutsick</td>
</tr>
<tr>
<td>05/20/2004</td>
<td>2004</td>
<td>Generation for beta</td>
<td>Alex Kutsick</td>
</tr>
</tbody>
</table>
# Table of Contents

1. Exercise: Subsystem Design
   
   1.1 BankSystem
      1.1.1 Interface Realizations
      1.1.2 Subsystem Dependencies Class Diagram
   
   1.2 PrintService
      1.2.1 Interface Realizations
      1.2.2 Subsystem Dependencies Class Diagram
   
   1.3 ProjectManagementDatabase
      1.3.1 Interface Realizations
      1.3.2 Subsystem Dependencies Class Diagram
Payroll System Subsystem Design Solution

1. Exercise: Subsystem Design

1.1 BankSystem

1.1.1 Interface Realizations

IBankSystem::deposit

1. deposit(Paycheck, BankInformation)

1.2. create(Deposit, paycheckAmount, BankInformation.routingNumber)

1.3. submit(BankTransaction)

1.3.1. // submit(BankTransaction)

© Copyright IBM Corp. 2004
IBankSystem::deposit

The formal interfaces with the external Bank System would need to be defined early in the process and represented here. Such definition is considered out of scope for this course.

IBankSystem
1.1.2 Subsystem Dependencies Class Diagram

Bank System Dependencies

To access the subsystem interface

1.2 PrintService

1.2.1 Interface Realizations

IPrintService::print
PrintService

Client

Client

PrintService

---

Printer

Printer

Paycheck

Employee

Paycheck

Create an image of the Paycheck from the given Paycheck for the specified Printer

Build an image of the Paycheck from the retrieved data for a specific Printer

Print the generated image

1. print(Paycheck, String)

1.1. // create(Paycheck)

1.1.1. // getEmployee()

1.1.2. // getEmployeeName()

1.1.3. // get Employee ID()

1.1.4. // getAmount()

1.1.5. // buildPrintImage()

Include similar operations for all Employee information that is to appear on the printed Paycheck

Include similar operations for all Paycheck information that is to appear on the printed Paycheck

The formal interfaces with external printers would need to be defined early in the process and represented here. Such definition is considered out of scope for this course.

The formal interfaces with external printers would need to be defined early in the process and represented here. Such definition is considered out of scope for this course.

The formal interfaces with external printers would need to be defined early in the process and represented here. Such definition is considered out of scope for this course.
IPrintService

<<interface>>
IPrintService
(from External System Interfaces)
+ print(aPaycheck : Paycheck, onPrinter : String)

<<subsystem proxy>>
PrintService
+ print(aPaycheck : Paycheck, onPrinter : String)

PrinterInterface
+ // print(theImage)

PrinterImage

PaycheckPrinterImage
+ // create(fromPaycheck : Paycheck)
+ // buildPrintImage()

Paycheck
- amount
+ // create with amount(forAmount : float) : Paycheck
+ // getAmount()
+ // getEmployee() : Employee

Employee
- name
- employee id : int
- social security number
- address
+ // get Employee ID() : int

For every printer the system must communicate with, a PrinterInterface, PrinterImage, and a PaycheckPrinterImage class must be defined.

Anything that can be printed must have a specific PrinterImage class defined, and that class must inherit from the base PrinterImage class.

© Copyright IBM Corp. 2004
1.2.2 Subsystem Dependencies Class Diagram

PrintService Dependencies

To access the subsystem interface

<<subsystem>>
PrintService
(from Business Services)

External System Interfaces
(from Business Services)

Payroll Artifacts
(from Business Services)

1.3 ProjectManagementDatabase

1.3.1 Interface Realizations

IProjectManagementDatabase::getChargeNumbers
ProjectManagement
   DatabaseClient

1. getChargeNumbers(String)

ProjectManagementDatabase

1.1. getChargeNums(string)

DBChargeNumbers

1.1.1. createStatement()
1.1.2. executeQuery(String)
1.1.3. new()
1.1.4. getString()
1.1.5. getString()
1.1.6. new(projectName, value)
1.1.7. add(ChargeNum)

: ChargeNumList

: ResultSet

: Statement

: ProjectManagement

© Copyright IBM Corp. 2004
IProjectManagementDatabase::getChargeNumbers

```
1. getChargeNumbers(String)
1.1. getChargeNums(string)
1.1.1. createStatement()
returns a Statement
1.1.2. executeQuery(String)
DBChargeNumbers builds the sql statement to retrieve the charge
numbers and passes it to Statement to be executed. A ResultSet is
returned.
1.1.3. resultSet()
Create a list to hold all retrieved data
1.1.4. getString()
Retrieve the project name for the charge number
1.1.5. getString()
Retrieve the value for the charge number
1.1.6. new(projectName, value)
Build the class from the retrieved data.
1.1.7. add(ChargeNum)
Add the retrieved charge number to the list to be returned
```

To read a class, the ProjectManagementDatabase proxy class asks the DBChargeNumbers class for the list of available charge numbers from the database. The DBChargeNumbers creates a new statement using the Connection class createStatement() operation. The statement is executed and the data is returned in a ResultSet object. The DBChargeNumbers then creates a new instance of the ChargeNum persistent class and populates it with the retrieved data. The data is returned in a collection object, the ChargeNumList class.

IProjectManagementDatabase::initialize

```
1. initialize()
1.1. initialize()
1.1.1. getConnection(url, user, pass)
1.1.1.1. // getConnection()
```

The formal interfaces with the external Project Management System would need to be defined early in the process and represented here. Such definition is considered out of scope for this course.
IProjectManagementDatabase::initialize

A "plural" DBClass was defined because the charge numbers always are retrieved as a set (a list of the available charge numbers).

IProjectManagementDatabase

<<Interface>>

+ getChargeNumbers(criteria : String) : chargeNumList

<<subsystem proxy>>

ProjectManagementDatabase

+ getChargeNumbers(criteria : String) : chargeNumList
+ initialize()

ChargeNumList

+ new()
+ add(theChargeNum : ChargeNum)

DBChargeNumbers

+ getChargeNums(criteria : string) : ChargeNumList
+ initialize()

ChargeNum

- projectName
- value
+ // getProjectName()
+ // getValue()
+ new(projectName, value)

ResultSet

+ getString()
1.3.2 Subsystem Dependencies Class Diagram

ProjectManagementDatabase Dependencies

To access the subsystem interface

```
<<subsystem>>
ProjectManagementDatabase
(from Business Services)
```

External System Interfaces
(from Business Services)

```
java.sql
(from Middleware)
```