Politechnika Warszawska

Wydział Elektryczny

Informatyka Stosowana

Sprawozdanie z projektu 1

Testowanie oprogramowania (1DI1542)

Michał Bogusz
Krzysztof Dąbrowski
Adrian Krzeszewski
Mateusz Łuczak
Paweł Osiński

Spis treści

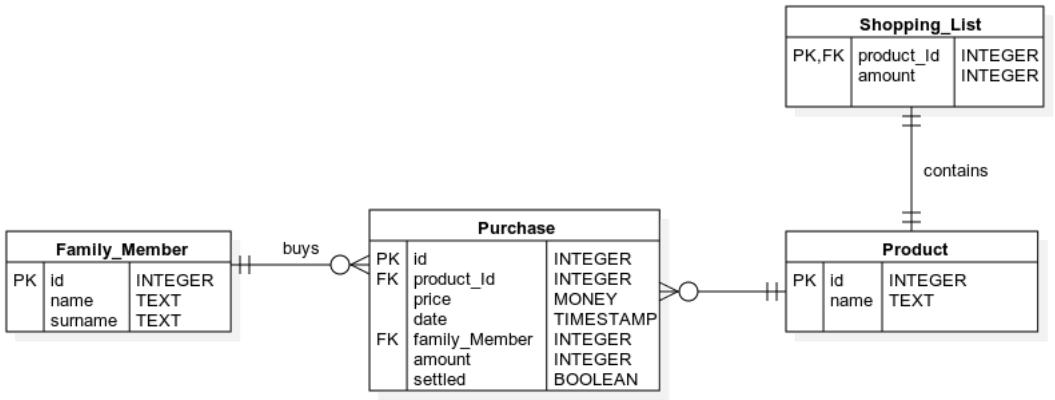
1 Opis projektu

Podczas wykonywania projektu stworzono prostą aplikację służącą do zarządzania budżetem domowym. Wszyscy użytkownicy systemu – domownicy – mają możliwość edytowania listy zakupów oraz raportowania dokonanych zakupów.

2 Baza danych

Zaprojektowana do tego projektu baza danych pozwala wypełnić wszystkie opisane założenia. Składa się z 4 tabel zawierających rekordy reprezentujące:

- członków rodziny;
- produkty;
- wykonane zakupy;
- listę zakupów.



Rysunek 1: Model bazy danych

3 Funkcjonalność programu

W programie, użytkownik ma dostęp do następujących funkcji:

- dodawanie członków rodziny
- dodawanie produktów
- rejestrowanie zakupów
- wyświetlanie listy członków rodziny

- wyświetlanie listy produktów
- wyświetlanie listy zakupów
- wyświetlanie raportu zakupów
- usuwanie członków rodziny
- usuwanie produktów

4 Architektura systemu

Projekt został wykonany z użyciem silnika Docker. Baza danych implementujących standard SQL jest zarządzana za pomocą systemu PostgreSQL. Baza jest uruchamiana jako kontener silnika Docker. Warstwa aplikacyjna projektu została wykonana w języku Java. W celu zapewnienia sprawnego współdziałania napisanych skryptów z systemem PostgreSQL użyty został framework Hibernate, do którego dostęp został zapewniony poprzez metody Javy, napisane ściśle według paradygmatu CRUD. Do testów użyto biblioteki JUnit, a do zapewnienia konsolidacji wszystkich elementów – narzędzia Maven.

5 Testy

Część aplikacyjna została przetestowana ze szczególnym naciskiem na sprawdzenie poprawności zaimplementowania metod opartych na paradygmacie CRUD. Wszystkie metody implementujące jego założenia zostały opatrzone testami jednostkowymi, które uwzględniają scenariusze mające na celu sprawdzenie kluczowych funkcjonalności aplikacji.

Ponieważ CRUD jest paradygmatem, który z swojej definicji zapewnia odpowiednie reguły współpracy różnych warstw programu, jego testowanie – mimo, że może być przeprowadzane ściśle pod kątem czterech składowych paradygmatu – **nie może się odbyć w oderwaniu od całości systemu**. Warto jednak zauważyć, że z powodu mnogości rozwiązań, za pomocą których możliwe jest powiązanie aplikacji z bazą danych, możliwe jest takie stworzenie testów, aby sprawdzać poprawność CRUD niezależnie od standardu i sposobu komunikacji z bazą danych. Takie rozwiązanie wydaje się atrakcyjnym sposobem testowania poprawności komunikacji, gdyż aplikacja może korzystać z różnych typów baz danych lub sposobów łączenia się z bazami, jak również w sytuacjach, gdy brana jest pod uwagę możliwość zmiany obszaru bazodanowego projektu w przyszłości.

6 Podsumowanie

Użycie silnika Docker pozwala na szybkie i wygodne zbudowanie systemu, który charakteryzuje się wysoką przenaszalnością. Kontenery Docker zapewniają modularną budowę systemu, co przekłada się na większą elastyczność, oraz pozwala zmniejszyć

liczbę potrzebnych do zainstalowania oraz skonfigurowania narzędzi, przed przystąpieniem do pracy nad projektem. Paradygmat CRUD okazuje się przydatny podczas tworzenia połączenia między warstwą aplikacyjną projektu a bazą danych, gdyż ułatwia wyznaczenie zakresu zadań jakie muszą być wykonane aby zbudować kompletny system. Framework Hibernate pozwala na wygodne mapowanie obiektów w języku Java na wiersze bazodanowe. Takie rozwiązanie zapewnia wygodniejszy sposób zarządzania danymi, oraz nie nakłada na użytkownika obowiązku znajomości struktury języka SQL.