Employee Evaluation System

Team 6: Lance Elliot, Doug Holloway,

Phyllistine McCrary, Shyam Patel

Spring 2019

**Planning & Scheduling**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assignee Name | Task | Duration (hours) | Dependency | Notes |
| Lance Elliot | Revised Class Diagrams, Testing | 4 | Database & Web Page Implementation | 100 |
| Doug Holloway | Backend Implementation,  Test Specs | 4 | None, Class Diagrams | 100 |
| Phyllistine McCrary  (coordinator) | Problem Statement & Requirements, Architectural Modeling, Report | 4 | None | 100 |
| Shyam Patel | Behavioral Modeling, Frontend Implementation | 4 | Architectural & Class Models, Database Implementation | 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 be unique amongst competitors as it will feature an assortment of analytic reporting options for individuals and groups within each organization using our system based on scheduled and year-round impromptu evaluations, customizable evaluation forms, and feedback that connects employees with workshops and events to improve their skills and further each company’s goals. 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 be 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 algorithms, services, and interactions between each component and user will be the most difficult and interesting aspect of this project.

**Requirements**

Feature: Company Registration

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

Inputs: First Name, Last Name, Company Name, 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 company’s information is secure.

Feature: Company Management

Introduction: This will establish and modify details about each company and its employees for database management and use. It will prompt the employer for information about their company’s position structure and each employee’s personal information. Without this function the employer cannot utilize all the analytic options of the system.

Inputs: All company positions & departments and each employee’s name, email, department, position, and hire date.

Outputs: Confirmation of data storage or modification or error message

Requirements Description: A database to store and relate all data input. Encryption. Initial Setup and Modification Web Pages.

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 each employee’s participation cannot be monitored and each employer could not securely review their employees’ evaluations.

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.

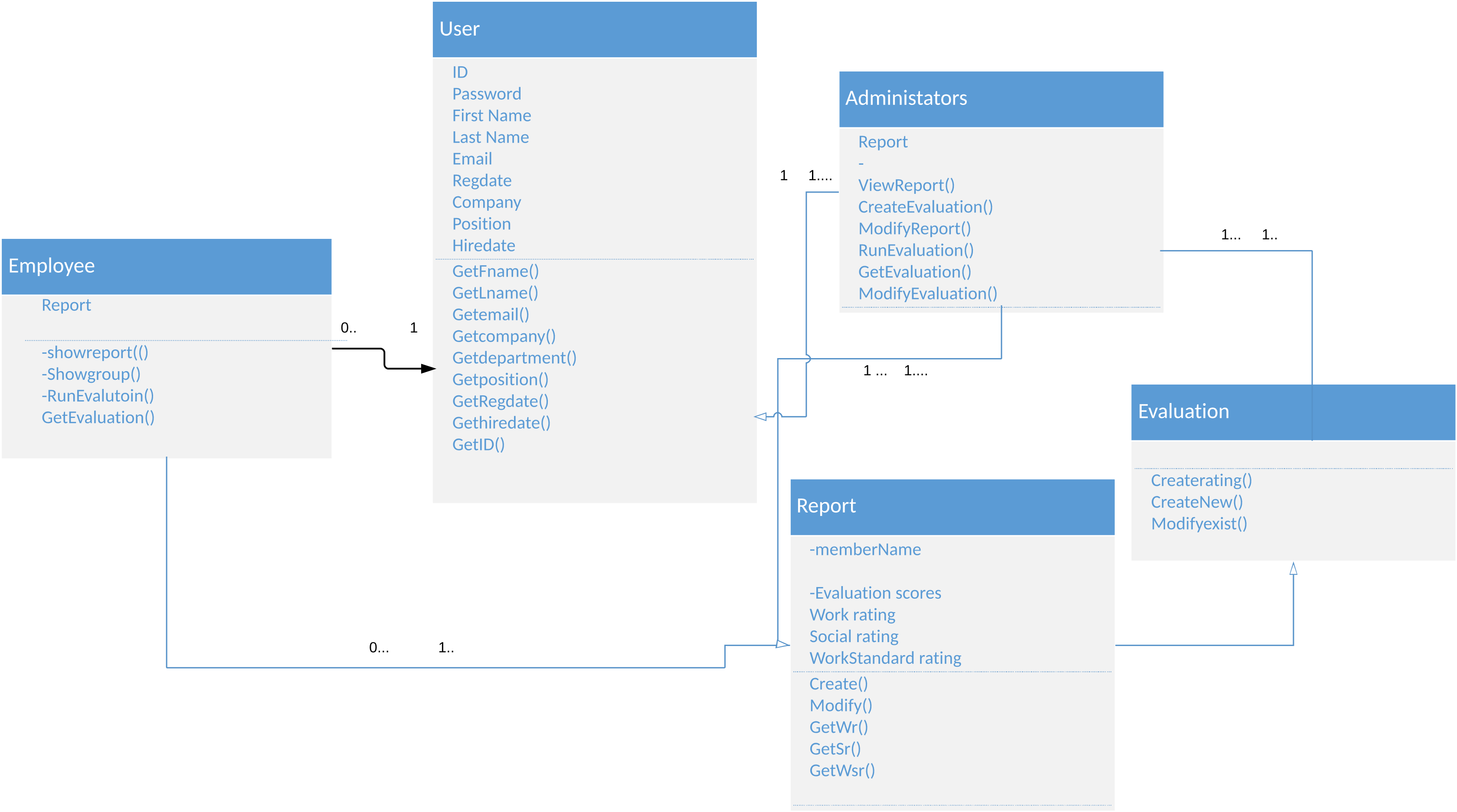
Inputs: Queries from the UI

Outputs: Information to UI or employers

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

**System Modeling**

Class Diagram



1. The objects we are using are Employee, Administrators, Report and Evaluation.
2. Employee and Administrators objects have a Report and an Evaluation association.
3. The multiplicity between the Administrators and Employee to the Evaluation and Report are 1 so far.
4. Evaluation object hold 3 Arraylist[]. The Report has an Evaluation object, integer rating, and Arraylist[] rating. Administrators has Evaluation and Report objects with and share private Strings first name, last name, Id, password, email, company, position, with Employees.
5. The operations for the User class that is the super of Administrators and Employee are GetFname(), GetLname(), Getemail(), Getcompany(), Getdepartment(), Getposition(), GetRegdate(), Gethiredate(), GetID(). The Administrators class contains operations ViewReport(), CreateEvaluation(), ModifyReport(), RunEvaluation(), GetEvaluation(), ModifyEvaluation().

Report class contains: Create(), Modify(), GetWr(), GetSr(), GetWsr(). And a method to find the weighted average of the evaluation ratings.

The Evaluation class contains: createMetric(), deleteMetric(), clearMetric(), setMetric(), createRating(), deleteRating(), clearRating(), setRating(), createweight(), deleteWeight(), clearWeight(), setWeight()

The Administrators class contains: ViewReport(), CreateEvaluation(), ModifyReport(), RunEvaluation(), GetEvaluation(), ModifyEvaluation()

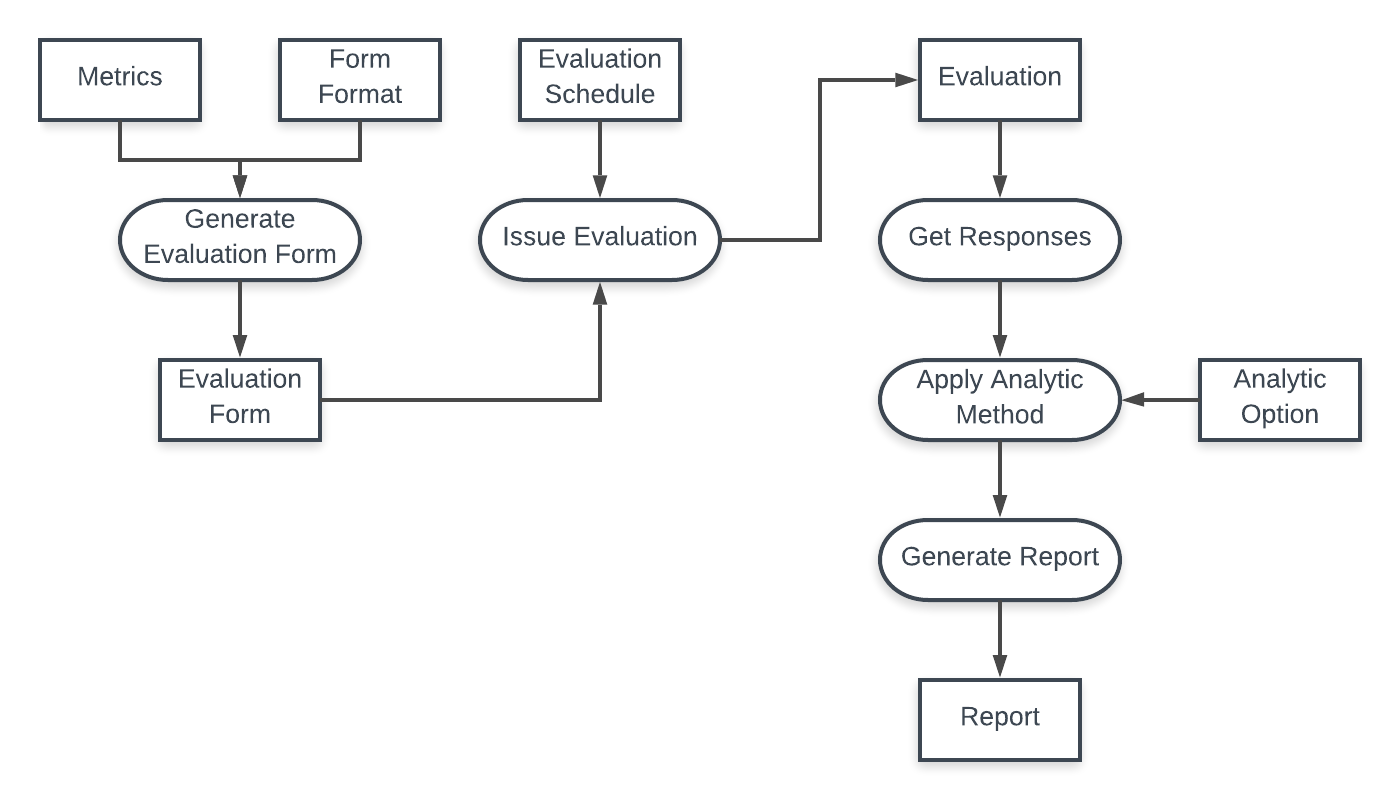
Employee class contains: showreport(), Showgroup(), RunEvalutoin(), GetEvaluation()

Database Specification & Analysis

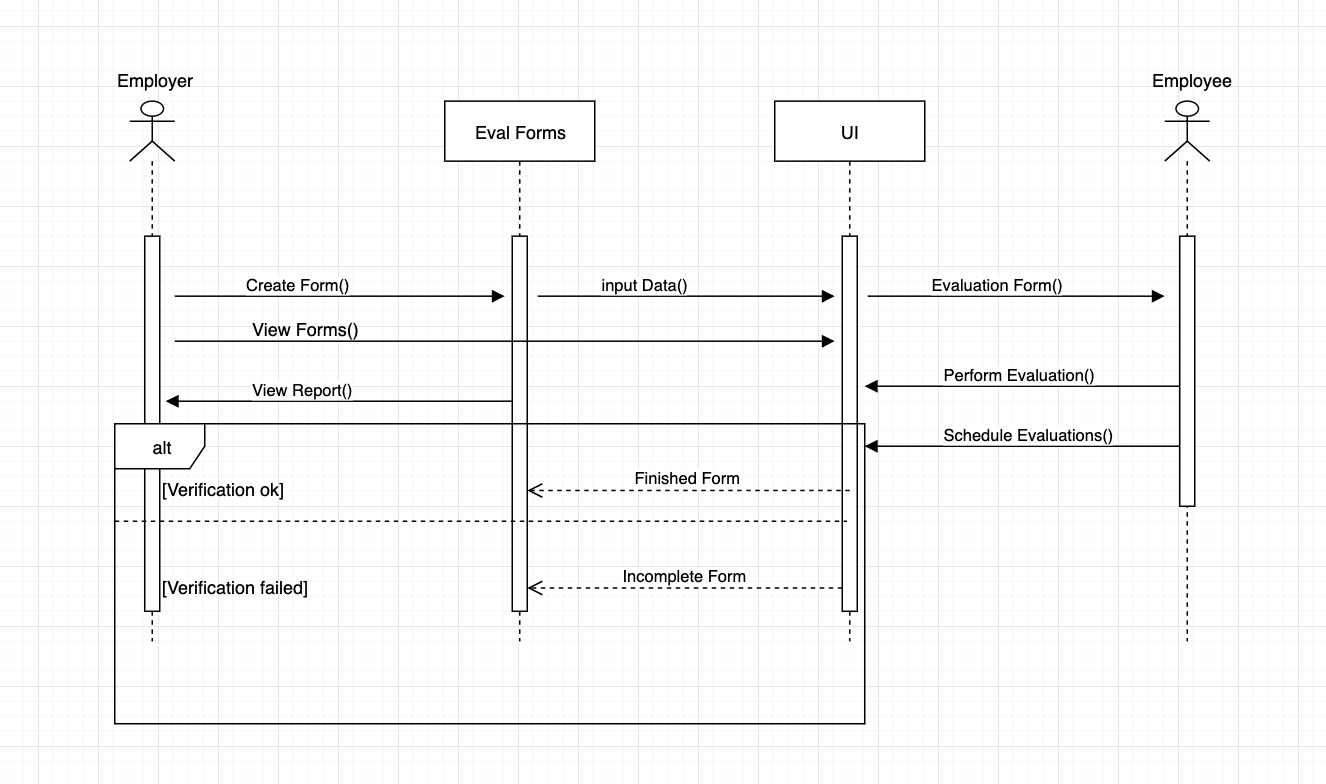
Hosted by Amazon RDS. The database is a db.t2.micro instance with 2 Gb storage. The address is: arn:aws:rds:us-east-1:400073246240:db:group6-spring-19. The metrics and notes section will ultimately be decided by the client so we will make space but cannot say exactly what they will be as they will change in a case to case basis. Our current plan for our tables and which will be primary and foreign keys are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

Pipeline Architecture Model



Sequence Diagram of Evaluation Creation and UI



**Implementation**

**Testing**

Test case: Creating a form

**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)

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

**Appendix**

Github Project

