Тема:

Єдиний національний реєстр підприємств щодо яких порушено провадженння у справі про банкрутство

Визначення, акроніми та скорочення

ППППСБ - підприємства, щодо яких порушено провадження у справі про банкрутство

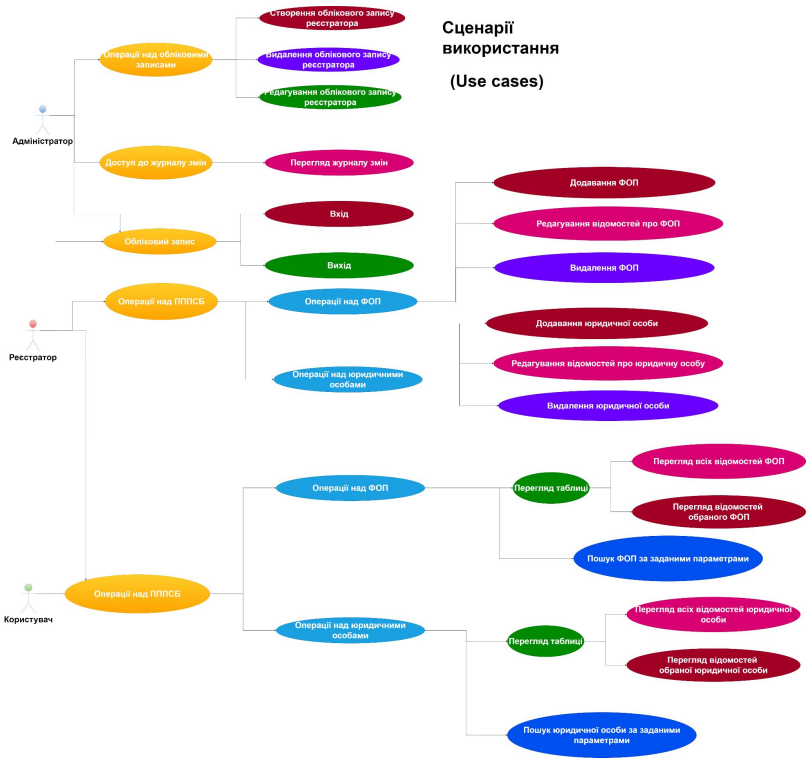
ФОП - фізична особа-підприємець

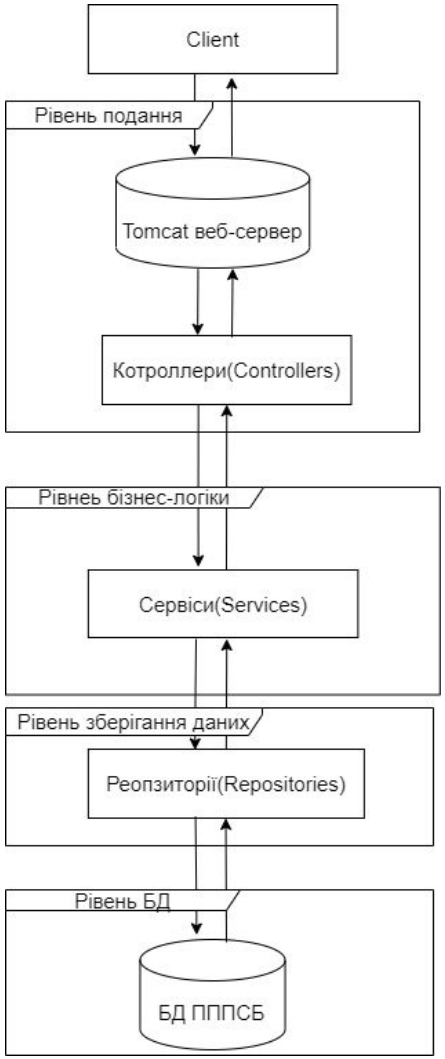
Мета - альтернативне рішення!

https://www.kmu.gov.ua/service/vidomosti-z-reyestru-bankrutstva

Перспектива

Варіанти використання





Опис архітектури

Детальний опис компонентів програмної системи

Діаграми класів

Controllers

PhysicalPersonEntepreneursController

+ PhysicalPersonEntepreneursController(PhysicalPersonEntrepreneursService, PassportSeriesRepo, AuthorizationService):

+ physPersService: PhysicalPersonEntrepreneursService
+ authorizationService: AuthorizationService
+ passportSeriesRepo: PassportSeriesRepo

+ create(PhysicalPersonEntrepreneur): String
+ edit(int, Model): String
+ index(Model): String
+ delete(int, Model): String
+ edit(PhysicalPersonEntrepreneur): String
+ create(Model): String
+ deleteConfirmed(int): String
+ details(int, Model): String

HomeController

+ HomeController(AuthorizationService):

+ successMessageKey: String
+ failMessageKey: String
+ authorizationService: AuthorizationService

+ index(Model): String
+ failed(Model): String
+ success(Model): String

AuthenticationController

+ AuthenticationController(AuthorizationService):

+ currentUserKey: String
+ authorizationService: AuthorizationService

+ SignOut(): String
+ Index(Model): String
+ SignIn(SignIn): String

JuridicalPersonsController

+ JuridicalPersonsController(JuridicalPersonsService, OwnershipTypesRepo, ActivityKindsRepo, AuthorizationService):

+ authorizationService: AuthorizationService
+ jurPersService: JuridicalPersonsService
+ ownershipTypesRepo: OwnershipTypesRepo
+ activityKindsRepo: ActivityKindsRepo

+ edit(JuridicalPerson): String
+ create(Model): String
+ create(JuridicalPerson): String
+ edit(int, Model): String
+ index(Model): String
+ delete(int, Model): String
+ details(int, Model): String
+ deleteConfirmed(int): String

RegistratorsController

+ RegistratorsController(UsersService, RolesRepo, PassportSeriesRepo, AuthorizationService):

+ rolesRepo: RolesRepo
+ usersService: UsersService
+ authorizationService: AuthorizationService
+ passportSeriesRepo: PassportSeriesRepo

+ create(Model): String
+ deleteConfirmed(int): String
+ index(Model): String
+ edit(User): String
+ edit(int, Model): String
+ delete(int, Model): String

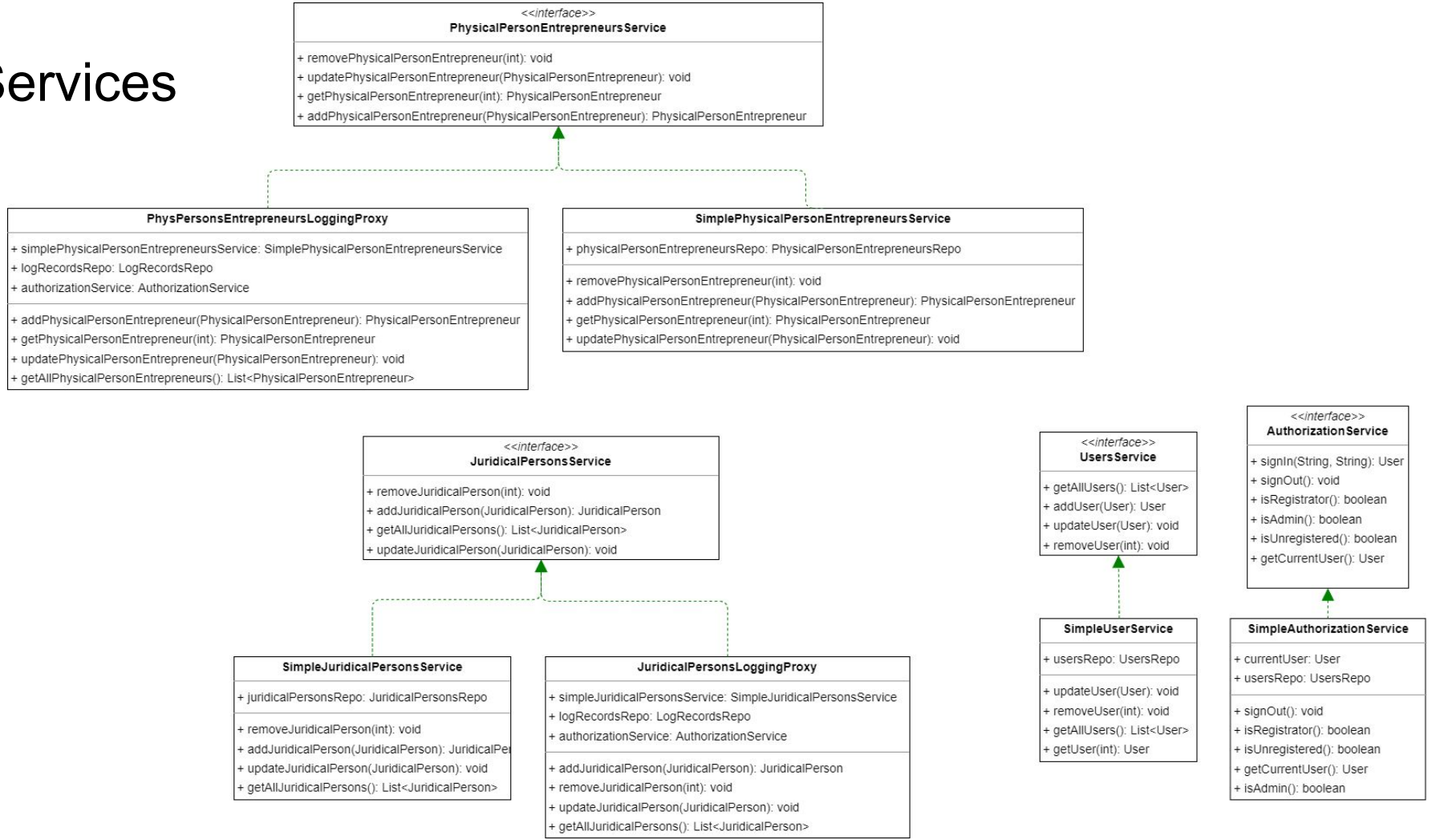
LogsController

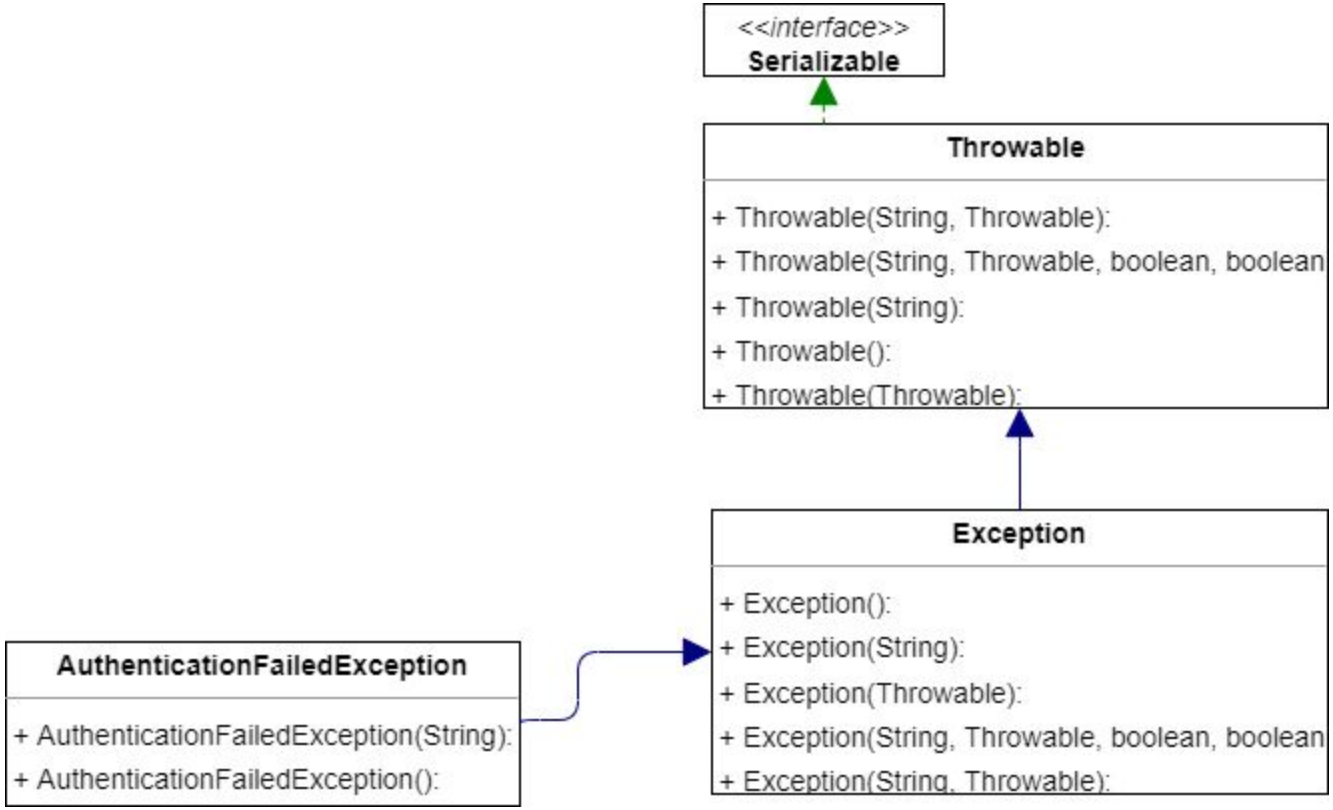
+ LogsController(LogRecordsRepo, AuthorizationService):

+ logRecordsRepo: LogRecordsRepo
+ authorizationService: AuthorizationService

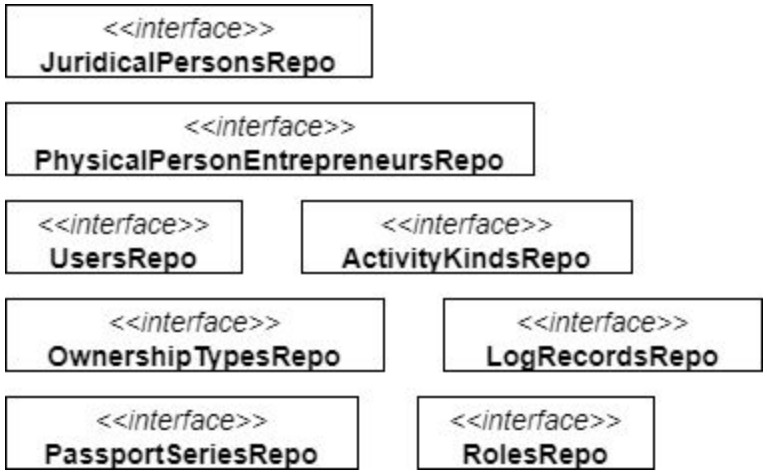
+ index(Model): String

Services

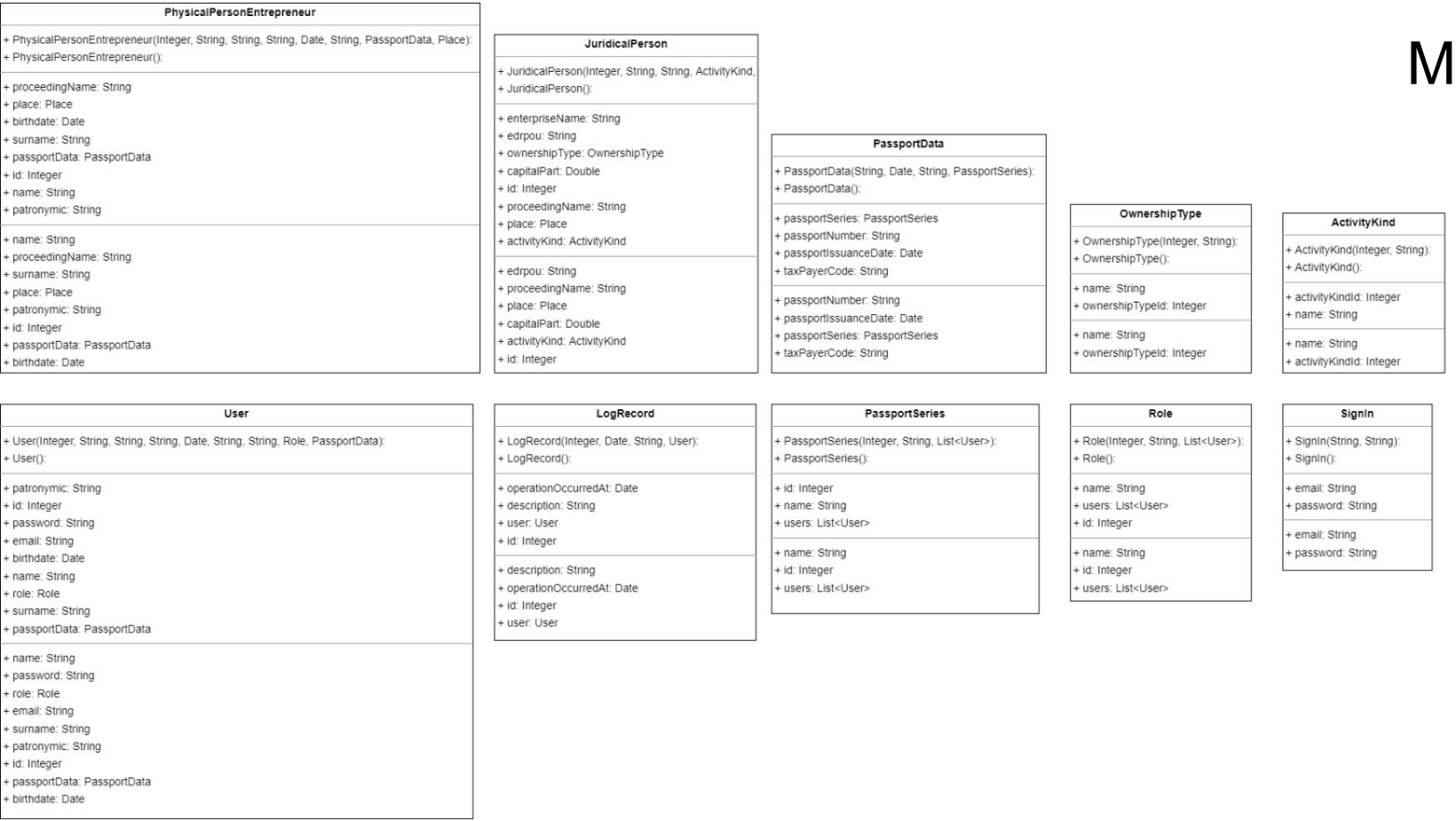

<<interface>>
PhysicalPersonEntrepreneursService
+ removePhysicalPersonEntrepreneur(int): void
+ updatePhysicalPersonEntrepreneur(PhysicalPersonEntrepreneur): void
+ getPhysicalPersonEntrepreneur(int): PhysicalPersonEntrepreneur
+ addPhysicalPersonEntrepreneur(PhysicalPersonEntrepreneur): PhysicalPersonEntrepreneur
PhysPersonsEntrepreneursLoggingProxy
+ simplePhysicalPersonEntrepreneursService: SimplePhysicalPersonEntrepreneursService
+ logRecordsRepo: LogRecordsRepo
+ authorizationService: AuthorizationService
+ addPhysicalPersonEntrepreneur(PhysicalPersonEntrepreneur): PhysicalPersonEntrepreneur
+ getPhysicalPersonEntrepreneur(int): PhysicalPersonEntrepreneur
+ updatePhysicalPersonEntrepreneur(PhysicalPersonEntrepreneur): void
+ getAllPhysicalPersonEntrepreneurs(): List<PhysicalPersonEntrepreneur>
SimplePhysicalPersonEntrepreneursService
+ physicalPersonEntrepreneursRepo: PhysicalPersonEntrepreneursRepo
+ removePhysicalPersonEntrepreneur(int): void
+ addPhysicalPersonEntrepreneur(PhysicalPersonEntrepreneur): PhysicalPersonEntrepreneur
+ getPhysicalPersonEntrepreneur(int): PhysicalPersonEntrepreneur
+ updatePhysicalPersonEntrepreneur(PhysicalPersonEntrepreneur): void
<<interface>>
JuridicalPersonsService
+ removeJuridicalPerson(int): void
+ addJuridicalPerson(JuridicalPerson): JuridicalPerson
+ getAllJuridicalPersons(): List<JuridicalPerson>
+ updateJuridicalPerson(JuridicalPerson): void
SimpleJuridicalPersonsService
+ juridicalPersonsRepo: JuridicalPersonsRepo
+ removeJuridicalPerson(int): void
+ addJuridicalPerson(JuridicalPerson): JuridicalPer
+ updateJuridicalPerson(JuridicalPerson): void
+ getAllJuridicalPersons(): List<JuridicalPerson>
JuridicalPersonsLoggingProxy
+ simpleJuridicalPersonsService: SimpleJuridicalPersonsService
+ logRecordsRepo: LogRecordsRepo
+ authorizationService: AuthorizationService
+ addJuridicalPerson(JuridicalPerson): JuridicalPerson
+ removeJuridicalPerson(int): void
+ updateJuridicalPerson(JuridicalPerson): void
+ getAllJuridicalPersons(): List<JuridicalPerson>
<<interface>>
UsersService
+ getAllUsers(): List<User>
+ addUser(User): User
+ updateUser(User): void
+ removeUser(int): void
SimpleUserService
+ usersRepo: UsersRepo
+ updateUser(User): void
+ removeUser(int): void
+ getAllUsers(): List<User>
+ getUser(int): User
<<interface>>
AuthorizationService
+ signIn(String, String): User
+ signOut(): void
+ isRegistrator(): boolean
+ isAdmin(): boolean
+ isUnregistered(): boolean
+ getCurrentUser(): User
SimpleAuthorizationService
+ currentUser: User
+ usersRepo: UsersRepo
+ signOut(): void
+ isRegistrator(): boolean
+ isUnregistered(): boolean
+ getCurrentUser(): User
+ isAdmin(): boolean


<<interface>>
Serializable
Throwable
+ Throwable(String, Throwable):
+ Throwable(String, Throwable, boolean, boolean
+ Throwable(String):
+ Throwable():
+ Throwable(Throwable):
Exception
+ Exception():
+ Exception(String):
+ Exception(Throwable):
+ Exception(String, Throwable, boolean, boolean
+ Exception(String, Throwable):
AuthenticationFailedException
+ AuthenticationFailedException(String):
+ AuthenticationFailedException():

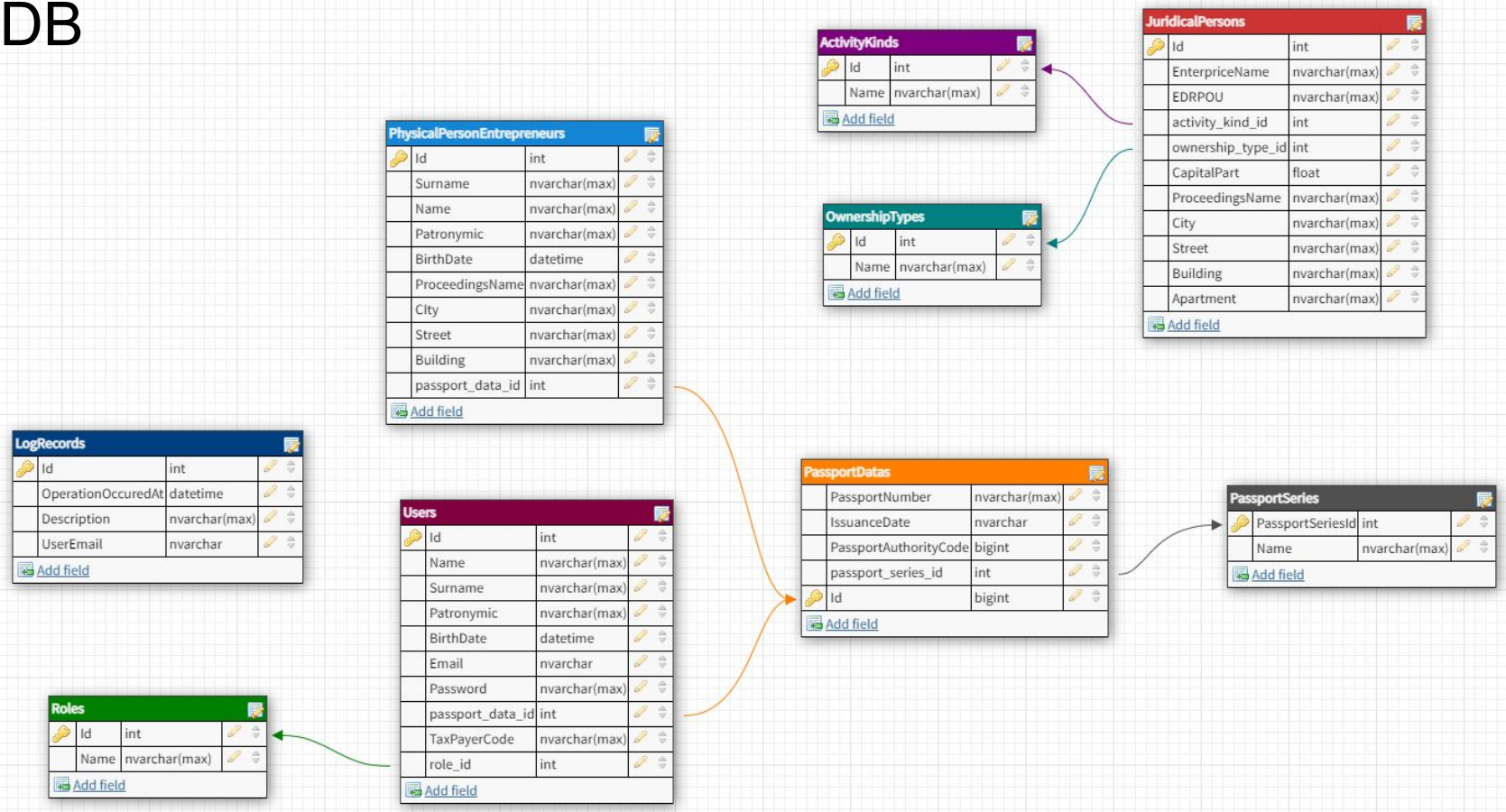
Repositories



Models


PhysicalPersonEntrepreneur
+ PhysicalPersonEntrepreneur(Integer, String, String, String, Date, String, PassportData, Place):
+ PhysicalPersonEntrepreneur():
+ proceedingName: String
+ place: Place
+ birthdate: Date
+ surname: String
+ passportData: PassportData
+ id: Integer
+ name: String
+ patronymic: String
+ name: String
+ proceedingName: String
+ surname: String
+ place: Place
+ patronymic: String
+ id: Integer
+ passportData: PassportData
+ birthdate: Date
JuridicalPerson
+ JuridicalPerson(Integer, String, String, ActivityKind,
+ JuridicalPerson():
+ enterpriseName: String
+ edrpou: String
+ ownershipType: OwnershipType
+ capitalPart: Double
+ id: Integer
+ proceedingName: String
+ place: Place
+ activityKind: ActivityKind
+ edrpou: String
+ proceedingName: String
+ place: Place
+ capitalPart: Double
+ activityKind: ActivityKind
+ id: Integer
PassportData
+ PassportData(String, Date, String, PassportSeries):
+ PassportData():
+ passportSeries: PassportSeries
+ passportNumber: String
+ passportIssuanceDate: Date
+ taxPayerCode: String
+ passportNumber: String
+ passportIssuanceDate: Date
+ passportSeries: PassportSeries
+ taxPayerCode: String
OwnershipType
+ OwnershipType(Integer, String):
+ OwnershipType():
+ name: String
+ ownershipTypeId: Integer
+ name: String
+ ownershipTypeId: Integer
ActivityKind
+ ActivityKind(Integer, String):
+ ActivityKind():
+ activityKindId: Integer
+ name: String
+ name: String
+ activityKindId: Integer
Place
+ Place(String, String, String, String):
+ Place():
+ city: String
+ building: String
+ apartment: String
+ street: String
+ city: String
+ street: String
+ building: String
+ apartment: String
User
+ User(Integer, String, String, String, Date, String, String, Role, PassportData):
+ User():
+ patronymic: String
+ id: Integer
+ password: String
+ email: String
+ birthdate: Date
+ name: String
+ role: Role
+ surname: String
+ passportData: PassportData
+ name: String
+ password: String
+ role: Role
+ email: String
+ surname: String
+ patronymic: String
+ id: Integer
+ passportData: PassportData
+ birthdate: Date
LogRecord
+ LogRecord(Integer, Date, String, User):
+ LogRecord():
+ operationOccurredAt: Date
+ description: String
+ user: User
+ id: Integer
+ description: String
+ operationOccurredAt: Date
+ id: Integer
+ user: User
PassportSeries
+ PassportSeries(Integer, String, List<User>):
+ PassportSeries():
+ id: Integer
+ name: String
+ users: List<User>
+ name: String
+ id: Integer
+ users: List<User>
Role
+ Role(Integer, String, List<User>):
+ Role():
+ name: String
+ users: List<User>
+ id: Integer
+ name: String
+ id: Integer
+ users: List<User>
SignIn
+ SignIn(String, String):
+ SignIn():
+ email: String
+ password: String
+ email: String
+ password: String

DB


ActivityKinds
Id int
Name nvarchar(max)
Add field
JuridicalPersons
Id int
EnterpriceName nvarchar(max)
EDRPOU nvarchar(max)
activity_kind_id int
ownership_type_id int
CapitalPart float
ProceedingsName nvarchar(max)
City nvarchar(max)
Street nvarchar(max)
Building nvarchar(max)
Apartment nvarchar(max)
Add field
PhysicalPersonEntrepreneurs
Id int
Surname nvarchar(max)
Name nvarchar(max)
Patronymic nvarchar(max)
BirthDate datetime
ProceedingsName nvarchar(max)
City nvarchar(max)
Street nvarchar(max)
Building nvarchar(max)
passport_data_id int
Add field
OwnershipTypes
Id int
Name nvarchar(max)
Add field
LogRecords
Id int
OperationOccuredAt datetime
Description nvarchar(max)
UserEmail nvarchar
Add field
Users
Id int
Name nvarchar(max)
Surname nvarchar(max)
Patronymic nvarchar(max)
BirthDate datetime
Email nvarchar
Password nvarchar(max)
passport_data_id int
TaxPayerCode nvarchar(max)
role_id int
Add field
PassportDatas
PassportNumber nvarchar(max)
IssuanceDate nvarchar
PassportAuthorityCode bigint
passport_series_id int
Id bigint
Add field
PassportSeries
PassportSeriesId int
Name nvarchar(max)
Add field
Roles
Id int
Name nvarchar(max)
Add field

Стек