

# - Graphical User Interfaces in Java -

## CashHive - E-Wallet Application

Davide SCACCIA, Danilo SPERA, Arcangelo MAURO, Giovanni ROMANO, Arthur  
PHOMMACHANH

January 18, 2025

Brno University of Technology



## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	What is CashHive? . . . . .	2
<b>2</b>	<b>Product Overview</b>	<b>2</b>
2.1	What Can CashHive Do? . . . . .	2
2.2	Technology Stack . . . . .	2
<b>3</b>	<b>Installation and Deployment</b>	<b>3</b>
3.1	Project Structure . . . . .	3
3.2	Installation Steps . . . . .	3
<b>4</b>	<b>References</b>	<b>4</b>

# 1 Introduction

## 1.1 What is CashHive?

CashHive is an e-wallet application designed for individuals to effectively manage their personal finances. Its most important features are budgeting, money transfers, and overlooking how you spend your finances!

# 2 Product Overview

## 2.1 What Can CashHive Do?

CashHive offers the following features:

### 1. User Registration and Authentication:

- Users can securely register, log in, and access their personal dashboard.
- Email verification during registration.
- Optional two-factor authentication for added security during login.

### 2. Dashboard Overview:

- Provides a summary of financial data, including:
  - Current account balance.
  - Budget allocation and tracking.
  - Savings in the Piggy Bank.

### 3. Money Transfers:

- Transfer money to other users securely.
- Request money from friends or family.

### 4. Deposit Money:

- Deposit funds from an external bank account using an API.
- Generate QR codes with Short Payment Descriptors for quick deposits.

### 5. Withdraw Money:

- Withdraw funds from the e-wallet to a registered bank account.

### 6. Budget Management:

- Set monthly budgets and allocate them to different categories (e.g., food, transport, entertainment).

### 7. Transaction History:

- View a detailed history of all transactions, including date, amount, and type.

### 8. Piggy Bank Savings:

- Save for specific goals while locking the funds for a defined period.
- Monitor progress toward savings goals.

## 2.2 Technology Stack

- Backend: Jakarta EE, Hibernate
- Frontend: PrimeFaces (JSF)
- Database: PostgreSQL
- Server: GlassFish Server
- Build Tool: Maven

## 3 Installation and Deployment

### 3.1 Project Structure

The project folder contains two main subprojects:

1. **e-wallet:** This is the main application that provides the core e-wallet functionality, including money transfers, piggy bank management, budgeting, and transaction tracking.
2. **BankSimulator:** This is an optional utility to simulate deposits into user accounts using an API.

### 3.2 Installation Steps

1. **Clone the Repository or Unzip the Folder:**

```
git clone https://github.com/Obli04/GUI-Repository.git
```

2. **Build the e-wallet Application:**

```
cd e-wallet
mvn package
```

3. **(Optional) Build the BankSimulator:**

```
cd bank-simulator
mvn package
```

4. **Set Up JDBC Connection Pool in GlassFish:**

- Host a GlassFish Server (we used both 7.0.12 and 7.0.21 - and it works fine).
- On the GlassFish Server Console, create a new JDBC Connection Pool.

Property	Value
Pool Name	PostgresPool
Resource Type	javax.sql.ConnectionPoolDataSource
Datasource Classname	org.postgresql.ds.PGSimpleDataSource
User	defaultdb_owner
Password	zG10j8VLRHWc
DatabaseName	defaultdb
ServerName	ep-winter-wind-a2apa1eh.eu-central-1.aws.neon.tech

Table 1: JDBC Connection Pool Configuration

5. **Create a JDBC Resource**

- Click on new and insert into JNDI Name: *jdbc/PostgresPool*
- Select Pool Name the JDBC Connection Pool you just created: *PostgresPool*
- Confirm the creation by clicking on Ok.

6. **Deploy the Application:**

- Navigate to the GlassFish Server Admin Console ('http://localhost:4848').
- Upload the '.war' file from the 'target/' folder.

7. **Access the Application:**

```
http://localhost:8080/e-wallet
```

## 4 References

1. Jakarta EE Documentation: <https://jakarta.ee/learn/docs/jakartaee-tutorial/current/index.html>
2. PrimeFaces Documentation: <https://www.primefaces.org/showcase/getstarted.xhtml>
3. PostgreSQL Documentation: <https://www.postgresql.org/docs/>
4. GlassFish Documentation: <https://glassfish.org/documentation.html>