

Лаб 3 Сидорін Д.

Завдання

1. Розділити додаток на основні компоненти/модулі:

Для вашого симулятора банку можна виділити такі основні компоненти:

- **Користувач (User)** – представляє клієнта банку, який взаємодіє з системою.
- **Обліковий запис банку (BankAccount)** – обробляє операції з рахунками (поповнення, зняття коштів, баланс).
- **Транзакції (Transaction)** – здійснює всі фінансові операції (переклади, списання, депозити).
- **Інтерфейс користувача (UI)** – відображає інформацію користувачу, включаючи баланс рахунків, доступні транзакції тощо.
- **Авторизація та аутентифікація (Auth)** – управління доступом користувачів (реєстрація, вхід, вихід).

Структура проекту:

bank-simulator:

lab3:

components

auth.js

bankAccount.js

transaction.js

user.js

ui

index.js

data

users.json

accounts.json

transactions.json

2. Описати взаємодію компонентів:

Між компонентами буде взаємодіяти наступним чином:

- **User** має доступ до **Auth** для реєстрації та входу.
- **BankAccount** зберігає інформацію про користувача та його рахунки.
- **Transaction** взаємодіє з **BankAccount** для виконання фінансових операцій.
- **UI** відображає інформацію з **BankAccount** та **Transaction**.

3. Створіть діаграму компонентів ПЗ:

Для опису взаємодії компонентів можна створити **діаграму компонентів ПЗ**. Вона буде виглядати приблизно так:

- **User** — взаємодіє з → **Auth**
- **User** — взаємодіє з → **BankAccount**
- **BankAccount** — взаємодіє з → **Transaction**
- **Transaction** — взаємодіє з → **BankAccount**
- **UI** — взаємодіє з → **BankAccount** та **Transaction**

4. Описати дані та їх зв'язки (ER діаграма):

ER діаграма описує структуру даних та їхні зв'язки між собою. У вашому випадку:

- **User**: має поля, такі як ім'я, email, пароль.
- **BankAccount**: має поля, такі як номер рахунку, баланс, тип рахунку.
- **Transaction**: має поля, такі як тип операції, сума, дата.

Приклад ER діаграми:

[User] 1 - ∞ [BankAccount] 1 - ∞ [Transaction]

- Один **User** може мати кілька **BankAccount**.
- Один **BankAccount** може мати кілька **Transaction**.

5. Опис сценаріїв роботи додатку:

Тепер потрібно описати, як дані оновлюються/змінюються/агрегуються на основі сценаріїв:

Ключові сценарії:

1. Реєстрація користувача:

- Користувач вводить ім'я, email і пароль.
- Додаток перевіряє наявність користувача в базі та створює нового користувача.
- Данні користувача зберігаються в **users.json**.

2. Авторизація користувача:

- Користувач вводить email та пароль.
- Додаток перевіряє правильність введених даних.
- Якщо вірно, користувач отримує доступ до своїх **BankAccount**.

3. Створення рахунку:

- Користувач може створити новий рахунок.
- Додаток додає новий рахунок до **accounts.json** для відповідного користувача.

4. Транзакції:

- Користувач може зробити транзакцію (поповнення, зняття, переказ).
- **Transaction** зберігається в **transactions.json** та оновлюється баланс на рахунку.