# PERSONAL PORTFOLIO – N8947791

## Code Snippets

This code is written in Java and based on the knowledge gained in IAB130-Databases.

The code establishes a JDBC driver, creates a database for storing the username and passwords of the user and then connects to that database.

The rest of the code is in the folder.

### Establish a connection

**static** **final** String ***JDBC\_DRIVER*** = "com.mysql.jdbc.Driver";

// Address of the database.

**static** **final** String ***DATABASE\_URL*** = "jdbc:mysql://localhost/";

**static** **final** String ***DATABASE\_NAME*** = "MediaVault";

**static** Connection *connection*=**null**;

**static** PreparedStatement *preparedStatement*=**null**;

**static** ResultSet *resultSet* = **null**;

// The admin’s username and password is entered.

**static** **final** String ***USERNAME*** = "username";

**static** **final** String ***PASSWORD*** = "password";

### Create database

**public** **static** **void** createDatabase() **throws** SQLException, ClassNotFoundException {

String myDriver = "com.mysql.jdbc.Driver";

Class.forName(myDriver);

*connection* = DriverManager.getConnection(***DATABASE\_URL***, ***USERNAME***, ***PASSWORD***);

Statement statement = *connection*.createStatement();

statement.executeUpdate("CREATE DATABASE IF NOT EXISTS " + ***DATABASE\_NAME*** + ";");

statement.close();

System.out.println("Database is created.”);

}

### Connect to database

**public** **static** **void** connectToDatabase() **throws** SQLException, ClassNotFoundException {

//Register JDBC driver

String myDriver = "com.mysql.jdbc.Driver";

Class.forName(myDriver);

*connection* = DriverManager.getConnection(***DATABASE\_URL*** + ***DATABASE\_NAME***, ***USERNAME***, ***PASSWORD***);

Statement statement = *connection*.createStatement();

statement.executeUpdate("CREATE DATABASE IF NOT EXISTS " + ***DATABASE\_NAME*** + ";");

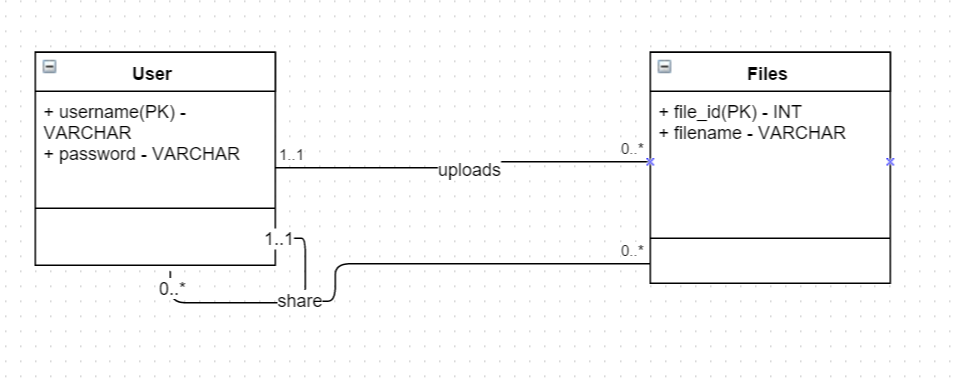
System.out.println("Connected to the database.”);

### }

## DATABASE DESIGN

MySQL was used to store the usernames and passwords of the user along with the file details like users who have access to it, etc. An E-R diagram was drawn to come up with the following relational schema. Database design is required to estimate what details are to be stored.

### ENTITY-RELATIONSHIP MODEL



### RELATIONAL SCHEMA

The users are asked to register for an account using a username, password and email-address. The passwords are stored in a hash along with a salt.

User( username, password, email\_id )

Each file that is uploaded is recorded in the database with an auto-generated file id along with the file name and the uploader’s username.

Files( file\_id, file\_name, owner(FK) )

Information about the files shared along with the access rights like write, read, etc is also stored in the MySQL database MediaVault.

Shared\_Files( file\_id(PK), shared\_to\_username(PK), owner(PK), access )

## USER STORIES, REVIEW LETTER

I contributed to user stories regarding the encryption of data and storing the hash-value of passwords with salt rather than just plain-text passwords. Also, prioritized the user stories along with the other team members. The introduction, conclusion and some questions on the business review letter were done by me. I have worked with the team to review our sprint plans and discussed with them on what would be the best way to divide the implementation into two releases such that release one would be functional and profitable to the client.

I have took part in discussions regarding the environment for coding and the optimal method to store the files that are uploaded by the user. Also took part in discussions involving negotiations with the client team about the release plans, sprints and user stories.

## PROJECT ANALYSIS

I have used the Waterfall model (Requirements, Design, Implementation and Verification) to break down the project development process and have used a tree diagram to illustrate the different components essential for the application to function. I have included the tree diagram in the folder.

## PROJECT IMPLEMENTATION AND TESTING

I have been searching for an efficient method (platform) to store the files or data that the user uploads. Currently, the team is planning to use Google App Engine for storage. I am trying to improvise this design decision by trying to find a computing platform that is efficient, has a large storage capacity (since, media files are large), cost-effective and is reliable and fast. This decision is important for this project as the core concept of this project is to provide a medium for storage of media files. If an appropriate platform is not found to serve this purpose, the whole application could come crashing down.

Though, I couldn’t test the application using Selenium, due to platform issues, I have manually tested and verified the application against the acceptance criteria. Also, run it in different web browsers, to see if it is compatible.