Employee Payroll System

Bobby Sharp

Grant Parker

Houston Walley

Ian Fair

Lucas Elmore

**Table of Contents**

**Domain Analysis**

**Problem Statement**

**Conceptual Domain Model (i.e., Class Domain Model)**

* + 1. **Classes, Associations, Attributes**
    2. **Generalization and Specialization**
    3. **UML OCL Constraints (at least 5 constraints)**

**Domain State Model**

**Application Analysis**

**Application Interaction Model (at least 15 top-level use cases)**

* + 1. **Use Cases**
    2. **Scenarios**
    3. **Sequence Diagrams**

**Application Class Model**

* + - 1. **Classes, Associations, Attributes**
      2. **Generalization and Specialization**
      3. **UML OCL Constraints (at least 5 constraints)**

**Application State Model**

**Model Review (1-2 pages) - Evaluation of your models for completeness, consistency, and correctness criteria (within and across diagrams)**

Domain Analysis

Problem Statement

We are to design software for a simple employee payroll system for an organization. The payroll system will be used to keep track of the payments distributed to each employee of the organization, and provide reporting capabilities to users and administrators. The organization will have many different types of employees. An employee can be temporary or permanent, and can be paid on an hourly, two week, or monthly basis. Employees can also have various grades corresponding to their position in the company. Employees can also have various levels of access to the system. An administrator can read and modify the data in the payroll system. Some employees may have additional privileges, including the ability to override.

The system will support different types of payment. These include remuneration (additional pay offered for completion of specific services) and overtime pay (1.5 times normal pay rate) for employees who work more than their allotted work schedule. The system also allows for other unspecified forms of bonus payment.

The system keeps track of all deductions for each employee. Each deduction will be automatically removed from the employee’s pay for all future paychecks. The system allows the user to add new deduction types for each employee. A user can also remove deductions for an employee. The system allows for deductions from taxes, charitable contributions, retirement, or other unspecified deductions. The system can total all income for a given employee’s paycheck, along with the deductions and bonus payment for the paycheck. The system can compute an employee’s net take-home pay by subtracting all deductions from the employee’s total income.

The payroll system can calculate many additional values for a given paycheck. It can compute the total charitable contributions from a paycheck. It can also compute all contributions to an employee’s retirement fund for each paycheck. Additionally, the system can compute all taxes paid for a given paycheck. The payroll system can also compute an employee’s base pay for a given paycheck and an employee’s overtime and remuneration pay for that paycheck.

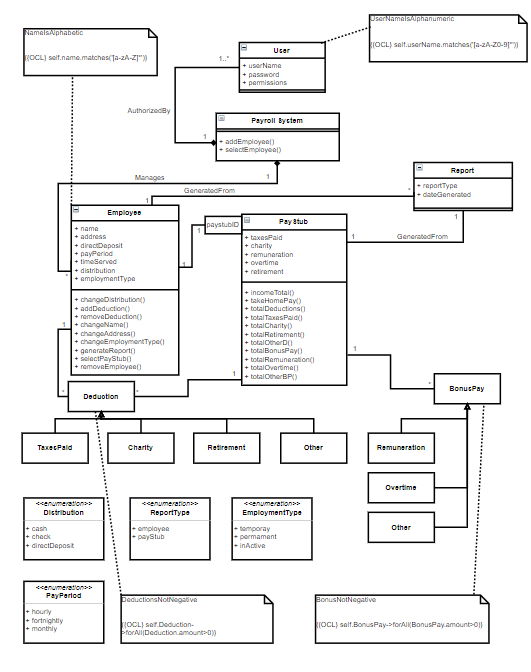
The payroll system will allow a user to change the method of distributing paychecks. Different paycheck distribution styles supported by the system include distributing paychecks as cash, direct deposits, and checks. The method of paycheck distribution is unique to each employee.

The payroll system allows a user with appropriate permissions to edit various aspects of an employee’s record. A user can edit an employee’s name and/or mailing address and these changes will be reflected in the system. A qualified user can also change an employee’s employment type, permissions, payment frequency, or preferred method of payment distribution.

New employees can be added to the payroll system through registration. Registration of a new employee requires the user to provide the employee’s name, address, employment type, payment frequency, and preferred method of payment distribution. If the required information is not provided, then registration fails and the system will ask for the new employee’s information again. Employees can also be removed from the system if they are no longer actively working for the organization. If an employee is removed, his or her record will stay in the payroll system (for legal/tax purposes) but their employment status will be updated to inactive.

The payroll system provides reporting capabilities, allow the user to generate a report for an employee that contains that employee’s position, grade, employment status (temporary/permanent), pay period, privileges, base pay, additional payments, deductions, charitable contributions, retirement contributions, taxes, and net take-home pay. A user can also generate a report for a specific paystub of a given employee that includes the employee’s name, the date of the pay period for the associated paystub, and the employee’s take home pay, base pay, deductions, and bonus pay for the pay period. When specifying which paystub to generate a report for, the system will prompt the user for a paystubID. This paystubID is unique among all paystubs for a given employee, though 2 different employees could have paystubs with the same ID (this won’t be a problem as paystubs are identified by both their paystubID AND the employee they are associated with). A paystubID is generated by converting the first day of its corresponding pay period into a whole number (e.g. a paystub for a given employee’s pay period that started on September 30, 2019 would have a paystubID of “20190930”). The payroll system can generate a report of all employees currently registered in the system. A report on all current employees will not contain any inactive employees.

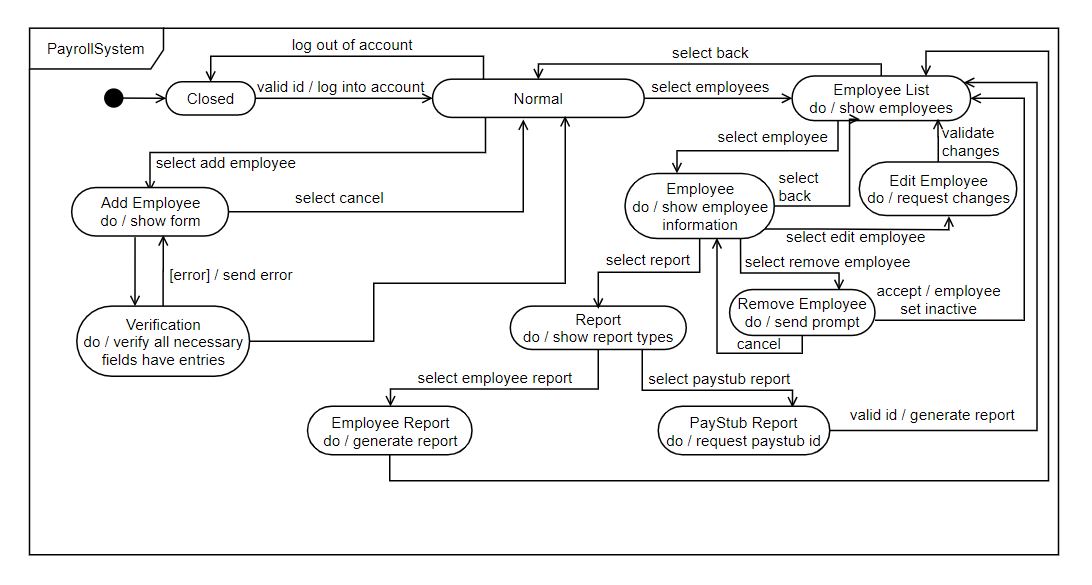
Domain Class Model



Domain State Models

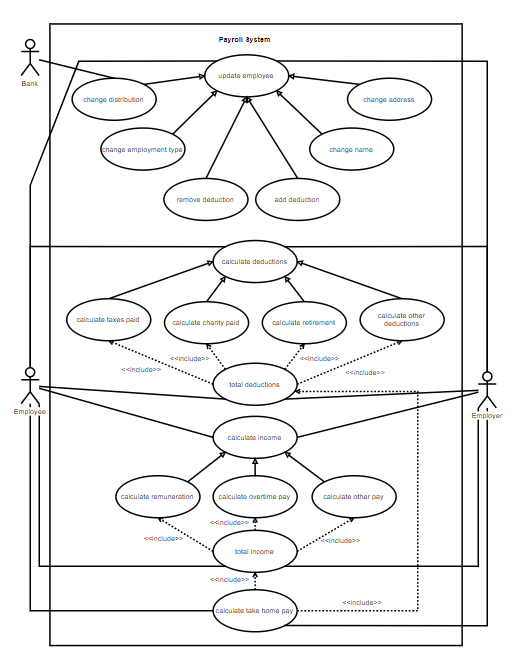
Our model has 1 class with multiple potential states, thus has 1 state diagram for the class PayrollSystem.

# PayrollSystem



**Application Interaction Model**

**Use Cases**

****

**Use Case:** Payroll System

**Summary:** The Employer (user) and Employee (user) can select to make certain changes to the payroll system.

**Actors:** The Employer, The Employee

**Preconditions:** The Payroll System is waiting user to perform an action.

**Description:** A user (employer) or user (employee) starts by attempting to log in to the payroll system. The System can then ask for credentials from the user (employer) or user (employee). The user can then submit login credentials, the payroll system then can either accept the login credentials if correct, if incorrect the payroll system rejects the users attempt to log in and then logs out. If credentials submitted by user were validated the user can then select employees from the user interface. The system then recognizes the user has read permission and displays the employee list. The user can then select an employee from the list. The user decides to select an employee from the list to which the system then displays employee info, the user can then choose to edit the employee in the list by adding info, removing the employee from the list, changing the name, changing the address, generate report. The user can also edit the employee’s info to change distribution. If change distribution is selected the system prompts the user to select the new form of payment distribution such as direct deposit, from there the user can select to add or remove deductions. The system then prompts the user to add or remove deduction info and to select deduction type, user can enter a deduction value and name of deduction. The user can also select to Generate report from the employee list on a specific employee. The user then selects report type, then paystub report, the system then prompts the user to enter paystub id. User then enters paystub ID which the system can reject or validate if correct info submitted. The system makes necessary changes submitted by the user.

**Exceptions:** *Don’t have permission*: If the user provide invalid login credentials, the system does not give the user permission access to the system.

*Cancelling*: The User can cancel any changes or edits made to the system.

*Invalid edit*: An invalid edit can be made such as incorrect deduction type value which can’t be less than *0*.

**Postconditions:** The system is waiting for user to perform an action.

**Scenarios and Sequence Diagrams**

1. Bobby Sharp attempts to log into the payroll system. System asks for credentials.

Bobby provides the wrong credentials.

System rejects credentials and returns to “closed” state.

A close up of text on a white background

Description automatically generated

2. Bobby Sharp attempts to log into the payroll system. System asks for credentials.

Bobby provides his credentials. System validates login.

Bobby logs out. System returns to “closed” state.

A screenshot of a cell phone

Description automatically generated

3. Bobby Sharp attempts to log into the payroll system. System asks for credentials.

Bobby provides his credentials. System validates login.

Bobby chooses “select employees” from the user interface.

System recognizes that the user does not have permission to access employee list, and rejects the request.

Bobby logs out. System returns to “closed” state.

A close up of text on a white background

Description automatically generated

4. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A screenshot of text

Description automatically generated

5. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Bobby Sharp” from the employee list. System displays employee info.

Grant selects “remove employee”. The employment type of “Bobby Sharp” is set to “inactive” in the system.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A screenshot of a cell phone

Description automatically generated

6. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Ian Fair” from the employee list. System displays employee info and options.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A close up of a white wall

Description automatically generated

7. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Ian Fair” from the employee list. System displays employee info and options.

Grant selects “edit employee”. System displays edit options.

Grant selects “Change distribution”. System prompts him to select the new form of payment distribution.

Grant selects “cash”. The system sets the payment distribution of “Ian Fair” to “cash”.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A screenshot of text

Description automatically generated

8. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Ian Fair” from the employee list. System displays employee info and options.

Grant selects “edit employee”. System displays edit options.

Grant selects “add deduction”. System prompts him for deduction info.

Grant selects “charity” for deduction type and $30 for deduction amount. He types “Kars 4 Kids” as the deduction name.

System adds new deduction (Name: “Kars 4 kids”, type: charity, amount: $30) for “Ian Fair”.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A close up of text on a white background

Description automatically generated

9. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Ian Fair” from the employee list. System displays employee info and options.

Grant selects “edit employee”. System displays edit options.

Grant selects “add deduction”. System prompts him for deduction info.

Grant selects “charity” for deduction type and $-30 for deduction amount. He types “Kars 4 Kids” as the deduction name.

System rejects new deduction: deduction amount must be greater than zero.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A close up of text on a white background

Description automatically generated

10. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Ian Fair” from the employee list. System displays employee info and options.

Grant selects “edit employee”. System displays edit options.

Grant selects “change name”. System asks for new name.

Grant enters “Houston Walley” as the new name.

System changes name from “Ian Fair” to “Houston Walley”.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A close up of text on a white background

Description automatically generated

11. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Houston Walley” from the employee list. System displays employee info and options.

Grant selects “edit employee”. System displays edit options.

Grant selects “Change address”. System prompts grant to provide new address.

Grant provides “424 Big Boy Way” as new address. System updates address.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A screenshot of a cell phone

Description automatically generated

12. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Houston Walley” from the employee list. System displays employee info.

Grant selects “generate report”. System prompts user for the report type.

Grant selects “employee report”. System generates a report with all employee information for “Houston Walley”.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A close up of text on a white background

Description automatically generated

13. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Houston Walley” from the employee list. System displays employee info.

Grant selects “generate report”. System prompts user for the report type.

Grant selects “paystub report”. System prompts Grant for a paystub ID.

Grant enters “20200124” as the paystub ID. System generates a report containing Houston Walley’s paystub information for the pay period starting on January 24, 2020.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A close up of a white wall

Description automatically generated

14. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Houston Walley” from the employee list. System displays employee info.

Grant selects “generate report”. System prompts user for the report type.

Grant selects “paystub report”. System prompts Grant for a paystub ID.

Grant enters “20200127” as the paystub ID. System rejects paystub ID: no paystub for Houston Walley corresponding to a pay period starting on January 27, 2020.

Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

A close up of a white wall

Description automatically generated

15. Grant Parker attempts to log into the payroll system. System asks for credentials.

Grant provides his credentials. System validates login.

Grant chooses “select employees” from the user interface.

System recognizes the user has read permission and displays the employee list.

Grant selects “Lucas Elmore” from the employee list. System displays employee info and options.

Grant selects “edit employee”. System displays edit options.

Grant selects “remove deduction”. System prompts him for name of deduction to be removed.

Grant selects “401k”. System removes deduction “401k” from employee “Lucas Elmore”.

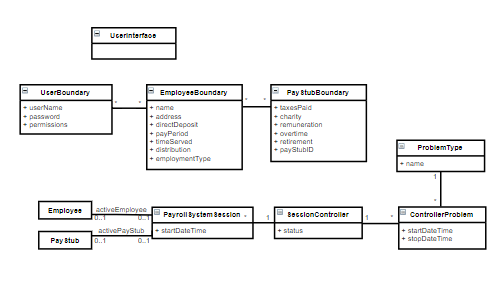
Grant selects “back” on the user interface. Payroll System returns to the main screen.

Grant logs out. System returns to “closed” state.

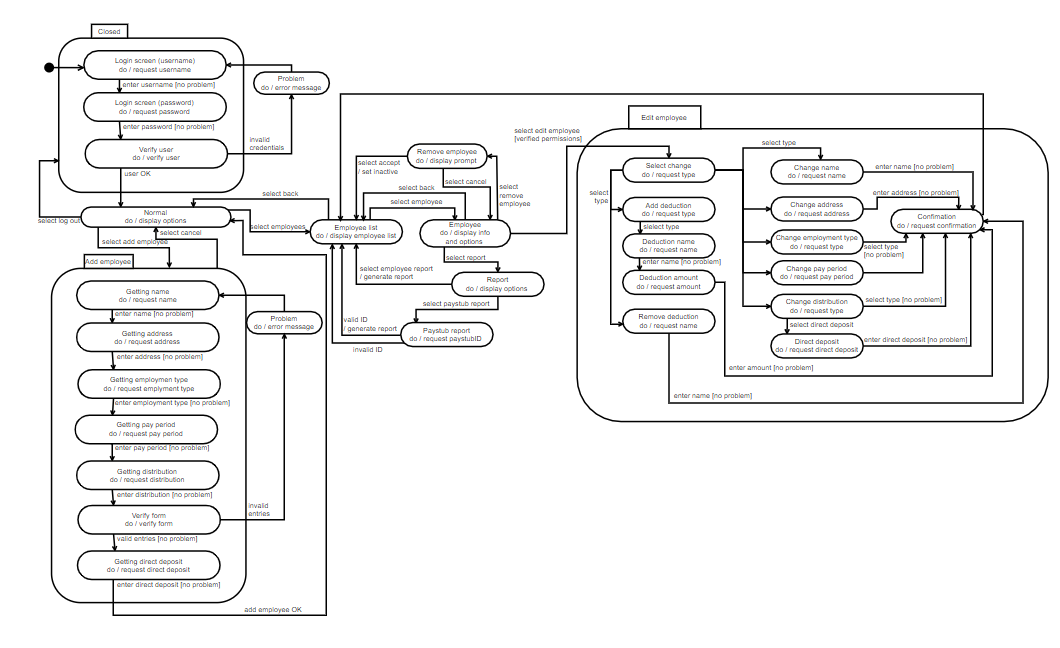
A screenshot of a cell phone

Description automatically generated

**Application Class Model**

****

**Application State Model**

****

**Model Review**

Our model allows for the implementation of a payroll system that satisfies all of the criteria of the problem statement. As shown by our Domain Class Model, our payroll system is composed of many employees, and one or more payroll system users. Each employee in the system has a record of multiple paystubs, with each paystub identified by a paystubID (this is illustrated by the 1-to-1 association between Employee and PayStub with a paystubID as a qualified association of PayStub).

A qualified user can enter the log into the payroll system and read/write data if their permissions allow it. The UserInterface (specified in the Application Class Model) allows a user to navigate through the system. A user can select ‘Employees’ to view a list of past and present employees, with each employee specified as temporary, permanent, or inactive (used to keep track of the paystubs of employees no longer working for the organization, for tax purposes).

The PayrollSystem itself has two functions: one to add an employee to the payroll system and one to select an employee from the employee list. All other functions are handled by the employee or PayStub objects themselves. If a user is on the main screen (“Normal” in the state diagrams), they can choose to add an employee. This will take the user to the “Add employee” window, where they will be prompted to provide all the necessary information to add a new employee to the PayrollSystem. If any of the entries are invalid, they will be prompted to enter the information again. They can also choose to cancel the operation at any time. If all entries are valid, then the employee will be added to the system and the user will be taken to the updated employee list.

If a user is currently viewing the employee list, they can select an individual employee (using the PayrollSystem’s selectEmployee() function). Once an employee is selected, the PayrollSystem sets that employee as the current active employee (see Application Class Model) for a session and displays that employee’s information (i.e. the attributes of the Employee class in our Domain Class Model). From here, a qualified user can view and make changes to an employee’s personal information. The user can also choose to generate a report on the employee’s general information or a report on an employee’s paystub for a given pay period.

If an employee is currently selected (i.e. there is an active employee for the current PayrollSession), then a qualified user can select a paystub (selectPayStub() function), and be provided a paystub matching the payStubID they provide. From here the user can view the paystub information. The PayStub class includes many functions used in displaying the information required when an individual paystub is viewed. These functions allow a user to view the bonus pay and deductions for a given paystub, along with viewing the totals of each individual type of deduction or bonus pay (e.g. total charity, total taxes paid, etc.). Deduction and Bonus Pay amounts must not be less than zero (see OCL constraints on Domain Class Model).

An employee’s information can be edited by a qualified user by selecting the employee from the employee list and then selecting the option “Edit employee”. As seen in the Application State Model, this will bring the user to an editing window that presents the user with various changes that they can make to the employee’s information. From this window, the user can select the change that they wish to make to the employee’s information. If the changes are valid, the PayrollSystem will ask the user for a final confirmation of the changes. If the user confirms, then the employee’s information will be updated and the user will be returned to the employee list.

While an employee is selected, the user can also choose to remove the employee. If this option is selected, the user will be prompted to provide confirmation. If the user confirms, then the employee will stay within the system but their employmentType attribute will be set to “inactive”. This allows the system to maintain a record of past employees for fiscal or tax purposes. Upon removal, the user will be returned to the updated employee list.

A user can choose to generate a report from within the Employee window (i.e. when an employee is selected). From here, they will be prompted to provide the type of report they would like to generate. If the user selects “Employee report”, then a report containing all of the employee’s information (name, address, employment type, etc.) will be generated and the user will then be returned to the employee list. The user can also select the “PayStub report” option. If the user chooses this option, they will be prompted to provide the paystubID corresponding to the paystub that they would like a report of. The paystubID is a numeric representation of the first day of a paystub’s pay period (i.e. a pay period starting on September 30, 2019 would have a paystubID of “20190930” ). If a valid paystubID is provided, the PayrollSystem will generate a report containing the employee’s name and all the employee’s payment information (base pay, deductions, bonus pay) for the given pay period. The user will then be returned to the employee list.

Every time the PayrollSystem is powered on (i.e. enters the “Normal” state), a new PayrollSystemSession is established, recording the start time of the session. Each session is managed by a SessionController that is used to manage the control flow of the session. As seen in the Application Class Model, a session can have 0 or 1 active employees and 0 or 1 active Paystubs at any given time. These correspond to the currently selected Employee or Paystub. Employee functions will only work if the session has an active employee selected.

Within our domain class model, the class Deduction is associated with both Employee and PayStub, while the class BonusPay is associated only with PayStub. This is because deductions should stay unchanged from pay period to pay period, unless a change is manually made by a user. However, BonusPay will vary with each paystub. The association multiplicities reflect that each paystub can have many different deductions and many different forms of bonus pay. EmploymentType, ReportType, Distribution, and PayPeriod are all represented as enumerations, as each object can take on only one of their associated values (e.g. a single employee can have only one employment type or distribution type). This model allows for all employee and payment data to be easily accessible and modifiable through our UserInterface.