

VALATRON – projekt test

ZADÁNÍ

Bylo zadáno zpracování, návrh databázové struktury a implementace rozhraní pro registraci zákazníků s přiřazením čísla věrnostní karty, implementace vyhledávání zákazníka dle jména, příjmení a čísla věrnostní karty a dále vytvoření reportů z informací z databázové struktury. Zadání je definováno v samostatném dokumentu.

ZÁKLADNÍ NÁVRH

ROZVRŽENÍ APLIKACE A POUŽITÍ TECHNOLOGIÍ

- Databázová struktura: MySQL verze 5.7
- Implementační jazyk aplikace: PHP 7.2
- Framework: Symfony 4.3.4, pro přístup k databázi použití ORM-Doctrine

Oproti zadání, byl zvolen framework Symfony 4.3.x z důvodu možné budoucí rozšiřitelnosti aplikace, především využití balíčků pro přístup do databáze ORM/Doctrine, zpracování formulářů Form adp.

NÁVRH OOP

Byly navrženy tyto ORM entity:

- Customer – objekt s daty o zákazníkovi v systému
- Card – objekt s daty karet v systému
- Purchase – objekt s daty objednávek
- Good – objekt s daty zakoupeného zboží

Dále byla vytvořena třída modelu dat App\Classes\Report která poskytuje pro controller data pro zobrazení reportu.

Relace byly implementovány takto:

Card – Customer, relace M:N, Card v tabulce card_customer je unique (není možné přiřadit jednu kartu více zákazníkům)

Card – Purchase, relace 1:N

Purchase – Good, relace 1:N

ÚLOŽIŠTĚ DAT

Bylo implementováno úložiště dat MySQL 5.7 (možno použít i verzi 8). Struktura databáze je znázorněna v příloze 1 – Diagram databáze

Oproti základním datovým tabulkám byly vytvořeny dvě funkce pro získání údajů o cenách objednávek a jeden pohled reporttopcustomers, který zobrazuje nejlepších 10 zákazníků za posledních 30 dnů.

UŽIVATELSKÉ ROZHRANÍ

Dvě třídy CustomerFormType a FindCustomerFormType definují formuláře pro vstup uživatelských dat.

CONTROLLERY

Pro jednoduchost a názornost byl vytvořen jeden controller (App\Controller\IndexController) který zajišťuje všechny požadavky. Routové cesty byly pro názornost vypsány k jednotlivým obsluhujícím metodám jako anotace.

Tento controller (App\Controller\IndexController) má tyto obsluhující metody

registrationCustomer() : Response

Metoda zobrazí registrační formulář pro zákazníka. Formulář obsahuje všechny požadované údaje, jako jméno, příjmení, adresa, email, telefon a číslo věrnostní karty. Po odeslání formuláře je zpracování formuláře předáno metodě saveCustomer.

saveCustomer(Request \$request): Response

Je provedena kontrola existence věrnostní karty v tabulce card. Pokud je karta existující, dojde k zjištění, zda registrovaný zákazník již v systému existuje. Pokud ano, je karta přiřazena k existujícímu zákazníkovi, jinak je vytvořen nový záznam v tabulce customer.

findCustomerForm(Request \$request): Response

Funkce zobrazí formulář pro vyhledání zákazníka na základě zadaných parametrů jméno, příjmení a nebo číslo věrnostní karty.

reportCustomer(): Response

Funkce zobrazí požadovaný report a to počet zákazníků a tabulku top 10 zákazníků s největším obratem.

OSTATNÍ INFORMACE

Žádné další implementace nebyly řešeny, neboť nebyly v zadání požadovány.

NÁVRH ZPRACOVATELE ÚKOLU

U aplikace navrhuji vytvořit možnost registrace zákazníka se zadáním login a hesla. Po přihlášení tak bude mít přehled o svých věrnostních kartách. Nebo tyto data poskytnout na základě požadavku například přes rozhraní SOAP.

Aplikace není nijak v rámci php chráněna proti neoprávněnému zneužití dat, proto navrhuji zabezpečit přístup v případě produkčního nasazení.

příloha č. 1 - Diagram datového modelu databáze

Návrh relační databáze MySQL

Předpokládaná verze databáze: MySQL 8.0

Datum vytvoření: 23.9.2019

Navrhl a zpracoval: Zajíc Jan, zajic@vejhybka.cz, +420 721 733 625

