Mini Project - 1

|  |  |
| --- | --- |
| **Student Name/ID Number:** | Syukur Sidiq Nur Alam bdse-0922-089 |
| **Unit Number and Title:** | ACWD Module 4 – Database Design & Implementation |
| **Academic Year:** | 2023 |
| **Unit Assessor:** |  |
| **Project Title:** | Database Design for Community Portal |
| **Issue Date:** | 17/01/2023 |
| **Submission Date:** | 17/01/2023 |
| **Internal Verifier Name:** |  |
| **Date:** | 17/01/2023 |

|  |
| --- |
| **Learner declaration** |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student signature: Date:17/01/2023 |

|  |
| --- |
| **Purpose of this project** |
| **Purpose of this project**  To demonstrate your capabilities in the following areas:   * Design a Database for Community Portal |
| **Submission Format** |
| 1. Description of Entities in the database 2. Screen capture of the ER Diagram 3. Screen capture of Data 4. Screen capture of Normalized Database Design 5. Documentation of Relationships |
| **Project Brief & Guidance** |
| **Scenario:**  **Refer to the Project Scenario for the Module Project**  You have been approached by ‘ABC Jobs Pte Ltd’ as a website developer to develop a community portal for Software Developers. The project will be carried over through Module 3, Module 4, Module 5 and Capstone project. For this module the scope is to Design, Develop, Implement & Document Apache Struts Framework Website.  The Scope of the Project is to design a Community Portal Similar to Linkedin.com. Users will be able to register in the portal using the Registration Page. Users of the portal can search for other users using various parameters such as First Name, Last Name, Company Name, City & Country. Users will be able to view the Public Profile of users after searching them. The portal allow users to login, request for forgotten password and Update their profile information  The scope of the mini project is to design the database.  **The overview of the project is as below**  There are 2 types of users in this Community portal. They are   1. Software Programmer 2. Administrator   **Software Programmer should be able to perform following functions in the portal** |

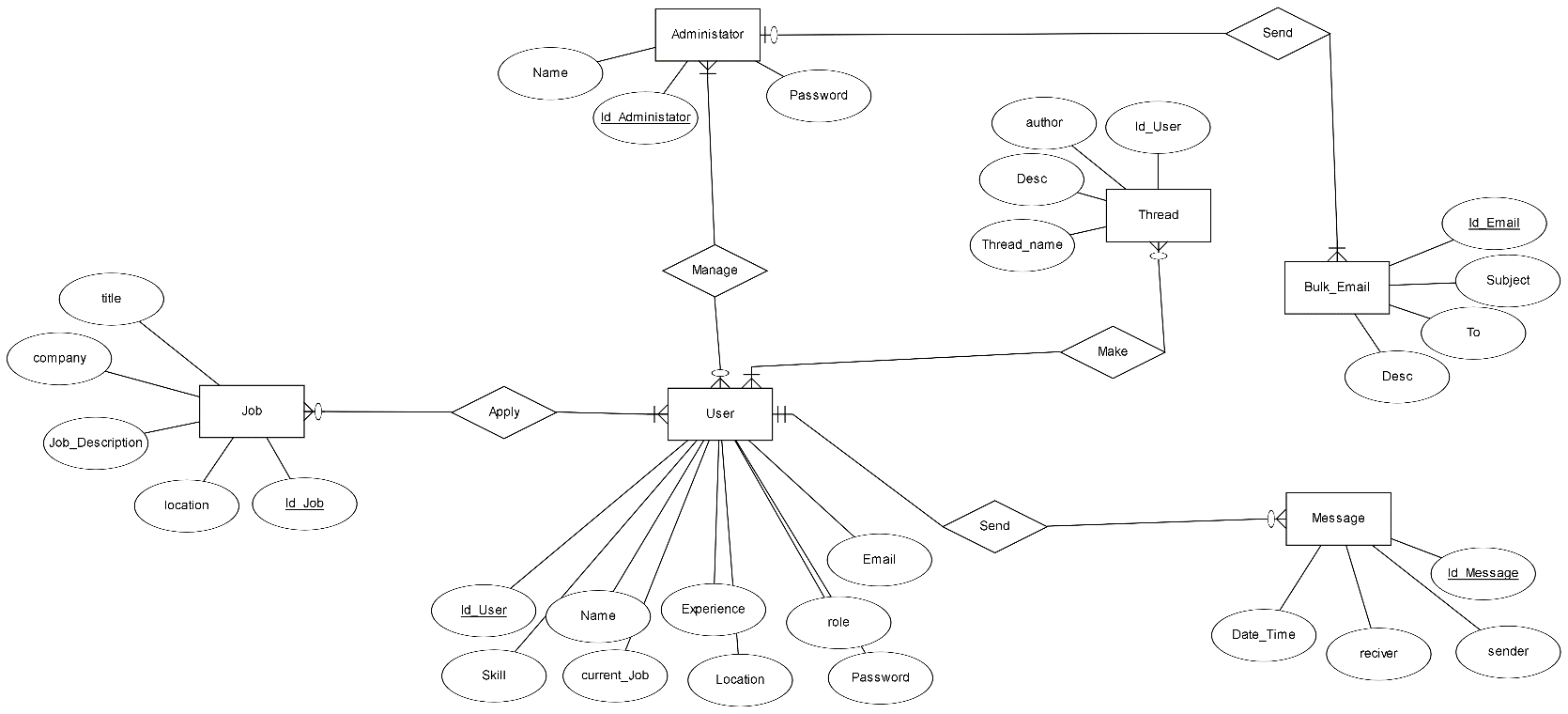
|  |
| --- |
| 1. Allow the programmers to register in the portal, show a thank you page & send a registration confirmation email. 2. Search & Find Other Programmers after login and view their profile. 3. Provide Login Page 4. Provide password retrieval functionality. 5. Update their Profile after logging in.   Following Functionality is part of Database Design & Will be enhanced as features as required in the Capstone Project, Depending on time availability.   1. Send Messages to Each Other on the Portal 2. Create Threads & Post Replies to a Thread 3. Post Job Opportunities in the Portal   **Administrator should be able to perform following functions in the portal**   1. Administer user data. 2. Send bulk email inviting programmers to register on the community portal   **The portal consist of the following Key pages (For Reference)**   1. Community Portal Home Page 2. Registration Page 3. Registration Confirmation Page 4. Update Profile Page 5. Search Users Page 6. List Search Results 7. Public Profile Page 8. Registration Confirmation Email 9. Login Page 10. Forget Password Page 11. Design the Forget Password Confirmation Page   Following Functionality is part of Database Design and will not be part of development & Will be enhanced as features are required in the Capstone Project, Depending on time availability.   1. Send Messages 2. Read Messages 3. Post in Message Board 4. List Message Board 5. Read A Thread 6. Post Job Opportunities 7. List Job Opportunities & Responses   **The scope of this assignment**  The scope is to design the community portal database.   * 1. Identify the Entities in the Community Portal Database   2. Create ER Diagram for the Community Portal Database |

1. Create Sample data for all tables in the community portal (At least 3 tables)
2. Normalize the complete database to 3rd normal form (Provide screen capture of the normalization done)
3. Identify the relationships in tables & document them
4. **Task 1 -** Identify the Entities in the Community Portal Database

These are entities which I’ve created for the model of community portal database:

|  |  |  |
| --- | --- | --- |
| Entities | Description | Pages |
| **User** | It contains information about registering and login | **Registration page, Login Page.** |
| **Admnistrator** | Admin can administer users and send bulk email | Administrator Page |
| **Bulk Email** | To store the e-mail information of each user registered in the system. | Registration and Login Page |
| **Job** | It provides details about the user's job prospects. | Job Page |
| **Message** | Users can send and receive messages | Messages Page |
| **Thread** | we can create a forum in the community portal | Thread Page |

1. **Task 2 -** Create ER Diagram for the Community Portal Database



1. **Task 3 -** Create Sample data for all tables in the community portal (At least 3 tables)
   * + 1. **Sample Table –** User

| **User ID** | **Email ID** | **Name\_user** | **Email** | **Skill** | **Current Job** | **Location** | **Experience** | **Password\_user** | **Roles** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | Ahmad | [user1@gmail.com](mailto:user1@gmail.com) | Java ,PHP, CSS | Software Engineer | Indonesia | 10 years | Anjay12 | Admin |
| 2 | 2 | Joni | [user2@gmail.com](mailto:user2@gmail.com) | Ruby, Java Script | Software Engineer | Singapore | 5 years | Anjay123 | User |
| 3 | 3 | kusno | [user3@gmail.com](mailto:user3@gmail.com) | C++, Python | Software Engineer | Malaysia | 5 years | Anjay1234 | User |
| 4 | 4 | Andi | User4@gmail.com | C++, Java Script | Software Engineer | India | 5 years | Anjay12345 | User |
| 5 | 5 | Mamat | User5@gmail.com | HTML, Python | Software Engineer | Myanmar | 5 years | Anjay123456 | User |

* + - 1. **Sample Table - Message**

| **ID** | **Date Time** | **Sender** | **Receiver** |
| --- | --- | --- | --- |
| 1 | 2021-01-01 10:00 | John | Ahmad |
| 2 | 2021-01-02 11:00 | Jane | Joni |
| 3 | 2021-01-03 12:00 | Michael | Kusno |
| 4 | 2021-01-04 13:00 | Paul | Andi |
| 5 | 2021-01-05 14:00 | Adam | Mamat |

* + - 1. **Sample Table –Bluk Email**

| **ID** | **Email Subject** | **Name Email** | **Email To** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | Meeting Reminder | [Ahmad@gmail.com](mailto:Ahmad@gmail.com) | [John@gmail.com](mailto:%20John@gmail.com) | Reminder for meeting at 2 PM |
| 2 | Report | [Joni@gmail.com](mailto:Joni@gmail.com) | [Jane@gmail.com](mailto:%20Jane@gmail.com) | Attached is the sales report for the month |
| 3 | Holiday Request | [Kusno@gmail.com](mailto:%20Kusno@gmail.com) | [Michael@gmail.com](mailto:Michael@gmail.com) | Requesting for holiday from 15-20 June |
| 4 | Permission | [Andi@gmail.com](mailto:Andi@gmail.com) | [Paul@gmail.com](mailto:Paul@gmail.com) | sorry I'm sick I can't work today |
| 5 | did not attend the meeting | [Mamat@gmail.com](mailto:Mamat@gmail.com) | [Adam@gmail.com](mailto:Adam@gmail.com) | Sorry sir, I was stuck in the toilet for 2 hours |

1. **Task 4 –** Normalize the complete database to 3rd normal form (Provide screen capture of the normalization done)
   * + 1. **Table User**

* **1NF**

1NF ensures that every attribute in the table is atomic (cannot be broken down further) and there are no repeating groups.

| **User ID** | **Email** | **Name\_user** | **Skill** | **Current Job** | **Location** | **Experience** | **Password\_user** | **Roles** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | [user1@gmai.com](mailto:user1@gmai.com) | Ahmad | Java | Software Engineer | Indonesian | 10 years | Anjay12 | Admin |
| 2 | [user2@gmai.com](mailto:user2@gmai.com) | Joni | Ruby | Software Engineer | Singapore | 5 years | Anjay123 | User |
| 3 | [user3@gmai.com](mailto:user3@gmai.com) | Kusno | C++ | Software Engineer | Malaysia | 5 years | Anjay1234 | User |

* **2NF**

2NF ensures that all non-primary key columns are dependent on the entire primary key.

| **User ID** | **Email** | **Name\_user** | **Skill** | **Current Job ID** |
| --- | --- | --- | --- | --- |
| 1 | [user1@gmai.com](mailto:user1@gmai.com) | Ahmad | Java | 1 |
| 2 | [user2@gmai.com](mailto:user2@gmai.com) | Joni | Ruby | 2 |
| 3 | [user3@gmai.com](mailto:user3@gmai.com) | Kusno | C++ | 3 |

| **Current Job ID** | **Current Job** | **Location** | **Experience** |
| --- | --- | --- | --- |
| 1 | Software Engineer | Indonesian | 10 years |
| 2 | Software Engineer | Singapore | 5 years |
| 3 | Software Engineer | Malaysia | 5 years |

| **User ID** | **Password\_user** | **Roles** |
| --- | --- | --- |
| 1 | Anjay12 | Admin |
| 2 | Anjay123 | User |
| 3 | Anjay1234 | User |

* **3NF**

3NF ensures that every non-primary key column is only dependent on the primary key and there is no unnecessary data in the table.

| **User ID** | **Email ID** | **Name\_user** | **Skill ID** |
| --- | --- | --- | --- |
| 1 | [user1@gmai.com](mailto:user1@gmai.com) | Ahmad | 1 |
| 2 | [user2@gmai.com](mailto:user2@gmai.com) | Joni | 2 |
| 3 | [user3@gmai.com](mailto:user3@gmai.com) | Kusno | 3 |

| **Skill ID** | **Skill** |
| --- | --- |
| 1 | Java |
| 2 | Ruby |
| 3 | C++ |

| **Current Job ID** | **Current Job** | **Location ID** | **Experience** |
| --- | --- | --- | --- |
| 1 | Software Engineer | 1 | 10 years |
| 2 | Software Engineer | 2 | 5 years |
| 3 | Software Engineer | 3 | 5 years |

| **Location ID** | **Location** |
| --- | --- |
| 1 | Indonesian |
| 2 | Singapore |
| 3 | Malaysia |

| **User ID** | **Password\_user** | **Role ID** |
| --- | --- | --- |
| 1 | Anjay12 | 1 |
| 2 | Anjay123 | 2 |
| 3 | Anjay1234 | 3 |

| **Role ID** | **Roles** |
| --- | --- |
| 1 | Admin |
| 2 | User |
| 3 | User |

* + - 1. **Table Message**
* **1NF**

1NF ensures that every attribute in the table is atomic (cannot be broken down further) and there are no repeating groups.

| **Id Message** | **Date Time** | **Sender** | **Receiver** |
| --- | --- | --- | --- |
| 1 | 01/01/2022 | John | Ahmad |
| 2 | 01/01/2022 | Jane | Joni |
| 3 | 01/02/2022 | Michael | Kusno |

* **2NF**

2NF ensures that all non-primary key columns are dependent on the entire primary key

| **Id Message** | **Date Time** | **Sender ID** | **Receiver ID** |
| --- | --- | --- | --- |
| 1 | 01/01/2022 | 1 | 2 |
| 2 | 01/01/2022 | 2 | 3 |
| 3 | 01/02/2022 | 1 | 3 |

| **Sender ID** | **Sender** |
| --- | --- |
| 1 | Ahmad |
| 2 | Joni |

| **Receiver ID** | **Receiver** |
| --- | --- |
| 2 | Joni |
| 3 | Kusno |

* **3NF**

3NF ensures that every non-primary key column is only dependent on the primary key and there is no unnecessary data in the table.

| **Id Message** | **Date Time** | **Sender ID** | **Receiver ID** |
| --- | --- | --- | --- |
| 1 | 01/01/2022 | 1 | 2 |
| 2 | 01/01/2022 | 2 | 3 |
| 3 | 01/02/2022 | 1 | 3 |

| **Sender ID** | **Name** |
| --- | --- |
| 1 | John |
| 2 | Jane |

| **Receiver ID** | **Name** |
| --- | --- |
| 1 | Ahmad |
| 2 | Joni |

* + - 1. **Table Bulk Email**
* **1NF**

1NF ensures that every attribute in the table is atomic (cannot be broken down further) and there are no repeating groups.

| **Id Email** | **Email Subject** | **Name Email** | **Email To** | **Description** |
| --- | --- | --- | --- | --- |
| 1 | Meeting | John | Jane | Discuss Project |
| 2 | Reminder | Jane | Michael | Submit Report |
| 3 | Invitation | Michael | John | Attend Conference |

* **2NF**

2NF ensures that all non-primary key columns are dependent on the entire primary key

| **Id Email** | **Email Subject** | **Name Email ID** | **Email To ID** |
| --- | --- | --- | --- |
| 1 | Meeting | 1 | 2 |
| 2 | Reminder | 2 | 3 |
| 3 | Invitation | 3 | 1 |

| **Name Email ID** | **Name Email** |
| --- | --- |
| 1 | John |
| 2 | Jane |
| 3 | Michael |

| **Email To ID** | **Email To** |
| --- | --- |
| 2 | Jane |
| 3 | Michael |
| 1 | John |

| **Id Email** | **Description** |
| --- | --- |
| 1 | Discuss Project |
| 2 | Submit Report |
| 3 | Attend Conference |

* **3NF**

3NF ensures that every non-primary key column is only dependent on the primary key and there is no unnecessary data in the table.

| **Id Email** | **Email Subject ID** | **Name Email ID** | **Email To ID** |
| --- | --- | --- | --- |
| 1 | 1 | 1 | 2 |
| 2 | 2 | 2 | 3 |
| 3 | 3 | 3 | 1 |

| **Email Subject ID** | **Email Subject** |
| --- | --- |
| 1 | Meeting |
| 2 | Reminder |
| 3 | Invitation |

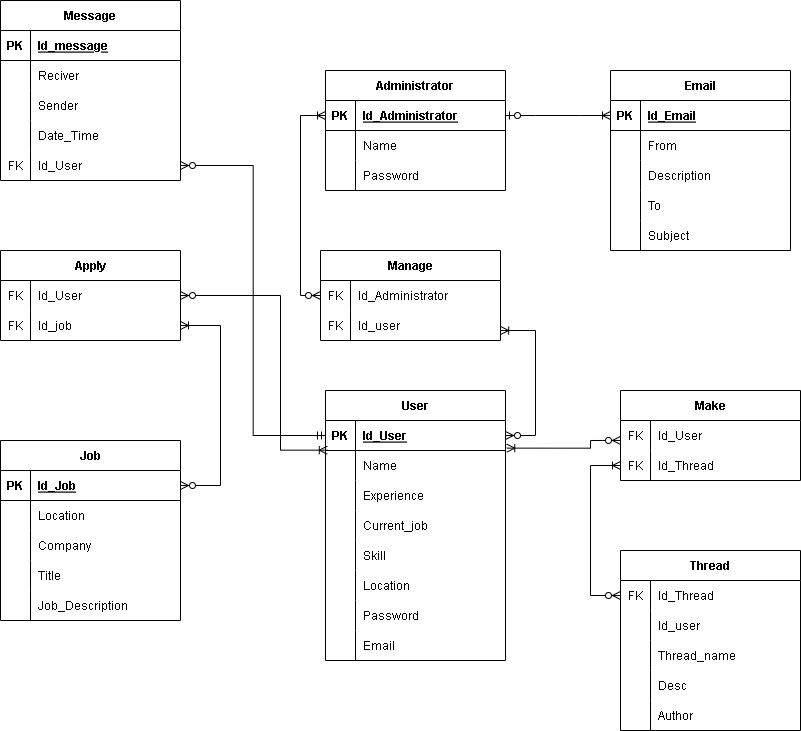
| **Name Email ID** | **Name** |
| --- | --- |
| 1 | John |
| 2 | Jane |
| 3 | Michael |

| **Email To ID** | **Email** |
| --- | --- |
| 2 | Jane |
| 3 | Michael |
| 1 | John |

| **Id Email** | **Description** |
| --- | --- |
| 1 | Discuss |
| 2 | Submit Report |
| 3 | Attend Conference |

**Task 5 -** **Identify the relationships in tables & document them**

**Logical database design - relational schema**

****