Návrh webové aplikace – Internetové bankovnictví

Obsah

[Funkční požadavky 1](#_Toc212296846)

[Entity 1](#_Toc212296847)

[Uživatel 1](#_Toc212296848)

[Účet 2](#_Toc212296849)

[Karty 2](#_Toc212296850)

[Transakce 3](#_Toc212296851)

# Funkční požadavky

Aplikace bude podporovat registraci a přihlašování uživatele. Vytváření bankovních účtů, transakcí, karet uživatelů a zobrazování historie transakcí. Uživatel bude mít možnost zobrazit své bankovní účty. V každém bankovním účtu bude mít veden přehled transakcí, bankovních karet přiřazených k účtu. Bude mít možnost skrze bankovní účet provádět platby a přijímat platby. Každý bankovní účet bude přiřazen ke specifické bance.

# Entity

Tabulky, které budou v databázi a jejich parametry.

## Uživatel

Uživatel má bankovní účty, transakce a historii přihlášení. Nazvána User.

public class User

{

public int Id { get; set; }

public string Username { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string Tel { get; set; }

public string PasswordHash { get; set; }

public DateTime DateOfBirth { get; set; }

public DateTime CreatedAt { get; set; } = DateTime.UtcNow;

public DateTime? UpdatedAt { get; set; }

public UserType UserType { get; set; } = UserType.USER;

// Relationships

public IList<Account> Accounts { get; set; } = new List<Account>();

[NotMapped]

public string FullName => $"{FirstName} {LastName}";

}

## Účet

Obsahuje relaci k uživateli. Vazba bude typu 1:N, kdy jeden uživatel bude mít N bankovních účtů. Nazvána Account.

public class Account

{

public int Id { get; set; }

// Account balance

public decimal Balance { get; set; }

// Account details

public string AccountNumber { get; set; }

public string BankCode { get; set; }

public BankAccountType Type { get; set; }

public DateTime CreatedAt { get; set; } = DateTime.UtcNow;

// Relationships

public int UserId { get; set; }

public User User { get; set; }

public IList<Card> Cards { get; set; } = new List<Card>();

public IList<Transaction> Transactions { get; set; } = new List<Transaction>();

}

## Karty

Obsahuje kreditní/debetní karty přiřazené k bankovnímu účtu. Vazba bude 1:N, kdy jeden účet může mít více karet. Nazvána Card.

public class Card

{

public int Id { get; set; }

// Card details

[MaxLength(16)]

public string CardNumber { get; set; }

public DateTime ExpirationDate { get; set; }

[MaxLength(3)]

public string SecurityCode { get; set; }

public CardType Type { get; set; }

public string CardHolderName { get; set; }

public bool IsBlocked { get; set; } = false;

public DateTime CreatedAt { get; set; } = DateTime.UtcNow;

// Relationships

public int AccountId { get; set; }

public Account Account { get; set; }

}

## Transakce

Obsahuje relaci k bankovním účtům. Vazba bude typu 1:N, kdy jeden bankovní účet bude mít N transakcí. Nazvána Transaction.

public class Transaction

{

public int Id { get; set; }

// Accounts details

public string FromAccountNumber { get; set; }

public string FromBankCode { get; set; }

public string ToAccountNumber { get; set; }

public string ToBankCode { get; set; }

// Symbols

[MaxLength(10)]

public string ConstantSymbol { get; set; }

[MaxLength(10)]

public string VariableSymbol { get; set; }

// Transaction details

public decimal Amount { get; set; }

public DateTime CreatedAt { get; set; } = DateTime.UtcNow;

public TransactionType TransactionType { get; set; }

public TransactionStatus Status { get; set; }

// Relationship

public int AccountId { get; set; }

public Account Account { get; set; }

}