Beschrijving van de klasse database bewerkingen.

Methode:

De methode LaadSqlDatabase(), het is de bedoeling dat deze methode de database met simulatie gegevens van de SQL database server in leest als er en connectie met de server is

public void laadSqlDatabase()

{

Connection con = openSQLConnection();

if (con != null)

{

// Code om de gegevens in te lezen

}

}

Wanneer men een simulatie wil opslaan, dan wordt er eerst gekeken of er al een database bestaat met dezelfde naam. Indien de naam al bestaat krijgt de gebruiker de keuze om de bestande database te verwijderen. Indien de naam niet bestaat maakt deze methode een database aan met de bijbehorende tabellen.

public void maakSqlDatabase(String naam)

{

…

…

}

Deze methode is gemaakt om een database van de server te verwijderen indien deze bestaat.

public void DropDatabase(String naamDatabase)

{

…..

….

}

Deze method wordt gebruikt om te kijken of een bepaalde database al aanwezig is op de sql server.

Indien de database met opgegeven naam al aanwezig is op de server dan retourneert de methode true anders retourneert de methode false

public boolean checkDBExists(String dbName)

{

……

….

}

De methode die in alle andere methodes gebruikt wordt om de verbinding met de server tot stand te brengen is:

private Connection openSQLConnection() {

String url = "jdbc:mysql://localhost:3306/";

String user = "root";

String passw = "sql";

try {

//registreer JDBC Driver

Class.forName("java.sql.Driver");

//Open de verbinding

return DriverManager.getConnection(url, user, passw);

} catch (ClassNotFoundException | SQLException ex) {

JOptionPane.showMessageDialog(null, "Er is iets misgegaan met het openen van de connectie : " + ex.getMessage(), "ERROR", JOptionPane.ERROR\_MESSAGE);

}

return null;

}

Indien de verbinding geopend kan worden retourneert de methode “de verbinding” indien dit mislukt wordt er “null” geretourneerd

Let op dat de usernaam en password voor de SQL server toegang hardcodes zijn.