Employee Evaluation System

Team 6: Lance Elliot, Doug Holloway,

Phyllistine McCrary, Shyam Patel

Spring 2019

**Planning & Scheduling**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assignee Name | Task | Duration (hours) | Dependency | Evaluation |
| Lance Elliot | Backend, Testing & Test Specs, Class Diagrams | 29.5 | Database & Web Page Implementation, System Requirements | 100 |
| Doug Holloway  (coordinator) | Database, Test Specs, Github and File Management | 19.5 | System Requirements, Backend Implementation | 100 |
| Phyllistine McCrary | Frontend, Architectural Modeling, Report & Diagrams, Testing, System Revisions | 32.5 | None | 100 |
| Shyam Patel | Behavioral Modeling, Frontend, Problem Statement Specification | 24.5 | System Requirements, Architectural & Class Models | 100 |

**Problem Statement**

Our product gives employers the ability to evaluate employees based on their performance in the work environment. This evaluation system assesses employees based on criteria such as their dependability, productivity, work ethic, attitude and cooperation. We also give the option to write a brief description of the employees in terms of areas like their accomplishments, suggestions, changes or improvements that can be made. This system will feature an assortment of analytic reporting options for individuals and groups within each organization using our system based on customizable evaluation forms, and it will provide employees feedback from their employers. Evaluations can help employees increase their commitment to the company and productivity. It shows that the firm cares for their workers, and their effort is being recognized. Our system can help develop employees’ careers by allowing managers to see which areas they are lacking and need improvement; they will then be able to create plans for those specific employees to mature their career and meet goals that benefit the company. Fulfilling the explicit and implicit needs of our clients will require a variety of resources and techniques. Our evaluation system will feature a database to store and query entries and various webpages to interface and display each service. Even though components of our system will be simple the interactions between each component and user will be complex in implementation.

**Changes**

Our problem statement has mostly remained consistent throughout the project development. However, we have simplified some implementations, and plans to implement impromptu evaluation and feedback that links to/recommends workshops for employees have been cut entirely.

**Requirements**

**User Requirements**

1. Each user (employees and employers) must be able to access the system.
2. Employers must be able to create and customize evaluations forms for employees to complete.
3. Employers must be able to view completed evaluations and provide feedback to employees accordingly.
4. Employees must be able to complete evaluations.
5. Employees must be able to view feedback from their employer.

**System Requirements**

* 1. Each user will be able to signup for access to the system.
  2. All users will be able to login to access particular features of the system and view artifacts (feedback, evaluations & reports) of interest depending on their role in the organization.

2.1. An evaluation form creator will be made to save the metrics and types of questions an employer would like employees to respond to.

2.2. The generated evaluation forms will be saved in the database.

2.3. Employers will be shown the list of evaluations they’ve created and be able to load an evaluation form into the form creator for modification.

3.1. Employers will be able view a list of completed evaluations from employees.

3.2. Employers will be able to provide feedback to that is saved to the database for each employee.

4.1. Employees will be notified of an evaluation form they must complete.

4.2. Employees will be able to complete available evaluations and the responses will be stored in the database.

5.1. Employees will be shown a list of feedback from the employer.

**Requirements**

Feature: Registration

Introduction: User registration is needed to establish a user’s access to the system. It will prompt the user for personal information to be stored in the database. Without this function a user cannot use the system.

Inputs: First Name, Last Name, Department, Position, Email, and Password

Outputs: Confirmation of registration or error message indicating invalid field values

Requirements Description: A database to store each field’s data. Some form of encryption is needed so each user’s information is secure.

Feature: User Login

Introduction: User login is needed so each user can access the system. It will grant access or display a permission error depending on the inputs. Without this function users cannot access the system.

Inputs: Employee username and password.

Outputs: Redirection to system to confirm access or Error message to reject permission

Requirements Description: A database with each user’s credentials is needed. There should also be a different page the user is redirected to whenever their login is successful. If a user forgets their login info, a resetting mechanism or admin contact should be used. Lastly some form of encryption is needed so each user’s information is secure.

Feature: Create Evaluation Form

Introduction: The creation of the evaluation form is essential for the purpose of the project. Without a customizable form each employer cannot evaluate their employees in personally significant way.

Inputs: Specifications/Selections on evaluation metrics and fields of interest for the employer

Outputs: Copy of evaluation form and verification or error message

Requirements Description: A database to store each form and the information to be entered by each employee. An interface to allow each employer to easily create forms. Pages/interfaces to display the evaluation form upon completion or error message upon failure.

Feature: User Interface

Introduction: A user interface is essential to a user-friendly site. It will enable easier navigation & data input and display essential features such as feedback for employees and analytic reports for employers.

Inputs: mouse clicks & text entry from users

Outputs: Several screens display employer and employee feedback and analytics diagrams, screens for filling, creating, and moderating evaluation forms.

Requirements Description: Several webpages to display information from the database and take user input.

Feature: Database

Introduction: A database is essential to proper information management. The database will store all information related to users and evaluation forms, and without it, there would be no storage mechanism for the user to use.

Inputs: Queries from the UI

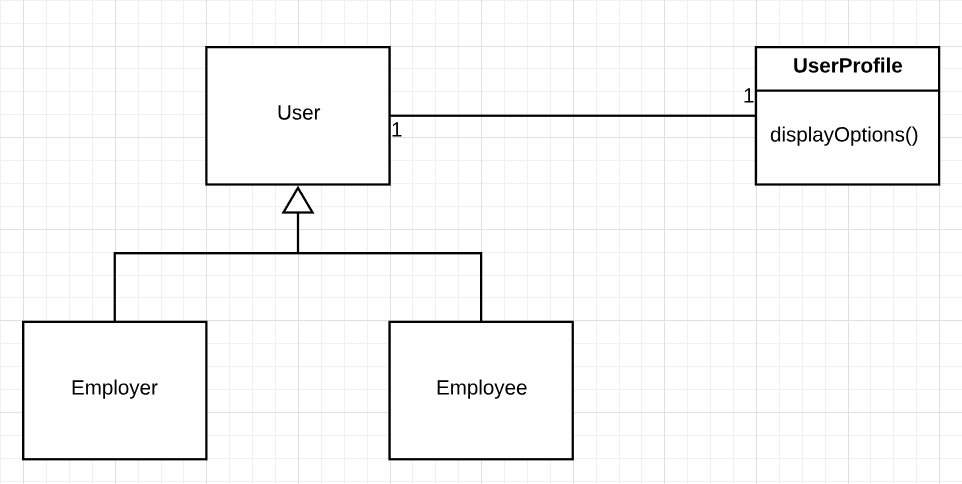
Outputs: Information to UI or employers

Requirements Description: Query functions related to each field in the database

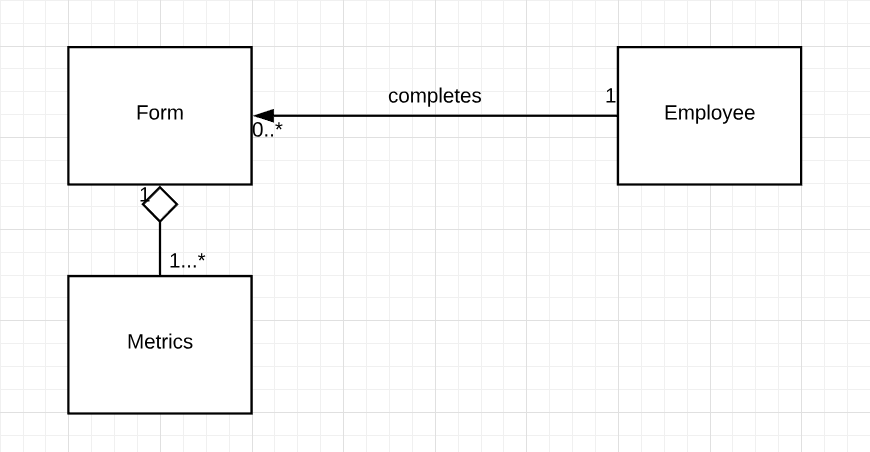
**System Modeling**

**Class Diagrams**

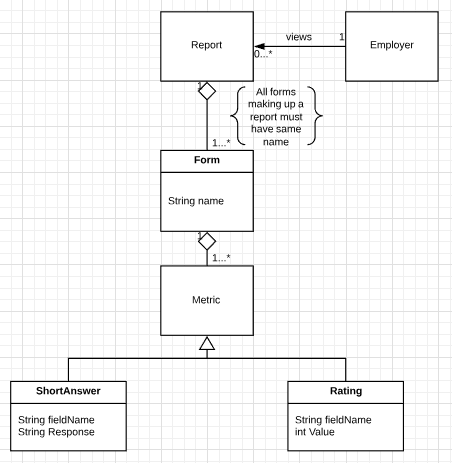
Login Display Case

****

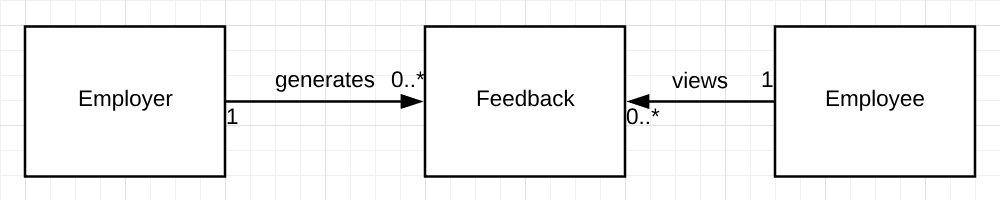
Submit Evaluation Form Case

****

View Report Case

****

Feedback

****

**Database Specification & Analysis**

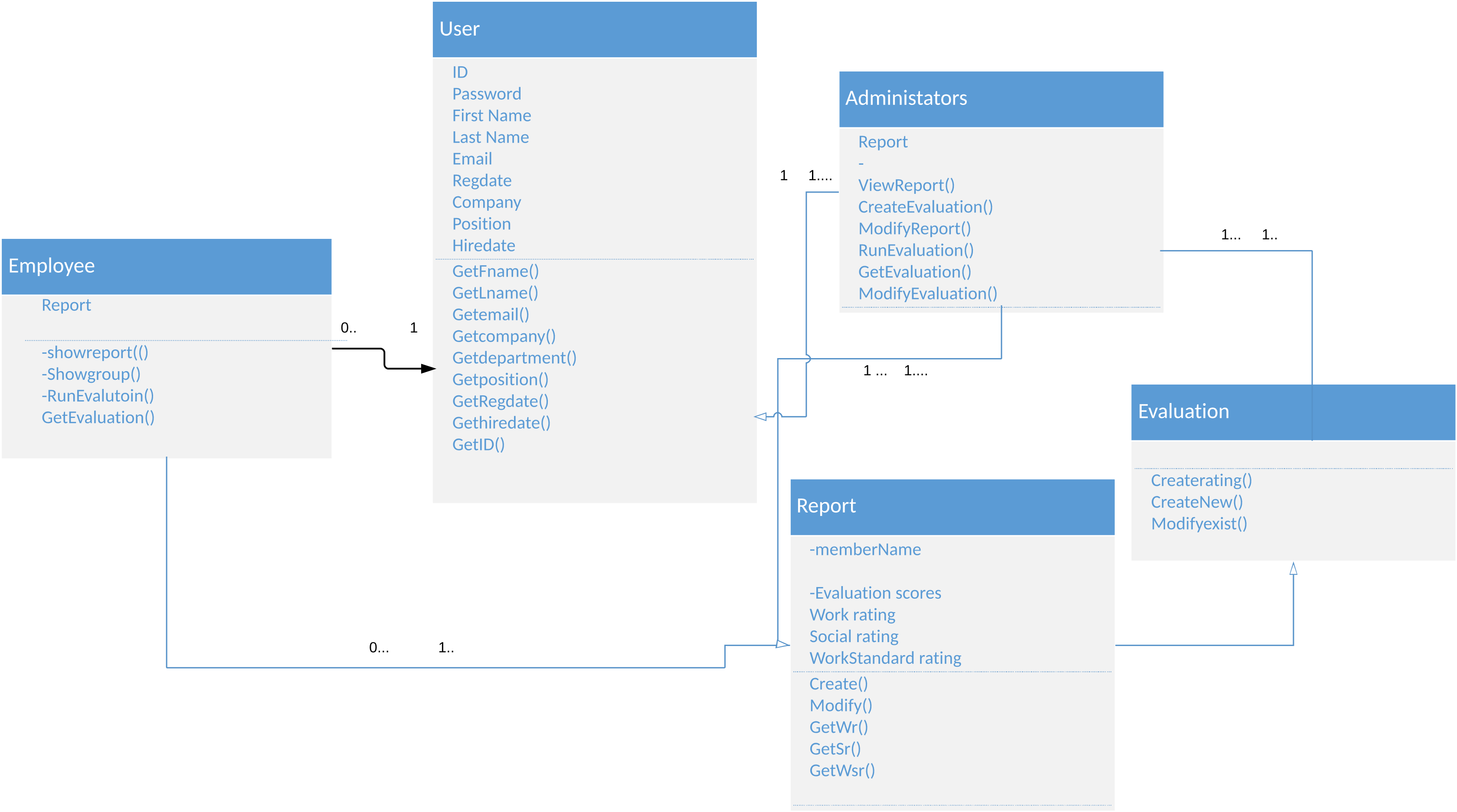
Hosted by Google. The address is: 104.196.56.178.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| users Table | Primary Key |  | company Table | foreign key |
|  |  |  |  |  |
| user\_id | int |  | department | varchar(60) |
| email | varchar(60) |  | company | varchar(60) |
| first\_name | varchar(20) |  | position | varchar(60) |
| last\_name | varchar(40) |  | hire\_date | datetime |
| password | char(40) |  |  |  |
| registration\_date | datetime |  | metrics Table | foreign key |
|  |  |  |  |  |
| notes Table | foreign key |  | 10-20 metrics | int |
|  |  |  | metric\_avg | int |
| 1-5 note/short ans | mediumtext |  | overall\_avg | int |

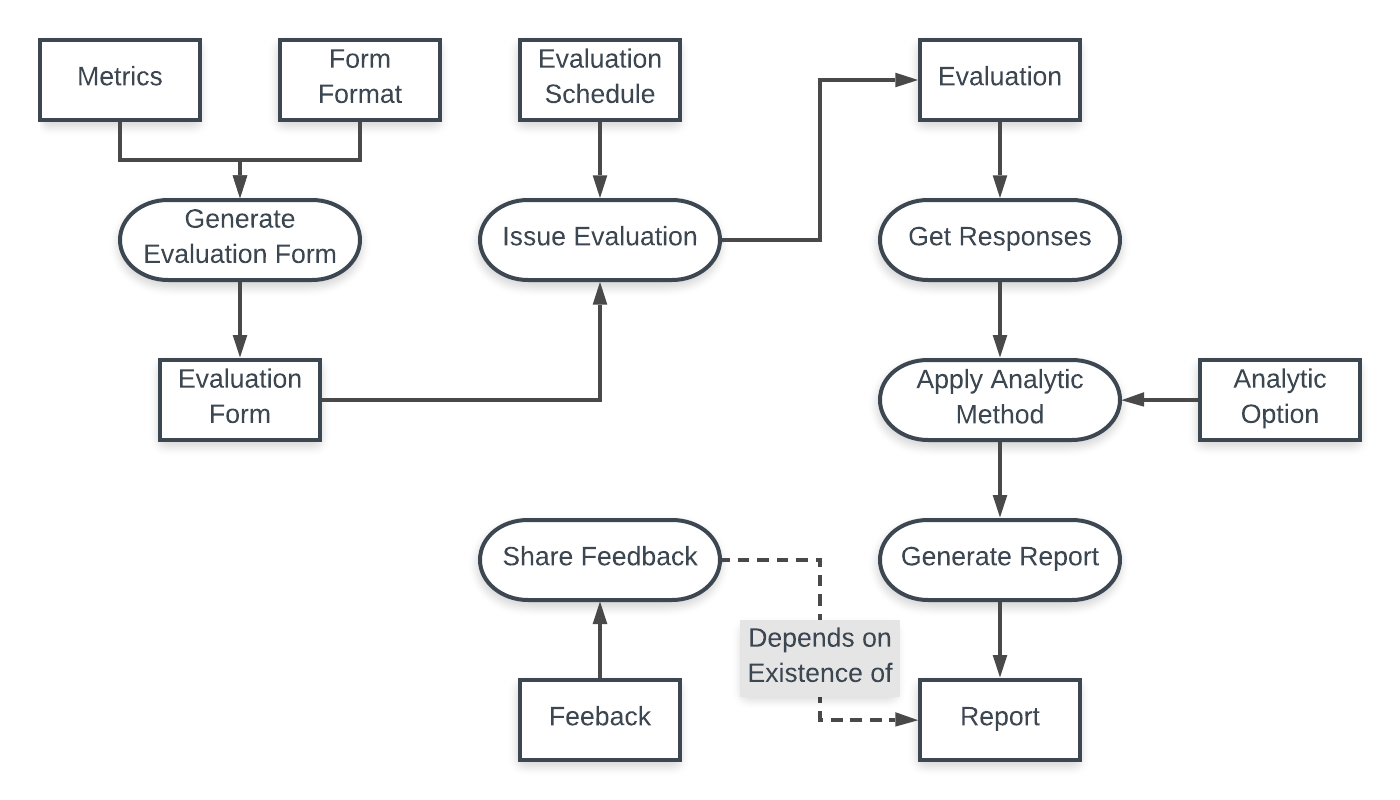
**Architecture Modeling**

Views:

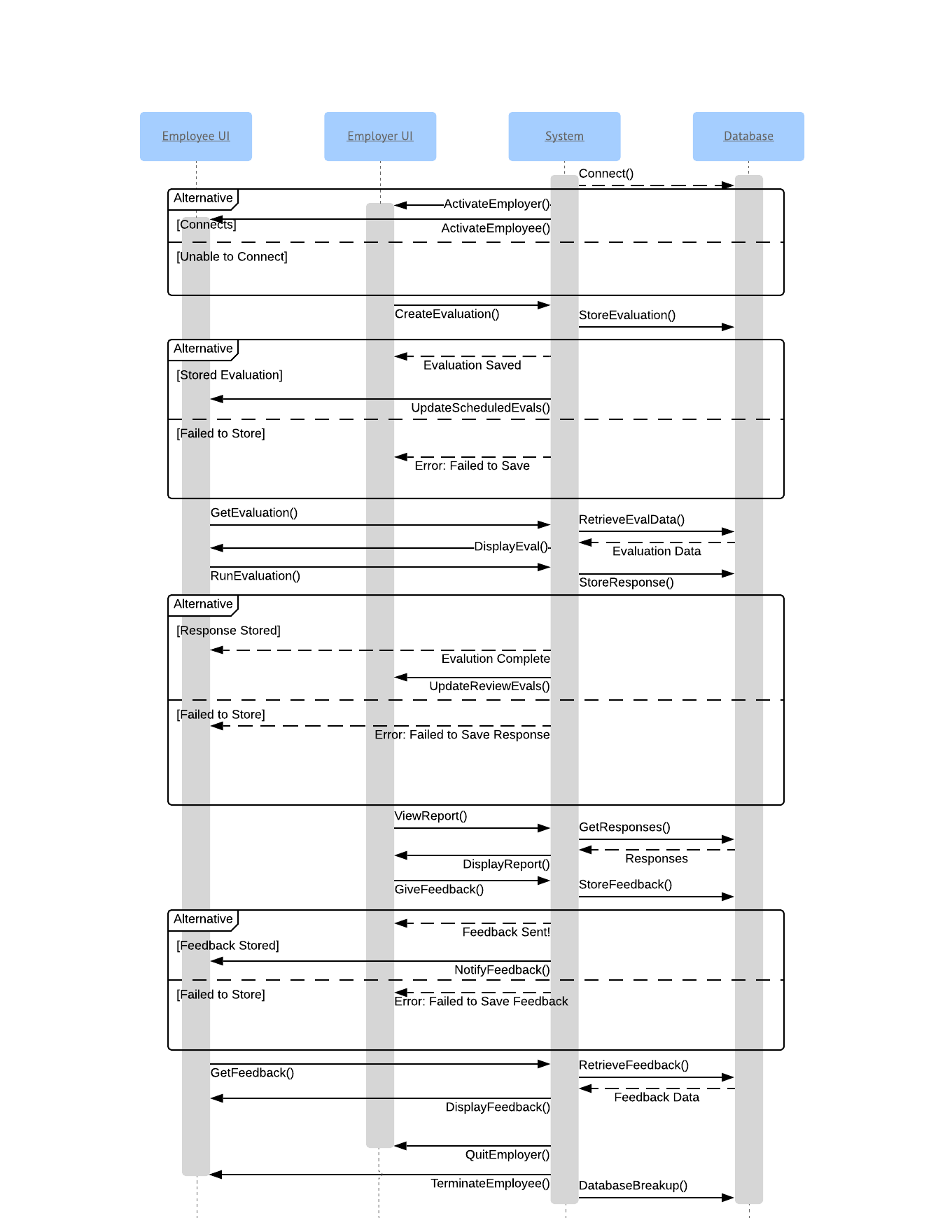
Class Diagram for Logical View:



Pipeline Diagram for Development View:



Sequence Diagram for Process View:



System Components for Physical View:

* Apache Tomcat Server – for web app function
* Java Servlet – for logic and process creation and handling
* HTML/CSS & JS Files – for UI development
* Google MySQL Database

Design Patterns:

Name: User Login

Problem: Loading an Employee vs Employer profile page for a user

Solution: Behavioral – Strategy 🡪 Check role in company and select related page

Name: Creating and Managing Forms, Reports/Submissions, & Feedback

Problem The lifespan and management of instances of artifacts.

Solution: Creational – Singleton 🡪 Create single instance of artifacts and provide services and users with access to request their display & modification.

Name: Updating Lists

Problem: Updating lists for Employers and Employees of available evaluations, submissions/reports, and feedback

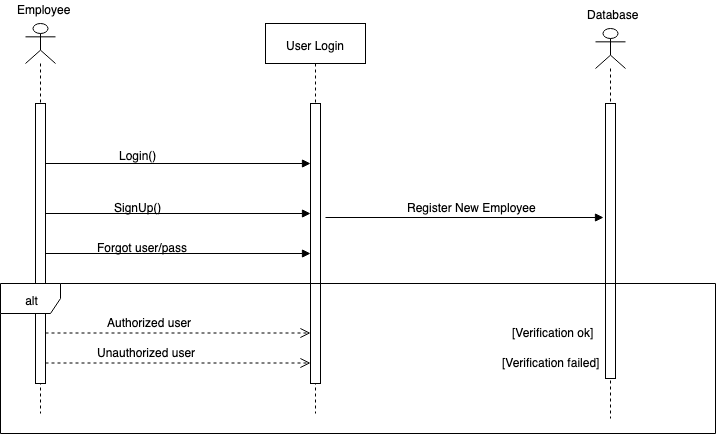
Solution: Behavioral – Observer 🡪 Notify and update lists when a change in the related database tables occur.

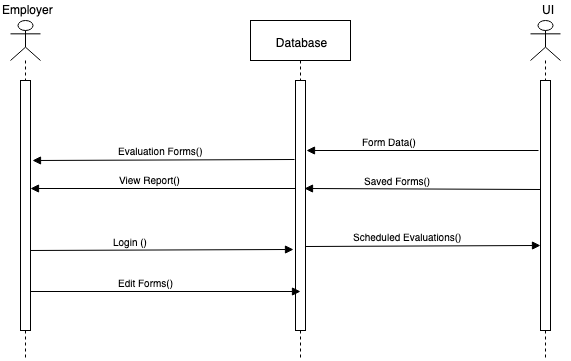
Name: Transform Form Data

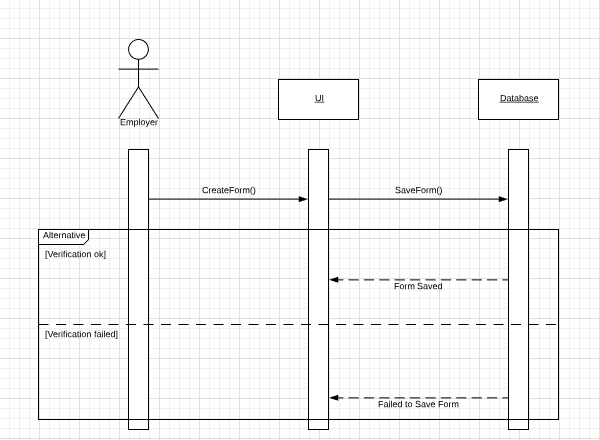
Problem: Display and transform form data into various artifacts like a form and a report.

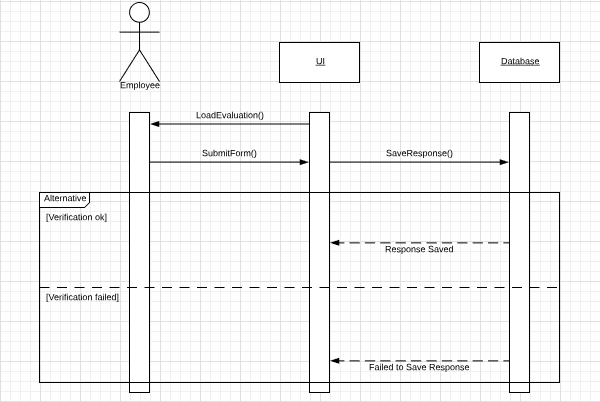
Solution: Structural – Adapter 🡪 Depending on the page that wishes to use the form data, adapt and transform the data accordingly.

**Behavioral Modeling**

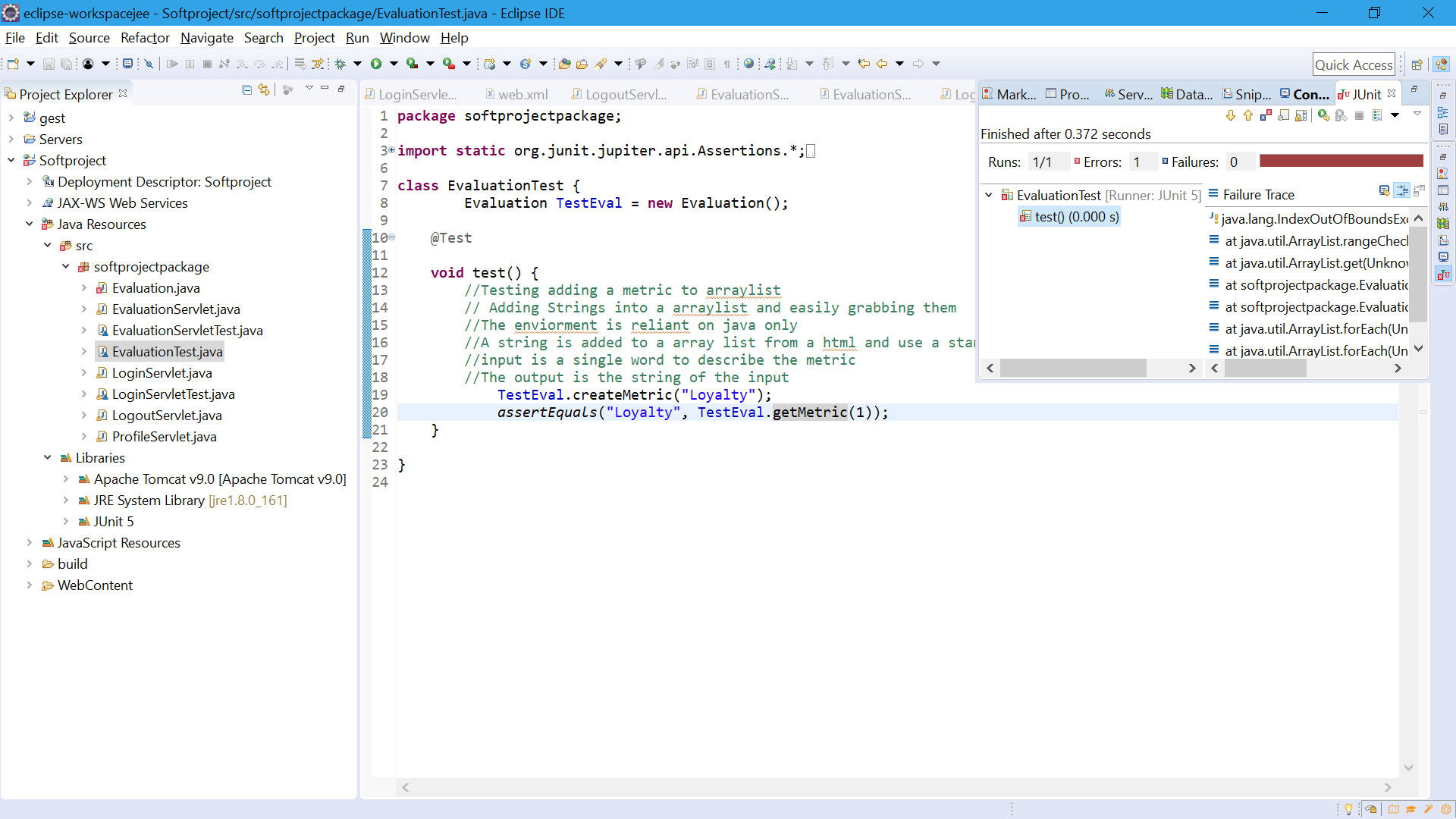


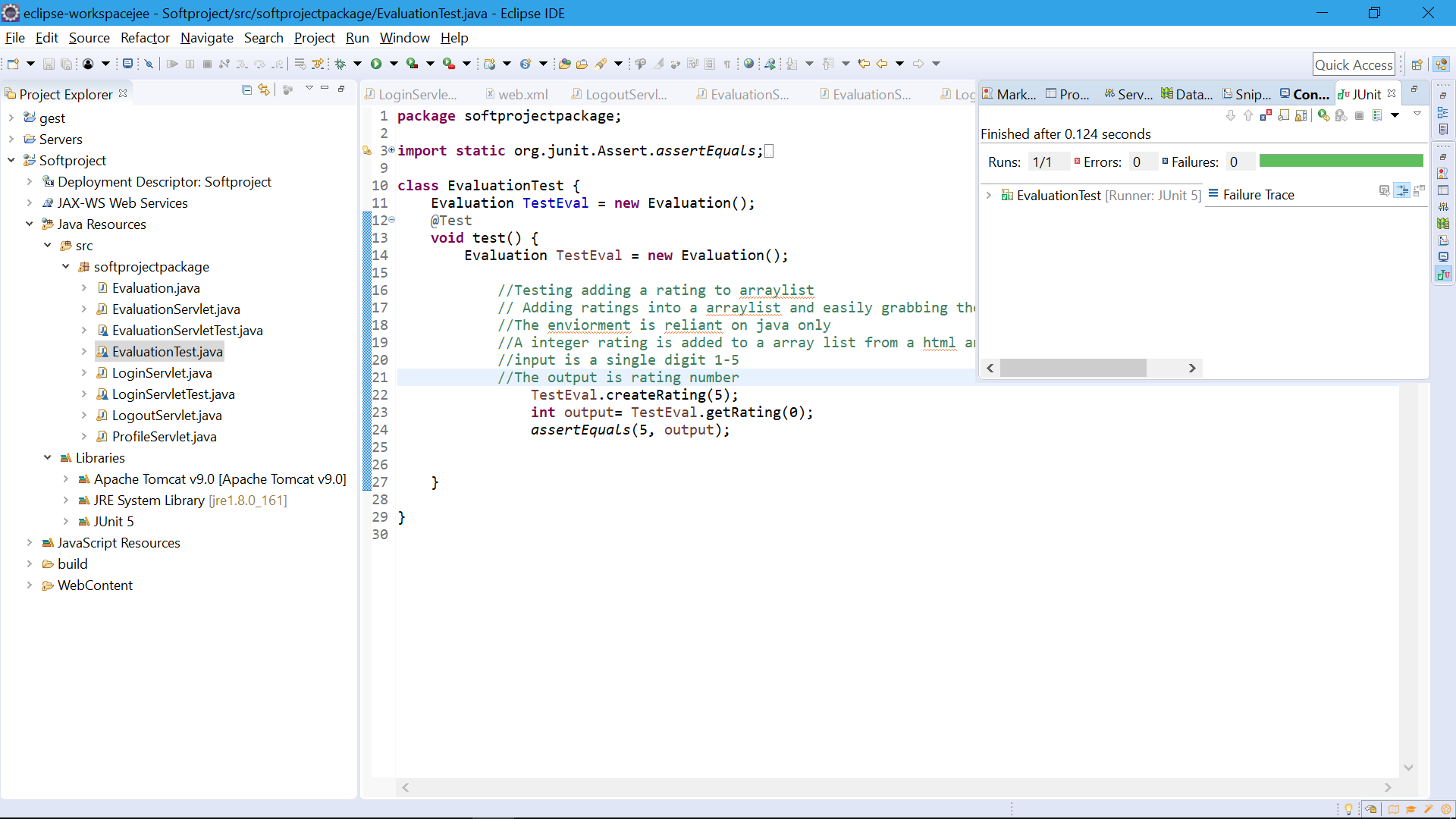


****

****

**Implementation**





**Testing**

Test case: Sign-up

**Hire date:**

Input expected: date(mm/dd/yyyy)

Test spec: date w/ month >12; day >31; year >2019; date in improper format; date w/ letter(s) or symbol(s)

Test cases: (14/25/2017), (02/43/2019), (04/13/2022), (2015/12/23), (1O/01/2017), (%5/10/2005)

**Incomplete form:**

Input expected: all required fields filled

Test spec: First name, last name, email, department, company, position, and hire date all filled

Test cases: Leave each one and then multiples blank in sequence

**Name entry:**

Input expected: Letters and symbols

Input partition: Letters >= 40 char

Test spec: Name with number, name longer than 40 char

Test cases: (St3v3n), (Hredfgtedgidneidgkdhsjtbendhgjdkevnfhskdghddkcniehvndekcn)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug** | **The test that uncovered the bug** | **Description of the bug** | **Action to fix the bug** |
| Test 1-2 had a bug. Assigning user as employer or employee. | Signing up different types of employees. | Department and position fields were logically confusing the assignment of the user type. | Correct logic. |

Test case: Creating a form

**Creating Metrics:**

Input expected: Int >0 to generate a number of metrics of selected type

Test spec: int= 0/int >0

Test cases: (0), (1), (10)

**Saving Metrics:**

Input expected: Strings of metric names and their types

Test cases: Generating different size forms of varying metrics type numbers

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug** | **The test that uncovered the bug** | **Description of the bug** | **Action to fix the bug** |
| Test 1 had a bug. Couldn’t grab the metric strings | A simple call to the get method after adding a metric to the array list | The get method only got a single index of the array. | Create a new get method using a for loop and string length |

Test case: Performance review

**Filling out review:**

Input expected: All required fields in forms filled out

Input partition: 10-20 metrics req, optional 1-4 short answer req

Test spec: leaving required field blank

Test cases: Trying different sized forms leaving required fields blank

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug** | **The test that uncovered the bug** | **Description of the bug** | **Action to fix the bug** |
| Test 1 had a bug.  Field names were being changed and response was shorter than expected. | Checking string to represent form field and value entered. | Instead of field name, field type was being grabbed and overwriting similar type responses. | Change parsing of field info to grab correct information. |

**Appendix**

Github Project

