

Expense Management System - Documentation

Overview

This documentation covers the Java-based Expense Management System with user authentication and expense tracking capabilities.

Class Documentation

1. DataBaseConnection.java

Purpose: Manages database connections

Methods:

- getConnection(): Returns a connection to the MySQL database

Configuration via constants:

```
private static final String URL = "jdbc:mysql://localhost:3306/your_database_name";
```

```
private static final String USER = "your_username";
```

```
private static final String PASSWORD = "your_password";
```

2. LoginFrame.java

Purpose: Handles user authentication

UI Components:

- Username and password fields

- Login button

- Signup navigation button

Key Methods:

- Validates credentials against database

- On success: Launches ExpenseManager with user ID

- On failure: Shows error message

3. SignupFrame.java

Purpose: Handles new user registration

UI Components:

- Full name, email, username, password fields

- Registration button

Expense Management System - Documentation

- Login navigation button

Key Methods:

- `userExists()`: Checks for duplicate users
- `verifyDataInserted()`: Debug helper
- Validates and inserts new user records

4. ExpenseManager.java

Purpose: Main application dashboard

Features:

- Expense management (add/view/delete)
- Monthly reporting
- Account management

UI Components:

- Card layout for different views
- Menu bar with:
 - Expenses menu
 - Reports menu
 - Account menu (with "Return to Welcome")

Key Methods:

- `showMonthlyReport()`: Generates category summaries
- Expense CRUD operations
- Navigation between views

Database Schema

Tables:

account (User accounts)

```
CREATE TABLE account (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    full_name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,
```

Expense Management System - Documentation

```
username VARCHAR(50) NOT NULL UNIQUE,  
password VARCHAR(100) NOT NULL  
);  
  
expenses (User expenses)  
CREATE TABLE expenses (  
    expense_id INT AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,  
    amount DECIMAL(10,2) NOT NULL,  
    category VARCHAR(50) NOT NULL,  
    description TEXT,  
    expense_date DATE NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES account(id)  
);
```

Setup Instructions

Prerequisites:

- Java JDK 8+
- MySQL Server
- MySQL Connector/J

Database Setup:

```
CREATE DATABASE expense_manager;  
USE expense_manager;  
-- Create tables (see schema above)
```

Configuration:

Update DataBaseConnection.java with your:

- Database name
- MySQL username

Expense Management System - Documentation

- MySQL password

Running the Application:

```
javac *.java
```

```
java LoginFrame
```

Security Considerations

Password Security:

- Currently stores plain text passwords (not production-ready)
- Recommended to implement password hashing (e.g., BCrypt)

SQL Injection:

- Uses PreparedStatement for all queries
- Properly parameterized inputs

Session Management:

- Simple window-based navigation
- No token-based session handling

Extension Points

Future Enhancements:

- Password recovery
- Expense categories management
- Data export (CSV/PDF)
- Charts and visualizations
- Multi-user expense sharing

Technical Debt:

- Move database operations to separate DAO classes

Expense Management System - Documentation

- Implement proper MVC separation
- Add logging framework
- Connection pooling

Troubleshooting

Common Issues:

Database Connection Problems:

- Verify MySQL service is running
- Check connection credentials
- Ensure MySQL connector is in classpath

UI Rendering Issues:

- Verify Java version compatibility
- Check for missing Swing dependencies

Data Not Persisting:

- Check auto-commit settings
- Verify user has write permissions
- Check for silent SQL exceptions