

Projekt aplikacji opartej o relacyjną bazę danych

<tu wpisz tytuł>

Jarosław Nigiel, Adam Friedensberg

semestr zimowy 2016/2017

Spis treści

1	Analiza problemu/zagadnienia	1
2	Wymagania funkcjonalne	1
3	Wymagania niefunkcjonalne	2
4	Projekt bazy danych (diagram UML)	3
5	Legenda do diagramu UML	3
6	Scenariusze użycia	5

1 Analiza problemu/zagadnienia

Zaprezentowana aplikacja ma za zadanie pomóc użytkownikowi w kontrolowaniu miesięcznych wydatków i przy ustalonym budżecie liczyć, ile pieniędzy pozostało jeszcze dostępnych do końca ustalonego miesiąca rozliczeniowego. Aplikacja nie ma na celu ograniczać użytkownika, lecz stanowić pomoc informacyjną i działać jako historia dokonanych płatności.

2 Wymagania funkcjonalne

Przedstawiona poniżej lista określa najważniejsze możliwości funkcjonalne aplikacji:

- Możliwość zalogowania się, bądź utworzenia nowego konta użytkownika
- Zdarność do usunięcia konta z poziomu administratora
- Istnienie trzech rodzajów użytkowników: Podstawowych, Premium, Administratorów
- Ustalenie miesięcznego budżetu do rozdysponowania przez użytkownika
- Wprowadzenie do bazy rekordów o dokonanych płatnościach wraz z ich kosztem (zakupy w sklepie)
- Śledzenie historii dokonanych transakcji z dokładną datą wprowadzenia rekordów
- Ustalanie aktualnego budżetu na podstawie płatności z bazy
- Możliwość sortowania rekordów, w celu wyszukania ceny i ilości zakupionych dóbr, usług
- Zdarność do uwzględnienia w rozliczanym budżecie płatności stałych z ustawioną datą realizacji

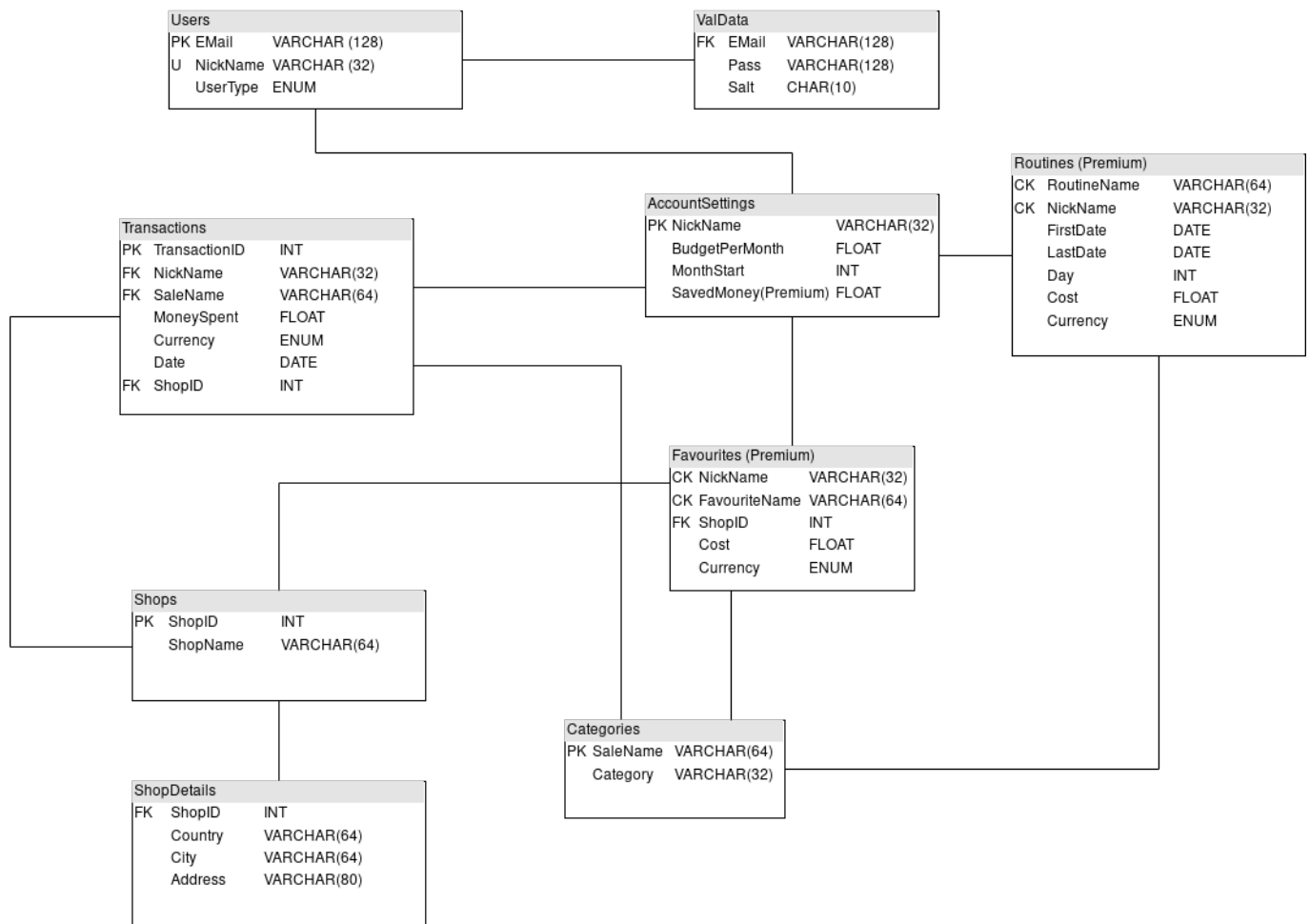
- Ustawianie początku i końca miesiąca rozliczeniowego w aplikacji
- Włączenie przez użytkownika opcji liczenia oszczędności, jako pozostałości niewydanego budżetu z poprzednich miesięcy i przeznaczenie ich do konkretnego celu
- Możliwość dodania do bazy danych konkretnego sklepu, bądź produktu w celu ich wyszukania w przyszłości bądź wybrania z dostępnej listy przy wprowadzaniu rekordów o transakcjach
- Wykorzystanie z bazy danych unikalnej dla użytkownika listy produktów, usług oraz lokalizacji ich nabycia w celu skrócenia czasu wprowadzania rekordów o płatnościach przez klienta
- Umożliwienie użytkownikowi opcji "błyskawicznego" dodania rekordu o płatnościach ustawionych jako ulubione, w przypadku czynności wykonywanych z dużą częstotliwością, np. poranna kawa w kawiarni blisko uczelni, miejsca pracy, etc.

3 Wymagania niefunkcjonalne

Gotowa aplikacja będzie spełniała dodatkowe wymagania podane poniżej:

- Zapewnienie bezpieczeństwa związanego z ochroną danych użytkowników
- Utrzymanie spójności i stabilności podczas korzystania z programu
- Korzystanie z aplikacji poprzez intuicyjny i spełniający swoją rolę interfejs
- Zapewnienie przystępnej dla użytkownika wydajności programu
- Kontrolowanie niezawodności działania aplikacji wraz z pomocą techniczną

4 Projekt bazy danych (diagram UML)



5 Legenda do diagramu UML

PK - Primary Key - Klucz główny

FK - Foreign Key - Klucz obcy

CK - Composite Key - Klucz łączony

U - Unique - Wartość unikalna

Users - Przechowuje dane użytkowników

- EMail - adres e-mail
- NickName - nazwa użytkownika
- UserType - rodzaj użytkownika: Podstawowy, Premium, Admin

ValData - Przechowuje dane uwierzytelnienia użytkowników

- EMail - adres e-mail
- Pass - hash od hasła użytkownika
- Salt - sól dołączana przy hashowaniu hasła

AccountSettings - Przechowuje dane o ustawieniach konta

- NickName - nazwa użytkownika
- BudgetPerMonth - dostępny miesięczny budżet
- MonthStart - dzień miesiąca, rozpoczynający okres rozliczeniowy
- SavedMoney(Premium) - Oszczędności, usługa dostępna przy koncie Premium

Transactions - Przechowuje rekordy transakcji użytkowników

- TransactionID - numer transakcji
- NickName - nazwa użytkownika
- SaleName - nazwa produktu, usługi
- MoneySpent - koszt kupna produktu, usługi
- Currency - waluta
- Date - data transakcji
- ShopID - numer identyfikacyjny sklepu

Routines (Premium) - Przechowuje dane o płatnościach stałych użytkowników Premium

- RoutineName - nazwa płatności stałej
- NickName - nazwa użytkownika
- FirstDate - data ustanowienia płatności stałej
- LastDate - data zakończenia płatności stałej
- Day - dzień miesiąca wykonania płatności stałej
- Cost - koszt płatności
- Currency - waluta

Favourites (Premium) - Przechowuje dane o ulubionych płatnościach użytkowników Premium

- NickName - nazwa użytkownika
- FavouriteName - nazwa płatności
- ShopID - numer identyfikacyjny sklepu
- Cost - koszt płatności
- Currency - waluta

Categories - Przechowuje dane o kategoriach produktów, usług podanych w transakcjach

- SaleName - nazwa płatności
- Category - nazwa kategorii

Shops - Przechowuje podstawowe dane o sklepach i miejscach świadczenia usług

- ShopID - numer identyfikacyjny sklepu
- ShopName - nazwa sklepu

ShopDetails - Przechowuje szczegółowe dane o sklepach i miejscach świadczenia usług

- ShopID - numer identyfikacyjny sklepu
- Country - adres sklepu; kraj
- City - adres sklepu; miasto
- Address - adres sklepu; ulica i numer

6 Scenariusze użycia

Aplikacja ma duże możliwości, natomiast podane poniżej są przykładowe scenariusze jej użycia:

1. Użytkownik zakłada konto, ustala miesięczny budżet do rozdysponowania, następnie w danym miesiącu wprowadza dane o opłaconych usługach.
2. Klient postanawia sprawdzić w jakim sklepie zakupił produkt po najniższej cenie, kiedy transakcja odbyła się ponad miesiąc wcześniej. Sprawdza historię płatności, ogranicza wyniki do produktu i sortuje po dacie.
3. Klient zastanawia się ile miesięcznie płaci za poranną kawę, którą pije od czasu do czasu przed rozpoczęciem pracy, więc ustawia tę czynność jako ulubioną i sprawdza sumę kosztów pod koniec miesiąca.
4. Student chce oszczędzić pewną sumę z dostępnego dla niego budżetu, a po paru miesiącach, wydać ją na zakup dobrej książki, więc włącza opcję liczenia oszczędności w aplikacji.
5. Użytkownik włącza do miesięcznych rachunków stałą płatność za Internet, abonament telefoniczny etc. W konkretnym ustalonym dniu, system odejmuje od pozostałego budżetu koszt opłacenia usługi.