

Stories:

1. As a user, I need to perform CRUD (Create, Read, Update, Delete) on Employee Demographic Information, including Name, Address, Phone, and Emails.
 - 1.1. Create a new Employee
 - 1.1.1. Make a database to store Employee information, Demographic Info includes:
 - 1.1.1.1. Name
 - 1.1.1.2. Address
 - 1.1.1.3. Phone Numbers
 - 1.1.1.4. Emails
 - 1.1.1.5. Member ID (could make it easier to search for employees)
 - 1.1.1.6. Current Team (to know what the employee is currently working on or who they're working with)
 - 1.1.2. Need a user interface (GUI) to collect the data
 - 1.1.3. Need a save function
 - 1.1.3.1. When saved database is updated with a new Employee record
 - 1.2. Read
 - 1.2.1. Need to query the database to get Demographic information:
 - 1.2.1.1. Name
 - 1.2.1.2. Address
 - 1.2.1.3. Phone Numbers
 - 1.2.1.4. Emails
 - 1.2.1.5. Member ID
 - 1.2.1.6. Current Team
 - 1.2.2. Need a GUI to select certain employees
 - 1.2.3. Need a search feature
 - 1.2.4. Maybe organize based on teams
 - 1.3. Update
 - 1.3.1. Need to edit a employees information (should be able to edit any and all info on employee not just demographic)
 - 1.3.2. Maybe make edit a button that has to be pressed, like a mode change
 - 1.4. Delete
 - 1.4.1. Make a delete button
 - 1.4.1.1. Add a confirmation if delete is pressed to make sure that no one is accidentally deleted
 - 1.4.2. When confirmed delete specified employee from database
 2. As a user, I need to perform CRUD on Employee Job History, including current and past jobs.
 - 2.1. Create a Job History section under chosen Employee
 - 2.1.1. Info for job history will be added to a section in the database, info needed:
 - 2.1.1.1. Company Name
 - 2.1.1.2. Supervisor
 - 2.1.1.3. Teammate Names (optional)
 - 2.1.1.4. Length on the Job
 - 2.1.1.5. Job Title
 - 2.1.1.6. Role on Team
 - 2.1.1.7. Critical Skills and their Level (Beginner, Intermediate, Advanced)
 - 2.1.1.8. Soft Skills
 - 2.1.1.9. Talents of Employee
 - 2.1.2. Need a user interface GUI to collect data and add job history section
 - 2.1.3. Need a save feature (and notify user when save is complete)

2.2. Read Job History

2.2.1. Need to query the database for job history specifically info needed:

- 2.2.1.1. Company Name
- 2.2.1.2. Supervisor
- 2.2.1.3. Teammate Names (optional)
- 2.2.1.4. Length on the Job
- 2.2.1.5. Job Title
- 2.2.1.6. Role on Team
- 2.2.1.7. Critical Skills and their Level (Beginner, Intermediate, Advanced)
- 2.2.1.8. Soft Skills
- 2.2.1.9. Talents of Employee

2.2.2. Need a GUI to see the data in an organized manner

2.2.3. Need a sort feature to sort job's alphabetically or most recent

2.3. Update Job History

2.3.1. Need an edit button to allow user to edit employee job history

2.3.2. Need a save button to save data to database

2.4. Delete a specific job from history

2.4.1. Need a delete button to remove jobs from job history

2.4.1.1. Make sure to have a confirmation, like a "Are you sure you want to delete this?"

2.4.2. When confirmed delete specific job from job history in database (should update GUI)

3. As a user, I need to perform CRUD on Employee Evaluation Tracking.

3.1. Create

3.1.1. Need to be able to add notes to employees with required info:

- 3.1.1.1. Evaluating Supervisor
- 3.1.1.2. Date of Evaluation
- 3.1.1.3. Mental State of Employee
- 3.1.1.4. Supervisor notes on projected employee satisfaction, proficiency, or efficiency

3.1.2. Need a user interface GUI to allow users to create notes on employees with an "Add" button

3.1.3. Need a save feature (notify when save is complete)

3.1.4. Will add more to another section in database

3.2. Read

3.2.1. Users with Supervisor Class should be able to see evaluation notes

3.3. Update

3.3.1. Need to be able to edit evaluation notes, but should not be able to edit more than just the notes.

3.4. Delete

3.4.1. Need to be able to delete evaluations

3.4.1.1. Make sure there is a confirmation of deletion

4. As a user, I need to be able to learn a simple GUI .

5. As a sales representative, I need the GUI to be very vibrant and eye-catching to customers so I can sell it more effectively.

5.1. Add color to GUI.

6. As a manager, I need to be able to keep track of employee data.

6.1. "Manager" accounts need permission to view (but not edit) "Employee" (but not other) account data

6.1.1. GUI for employee info should be organized

7. As a user, I need to be able to easily open the app.

7.1. Ease of access

- 7.1.1. If on mobile, the app will be on the homescreen anyway.
- 7.1.2. If on desktop, the app should install with a desktop shortcut by default (rather than buried in files)

7.2. Optimization

- 7.2.1. The app should be optimized to improve load times on launch.

8. As a user, I need a secure login with username and password.

8.1. Secure login

- 8.1.1. The app will need a login screen to accept credentials.
 - 8.1.1.1. The app may hide entered passwords to prevent over-the-shoulder reading.
- 8.1.2. The app will need to query a database with account information with the entered credentials.
 - 8.1.2.1. The query will need to be encrypted for security.

8.2. Login info

- 8.2.1. Need somewhere to store large amounts of login information
 - 8.2.1.1. This will likely require a server, either one we own or rented.

9. As a manager, I want to be able to rate an employee's weekly performance.

9.1. "Employer" accounts need to be able to rate "Employee" accounts

- 9.1.1. Need a rate button that pops up on employee accounts for their employers, but no one else.
- 9.1.2. Need to store employer-employee relationship information
 - 9.1.2.1. Database for account relationships

9.2. Need to store rating info in a database

10. Create User

Classes and Object Design:

<u>Class</u>	<u>Parameters/Variables</u>	<u>Methods</u>
User	memberID/Username Password Company isSupervisor	getUsername() getPassword() getCompany() (bool) isSupervisor()
EmployeeDemographics	Name Address Phone Numbers Emails CurrentTeam	getName() getAddress() getPhoneNumbers() getEmails() getCurrentTeam()
EmployeeJobHistory	CompanyName Supervisor LengthOnJob Title Role CriticalSkills SoftSkills Talents	getCompanyName() getSupervisor() getLengthOnJob() getJobTitle() getRole() getCritSkills() getSoftSkills() getTalents()
EmployeeEvaluation (only accessible to managers)	EvaluatingSupervisor DateOfEval MentalState EmployeeNotes	getEvalSupervisor() getDate() getMentalState() getNotes()
DatabaseClass (abstract class)		addAccount(parameters) addEmployee(parameters) addJobHistory(parameters) addEvaluation(parameters) removeAccount(parameters) removeEmployee(parameters) removeJobHistory(parameters) removeEvaluation(parameters) updateAccount(parameters) updateEmployee(parameters) updateJobHistory(parameters) updateEvaluation(parameters)
guiMain	Labels Buttons TextFields	loginPage() userPage() evaluationPage()

GUI Concept Art:

