

User Story:

Your software development team is 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, and programming models. They currently have one HR admin person maintaining their employee data using dBeaver and MySQL scripts. They have about 40 full-time employees (no hourly, part-time). However, they plan to triple this amount within 18 months.

The software system requires features/functionality in to ensure things like:

- 1) A user login/authorization with this functionality:
 - a. For admin to enable CRUD functionality on the entire database
 - b. A general employee able to see their data (SELECT) only, no changing
- 2) Search for an employee using name, DOB, SSN, empid to show their information for editing (Admin only)
- 3) Search by employee to view their data. (Will require MySQL security...)
- 4) Update an employee's data once selected from a search result list. (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. (Admin only)
- 6) Several reports.
 - a. Full-time employee with pay statement history sorted by employee id, then pay date. (Admin see all and general employee see only their data)
 - b. Total pay for month by job title. (Admin only)
 - c. Total pay for month by Division.(Admin only)
- 7) Must use existing database (MySQL)

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

- 1) 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 and ID for state in 'Address' table. (Implies city and state tables)
 - 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 employee_job_titles.empid (foreign)
 - iv) employee_division.div_ID (primary) to division.ID (foreign)
 - v) employee_job_titles.job_title_id (primary) to job_titles.job_title_id (foreign)
 - vi) employees.empid (primary) to address.empid (foreign)
- (https://www.w3schools.com/mysql/mysql_create_index.asp)

- 2) A fully working UX in either console or GUI (Swing or JavaFX) to satisfy the employeeData MySQL database. 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.
- 3) 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 (list or collection)
 - b) Abstract classes, interfaces, aggregated, and inherited classes where used.
- 4) 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)
- 5) 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
- 6) Sequence diagrams (UML) for these:
 - a) Increase salary by % if current base is in a range amount
 - 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.)
- 7) A software design document in PDF that has the above items, each in a section(s), sub-section(s), and a table of contents (Easy to do in MS Word and export to PDF)
 - a) 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.
 - b) Your final report may have more and/or more elaborate diagrams. During work-in-class time, I need some beginning UML diagrams to see you are on the right track.

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)

Timeline for Deliverable Items:

1. **March 26, 2025: 11:59PM *Individual***: 150pts
 - 1.1. One UML use case describing the overall components and their connections
 - 1.2. One UML sequence diagram showing 'input an EmpID' to search the database
 - 1.3. Database schema diagram (Refer to #1 in Deliverables) generated from dBeaver.
(Ensures each person can run the database and understands normalization, primary-foreign key, schema design.)
2. **April 02, 2025: 11:59PM *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 #5 in Deliverables 'a, b, c'
3. **April 07, 2025: 11:59PM *Individual***: 110pts
 - 3.1. One UML sequence diagram from 'a' and one UML from 'b' in #6 Deliverables above
4. **April 25, 2025: 11:59PM (NO LATE PENALTY ALLOWED) *GROUP***: 150pts
 - 4.1. Company 'Z' needs a final software design document described in #7 Deliverables above (SDD template will be in iCollege)
5. **May 4, 2025: 11:59PM (NO LATE PENALTY ALLOWED) *GROUP***: 200pts
 - 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.