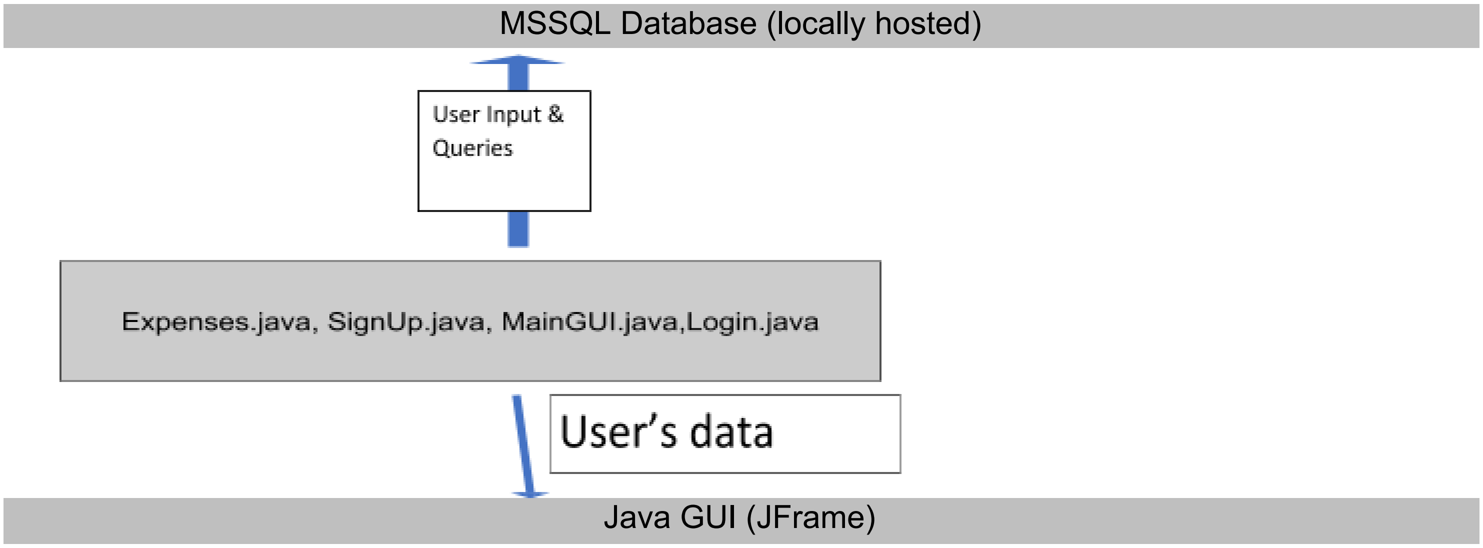
Samina, Noah, Jonathan

Budgeting Application

Database and Method Class Documentation

Application Diagram:


MSSQL Database (locally hosted)
User Input & Queries
Expenses.java, SignUp.java, MainGUI.java,Login.java
User’s data
Java GUI (JFrame)

Design Specification:

The program is designed to take the user’s input from the GUI and pass it to the database in MSSQL. The data is then stored based on the user’s personal information, and a unique table is created inside the database that is based on the user’s desired username. This unique table will hold the user’s expenses, salary, and savings goal.

The database is a MSSQL database that is locally hosted on the user’s computer. The database is titled “User_DB” and consists of one premade table, titled “User_Table”, along with a table that is generated once the user has signed up. The User_Table consists of 5 columns:

- userNameFL (varchar(50))
- username (varchar(50))
- userEmail (varchar(50))
- userPass (varchar(50))
- userAge (unused column)

The Expenses.java, SignUp.java, MainGUI.java, and Login.java have all of the callable methods that connect with our Database Directly. Below is a sample code that allows each method to independently connect with our locally hosted database.

```
String url = “jdbc:sqlserver://localhost:1433;databaseName=App_DB;integratedSecurity=true;”;

Connection conn = DriverManager.getConnection(url);
```

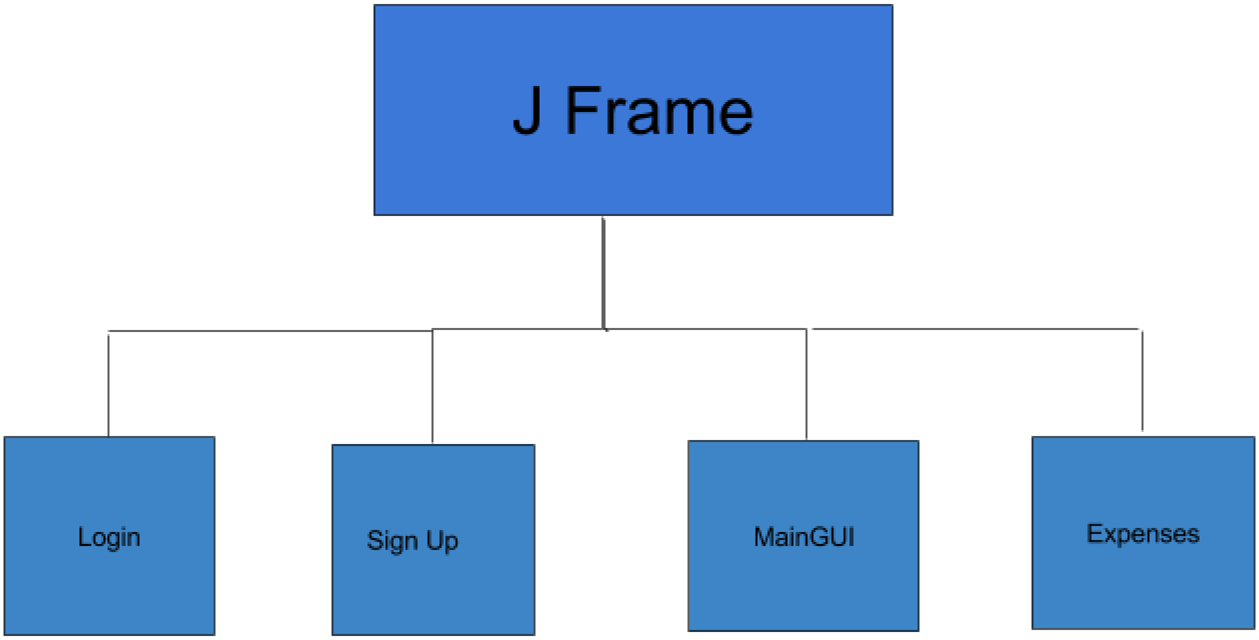
Here you can find many different methods to manipulate the database. Not all methods were included in the final version of the GUI, but they can be easily implemented in future versions/updates of the application. Our Java GUI made it easy to communicate between the MSSQL server and integrate our methods and queries.

```
try{

String url = “jdbc:sqlserver://localhost:1433;databaseName=App_DB;integratedSecurity=true;”;

Connection conn = DriverManager.getConnection(url);

Statement sta = conn.createStatement();

String getUser = “select userName from User_Table where userName = “ + username + “”;

ResultSet in = sta.executeQuery(getUser);
```

The GUI, documented elsewhere, communicates to the database by calling specific methods when buttons are pressed and taking user input in text fields and passing that information through the methodClass to the database. The specific methods were all attached to buttons that appear on the GUI and are mentioned in the User Manual

Our application can be used on both the Windows and MacOS. All the user needs is administration privileges on the computer, the Eclipse IDE and MS SQL Server downloaded on the machine. The Deployment article gives the user the instruction to set everything up, including connecting the IDE to the MSSQL database.

Graphical User Interface:



Login Frame:

- This Frame is the main window that opens when the program runs.
- It asks for the user's login and password.
- It contains three buttons: login- logs the user if they exist in the database and if the password matches. If the condition is true then it creates a new instance of mainGUI frame. signup- Asks the user to create a new profile in the database. Exit- exits the program using System.exit(0)
- The frame takes uses null layout. Components are hardcoded with .setBounds().

Sign Up Frame:

- This frame is used to take in user's Name, Age, email, username, and password.
- Uses JTextFields to input data.
- It contains two buttons. Continue button that takes the input in JTextField and passes it to the database. Exit button disposes of the frame.

MainGUI Frame:

- In this frame the user inputs their Salary and Savings Goals.
- The Frame contains two JTextFields to take in the users Salary and the user's Savings Goals.
- The Frame contains a JTextArea that displays database data.
- The Frame contains four Buttons. Update Button- Which updates the user's Salary and savings goals in the database. logout Button- Which returns the user to the login screen.

displayButton- Returns and prints the user’s salary and savings goal to the JTextArea. Expenses Button- Creates a new JFrame where the user can input their expenses.

Expenses Frame:

- In this frame the user inputs their expenses
- The frame contains two JTextFields to take in the users Expense Name and the expenses Value
- The frame contains two buttons add and close. add Button- Continuously creates a new Column with value in database for user. Close Button- Closes the expenses Frame

Layout:

- Login Frame uses a grid layout to clearly orient the buttons and textfields
- The other frames, (MainGUI, Expenses, SignUp) use no layout (null).
- These frames buttons and TextAreas are hardcoded with .setBounds();