

|  |  |
| --- | --- |
| **ASSIGNMENT** | |
| **Course Code** | CSC311A |
| **Course Name** | Database Systems |
| **Programme** | B.Tech |
| **Department** | CSE |
| **Faculty** | Ami Rai E. |
|  |  |

#### 

|  |  |
| --- | --- |
| **Name of the Student** | Shikhar singh |
| **Reg. No** | 17ETCS002168 |
| **Semester/Year** | 06/2017 |
| **Course Leader/s** | Ami Rai E. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Declaration Sheet** | | | | | | | | |
| Student Name | Shikhar singh | | | | | | | |
| Reg. No | 17ETCS002168 | | | | | | | |
| Programme | B. Tech | | | | | Semester/Year | 06/2017 | |
| Course Code | CSC311A | | | | | | | |
| Course Title | Database Systems | | | | | | | |
| Course Date |  | | to | |  | | | |
| Course Leader | Ami Rai E. | | | | | | | |
| **Declaration**  The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly. | | | | | | | | |
| Signature of the Student | |  | | | | | Date |  |
| Submission date stamp  (by Examination & Assessment Section) | |  | | | | | | |
| Signature of the Course Leader and date | | | | Signature of the Reviewer and date | | | | |
|  | | | |  | | | | |

# **Contents**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[**Declaration Sheet** ii](#_Toc38920998)

[**Contents** iii](#_Toc38920999)

[**List of Figures** 4](#_Toc38921000)

[**Question No. 1** 5](#_Toc38921001)

[1.1 Functional and Data Requirements: 5](#_Toc38921002)

[1.2 Implementation of database Tables: 9](#_Toc38921003)

[1.3 Implementation of GUI: 11](#_Toc38921004)

[1.4 Connection of front end with database: 13](#_Toc38921005)

[1.5 Conclusion: 16](#_Toc38921006)

# **List of Figures**

[Figure 1: ER diagram 9](#_Toc38920977)

[Figure 2: sql command to populate staff table 10](#_Toc38920978)

[Figure 3: sql command to populate staff table 10](#_Toc38920979)

[Figure 4: sql command to populate staff table 10](#_Toc38920980)

[Figure 5: staff table with data 10](#_Toc38920981)

[Figure 6: sql command to populate student table. 10](#_Toc38920982)

[Figure 7: student table with data 11](#_Toc38920983)

[Figure 8: sql command to populate department table 11](#_Toc38920984)

[Figure 9: department table with data 11](#_Toc38920985)

[Figure 10: login screen 1 11](#_Toc38920986)

[Figure 11: login screen after filling data 12](#_Toc38920987)

[Figure 12: student and staff home page 12](#_Toc38920988)

[Figure 13: team registration 12](#_Toc38920989)

[Figure 14: team registration 13](#_Toc38920990)

[Figure 15: Gui for viewing project and cancelation option 13](#_Toc38920991)

[Figure 16: code for submit button on registration screen 14](#_Toc38920992)

[Figure 17: code for student home page screen 14](#_Toc38920993)

[Figure 18: Code for logout in student home page screen 15](#_Toc38920994)

[Figure 19: project button on staff home page 15](#_Toc38920995)

[Figure 20: logout button for staff home page 15](#_Toc38920996)

[Figure 21: staff home page constructor 16](#_Toc38920997)

# **Question No. 1**

**Solution to Question No. 1:**

## Functional and Data Requirements:

|  |  |
| --- | --- |
| Requirement Tag | FR1 |
| Requirement Description | The system should allow the user to register in the system using his student ID |
| Dependent on Requirements | - |
| User/System interacting with the requirement | Staff |

|  |  |
| --- | --- |
| Requirement Tag | DR1 |
| Item Name | Student Name |
| Item Description (Where/How used) | The student will be entering his details |
| Item type | Char |
| User/System interacting with the item | student |
| Constraints (if any) | The value should be less than 50 characters. |

|  |  |
| --- | --- |
| Requirement Tag | DR2 |
| Item Name | student ID |
| Item Description (Where/How used) | The student will be entering his student Id. |
| Item type | Integer |
| User/System interacting with the item | Student |
| Constraints (if any) | The value should be an integer number. |

|  |  |
| --- | --- |
| Requirement Tag | DR3 |
| Item Name | Password |
| Item Description (Where/How used) | The student will be entering his password |
| Item type | Char |
| User/System interacting with the item | Employee |
| Constraints (if any) | The value should be combination of characters, digits and special characters. |

|  |  |
| --- | --- |
| Requirement Tag | FR2 |
| Requirement Description | The system should allow the registered user to login in the system, using his user ID and password |
| Dependent on Requirements | FR1 |
| User/System interacting with the requirement | student |

|  |  |
| --- | --- |
| Requirement Tag | DR1 |
| Item Name | Student ID |
| Item Description (Where/How used) | The verification of the student ID from the database. |
| Item type | Integer |
| User/System interacting with the item | student |
| Constraints (if any) | The value should be an integer number. |

|  |  |
| --- | --- |
| Requirement Tag | DR2 |
| Item Name | Password |
| Item Description (Where/How used) | The verification of the student password as in the feed of the database. |
| Item type | Char |
| User/System interacting with the item | student |
| Constraints (if any) | The value should combination of characters, digits and special characters. |

|  |  |
| --- | --- |
| Requirement Tag | FR3 |
| Requirement Description | The student should be able to book his project. |
| Dependent on Requirements | FR1- FR2 |
| User/System interacting with the requirement | student |

|  |  |
| --- | --- |
| Requirement Tag | DR1 |
| Item Name | Department Name |
| Item Description (Where/How used) | Used to distinguish between different Department verbally |
| Item type | Char |
| User/System interacting with the item | Head of department. |

|  |  |
| --- | --- |
| Requirement Tag | DR2 |
| Item Name | Department Id |
| Item Description (Where/How used) | Used to distinguish between different Department |
| Item type | char |
| User/System interacting with the item | Head of department. |

|  |  |
| --- | --- |
| Requirement Tag | FR4 |
| Requirement Description | The Student should be able to join a team. |
| Dependent on Requirements |  |
| User/System interacting with the requirement | Head of department. |

|  |  |
| --- | --- |
| Requirement Tag | DR1 |
| Item Name | Team Name |
| Item Description (Where/How used) | Used to distinguish between different teams verbally |
| Item type | Char |
| User/System interacting with the item | Team leader |

|  |  |
| --- | --- |
| Requirement Tag | DR2 |
| Item Name | Team Id |
| Item Description (Where/How used) | Used to distinguish between different teams |
| Item type | Integer |
| User/System interacting with the item | Team leader |

|  |  |
| --- | --- |
| Requirement Tag | FR5 |
| Requirement Description | The Project leader should be able to control or take projects. |
| Dependent on Requirements | FR4 |
| User/System interacting with the requirement | Project leader |

|  |  |
| --- | --- |
| Requirement Tag | DR1 |
| Item Name | Project ID |
| Item Description (Where/How used) | Used to distinguish between different projects |
| Item type | Integer |
| User/System interacting with the item | Project leader |
| Constraints (if any) | The value should be integer. |

|  |  |
| --- | --- |
| Requirement Tag | DR2 |
| Item Name | Project Name |
| Item Description (Where/How used) | Used to distinguish between different projects verbally |
| Item type | Character |
| User/System interacting with the item | Project leader |

|  |  |
| --- | --- |
| Requirement Tag | DR3 |
| Item Name | Category of project |
| Item Description (Where/How used) | Used to represent the types of the project |
| Item type | Character |
| User/System interacting with the item | Project leader |

|  |  |
| --- | --- |
| Requirement Tag | FR6 |
| Requirement Description | The Mentor should be able to control or take projects. |
| Dependent on Requirements | FR5 |
| User/System interacting with the requirement | Mentor |

|  |  |
| --- | --- |
| Requirement Tag | DR1 |
| Item Name | Mentor Name |
| Item Description (Where/How used) | Used to distinguish between different Mentors |
| Item type | character |
| User/System interacting with the item | Mentor |

|  |  |
| --- | --- |
| Requirement Tag | DR2 |
| Item Name | Mentor ID |
| Item Description (Where/How used) | Used to distinguish between different Mentors |
| Item type | character |
| User/System interacting with the item | Mentor |

An Entity-Relationship Diagram can be used to give a better understanding of the Database and the Relationship between various entities.

The entity relationship diagram for the given problem can be seen in the figure 1.

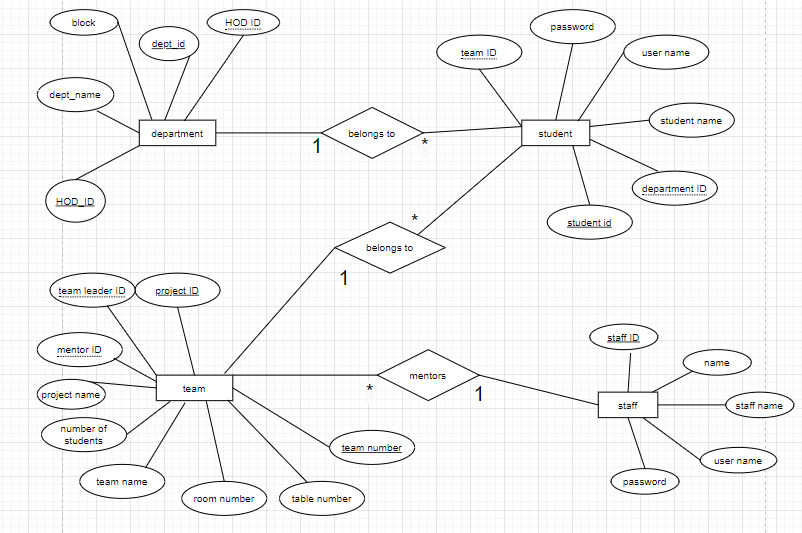


Figure 1: ER diagram

## Implementation of database Tables:

The tables were created using GUI interface provided by netbeans and populated using SQL commands. The code snippets of commands used are given below:

1. Populating staff table

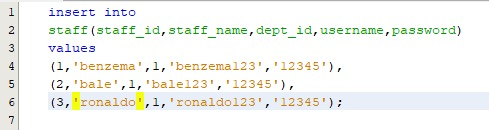


Figure 2: sql command to populate staff table

Here, department id is 1 for CSE, 2 for ECE and 3 for EEE. Figure 2 shows the sql command used to populate staff table for cse department. Similarly, data for other departments was populated. Figure 3 and figure 4 shows the same.

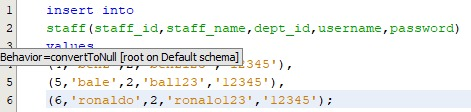


Figure 3: sql command to populate staff table

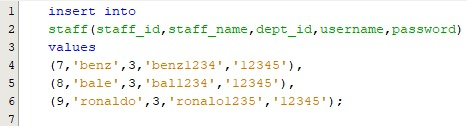


Figure 4: sql command to populate staff table

the staff table after being populated looks like this:

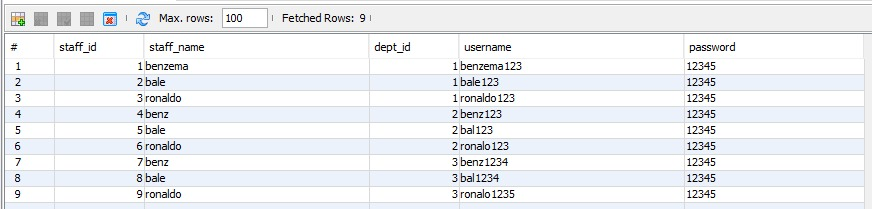


Figure 5: staff table with data

1. Populating student table :

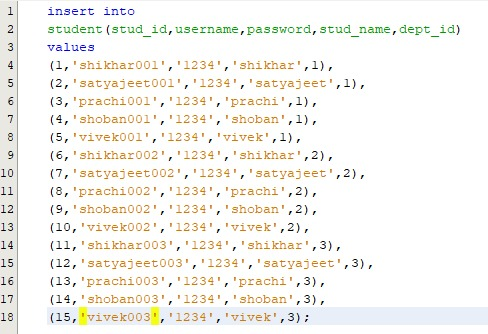


Figure 6: sql command to populate student table.

the student table after being populated looks like this:

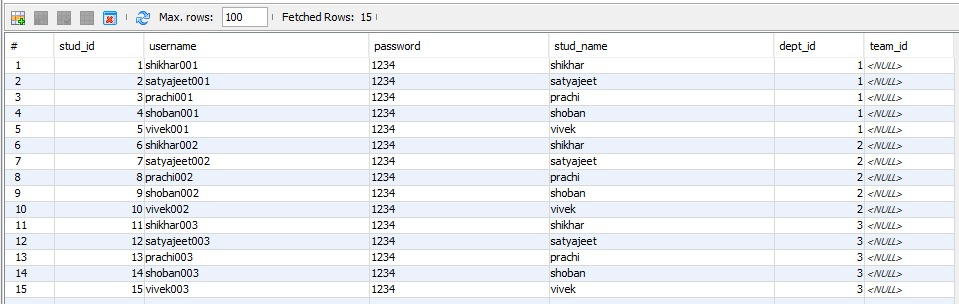


Figure 7: student table with data

1. Populating department table:

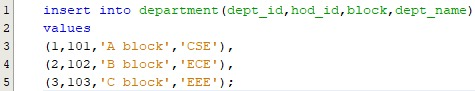


Figure 8: sql command to populate department table

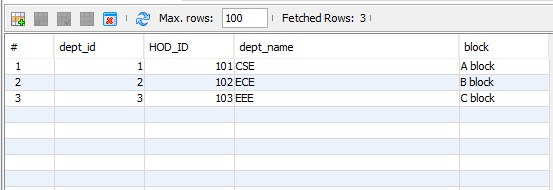


Figure 9: department table with data

## Implementation of GUI:

1. Login-screen interface for both student and staff.

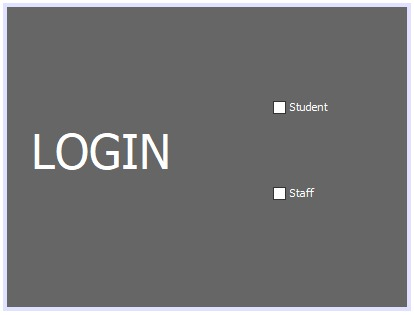


Figure 10: login screen 1



Figure 11: login screen after filling data

1. Home page for both student and staff

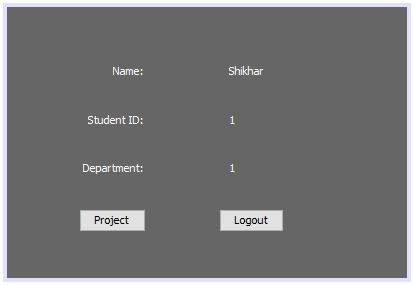


Figure 12: student and staff home page

1. GUI for team registration.

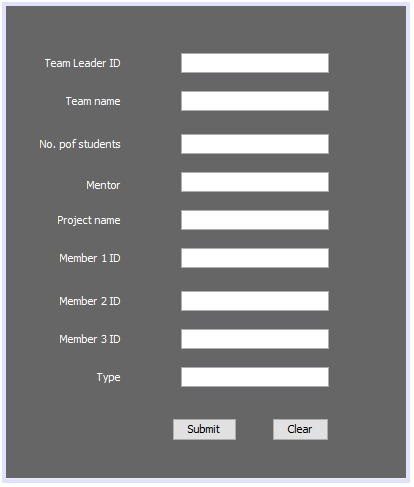


Figure 13: team registration

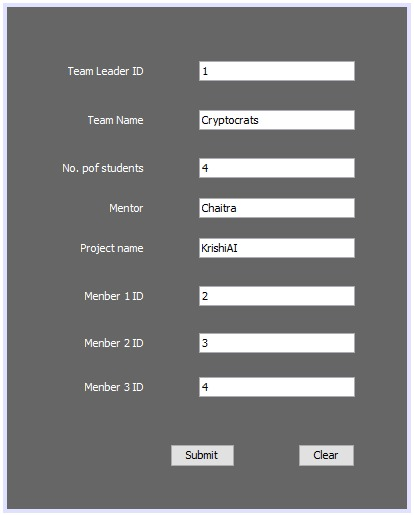


Figure 14: team registration

1. GUI for viewing project details

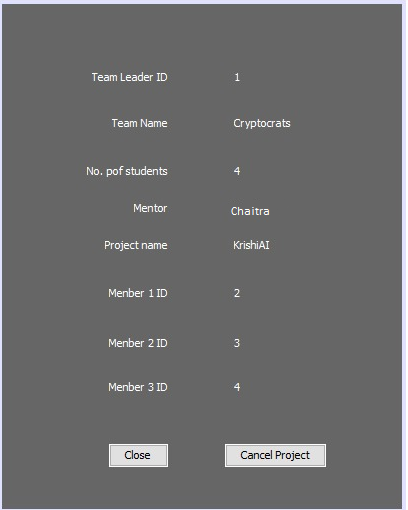
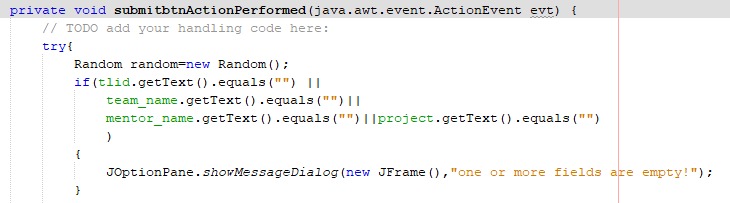
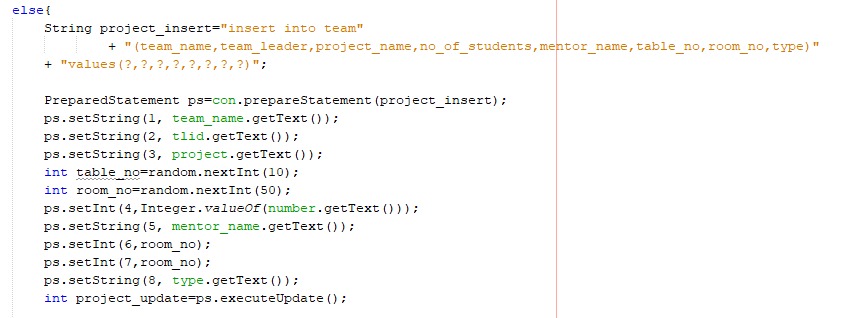


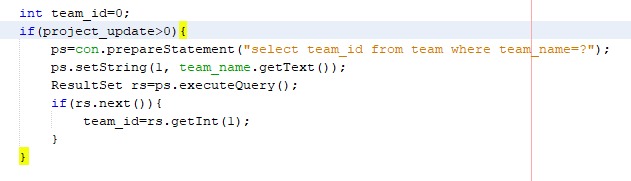
Figure 15: Gui for viewing project and cancelation option

## Connection of front end with database:

1. Snippet for Submit button on the registration page: passes all the entered values into the database.







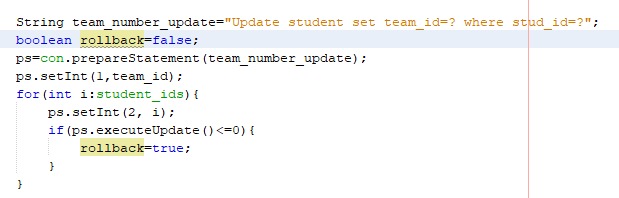


Figure 16: code for submit button on registration screen

1. Snippet for project button action in student home page

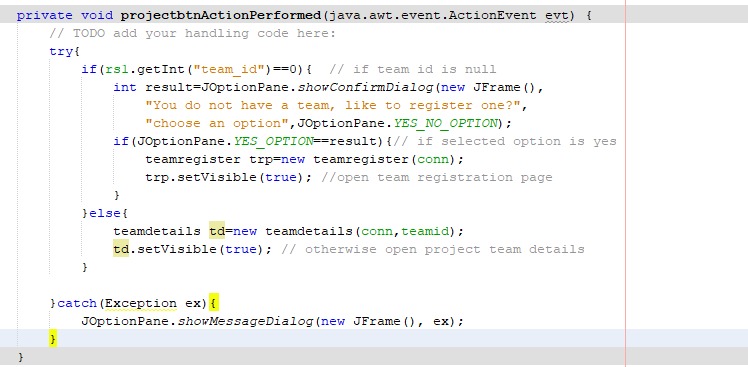


Figure 17: code for student home page screen

1. Snippet for logout in student home page screen

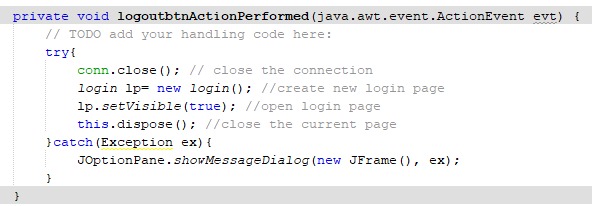


Figure 18: Code for logout in student home page screen

1. Snippet for Project button on staff home page

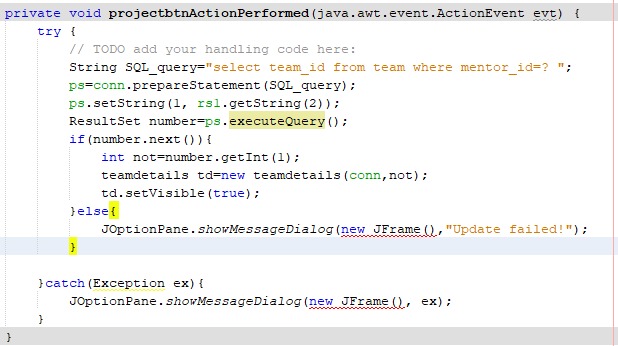


Figure 19: project button on staff home page

1. Snippet for logout button for staff home page

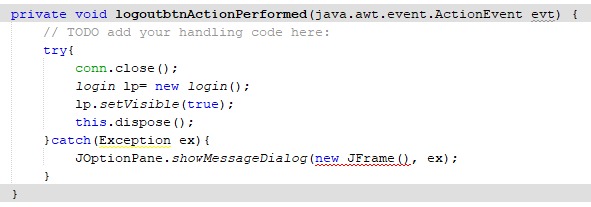
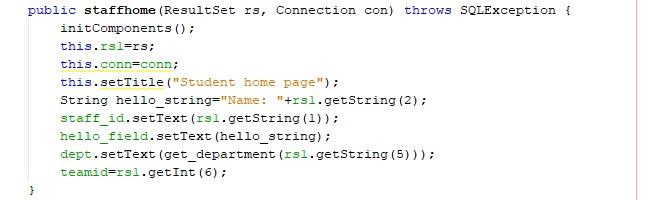


Figure 20: logout button for staff home page

1. Snippet for staff home page constructor



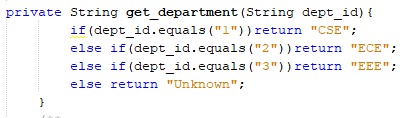


Figure 21: staff home page constructor

## Conclusion:

CONCLUSION:

A database management system is important because it manages data efficiently and allows users to perform multiple tasks with ease. A database management system stores, organizes and manages a large amount of information within a single software application.

The user interface (UI) is a critical part of any software product. When it is done well, users do not even notice it. When it is done poorly, users cannot get past it to efficiently use a product. To increase the chances of success when creating user interfaces, most designers follow interface design principles. Interface design principles represent high-level concepts that are used to guide software design.

The UI design principals are:

* Place users in control of the interface
* Make it comfortable to interact with a product
* Reduce cognitive load
* Make user interfaces consistent

LIMITATION:

the limitation of this application comes directly from the structure of SQL. SQL database is prone to SQL injections which can result in corruption of the database hence ruining the backbone of the project. Also, using the wild card operator, i.e. \* a person can view all the confidential data including passwords.

Another limitation is that the application stores passwords without hashing which is harmful in case of cyber-attacks as the passwords are stored in plaintext.

IMPROVEMENT:

The database can be changed from SQL to NoSQL databases like firebase or mongo dB, which provides more useful and modern features like easy scalability, more security, support for real-time changes etc.

Also the passwords should be passed through hash functions which provides proper encoding to the passwords before storing.

**Bibliography**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. <https://xd.adobe.com/ideas/process/ui-design/4-golden-rules-ui-design/>
2. https://www.manomayasoft.com/blog/item/210-what-is-the-importance-of-a-database-management-system
3. https://www.javaworld.com/article/3388036/what-is-jdbc-introduction-to-java-database-connectivity.html