CSc3350: Team Project FA2025. IOR: William Gregory Johnson, Ph.D.

User Story:

Your software development team is hired as consultants and tasked with delivering a new, working employee management system for company ‘Z’. The project requires a software design document as a set of designs for this software system involving its data schemas, Ux, general programming components and security. They currently have one HR admin person maintaining their employee data using dBeaver and MySQL scripts and not any security except for the HR admin password. They have about 55 full-time employees (no hourly or part-time). However, they plan to triple this amount within 18 months.

These are several software system features/functionality to be built:

1) A user logon/authorization with this functionality:
    a. Secure HR admin to ensure CRUD functionality on the entire database
    b. A general security for their employees to see their data (SELECT) only, no changing
2) Search for employee data using name, DOB, SSN, empid to show their information for editing: HR Admin only.
3) Search for employee data using name, DOB, SSN, empid to show their information for VIEWING: general employee only.
4) Update an employee’s data once selected from a search result list. (HR Admin only)
5) Update employee’s salary for an increase of a particular percentage only for a salary amount range. E.G., 3.2% for salary greater than, equal to 58K but less than 105K. (HR Admin only)
6) Several reports.
    a. Full-time employee can see their pay statement history sorted by most recent pay date. (General employee only)
    b. Total pay for month by job title. (HR Admin only)
    c. Total pay for month by Division.(HR Admin only)
    d. Employees hired within a given date range
7) Must use MySQL database.

Deliverable Items for Team Project (***Individual*** and ***GROUP***)

1) General, simple UML diagrams and dB schema diagram.
2) Database schema diagram from dBeaver showing the original and these extra items.
    a) Additional table:
        i) Address (street, city, state, zip) for employee (has empid as primary key) and additional demographic items: (gender, identified race, DOB, mobile/phone.)
        ii) This shall be split into other tables for city and state. It should use an ID for city (20 or less) and ID for state (all 50) in ‘Address’ table. (Implies cityand state tables)
        iii) Empid in Address defined as foreign key in employees table
    b) Create primary/foreign key connections:
        i) employees.empid (primary) to employee_division.empid (foreign)
        ii) employees.empid (primary) to payroll.empid (foreign)
        iii) employees.empid (primary) to address.empid (foreign)
        iv) employees.empid (primary) to employee_job_titles.empid (foreign)
        v) employee_division.div_ID (foreign) to division.ID (primary)

vi) employee_job_titles.job_title_id (foreign) to job_titles.job_title_id (primary) (https://www.w3schools.com/mysql/mysql_create_index.asp)

3) A fully working UX in either console or GUI (Swing or JavaFX) to satisfy the employeeData MySQL database. All code shall be Java. CORRECTION: No use of micro-frameworks, you CANNOT use HTML, java script, NO Node.js, React, ext… No extra credit will be applied.

   ~~*. Note: using HTML for the UX will require microframeworks, services, database (Node.js) to connect to MySQL. However, extra credit will be applied as percentage of your total individual points, TBD.*~~

4) Java class UML diagram showing code organization:
   a) Add new java classes to gather (get inputs) and store the employee information in a dynamic data structure like a list or collection.
   b) Show any abstract classes, interfaces, aggregated, and inherited classes where used.

5) Programming tasks (10) taken from the user story/scenario:
   a) These can be a one-to-one relation of a feature in the software system
   b) These can be a functional component (JDBC library, JavaFx library, interface class, abstract class)

6) Test cases that ensure the features/functions are going to work and update data correctly:
   a) Update employee data (make it general enough using your new classes for all employee data)
   b) Search for employee (admin user)
   c) Update salary for all employees less than a particular amount

   NOTE: Start with describing 'a,b,c' each as a detailed programming task, then create pass/fail test cases from each

7) Sequence diagrams (UML) for these:
   a) Increase salary by % for all employee salaries in a specified range
   b) Add new employee to employeeData database
      (In sequence diagrams, you will show components like UX, java classes, methods in main program, JDBC, microframeworks, etc.)

8) A software design document in PDF that has the above items, each in a section(s), sub-section(s), and a table of contents. A template is provided, and more details will be in the dropbox for iCollege assignment. Work-in-class-time will be used to develop UML items, dB schema, and java classes with your team members, then upload these into iCollege.

Operations Information:

- You may use any tool to draw/create UML diagrams. If you hand-draw them, be very neat to be readable and understandable. These will be graded on correct form and specifics of required elements.
- Your team recorded video shall be in a format compatible for Mac and windows: (MPEG-4, MPEG, WMV, AVI, MKV, 3GP, or MOV)

- Your code can only be in java using either a character interface (terminal window) or GUI (JavaFX or Swing)

Timeline for Deliverable Items:

1. **<u>11/02/2025: 11:59PM</u>** ***Individual***: 150pts
   1.1. One UML use case depicting the overall components and their connections (internal and external among these and the actors)
   1.2. One UML sequence diagram showing a regular employee 'input their EmpID' to search the database and return personal data.
   1.3. Database schema diagram (Refer to #1 in Deliverables) generated from a DBMS tool (dBeaver or SQL Workbench). Ensures each person can run the database and understands normalization, primary-foreign key, and schema design.

2. **<u>11/11/2025: 11:59PM (NO LATE PENALTY ALLOWED)</u>** ***GROUP***: 160pts
   2.1. Programming tasks, 5 from user story (Design document will require 10: can use these 5 both times)
   2.2. Test cases from #6 in Deliverables 'a, b, c'

3. **<u>11/16/2025: 11:59PM</u>** ***Individual***: 110pts
   3.1. Sequence diagrams from '#7 Deliverables 'a, b'

4. **<u>12/08/2025: 11:59PM (NO LATE PENALTY ALLOWED)</u>** ***GROUP***: 150pts
   4.1. Company 'Z' needs a final software design document described in #7 Deliverables above (SDD template will be in iCollege)

5. **<u>12/11/2025: 11:59PM (NO LATE PENALTY ALLOWED)</u>** ***GROUP***: 250pts
   5.1. One, continuous (no spliced parts) 10 to 15-minute video demonstration of the working software system (search, update, insert new employee, reports) where one or many of your team is walking through basic functionality and speaking while showing yourself on screen. Use very limited power point in the demo (no more than 5 slides) as I want to see you present a visual demonstration of the final software system.
   5.2. Video can be given by one or many team members.
   5.3. There will not be any late submissions. There is one final time and after that it is too late. Only ONE person need upload the video (single file). If multiple members upload, the latest 'time/date' will be graded as the final turn-in version.
   5.4. This is a recorded video in a format compatible for Mac and windows: (MPEG-4, MPEG, WMV, AVI, MKV, WebM, 3GP, or MOV) 1
   5.5. PLEASE show your code running in the video and your face talking and describing through the demonstration.
   5.6. UPLOAD your java files (points deducted if one monolithic program file)