uid: index title: ATM Application Documentation

ATM Application

This project simulates a basic ATM system with functionality for users and admins.

Features

- User registration and login
- Deposit and withdrawal
- View transaction history
- Admin controls (add/delete users)

Project Structure

- **ATMApp/** – contains the source code
- **ATMApp.Tests/** – unit tests
- **Controllers/** – API endpoints
- **Services/** – business logic

Getting Started

To run this app:

```
dotnet build
dotnet run --project ATMApp
```

Namespace ATMApp.DTOs

Classes

BalanceResultDto

BaseDto

CreateAccountDto

CreateUserDto

UpdateUserDto

UserLoginDTO

Class BalanceResultDto

Namespace: ATMApp.DTOs

Assembly: ATMApp.dll

```
public class BalanceResultDto
```

Inheritance

object ← BalanceResultDto

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

AccountId

```
public int? AccountId { get; set; }
```

Property Value

int?

Balance

```
public decimal? Balance { get; set; }
```

Property Value

decimal?

Message

```
public string Message { get; set; }
```

Property Value

string

Success

```
public bool Success { get; set; }
```

Property Value

bool

Class BaseDto

Namespace: ATMApp.DTOs

Assembly: ATMApp.dll

```
public class BaseDto
```

Inheritance

object ← BaseDto

Derived

CreateUserDto, UpdateUserDto

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

HolderName

```
public string HolderName { get; set; }
```

Property Value

string

Login

```
public string Login { get; set; }
```

Property Value

string

PinCode

```
public string PinCode { get; set; }
```

Property Value

string

Role

```
public UserRole Role { get; set; }
```

Property Value

UserRole

Class CreateAccountDto

Namespace: ATMApp.DTOs

Assembly: ATMApp.dll

```
public class CreateAccountDto
```

Inheritance

object ← CreateAccountDto

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

ClientID

```
public int ClientID { get; set; }
```

Property Value

int

IntialBalance

```
public decimal IntialBalance { get; set; }
```

Property Value

decimal

Status

```
public AccountStatus Status { get; set; }
```

Property Value

AccountStatus

Class CreateUserDto

Namespace: ATMApp.DTOs

Assembly: ATMApp.dll

```
public class CreateUserDto : BaseDto
```

Inheritance

object ← BaseDto ← CreateUserDto

Inherited Members

BaseDto.Login , BaseDto.PinCode , BaseDto.HolderName , BaseDto.Role , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

IntialBalance

```
public decimal IntialBalance { get; set; }
```

Property Value

decimal

Status

```
public AccountStatus Status { get; set; }
```

Property Value

AccountStatus

Class UpdateUserDto

Namespace: ATMApp.DTOs

Assembly: ATMApp.dll

```
public class UpdateUserDto : BaseDto
```

Inheritance

object ← BaseDto ← UpdateUserDto

Inherited Members

BaseDto.Role , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

HolderName

```
public string? HolderName { get; set; }
```

Property Value

string

Id

```
public int Id { get; set; }
```

Property Value

int

Login

```
public string? Login { get; set; }
```

Property Value

string

PinCode

```
public string? PinCode { get; set; }
```

Property Value

string

Status

```
public AccountStatus? Status { get; set; }
```

Property Value

AccountStatus?

Class UserLoginDTO

Namespace: ATMApp.DTOs

Assembly: ATMApp.dll

```
public class UserLoginDTO
```

Inheritance

object ← UserLoginDTO

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Login

```
public string Login { get; set; }
```

Property Value

string

PinCode

```
public string PinCode { get; set; }
```

Property Value

string

Namespace ATMApp.Data

Classes

ATMContext

ATMContextFactory

Class ATMContext

Namespace: ATMApp.Data

Assembly: ATMApp.dll

```
public class ATMContext : DbContext, IInfrastructure<IServiceProvider>,
IDbContextDependencies, IDbSetCache, IDbContextPoolable, IResettableService,
IDisposable, IAsyncDisposable
```

Inheritance

object ← DbContext ← ATMContext

Implements

IInfrastructure<IServiceProvider>, IDbContextDependencies, IDbSetCache, IDbContextPoolable, IResettableService, IDisposable, IAsyncDisposable

Inherited Members

DbContext.Set<TEntity>() , DbContext.Set<TEntity>(string) ,
DbContext.OnConfiguring(DbContextOptionsBuilder) ,
DbContext.ConfigureConventions(ModelConfigurationBuilder) , DbContext.SaveChanges() ,
DbContext.SaveChanges(bool) , DbContext.SaveChangesAsync(CancellationToken) ,
DbContext.SaveChangesAsync(bool, CancellationToken) , DbContext.Dispose() ,
DbContext.DisposeAsync() , DbContext.Entry<TEntity>(TEntity) , DbContext.Entry(object) ,
DbContext.Add<TEntity>(TEntity) , DbContext.AddAsync<TEntity>(TEntity, CancellationToken) ,
DbContext.Attach<TEntity>(TEntity) , DbContext.Update<TEntity>(TEntity) ,
DbContext.Remove<TEntity>(TEntity) , DbContext.Add(object) ,
DbContext.AddAsync(object, CancellationToken) , DbContext.Attach(object) ,
DbContext.Update(object) , DbContext.Remove(object) , DbContext.AddRange(params object[]) ,
DbContext.AddRangeAsync(params object[]) , DbContext.AttachRange(params object[]) ,
DbContext.UpdateRange(params object[]) , DbContext.RemoveRange(params object[]) ,
DbContext.AddRange(IEnumerable<object>) ,
DbContext.AddRangeAsync(IEnumerable<object>, CancellationToken) ,
DbContext.AttachRange(IEnumerable<object>) , DbContext.UpdateRange(IEnumerable<object>) ,
DbContext.RemoveRange(IEnumerable<object>) , DbContext.Find(Type, params object[]) ,
DbContext.FindAsync(Type, params object[]) ,
DbContext.FindAsync(Type, object[], CancellationToken) , DbContext.Find<TEntity>(params object[]) ,
DbContext.FindAsync<TEntity>(params object[]) ,
DbContext.FindAsync<TEntity>(object[], CancellationToken) ,
DbContext.FromExpression<TResult>(Expression<Func<IQueryable<TResult>>>) ,

DbContext.Database , DbContext.ChangeTracker , DbContext.Model , DbContext.ContextId , DbContext.SavingChanges , DbContext.SavedChanges , DbContext.SaveChangesFailed , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

ATMContext(DbContextOptions<ATMContext>)

```
public ATMContext(DbContextOptions<ATMContext> options)
```

Parameters

`options` DbContextOptions<ATMContext>

Properties

Account

```
public DbSet<Account> Account { get; set; }
```

Property Value

DbSet<Account>

Transactions

```
public DbSet<Transaction> Transactions { get; set; }
```

Property Value

DbSet<Transaction>

User

```
public DbSet<User> User { get; set; }
```

Property Value

DbSet <User>

Methods

OnModelCreating(ModelBuilder)

Override this method to further configure the model that was discovered by convention from the entity types exposed in DbSet<TEntity> properties on your derived context. The resulting model may be cached and re-used for subsequent instances of your derived context.

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

Parameters

`modelBuilder` ModelBuilder

The builder being used to construct the model for this context. Databases (and other extensions) typically define extension methods on this object that allow you to configure aspects of the model that are specific to a given database.

Remarks

If a model is explicitly set on the options for this context (via UseModel(IModel)) then this method will not be run. However, it will still run when creating a compiled model.

See Modeling entity types and relationships for more information and examples.

Class ATMContextFactory

Namespace: ATMApp.Data

Assembly: ATMApp.dll

```
public class ATMContextFactory : IDesignTimeDbContextFactory<ATMContext>
```

Inheritance

object ← ATMContextFactory

Implements

IDesignTimeDbContextFactory<ATMContext>

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

CreateDbContext(string[])

Creates a new instance of a derived context.

```
public ATMContext CreateDbContext(string[] args)
```

Parameters

`args` string[]

Arguments provided by the design-time service.

Returns

ATMContext

An instance of ATMContext.

Namespace ATMApp.Interfaces

Interfaces

IAccountRepository

IAdminservices

IAuthService

IClientService

ITransactionRepository

Iuser

Interface IAccountRepository

Namespace: ATMApp.Interfaces

Assembly: ATMApp.dll

```
public interface IAccountRepository
```

Methods

CreateAccount(Account)

```
Task<Account> CreateAccount(Account account)
```

Parameters

`account` Account

Returns

Task<Account>

DeleteAccountById(int)

```
Task<bool> DeleteAccountById(int accountId)
```

Parameters

`accountId` int

Returns

Task<bool>

GetAccountByClientID(int)

```
Task<Account> GetAccountByClientID(int clientId)
```

Parameters

`clientId` int

Returns

Task<Account>

GetAccountById(int)

```
Task<Account> GetAccountById(int id)
```

Parameters

`id` int

Returns

Task<Account>

UpdateAccount(Account)

```
Task<Account> UpdateAccount(Account account)
```

Parameters

`account` Account

Returns

Task<Account>

Interface IAdminservices

Namespace: ATMApp.Interfaces

Assembly: ATMApp.dll

```
public interface IAdminservices
```

Methods

AddUser(CreateUserDto)

```
Task<bool> AddUser(CreateUserDto createUserDto)
```

Parameters

`createUserDto` CreateUserDto

Returns

Task<bool>

DeleteUserAndAccount(int, string)

```
Task<bool> DeleteUserAndAccount(int userId, string confirmationInput)
```

Parameters

`userId` int

`confirmationInput` string

Returns

Task<bool>

GetAccount(int)

```
Task<Account> GetAccount(int id)
```

Parameters

`id` int

Returns

Task<Account>

GetUserByLogin(string)

```
Task<User> GetUserByLogin(string login)
```

Parameters

`login` string

Returns

Task<User>

UpdateUser(UpdateUserDto)

```
Task<bool> UpdateUser(UpdateUserDto updateUserDto)
```

Parameters

`updateUserDto` UpdateUserDto

Returns

Task<bool>

Interface IAuthService

Namespace: ATMApp.Interfaces

Assembly: ATMApp.dll

```
public interface IAuthService
```

Methods

Exit()

```
void Exit()
```

Login(UserLoginDTO)

```
Task<User> Login(UserLoginDTO userLogin)
```

Parameters

`userLogin` UserLoginDTO

Returns

Task<User>

Interface IClientService

Namespace: ATMApp.Interfaces

Assembly: ATMApp.dll

```
public interface IClientService
```

Methods

Deposit(int, decimal)

```
Task<bool> Deposit(int accountId, decimal amount)
```

Parameters

`accountId` int

`amount` decimal

Returns

Task<bool>

GetBalance(int)

```
Task GetBalance(int accountId)
```

Parameters

`accountId` int

Returns

Task

GetTransactionHistory(int)

```
Task<List<Transaction>> GetTransactionHistory(int accountId)
```

Parameters

`accountId` int

Returns

Task<List<Transaction>>

Withdraw(int, decimal)

```
Task<bool> Withdraw(int clientID, decimal amount)
```

Parameters

`clientID` int

`amount` decimal

Returns

Task<bool>

Interface ITransactionRepository

Namespace: ATMApp.Interfaces

Assembly: ATMApp.dll

```
public interface ITransactionRepository
```

Methods

AddTransaction(Transaction)

```
Task AddTransaction(Transaction transaction)
```

Parameters

`transaction` Transaction

Returns

Task

GetTransactionsByAccountId(int)

```
Task<List<Transaction>> GetTransactionsByAccountId(int accountId)
```

Parameters

`accountId` int

Returns

Task<List<Transaction>>

Interface Iuser

Namespace: ATMApp.Interfaces

Assembly: ATMApp.dll

```
public interface Iuser
```

Methods

AddUser(User)

```
void AddUser(User user)
```

Parameters

`user` User

GetAllPeople()

```
List<User> GetAllPeople()
```

Returns

List<User>

Namespace ATMApp.Migrations

Classes

CreateTablesAndRelations

A base class inherited by each EF Core migration.

Class CreateTablesAndRelations

Namespace: ATMApp.Migrations

Assembly: ATMApp.dll

A base class inherited by each EF Core migration.

```
[DbContext(typeof(ATMContext))]
[Migration("20250318031640_createTablesAndRelations")]
public class CreateTablesAndRelations : Migration
```

Inheritance

object ← Migration ← CreateTablesAndRelations

Inherited Members

Migration.InitialDatabase , Migration.TargetModel , Migration.UpOperations , Migration.DownOperations , Migration.ActiveProvider , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Remarks

See Database migrations for more information and examples.

Methods

BuildTargetModel(ModelBuilder)

Implemented to build the TargetModel.

```
protected override void BuildTargetModel(ModelBuilder modelBuilder)
```

Parameters

`modelBuilder` ModelBuilder

The ModelBuilder to use to build the model.

Remarks

See Database migrations for more information and examples.

Down(MigrationBuilder)

Builds the operations that will migrate the database 'down'.

```
protected override void Down(MigrationBuilder migrationBuilder)
```

Parameters

`migrationBuilder` MigrationBuilder

The MigrationBuilder that will build the operations.

Remarks

That is, builds the operations that will take the database from the state left in by this migration so that it returns to the state that it was in before this migration was applied.

This method must be overridden in each class that inherits from Migration if both 'up' and 'down' migrations are to be supported. If it is not overridden, then calling it will throw and it will not be possible to migrate in the 'down' direction.

See Database migrations for more information and examples.

Up(MigrationBuilder)

Builds the operations that will migrate the database 'up'.

```
protected override void Up(MigrationBuilder migrationBuilder)
```

Parameters

`migrationBuilder` MigrationBuilder

The MigrationBuilder that will build the operations.

Remarks

That is, builds the operations that will take the database from the state left in by the previous migration so that it is up-to-date with regard to this migration.

This method must be overridden in each class that inherits from Migration.

See Database migrations for more information and examples.

Namespace ATMApp.Models

Classes

Account

Transaction

User

Enums

AccountStatus

TransactionType

UserRole

Class Account

Namespace: ATMApp.Models

Assembly: ATMApp.dll

```
public class Account
```

Inheritance

object ← Account

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

ClientID

```
public int ClientID { get; set; }
```

Property Value

int

Id

```
public int Id { get; set; }
```

Property Value

int

IntialBalance

```
public decimal IntialBalance { get; set; }
```

Property Value

decimal

Status

```
public AccountStatus Status { get; set; }
```

Property Value

AccountStatus

Transactions

```
public List<Transaction> Transactions { get; set; }
```

Property Value

List<Transaction>

User

```
public User User { get; set; }
```

Property Value

User

Enum AccountStatus

Namespace: ATMApp.Models

Assembly: ATMApp.dll

```
public enum AccountStatus
```

Fields

`Active = 0`

`Disabled = 1`

Class Transaction

Namespace: ATMApp.Models

Assembly: ATMApp.dll

```
public class Transaction
```

Inheritance

object ← Transaction

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Account

```
public Account Account { get; set; }
```

Property Value

Account

AccountId

```
public int AccountId { get; set; }
```

Property Value

int

Amount

```
public decimal Amount { get; set; }
```

Property Value

decimal

Id

```
public int Id { get; set; }
```

Property Value

int

TimeStamp

```
public DateTime TimeStamp { get; set; }
```

Property Value

DateTime

Type

```
public TransactionType Type { get; set; }
```

Property Value

TransactionType

Enum TransactionType

Namespace: ATMApp.Models

Assembly: ATMApp.dll

```
public enum TransactionType
```

Fields

`Deposit = 0`

`Display = 2`

`Withdrawal = 1`

Class User

Namespace: ATMApp.Models

Assembly: ATMApp.dll

```
public class User
```

Inheritance

object ← User

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Account

```
public Account? Account { get; set; }
```

Property Value

Account

HolderName

```
public string HolderName { get; set; }
```

Property Value

string

Id

```
public int Id { get; set; }
```

Property Value

int

Login

```
public string Login { get; set; }
```

Property Value

string

PinCode

```
public string PinCode { get; set; }
```

Property Value

string

Role

```
public UserRole Role { get; set; }
```

Property Value

UserRole

Enum UserRole

Namespace: ATMApp.Models

Assembly: ATMApp.dll

```
public enum UserRole
```

Fields

`Admin = 1`

`Client = 0`

Namespace ATMApp.Repositories

Classes

AccountRepository

TransactionRepository

UserRepository

Interfaces

IUserRepository

Class AccountRepository

Namespace: ATMApp.Repositories

Assembly: ATMApp.dll

```
public class AccountRepository : IAccountRepository
```

Inheritance

object ← AccountRepository

Implements

IAccountRepository

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

AccountRepository(ATMContext)

```
public AccountRepository(ATMContext context)
```

Parameters

`context` ATMContext

Methods

CreateAccount(Account)

This methode will create an Account in Account Repo

```
public Task<Account> CreateAccount(Account newAccount)
```

Parameters

`newAccount` Account

Returns

Task<Account>

DeleteAccountById(int)

This methode will delete an Account by accountId in Account Repo

```
public Task<bool> DeleteAccountById(int accountId)
```

Parameters

`accountId` int

Returns

Task<bool>

GetAccountByClientID(int)

This methode will return an Account by clientId in Account Repo

```
public Task<Account> GetAccountByClientID(int clientId)
```

Parameters

`clientId` int

Returns

Task<Account>

GetAccountById(int)

This methode will return an Account by accountId in Account Repo

```
public Task<Account> GetAccountById(int id)
```

Parameters

`id` int

Returns

Task<Account>

UpdateAccount(Account)

This methode will update an Account by accountId in Account Repo

```
public Task<Account> UpdateAccount(Account account)
```

Parameters

`account` Account

Returns

Task<Account>

Interface IUserRepository

Namespace: ATMApp.Repositories

Assembly: ATMApp.dll

```
public interface IUserRepository
```

Methods

AddUser(User)

```
Task<User> AddUser(User newUser)
```

Parameters

`newUser` User

Returns

Task<User>

DeleteUserbyId(int)

```
Task<bool> DeleteUserbyId(int userId)
```

Parameters

`userId` int

Returns

Task<bool>

GetUserById(int)

```
Task<User> GetUserById(int id)
```

Parameters

`id` int

Returns

Task <User>

GetUserBylogin(string)

```
Task<User> GetUserBylogin(string login)
```

Parameters

`login` string

Returns

Task <User>

GetUserWithAccountByLogin(string)

```
Task<User?> GetUserWithAccountByLogin(string login)
```

Parameters

`login` string

Returns

Task <User>

UpdateUser(User)

```
Task<bool> UpdateUser(User user)
```

Parameters

`user` User

Returns

Task <bool>

Class TransactionRepository

Namespace: ATMApp.Repositories

Assembly: ATMApp.dll

```
public class TransactionRepository : ITransactionRepository
```

Inheritance

object ← TransactionRepository

Implements

ITransactionRepository

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

TransactionRepository(ATMContext)

```
public TransactionRepository(ATMContext context)
```

Parameters

`context` ATMContext

Methods

AddTransaction(Transaction)

This methode will add a transaction to an Account when ever the client draw or deposite money.

```
public Task AddTransaction(Transaction transaction)
```

Parameters

`transaction` Transaction

Returns

Task

GetTransactionsByAccountId(int)

```
public Task<List<Transaction>> GetTransactionsByAccountId(int accountId)
```

Parameters

`accountId` int

Returns

Task<List<Transaction>>

Class UserRepository

Namespace: ATMApp.Repositories

Assembly: ATMApp.dll

```
public class UserRepository : IUserRepository
```

Inheritance

object ← UserRepository

Implements

IUserRepository

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

UserRepository(ATMContext)

```
public UserRepository(ATMContext context)
```

Parameters

`context` ATMContext

Methods

AddUser(User)

This methode will return a User after we add the user to the database

```
public Task<User> AddUser(User user)
```

Parameters

`user` User

Returns

Task<User>

DeleteUserbyId(int)

This methode will return a bool after we delete the user by Id

```
public Task<bool> DeleteUserbyId(int id)
```

Parameters

`id` int

Returns

Task<bool>

GetUserById(int)

This methode will return a User by Id

```
public Task<User> GetUserById(int id)
```

Parameters

`id` int

Returns

Task<User>

GetUserBylogin(string)

This methode will return a User by login

```
public Task<User> GetUserBylogin(string login)
```

Parameters

`login` string

Returns

Task<User>

GetUserWithAccountByLogin(string)

This methode will return a User with their account by Login

```
public Task<User?> GetUserWithAccountByLogin(string login)
```

Parameters

`login` string

Returns

Task<User>

UpdateUser(User)

This methode will return a bool after updating a user

```
public Task<bool> UpdateUser(User user)
```

Parameters

`user` User

Returns

Task<bool>

Namespace ATMApp.Services

Classes

AdminServices

AuthService

ClientService

Class AdminServices

Namespace: ATMApp.Services

Assembly: ATMApp.dll

```
public class AdminServices : IAdminservices
```

Inheritance

object ← AdminServices

Implements

IAdminservices

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

AdminServices(ATMContext, IUserRepository, ITransactionRepository, IAccountRepository, IValidator<CreateUserDto>)

```
public AdminServices(ATMContext aTMContext, IUserRepository userRepository,
ITransactionRepository transactionRepository, IAccountRepository accountRepository,
IValidator<CreateUserDto> userValidator)
```

Parameters

`aTMContext` ATMContext

`userRepository` IUserRepository

`transactionRepository` ITransactionRepository

`accountRepository` IAccountRepository

`userValidator` IValidator<CreateUserDto>

Methods

AddUser(CreateUserDto)

This methode will add user in AdminSerivce.cs

```
public Task<bool> AddUser(CreateUserDto userDto)
```

Parameters

`userDto` CreateUserDto

Returns

Task<bool>

DeleteUserAndAccount(int, string)

This methode will Delete User and their account

```
public Task<bool> DeleteUserAndAccount(int accountId, string confirmationInput)
```

Parameters

`accountId` int

`confirmationInput` string

Returns

Task<bool>

GetAccount(int)

This methode will return a User after we add the user to the database

```
public Task<Account> GetAccount(int id)
```

Parameters

`id` int

Returns

Task<Account>

GetUserByLogin(string)

This methode will return a User by Login in AdminService

```
public Task<User> GetUserByLogin(string login)
```

Parameters

`login` string

Returns

Task<User>

UpdateUser(UpdateUserDto)

This methode will a bool after updating a User in AdminService

```
public Task<bool> UpdateUser(UpdateUserDto updateUserDto)
```

Parameters

`updateUserDto` UpdateUserDto

Returns

Task<bool>

Class AuthService

Namespace: ATMApp.Services

Assembly: ATMApp.dll

```
public class AuthService : IAuthService
```

Inheritance

object ← AuthService

Implements

IAuthService

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

AuthService(IUserRepository, IValidator<UserLoginDTO>)

```
public AuthService(IUserRepository userRepository, IValidator<UserLoginDTO> validator)
```

Parameters

`userRepository` IUserRepository

`validator` IValidator<UserLoginDTO>

Methods

Exit()

This methode will close the program

```
public void Exit()
```

Login(UserLoginDTO)

This methode will login our users

```
public Task<User> Login(UserLoginDTO userLogin)
```

Parameters

`userLogin` UserLoginDTO

Returns

Task<User>

Class ClientService

Namespace: ATMApp.Services

Assembly: ATMApp.dll

```
public class ClientService : IClientService
```

Inheritance

object ← ClientService

Implements

IClientService

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

ClientService(IAccountRepository, ITransactionRepository)

```
public ClientService(IAccountRepository accountRepository, ITransactionRepository
transactionRepository)
```

Parameters

`accountRepository` IAccountRepository

`transactionRepository` ITransactionRepository

Methods

Deposit(int, decimal)

This methode will return a bool after a client try to deposite money.

```
public Task<bool> Deposit(int clientId, decimal amount)
```

Parameters

`clientId` int

`amount` decimal

Returns

Task<bool>

GetBalance(int)

This methode will return will display the current balance of the account.

```
public Task GetBalance(int clientId)
```

Parameters

`clientId` int

Returns

Task

GetTransactionHistory(int)

This methode will return a list of transactions done by User using accountId

```
public Task<List<Transaction>> GetTransactionHistory(int accountId)
```

Parameters

`accountId` int

Returns

Task<List<Transaction>>

Withdraw(int, decimal)

This methode will allow the client to withdraw money

```
public Task<bool> Withdraw(int clientID, decimal amount)
```

Parameters

`clientID` int

`amount` decimal

Returns

Task<bool>

Namespace ATMApp.Validators

Classes

AddNewuserValidator

UserLoginValidator

Class AddNewuserValidator

Namespace: ATMApp.Validators

Assembly: ATMApp.dll

```
public class AddNewuserValidator : AbstractValidator<CreateUserDto>,
IValidator<CreateUserDto>, IValidator, IEnumerable<IValidationRule>, IEnumerable
```

Inheritance

object ← AbstractValidator<CreateUserDto> ← AddNewuserValidator

Implements

IValidator<CreateUserDto>, IValidator, IEnumerable <IValidationRule>, IEnumerable

Inherited Members

AbstractValidator<CreateUserDto>.Validate(CreateUserDto) ,
AbstractValidator<CreateUserDto>.ValidateAsync(CreateUserDto, CancellationToken) ,
AbstractValidator<CreateUserDto>.Validate(ValidationContext<CreateUserDto>) ,
AbstractValidator<CreateUserDto>.ValidateAsync(ValidationContext<CreateUserDto>, CancellationToken) ,
AbstractValidator<CreateUserDto>.CreateDescriptor() ,
AbstractValidator<CreateUserDto>.RuleFor<TProperty>(Expression<Func<CreateUserDto, TProperty>>) ,
AbstractValidator<CreateUserDto>.Transform<TProperty, TTransformed>(Expression<Func<CreateUserDto, TProperty>>, Func<TProperty, TTransformed>) ,
AbstractValidator<CreateUserDto>.Transform<TProperty, TTransformed>(Expression<Func<CreateUserDto, TProperty>>, Func<CreateUserDto, TProperty, TTransformed>) ,
AbstractValidator<CreateUserDto>.RuleForEach<TElement>(Expression<Func<CreateUserDto, IEnumerable<TElement>>>) ,
AbstractValidator<CreateUserDto>.TransformForEach<TElement, TTransformed>(Expression<Func<CreateUserDto, IEnumerable<TElement>>>, Func<TElement, TTransformed>) ,
AbstractValidator<CreateUserDto>.TransformForEach<TElement, TTransformed>(Expression<Func<CreateUserDto, IEnumerable<TElement>>>, Func<CreateUserDto, TElement, TTransformed>) ,
AbstractValidator<CreateUserDto>.RuleSet(string, Action) ,
AbstractValidator<CreateUserDto>.When(Func<CreateUserDto, bool>, Action) ,
AbstractValidator<CreateUserDto>.When(Func<CreateUserDto, ValidationContext<CreateUserDto>, bool>, Action) ,
AbstractValidator<CreateUserDto>.Unless(Func<CreateUserDto, bool>, Action) ,

AbstractValidator<CreateUserDto>.Unless(Func<CreateUserDto, ValidationContext<CreateUserDto>, bool>, Action) ,
AbstractValidator<CreateUserDto>.WhenAsync(Func<CreateUserDto, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<CreateUserDto>.WhenAsync(Func<CreateUserDto, ValidationContext<CreateUserDto>, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<CreateUserDto>.UnlessAsync(Func<CreateUserDto, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<CreateUserDto>.UnlessAsync(Func<CreateUserDto, ValidationContext<CreateUserDto>, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<CreateUserDto>.Include(IValidator<CreateUserDto>) ,
AbstractValidator<CreateUserDto>.Include<TValidator>(Func<CreateUserDto, TValidator>) ,
AbstractValidator<CreateUserDto>.GetEnumerator() ,
AbstractValidator<CreateUserDto>.EnsureInstanceNotNull(object) ,
AbstractValidator<CreateUserDto>.PreValidate(ValidationContext<CreateUserDto>, ValidationResult) ,
AbstractValidator<CreateUserDto>.RaiseValidationException(ValidationContext<CreateUserDto>, ValidationResult) ,
AbstractValidator<CreateUserDto>.OnRuleAdded(IValidationRule<CreateUserDto>) ,
AbstractValidator<CreateUserDto>.CascadeMode ,
AbstractValidator<CreateUserDto>.ClassLevelCascadeMode ,
AbstractValidator<CreateUserDto>.RuleLevelCascadeMode , object.Equals(object) ,
object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

AddNewuserValidator()

```
public AddNewuserValidator()
```

Class UserLoginValidator

Namespace: ATMApp.Validators

Assembly: ATMApp.dll

```
public class UserLoginValidator : AbstractValidator<UserLoginDTO>, IValidator<UserLoginDTO>,
IValidator, IEnumerable<IValidationRule>, IEnumerable
```

Inheritance

object ← AbstractValidator<UserLoginDTO> ← UserLoginValidator

Implements

IValidator<UserLoginDTO>, IValidator, IEnumerable <IValidationRule>, IEnumerable

Inherited Members

AbstractValidator<UserLoginDTO>.Validate(UserLoginDTO) ,
AbstractValidator<UserLoginDTO>.ValidateAsync(UserLoginDTO, CancellationToken) ,
AbstractValidator<UserLoginDTO>.Validate(ValidationContext<UserLoginDTO>) ,
AbstractValidator<UserLoginDTO>.ValidateAsync(ValidationContext<UserLoginDTO>, CancellationToken) ,
AbstractValidator<UserLoginDTO>.CreateDescriptor() ,
AbstractValidator<UserLoginDTO>.RuleFor<TProperty>(Expression<Func<UserLoginDTO, TProperty>>) ,
AbstractValidator<UserLoginDTO>.Transform<TProperty, TTransformed>(Expression<Func<UserLoginDTO, TProperty>>, Func<TProperty, TTransformed>) ,
AbstractValidator<UserLoginDTO>.Transform<TProperty, TTransformed>(Expression<Func<UserLoginDTO, TProperty>>, Func<UserLoginDTO, TProperty, TTransformed>) ,
AbstractValidator<UserLoginDTO>.RuleForEach<TElement>(Expression<Func<UserLoginDTO, IEnumerable<TElement>>>) ,
AbstractValidator<UserLoginDTO>.TransformForEach<TElement, TTransformed>(Expression<Func<UserLoginDTO, IEnumerable<TElement>>>, Func<TElement, TTransformed>) ,
AbstractValidator<UserLoginDTO>.TransformForEach<TElement, TTransformed>(Expression<Func<UserLoginDTO, IEnumerable<TElement>>>, Func<UserLoginDTO, TElement, TTransformed>) ,
AbstractValidator<UserLoginDTO>.RuleSet(string, Action) ,
AbstractValidator<UserLoginDTO>.When(Func<UserLoginDTO, bool>, Action) ,
AbstractValidator<UserLoginDTO>.When(Func<UserLoginDTO, ValidationContext<UserLoginDTO>, bool>, Action) ,
AbstractValidator<UserLoginDTO>.Unless(Func<UserLoginDTO, bool>, Action) ,

AbstractValidator<UserLoginDTO>.Unless(Func<UserLoginDTO, ValidationContext<UserLoginDTO>, bool>, Action) ,
AbstractValidator<UserLoginDTO>.WhenAsync(Func<UserLoginDTO, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<UserLoginDTO>.WhenAsync(Func<UserLoginDTO, ValidationContext<UserLoginDTO>, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<UserLoginDTO>.UnlessAsync(Func<UserLoginDTO, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<UserLoginDTO>.UnlessAsync(Func<UserLoginDTO, ValidationContext<UserLoginDTO>, CancellationToken, Task<bool>>, Action) ,
AbstractValidator<UserLoginDTO>.Include(IValidator<UserLoginDTO>) ,
AbstractValidator<UserLoginDTO>.Include<TValidator>(Func<UserLoginDTO, TValidator>) ,
AbstractValidator<UserLoginDTO>.GetEnumerator() ,
AbstractValidator<UserLoginDTO>.EnsureInstanceNotNull(object) ,
AbstractValidator<UserLoginDTO>.PreValidate(ValidationContext<UserLoginDTO>, ValidationResult) ,
AbstractValidator<UserLoginDTO>.RaiseValidationException(ValidationContext<UserLoginDTO>, ValidationResult) ,
AbstractValidator<UserLoginDTO>.OnRuleAdded(IValidationRule<UserLoginDTO>) ,
AbstractValidator<UserLoginDTO>.CascadeMode ,
AbstractValidator<UserLoginDTO>.ClassLevelCascadeMode ,
AbstractValidator<UserLoginDTO>.RuleLevelCascadeMode , object.Equals(object) ,
object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

UserLoginValidator()

```
public UserLoginValidator()
```

Namespace ATMApp.Views

Classes

AdminView

ClientView

Class AdminView

Namespace: ATMApp.Views

Assembly: ATMApp.dll

```
public class AdminView
```

Inheritance

object ← AdminView

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

AdminView(IAdminservices, IAuthService, Func<string>)

```
public AdminView(IAdminservices adminServices, IAuthService authService, Func<string> input)
```

Parameters

`adminServices` IAdminservices

`authService` IAuthService

`input` Func<string>

Methods

AskUserToEdit(string)

```
public static bool AskUserToEdit(string field)
```

Parameters

`field` string

Returns

bool

CreateUser()

Will create a User.

```
public Task<bool> CreateUser()
```

Returns

Task<bool>

DeleteAccount()

```
public Task<bool> DeleteAccount()
```

Returns

Task<bool>

Exit()

Exit the program.

```
public void Exit()
```

HandleCreateUserInput()

```
public static CreateUserDto HandleCreateUserInput()
```

Returns

CreateUserDto

HandleInputToUpudate()

```
public static UpdateUserDto HandleInputToUpudate()
```

Returns

UpdateUserDto

SearchForAccount()

search for account.

```
public Task SearchForAccount()
```

Returns

Task

Show()

```
public Task Show()
```

Returns

Task

UpdateAccount()

```
public Task<bool> UpdateAccount()
```

Returns

Task<bool>

Class ClientView

Namespace: ATMApp.Views

Assembly: ATMApp.dll

```
public class ClientView
```

Inheritance

object ← ClientView

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

ClientView(IClientService, IAuthService, Func<string?>?)

```
public ClientView(IClientService clientService, IAuthService authService, Func<string?>?
readLine = null)
```

Parameters

`clientService` IClientService

`authService` IAuthService

`readLine` Func<string>

Methods

Deposite(User)

```
public Task Deposite(User user)
```

Parameters

`user` User

Returns

Task

DisplayAccount(User)

```
public Task DisplayAccount(User user)
```

Parameters

`user` User

Returns

Task

Exit()

```
public void Exit()
```

Show(User)

```
public Task Show(User user)
```

Parameters

`user` User

Returns

Task

WithdrawMoney(User)

```
public Task WithdrawMoney(User user)
```

Parameters

`user` User

Returns

Task