



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

# **Skladový informační systém**

## **Návrhová dokumentace (Detailní design)**

Dokument vytvořen pro potřeby předmětu BI-SI1

Autoři: Róbert Selvek, Vojtěch Cahlík, Josef Hušek, Jan Lidák



## Obsah

1. Návrhový model tříd .....	6
1.1 skld .....	6
1.1.1 Bussiness vrstva .....	7
1.1.2 Datová vrstva .....	7
1.1.3 Prezentační vrstva .....	7
1.1.4 application .....	8
1.1.4.1 Class SKLDAPIApplication .....	8
1.1.4.2 Class SKLDAPIConfiguration .....	8
1.1.4.3 auth .....	9
1.1.4.3.1 Class UserAuthenticator .....	9
1.1.4.3.2 Class UserAuthorizer .....	9
1.1.4.4 core .....	10
1.1.4.4.1 Class Lineltem .....	10
1.1.4.4.2 Class Order .....	12
1.1.4.4.3 Class OrderIn .....	13
1.1.4.4.4 Class Product .....	14
1.1.4.4.5 Class ProductMovement .....	15
1.1.4.4.6 Class ProductPosition .....	16
1.1.4.4.7 Class User .....	17
1.1.4.4.8 Enumeration OrderState .....	18
1.1.4.5 db .....	18
1.1.4.5.1 Interface MovementDAO .....	19
1.1.4.5.2 Interface OrderInDAO .....	19
1.1.4.5.3 Interface PositionDAO .....	20
1.1.4.5.4 Interface ProductDAO .....	20
1.1.4.5.5 Interface UserDAO .....	20
1.1.4.5.6 postgres .....	21
1.1.4.5.6.1 Class PostgresMovementDAO .....	21
1.1.4.5.6.2 Class PostgresOrderInDAO .....	21
1.1.4.5.6.3 Class PostgresPositionDAO .....	22
1.1.4.5.6.4 Class PostgresProductDAO .....	22
1.1.4.5.6.5 Class PostgresUserDAO .....	23
1.1.4.6 operations .....	23
1.1.4.6.1 Class OrderInOperations .....	23
1.1.4.6.2 Class ProductOperations .....	24
1.1.4.6.3 exceptions .....	25
1.1.4.6.3.1 Class InvalidStateException .....	26
1.1.4.6.3.2 Class InvalidStateExceptionSupplier .....	26
1.1.4.6.3.3 Class NotFoundException .....	26
1.1.4.6.3.4 Class NotFoundExceptionSupplier .....	26
1.1.4.7 resources .....	26
1.1.4.7.1 Class AuthResource .....	27
1.1.4.7.2 Class OrderInResource .....	27
1.1.4.7.3 Class OrderInsResource .....	28
1.1.4.7.4 Class ProductResource .....	29
1.1.4.7.5 Class ProductsResource .....	29
1.1.4.7.6 Class RepresentationConverter .....	30
1.1.4.8 util .....	30
1.1.4.8.1 Class WebAppExceptionSupplier .....	31
1.1.5 client .....	31
1.1.5.1 Interface SkldClient .....	31



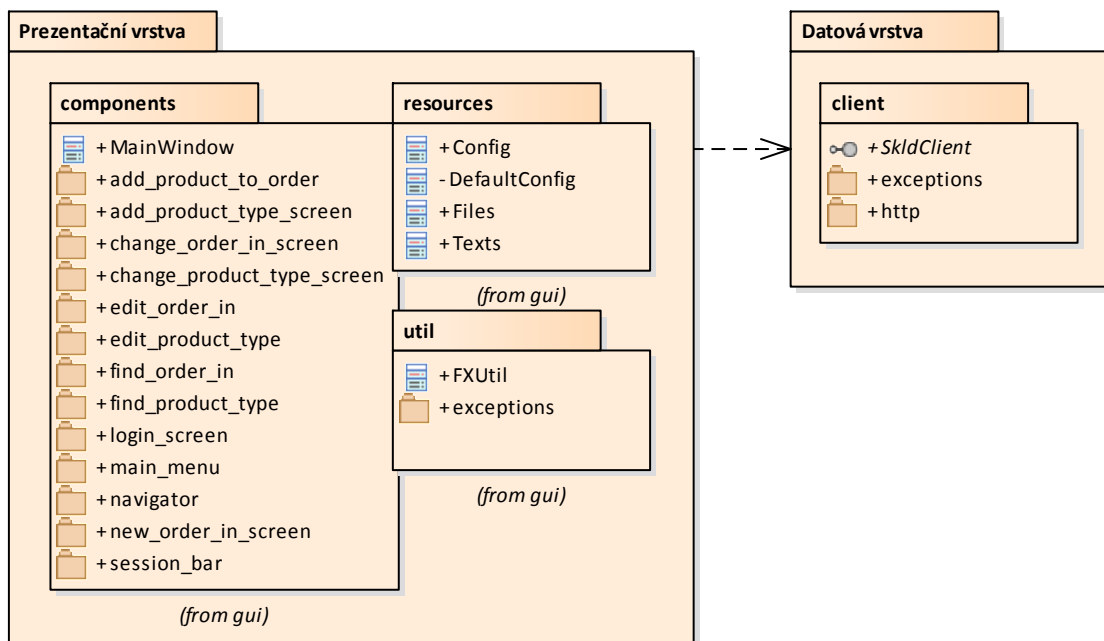
1.1.5.2	exceptions .....	33
1.1.5.2.1	Class APIException .....	33
1.1.5.2.2	Class BadRequestException .....	34
1.1.5.2.3	Class ForbiddenException .....	34
1.1.5.2.4	Class NonspecificException .....	34
1.1.5.2.5	Class NotFoundException .....	34
1.1.5.2.6	Class ServerException .....	34
1.1.5.2.7	Class UnauthorizedException .....	35
1.1.5.3	http .....	35
1.1.5.3.1	Class SkIdHttpClient .....	35
1.1.5.3.2	Enumeration HTTPMethod .....	37
1.1.6	gui .....	38
1.1.6.1	Class App .....	39
1.1.6.2	Class Component .....	39
1.1.6.3	Class Fragment .....	40
1.1.6.4	Class Frame .....	41
1.1.6.5	Class Handler .....	41
1.1.6.6	Class Screen .....	41
1.1.6.7	Class Window .....	42
1.1.6.8	Interface Notifyable .....	43
1.1.6.9	Interface Passable .....	43
1.1.6.10	Interface UI .....	43
1.1.6.11	Interface Wrapper .....	43
1.1.6.12	Enumeration FXMLFragmentType .....	44
1.1.6.13	Enumeration NotifyType .....	44
1.1.6.14	Enumeration PassResult .....	44
1.1.6.15	components .....	45
1.1.6.15.1	Class MainWindow .....	45
1.1.6.15.2	add_product_to_order .....	45
1.1.6.15.2.1	Class AddProductToOrderScreen .....	46
1.1.6.15.2.2	Class AddProductToOrderScreenHandler .....	47
1.1.6.15.2.3	Class AddProductToOrderWindow .....	48
1.1.6.15.3	add_product_type_screen .....	48
1.1.6.15.3.1	Class AddProductTypeFrame .....	49
1.1.6.15.3.2	Class AddProductTypeScreen .....	49
1.1.6.15.3.3	Class AddProductTypeScreenHandler .....	50
1.1.6.15.4	change_order_in_screen .....	50
1.1.6.15.4.1	Class ChangeOrderInFrame .....	51
1.1.6.15.4.2	Class ChangeOrderInScreen .....	51
1.1.6.15.4.3	Class ChangeOrderInScreenHandler .....	52
1.1.6.15.5	change_product_type_screen .....	53
1.1.6.15.5.1	Class ChangeProductTypeFrame .....	54
1.1.6.15.5.2	Class ChangeProductTypeScreen .....	54
1.1.6.15.5.3	Class ChangeProductTypeScreenHandler .....	55
1.1.6.15.6	edit_order_in .....	56
1.1.6.15.6.1	Class EditOrderInFragment .....	56
1.1.6.15.6.2	Class EditOrderInFragmentHandler .....	59
1.1.6.15.6.3	Class OrderInProductFragment .....	60
1.1.6.15.6.4	Class OrderInProductFragmentHandler .....	61
1.1.6.15.7	edit_product_type .....	61
1.1.6.15.7.1	Class EditProductTypeFragment .....	62
1.1.6.15.7.2	Class EditProductTypeFragmentHandler .....	64



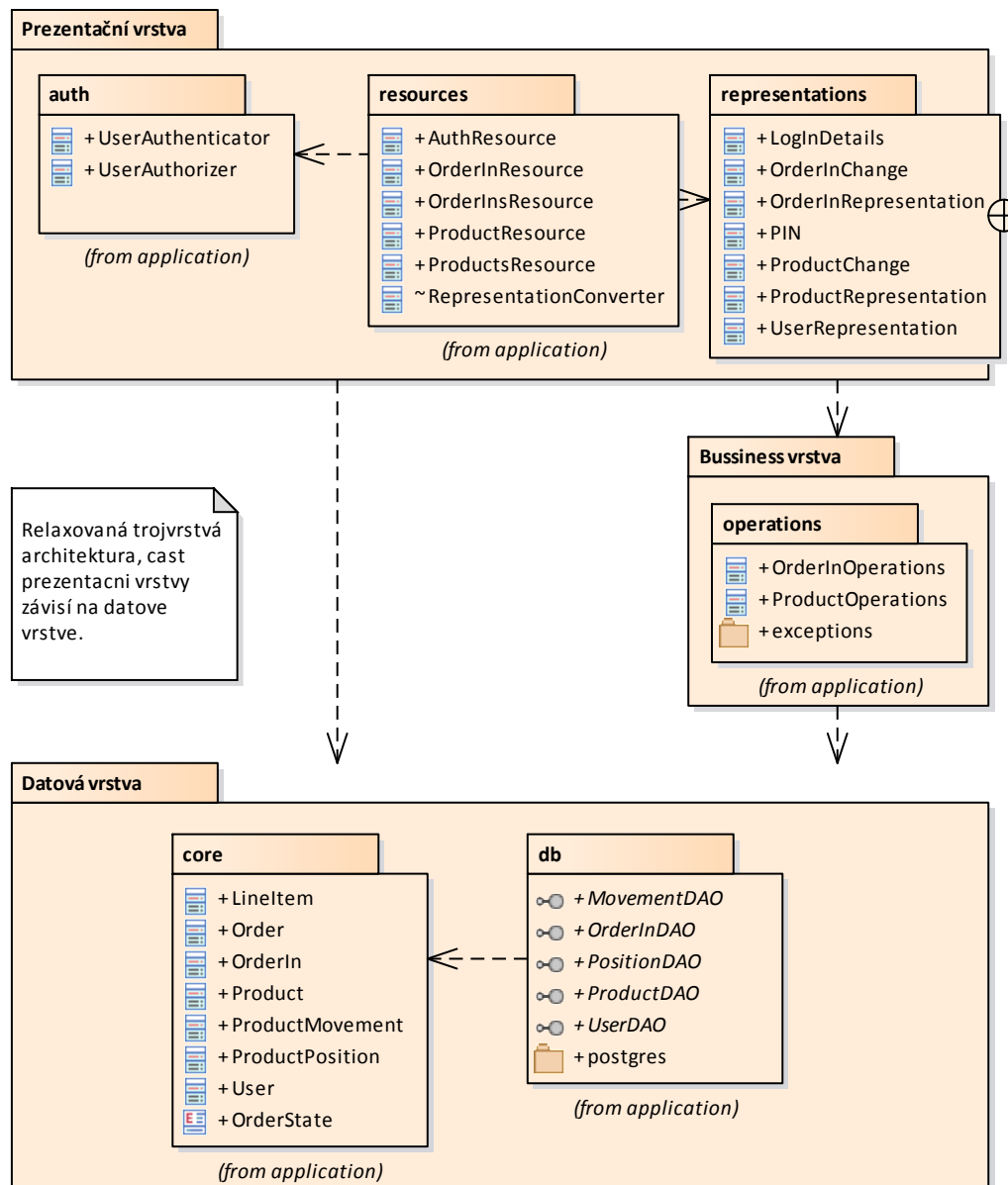
1.1.6.15.8	find_order_in	64
1.1.6.15.8.1	Class FindOrderInFragment	65
1.1.6.15.8.2	Class FindOrderInFragmentHandler	65
1.1.6.15.9	find_product_type	66
1.1.6.15.9.1	Class FindProductTypeFragment	67
1.1.6.15.10	login_screen	67
1.1.6.15.10.1	Class LoginScreen	68
1.1.6.15.10.2	Class LoginScreenHandler	69
1.1.6.15.11	main_menu	69
1.1.6.15.11.1	Class MainMenuScreen	70
1.1.6.15.11.2	Class MainMenuScreenHandler	71
1.1.6.15.12	navigator	71
1.1.6.15.12.1	Class NavigatorFrame	72
1.1.6.15.12.2	Class NavigatorFrameHandler	73
1.1.6.15.12.3	navigator_bar	74
1.1.6.15.12.3.1	Class NavigatorBarFragment	74
1.1.6.15.12.3.2	Class NavigatorBarFragmentHandler	74
1.1.6.15.13	new_order_in_screen	75
1.1.6.15.13.1	Class NewOrderInFrame	75
1.1.6.15.13.2	Class NewOrderInScreen	75
1.1.6.15.13.3	Class NewOrderInScreenHandler	76
1.1.6.15.14	session_bar	77
1.1.6.15.14.1	Class SessionBarFragment	77
1.1.6.15.14.2	Class SessionBarFragmentHandler	77
1.1.6.16	resources	78
1.1.6.16.1	Class Config	78
1.1.6.16.1.1	Class PropertyNames	79
1.1.6.16.2	Class DefaultConfig	79
1.1.6.16.3	Class Files	79
1.1.6.16.4	Class Texts	79
1.1.6.16.4.1	Class Alerts	79
1.1.6.16.4.2	Class ConsoleOutput	80
1.1.6.16.4.3	Class Windows	80
1.1.6.17	util	80
1.1.6.17.1	Class FXUtil	80
1.1.6.17.2	exceptions	81
1.1.6.17.2.1	Class InvalidInputException	81
1.1.7	representations	81
1.1.7.1	Class LoginDetails	82
1.1.7.2	Class OrderInChange	83
1.1.7.3	Class OrderInRepresentation	84
1.1.7.4	Class PIN	85
1.1.7.5	Class ProductChange	86
1.1.7.6	Class ProductRepresentation	86
1.1.7.7	Class UserRepresentation	87
2.	Model komunikace	89
2.1	Realizace UC2 - Vytvoření logistické objednávky	89
2.2	Realizace UC8 - Přidání nového typu zboží	91

# 1. Návrhový model tříd

## 1.1 skld



Obrázek 1 - Klient



Obrázek 2 - Server

### 1.1.1 Bussiness vrstva

### 1.1.2 Datová vrstva

### 1.1.3 Prezentační vrstva

## 1.1.4 application

SKLDAPIApplication	Ap	SKLDAPIConfiguration	Configuration
- hibernateBundle: HibernateBundle<SKLDAPIConfiguration> = new HibernateBu...	{r	- database: DataSourceFactory = new DataSourceF...	- jwtSecret: String

Obrázek 3 - application

### 1.1.4.1 Class SKLDAPIApplication

Vstupní bod serverové aplikace. Dedi nadřidu definovanou v Dropwizard. Inicializuje veskere potrebne balicky, injektuje potrebne objekty do resources a pomoci sve nadřidy spusti aplikacni server.

Název atributu	Datový typ	Popis
hibernateBundle	HibernateBundle<SKLDAPIConfiguration>	
Název metody	Návratový typ	Popis
getName	String	Vrací název aplikace. @return Název aplikace
initialize	void	Parametry: <b>bootstrap:</b> Bootstrap<SKLDAPIConfiguration> -
main	void	Hlavní metoda. Spustí aplikaci. Parametry: <b>args:</b> String[] - Argumenty předané z příkazové řádky
run	void	Parametry: <b>configuration:</b> SKLDAPIConfiguration - Parametry: <b>environment:</b> Environment -

### 1.1.4.2 Class SKLDAPIConfiguration

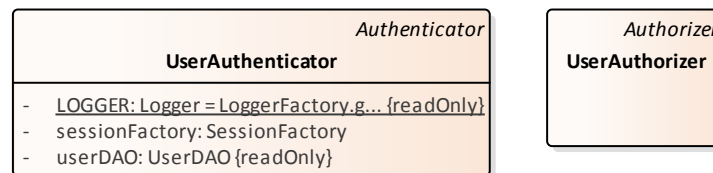
Obsahuje konfiguraci aplikace.

Název atributu	Datový typ	Popis
database	DataSourceFactory	
jwtSecret	String	
Název metody	Návratový typ	Popis
getDataSourceFactory	DataSourceFactory	
getJwtSecret	byte	
setDataSourceFactory	void	Parametry: <b>dataSourceFactory:</b> DataSourceFactory -



Název atributu	Datový typ	Popis
setJwtSecret	void	Parametry: <b>jwtSecret: String</b> -

### 1.1.4.3 auth



Obrázek 4 - auth

#### 1.1.4.3.1 Class UserAuthenticator

Overuje JWT data přijata z klienta a nahraje z databaze uzivatele s prislusnym heslem.

Název atributu	Datový typ	Popis
LOGGER	Logger	
sessionFactory	SessionFactory	
userDAO	UserDAO	
Název metody	Návratový typ	Popis
authenticate	Optional<User>	Validuje JWT data a pokusi se získat uzivatele s ID, které je uloženo jako identifikátor session. @return Uzivatel Parametry: <b>jwtContext: JwtContext</b> - Identifikátor session prislusneho requestu
UserAuthenticator		Konstruktor. Parametry: <b>dao: UserDAO</b> - UserDAO pro pristup k uzivatelum Parametry: <b>sessionFactory: SessionFactory</b> - Session pro pristup k databazi pres UserDAO. Viz dokumentace Dropwizard frameworku.

#### 1.1.4.3.2 Class UserAuthorizer

Implementace overeni, zda ma uzivatel prava k provedeni nejake akce.

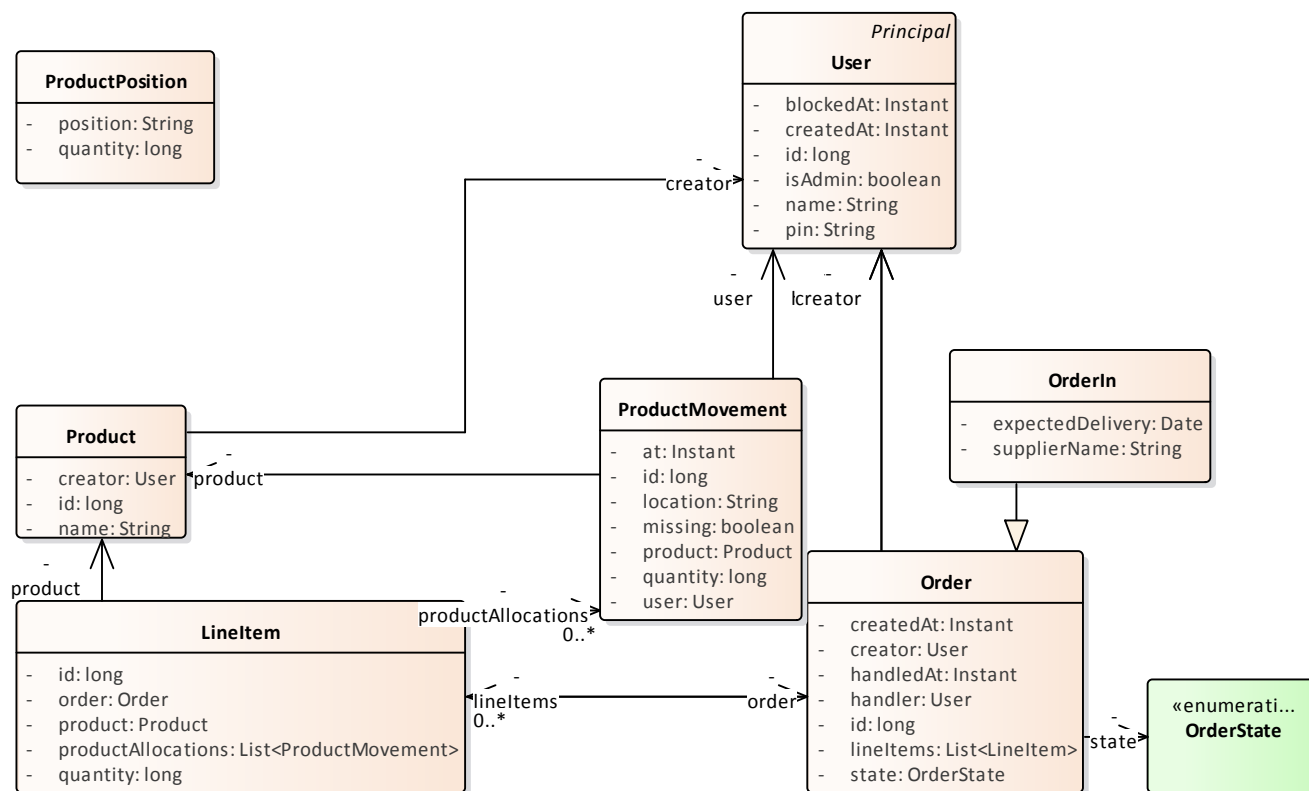
Název metody	Návratový typ	Popis
authorize	boolean	Zjistí, zda má uživatel specifikované minimální oprávnění. @return Parametry: <b>user: User</b> - Overovaný uživatel Parametry: <b>role: String</b> - Role, kterou uživatel musí mít k provedení





Název metody	Návratový typ	Popis
		prislusne operace

#### 1.1.4.4 core



Obrázek 5 - core

##### 1.1.4.4.1 Class Lineltem

Reprezentuje LineItem z doménového modelu.

Název atributu	Datový typ	Popis
id	long	
order	Order	
product	Product	
productAllocations	List<ProductMovement>	
quantity	long	
Název metody	Návratový typ	Popis

equals

boolean

Parametry:  
o: Object -



Název atributu	Datový typ	Popis
getId	long	
getOrder	Order	
getProduct	Product	
getProductAllocations	List<ProductMovement>	
getQuantity	long	
hashCode	int	
LineItem		Konstruktor.
LineItem		Konstruktor. Parametry: <b>quantity: long</b> - Mnozství produktu Parametry: <b>product: Product</b> - Produkt Parametry: <b>order: Order</b> - Příslušná objednávka
setId	void	Parametry: <b>id: long</b> -
setOrder	void	Parametry: <b>order: Order</b> -
setProduct	void	Parametry: <b>product: Product</b> -
setProductAllocations	void	Parametry: <b>productAllocations: List&lt;ProductMovement&gt;</b> -
setQuantity	void	Parametry: <b>quantity: long</b> -

#### 1.1.4.4.2 Class Order

Reprezentuje objednávku z doménového modelu.

Název atributu	Datový typ	Popis
createdAt	Instant	
creator	User	
handledAt	Instant	
handler	User	



Název atributu	Datový typ	Popis
id	long	
lineItems	List<LineItem>	
state	OrderState	
Název metody	Návratový typ	Popis
equals	boolean	Parametry: <b>o: Object</b> -
getCreatedAt	Instant	
getCreator	User	
getHandledAt	Instant	
getHandler	User	
getId	long	
getLineItems	List<LineItem>	
getState	OrderState	
hashCode	int	
lineItemMap	Map<Long, LineItem>	
Order		Konstruktor.
Order		Konstruktor. Parametry: <b>id: long</b> - ID objednávky Parametry: <b>creator: User</b> - Uživatel, který objednávku vytvořil
setCreatedAt	void	Parametry: <b>createdAt: Instant</b> -
setCreator	void	Parametry: <b>creator: User</b> -
setHandledAt	void	Parametry: <b>handledAt: Instant</b> -
setHandler	void	Parametry: <b>handler: User</b> -
setHandlingDetails	void	



Název atributu	Datový typ	Popis
		Parametry: <b>handler: User</b> -
setId	void	Parametry: <b>id: long</b> -
setLineItems	void	Parametry: <b>lineItems: List&lt;LineItem&gt;</b> -
setState	void	Parametry: <b>state: OrderState</b> -

#### 1.1.4.4.3 Class OrderIn

Reprezentuje logistickou objednávku z doménového modelu.

Název atributu	Datový typ	Popis
expectedDelivery	Date	
supplierName	String	
Název metody	Návratový typ	Popis
equals	boolean	Parametry: <b>o: Object</b> -
getExpectedDelivery	Date	
getSupplierName	String	
hashCode	int	
OrderIn		Konstruktor.
OrderIn		Konstruktor. Parametry: <b>id: long</b> - ID Parametry: <b>creator: User</b> - Uživatel, který objednávku zadal do systému Parametry: <b>supplierName: String</b> - Název dodavatele
setExpectedDelivery	void	Parametry: <b>expectedDelivery: Date</b> -
setSupplierName	void	Parametry: <b>supplierName: String</b> -



#### 1.1.4.4.4 Class Product

Reprezentuje produkt z doménového modelu.

Název atributu	Datový typ	Popis
creator	User	
id	long	
name	String	
Název metody	Návratový typ	Popis
equals	boolean	Parametry: <b>o: Object</b> -
getCreator	User	
getId	long	
getName	String	
hashCode	int	
Product		Konstruktor.
Product		Konstruktor. Parametry: <b>id: long</b> - ID Parametry: <b>name: String</b> - Nazv produktu Parametry: <b>creator: User</b> - Uzivatel, který produkt zadal do systemu
setCreator	void	Parametry: <b>creator: User</b> -
setId	void	Parametry: <b>id: long</b> -
setName	void	Parametry: <b>name: String</b> -

#### 1.1.4.4.5 Class ProductMovement

Reprezentuje umístění produktu z doménového modelu.

Název atributu	Datový typ	Popis
at	Instant	
id	long	
location	String	
missing	boolean	
product	Product	
quantity	long	



Název atributu	Datový typ	Popis
user	User	
Název metody	Návratový typ	Popis
equals	boolean	Parametry: <b>o: Object</b> -
getAt	Instant	
getId	long	
getLocation	String	
getProduct	Product	
getQuantity	long	
getUser	User	
hashCode	int	
isMissing	boolean	
ProductMovement		Konstruktor.
ProductMovement		Konstruktor. Parametry: <b>product: Product</b> - Umístěný produkt Parametry: <b>qty: long</b> - Množství produktu Parametry: <b>destination: String</b> - Místo, kde se produkt nachází po umístění Parametry: <b>creator: User</b> - Uživatel, který produkt zadal do systému
setAt	void	Parametry: <b>at: Instant</b> -
setId	void	Parametry: <b>id: long</b> -
setLocation	void	Parametry: <b>location: String</b> -
setMissing	void	Parametry: <b>missing: boolean</b> -



Název atributu	Datový typ	Popis
setProduct	void	Parametry: <b>product: Product</b> -
setQuantity	void	Parametry: <b>quantity: long</b> -
setUser	void	Parametry: <b>user: User</b> -

#### 1.1.4.4.6 Class ProductPosition

Pomocná třída, která popisuje určité množství nějakého (zde nespecifikovaného) produktu na specifikovaném místě.

Název atributu	Datový typ	Popis
position	String	
quantity	long	
Název metody	Návratový typ	Popis
equals	boolean	Parametry: <b>o: Object</b> -
getPosition	String	
getQuantity	long	
hashCode	int	
ProductPosition		Konstruktor. Parametry: <b>position: String</b> - Pozice Parametry: <b>quantity: long</b> - Množství produktu
setPosition	void	Parametry: <b>position: String</b> -
setQuantity	void	Parametry: <b>quantity: long</b> -

#### 1.1.4.4.7 Class User

Reprezentuje uživatele z doménového modelu.

Název atributu	Datový typ	Popis
blockedAt	Instant	
createdAt	Instant	
id	long	



Název atributu	Datový typ	Popis
isAdmin	boolean	
name	String	
pin	String	
Název metody	Návratový typ	Popis
		Overi, zda zadany PIN odpovida uzivatelove PINu. @return True zadany PIN odpovida uzivatelove PINu Parametry: <b>comparedPIN: String</b> - Porovnavany PIN
comparePIN	boolean	
equals	boolean	Parametry: <b>o: Object</b> -
getBlockedAt	Instant	
getCreatedAt	Instant	
getId	long	
getName	String	
hashCode	int	
isAdmin	boolean	
setAdmin	void	Parametry: <b>admin: boolean</b> -
setBlockedAt	void	Parametry: <b>blockedAt: Instant</b> -
setCreatedAt	void	Parametry: <b>createdAt: Instant</b> -
setId	void	Parametry: <b>id: long</b> -
setName	void	Parametry: <b>name: String</b> -
setPin	void	Parametry: <b>pin: String</b> -
User		Konstruktor.





Název atributu	Datový typ	Popis
User		Konstruktor. Parametry: <b>name: String</b> - Jmeno uzivatele Parametry: <b>PIN: String</b> - PIN kod, kterym se uzivatel prihlasuje

#### 1.1.4.4.8 Enumeration OrderState

Reprezentuje stav objednávky z doménového modelu.

Název atributu	Datový typ	Popis
OPEN		Informace o objednávce byly předány skladu provozním oddělením a čeká se na její zpracování.
CLOSED		Objednávka byla zpracována (naskladněna/odeslána).
REFUSED		Při zpracování objednávky došlo k problému a byla zrušena.
Název metody	Návratový typ	Popis
toString	String	

#### 1.1.4.5 db



Obrázek 6 - db

#### 1.1.4.5.1 Interface MovementDAO

Implementuje základní databázové operace nad MovementDAO objekty.

Název metody	Návratový typ	Popis
create	ProductMovement	Vloží do databáze nové umístění produktu (ProductMovement). @return Vytvořený ProductMovement (včetně případných automaticky generovaných hodnot) Parametry: <b>movement: ProductMovement</b> - ProductMovement
findById	Optional<ProductMovement>	Vrací umístění produktu (ProductMovement) se zadaným ID. @return ProductMovement se zadaným ID Parametry: <b>id: Long</b> - ID

#### 1.1.4.5.2 Interface OrderInDAO

Implementuje základní databázové operace nad OrderIn objekty.

Název metody	Návratový typ	Popis
--------------	---------------	-------



Název metody	Návratový typ	Popis
create	OrderIn	Vloží do databáze novou logistickou objednávku. @return Vytvořená logistická objednávka (včetně případných automaticky generovaných hodnot) Parametry: <b>order:</b> <b>OrderIn</b> - Logistická objednávka
findAll	List<OrderIn>	Vrací seznam všech logistických objednávek. @return Seznam objednávek
findById	Optional<OrderIn>	Vrací logistickou objednávku se zadaným ID. @return Logistická objednávka Parametry: <b>id:</b> <b>Long</b> - ID
removeLineItem	void	Odstraní LineItem z databáze. Tato metoda je umístěna v OrderInDAO, protože LineItems jsou zcela závislé na svých objednávkách. Parametry: <b>li:</b> <b>LineItem</b> - LineItem k odstranění

#### 1.1.4.5.3 Interface PositionDAO

Implementuje zjišťování pozic produktu. Tato třída neobsahuje metody pro tvoreň nových pozic nebo pokročilých queries jako v ostatních třídách v tomto balíčku, protože ProductPositions nejsou reprezentovány na DB úrovni.

Název metody	Návratový typ	Popis
findForProductId	List<ProductPosition>	Vrací pozice produktu se zadaným ID. @return Seznam pozic, na kterých je produkt umístěn Parametry: <b>productId:</b> <b>long</b> - ID produktu

#### 1.1.4.5.4 Interface ProductDAO

Implementuje základní databázové operace nad Product objekty.

Název metody	Návratový typ	Popis
create	Product	Vloží nový produkt do databáze. @return Vytvořený produkt (včetně případných automaticky generovaných hodnot) Parametry: <b>product:</b> <b>Product</b> - Vkládaný produkt
findAll	List<Product>	Vrací všechny produkty v databázi. @return Seznam produktu
findById	Optional<Product>	Vrací produkt se zadaným ID. @return Produkt Parametry: <b>id:</b> <b>Long</b> - ID produktu



#### 1.1.4.5.5 Interface UserDAO

Implementuje základní databázové operace nad User objekty.

Název metody	Návratový typ	Popis
create	User	Vloží do databáze nového uživatele. @return Uživatel (včetně případných automaticky generovaných hodnot) Parametry: <b>user: User</b> - Uživatel
findById	Optional<User>	Vrací uživatele se zadaným ID. @return Uživatel Parametry: <b>id: Long</b> - ID uživatele
findByPin	Optional<User>	Vrací uživatele, který se přihlašuje zadaným PINem. @return Uživatel Parametry: <b>pin: String</b> - PIN kód

#### 1.1.4.5.6 postgres



Obrázek 7 - postgres

#### 1.1.4.5.6.1 Class PostgresMovementDAO

Implementuje základní databázové operace nad MovementDAO objekty.

Název metody	Návratový typ	Popis
create	ProductMovement	Vloží do databáze nové umístění produktu (ProductMovement). @return Vytvořený ProductMovement (včetně případných automaticky generovaných hodnot) Parametry: <b>movement: ProductMovement</b> - ProductMovement
findById	Optional<ProductMovement>	Vrací umístění produktu (ProductMovement) se zadaným ID. @return ProductMovement se zadaným ID Parametry: <b>id: Long</b> - ID
PostgresMovementDAO		Konstruktor. Parametry: <b>factory: SessionFactory</b> - Factory specifikovaná v Dropwizard frameworku



#### 1.1.4.5.6.2 Class PostgresOrderInDAO

Implementuje základní databázové operace nad OrderIn objekty.

Název metody	Návratový typ	Popis
create	OrderIn	Vloží do databáze novou logistickou objednávku. @return Vytvořená logistická objednávka (včetně případných automaticky generovaných hodnot) Parametry: <b>order:</b> <b>OrderIn</b> - Logistická objednávka
findAll	List<OrderIn>	Vrací seznam všech logistických objednávek. @return Seznam objednávek
findById	Optional<OrderIn>	Vrací logistickou objednávku se zadaným ID. @return Logistická objednávka Parametry: <b>id:</b> <b>Long</b> - ID
PostgresOrderInDAO		Konstruktor. Parametry: <b>factory:</b> <b>SessionFactory</b> - Factory specifikovaná v Dropwizard frameworku
removeLineItem	void	Odstraní LineItem z databáze. Tato metoda je umístěna v OrderInDAO, protože LineItems jsou zcela závislé na svých objednávkách. Parametry: <b>li:</b> <b>LineItem</b> - LineItem k odstranění

#### 1.1.4.5.6.3 Class PostgresPositionDAO

Implementuje zjišťování pozic produktu. Tato třída neobsahuje metody pro tvoreni nových pozic nebo pokročilých queries jako v ostatních třídách v tomto balíčku, protože ProductPositions nejsou reprezentovány na DB úrovni.

Název metody	Návratový typ	Popis
findForProductId	List<ProductPosition>	Vrací pozice produktu se zadaným ID. @return Seznam pozic, na kterých je produkt umístěn Parametry: <b>productId:</b> <b>long</b> - ID produktu
PostgresPositionDAO		Konstruktor. Parametry: <b>factory:</b> <b>SessionFactory</b> - Factory specifikovaná v Dropwizard frameworku

#### 1.1.4.5.6.4 Class PostgresProductDAO

Implementuje základní databázové operace nad Product objekty.

Název metody	Návratový typ	Popis
create	Product	Vloží nový produkt do databáze. @return Vytvořený produkt (včetně případných automaticky generovaných hodnot) Parametry:



Název metody	Návratový typ	Popis
		<b>product: Product</b> - Vkládany produkt
findAll	List<Product>	Vrací všechny produkty v databázi. @return Seznam produktu
findById	Optional<Product>	Vrací produkt se zadaným ID. @return Produkt Parametry: <b>id: Long</b> - ID produktu
PostgresProductDAO		Konstruktor. Parametry: <b>factory: SessionFactory</b> - Factory specifikována v Dropwizard frameworku

#### 1.1.4.5.6.5 Class PostgresUserDAO

Implementuje základní databázové operace nad User objekty.

Název metody	Návratový typ	Popis
create	User	Vloží do databáze nového uživatele. @return Uživatel (včetně případných automaticky generovaných hodnot) Parametry: <b>user: User</b> - Uživatel
findById	Optional<User>	Vrací uživatele se zadaným ID. @return Uživatel Parametry: <b>id: Long</b> - ID uživatele
findByPin	Optional<User>	Vrací uživatele, který se přihlašuje zadaným PINem. @return Uživatel Parametry: <b>pin: String</b> - PIN kód
PostgresUserDAO		Konstruktor. Parametry: <b>factory: SessionFactory</b> - Factory specifikována v Dropwizard frameworku

#### 1.1.4.6 operations



OrderInOperations	ProductOperations
<ul style="list-style-type: none"><li>- movementDAO: MovementDAO {readOnly}</li><li>- orderInDAO: OrderInDAO {readOnly}</li><li>- productDAO: ProductDAO {readOnly}</li></ul>	<ul style="list-style-type: none"><li>- <u>LOGGER</u>: Logger = LoggerFactory.g... {readOnly}</li><li>- positionDAO: PositionDAO {readOnly}</li><li>- productDAO: ProductDAO {readOnly}</li></ul>
<ul style="list-style-type: none"><li>- sumMapKeys(): long</li></ul>	

Obrázek 8 - operations

#### 1.1.4.6.1 Class OrderInOperations

Implementuje operace související s logistickými objednávkami.

Název atributu	Datový typ	Popis
movementDAO	MovementDAO	
orderInDAO	OrderInDAO	
productDAO	ProductDAO	
Název metody	Návratový typ	Popis
close	OrderIn	Uzavře logistickou objednávku a zadá příslušné umístění produktu do systému podle předaných produktů z requestu Parametry: <b>handler: User</b> - Uživatel, který operaci zadal do systému Parametry: <b>request: OrderInRepresentation</b> - Uzavíraná logistická objednávka
create	OrderIn	Vytvoří novou logistickou objednávku. Parametry: <b>creator: User</b> - Uživatel, který operaci zadal do systému Parametry: <b>request: OrderInChange</b> -
edit	OrderIn	Upraví logistickou objednávku se zadaným ID. V současnosti smí být modifikován pouze status. Parametry: <b>change: OrderInChange</b> -
get	OrderIn	Vrací logistickou objednávku podle zadaného ID. Parametry: <b>id: long</b> - ID
getAll	List<OrderIn>	Vrací seznam všech logistických objednávek z databáze. @return Seznam logistických objednávek
OrderInOperations		Konstruktor. Parametry: <b>orderInDAO: OrderInDAO</b> - DAO pro přístup k logistickým objednávkám Parametry: <b>productDAO: ProductDAO</b> - DAO pro přístup k produktům Parametry: <b>movementDAO: MovementDAO</b> - DAO pro přístup k



Název atributu	Datový typ	Popis
		umístěním produktu
refuse	OrderIn	Zamítne logistickou objednávku. Parametry: <b>handler: User</b> - Uživatel, který operaci zadal do systému Parametry: <b>id: long</b> - ID zamítané logistické objednávky
sumMapKeys	long	Parametry: <b>m: Map&lt;String, Long&gt;</b> -

#### 1.1.4.6.2 Class ProductOperations

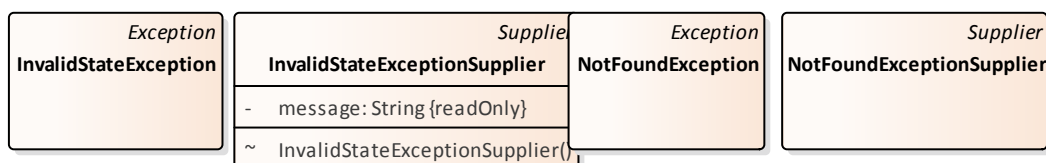
Implementuje operace související s produkty.

Název atributu	Datový typ	Popis
LOGGER	Logger	
positionDAO	PositionDAO	
productDAO	ProductDAO	
Název metody	Návratový typ	Popis
create	Product	Vytvoří nový produkt. @return Vytvořený produkt (včetně případných automaticky generovaných hodnot) Parametry: <b>creator: User</b> - Uživatel, který operaci zadal do systému Parametry: <b>product: ProductChange</b> - Objekt obsahující informace, podle kterých bude produkt vytvořen
edit	Product	Změní název produktu se zadaným ID. Parametry: <b>change: ProductChange</b> - Objekt obsahující informace, podle kterých bude produkt změněn
get	Product	Vrátí produkt se zadaným ID. @return Požadovaný produkt Parametry: <b>id: long</b> - ID produktu
list	List<Product>	Get a list of all the products that are in the database @return a list of all the products that are in the database
positionsForProduct	List<ProductPosition>	Vrátí seznam pozic, na kterých je produkt ve skladu umístěn. @return Seznam pozic Parametry: <b>id: long</b> - ID produktu
ProductOperations		Konstruktor. Parametry: <b>pDao: ProductDAO</b> - DAO pro přístup k produktům



Název atributu	Datový typ	Popis
		Parametry: <b>posDao:</b> <b>PositionDAO</b> - DAO pro přístup k pozicím

### 1.1.4.6.3 exceptions



Obrázek 9 - exceptions

#### 1.1.4.6.3.1 Class InvalidStateException

Objekt je v nepovolenem/nesmyslnem stavu.

Název metody	Návratový typ	Popis
InvalidStateException		Konstruktor. Parametry: <b>message:</b> <b>String</b> - Zprava vyjimky

#### 1.1.4.6.3.2 Class InvalidStateExceptionSupplier

Poskytovatel pro InvalidStateException.

Název atributu	Datový typ	Popis
message	String	
Název metody	Návratový typ	Popis
get	InvalidStateException	Vrací vytvorenou InvalidStateException. @return InvalidStateException
InvalidStateExceptionSupplier		Konstruktor. Parametry: <b>message:</b> <b>String</b> - Zprava zobrazena vyjimkou InvalidStateException

#### 1.1.4.6.3.3 Class NotFoundException

Hledany objekt nebyl nalezen.

Název metody	Návratový typ	Popis
NotFoundException		Konstruktor.

#### 1.1.4.6.3.4 Class NotFoundExceptionSupplier

Poskytovatel pro NotFoundException.

Název metody	Návratový typ	Popis
get	NotFoundException	Vrací vytvorenou NotFoundException. @return NotFoundException





Název metody	Návratový typ	Popis
NotFoundExceptionSupplier		Konstruktor.

### 1.1.4.7 resources

AuthResource	OrderInResource	OrderInsResource
<ul style="list-style-type: none"><li>- <code>LOGGER: Logger = LoggerFactory.g... {readOnly}</code></li><li>- <code>tokenKey: byte [] {readOnly}</code></li><li>- <code>userDAO: UserDAO {readOnly}</code></li></ul>	<ul style="list-style-type: none"><li>- <code>LOGGER: Logger = LoggerFactory.g... {readOnly}</code></li><li>- <code>orderInOps: OrderInOperations {readOnly}</code></li></ul>	<ul style="list-style-type: none"><li>- <code>LOGGER: Logger = LoggerFactory.g... {readOnly}</code></li><li>- <code>orderInOps: OrderInOperations {readOnly}</code></li></ul>

ProductResource	ProductsResource	RepresentationConverter
<ul style="list-style-type: none"><li>- <code>error404: WebAppExceptionSupplier = new WebAppExcep... {readOnly}</code></li><li>- <code>LOGGER: Logger = LoggerFactory.g... {readOnly}</code></li><li>- <code>productOps: ProductOperations {readOnly}</code></li></ul>	<ul style="list-style-type: none"><li>- <code>LOGGER: Logger = LoggerFactory.g... {readOnly}</code></li><li>- <code>productOps: ProductOperations {readOnly}</code></li></ul>	
<ul style="list-style-type: none"><li>- <code>generateRepresentation(): ProductRepresentation</code></li></ul>		

Obrázek 10 - resources

#### 1.1.4.7.1 Class AuthResource

Implementuje REST koncové body aplikace, které umožňují uživatelům získat bezpečnostní tokeny pro provedení dalších operací.

Název atributu	Datový typ	Popis
LOGGER	Logger	
tokenKey	byte	
userDAO	UserDAO	
Název metody	Návratový typ	Popis
		Konstruktor. Parametry: <b>tokenKey: byte[]</b> - Klic použitý v JWT encryption/signature algoritmech Parametry: <b>uDao: UserDAO</b> - DAO pro přístup k uživatelům v databázi
AuthResource		Overi zadany PIN a pri souhlasu vrati bezpecostni token a informace o prihlasenem uzivateli. Pri nespravnem PINu vyvola WebAppExceptionSupplier.
login	LoginDetails	@return LoginDetails s bezpecostnim tokenem a informacemi o prihlasenem uzivateli Parametry: <b>pinObject: PIN</b> - Deserializovany PIN

#### 1.1.4.7.2 Class OrderInResource

Implementuje REST koncové body aplikace, které umožňují uživatelům vyhledávat a upravovat logistické objednávky.

Název atributu	Datový typ	Popis
----------------	------------	-------



Název atributu	Datový typ	Popis
LOGGER	Logger	
orderInOps	OrderInOperations	
Název metody	Návratový typ	Popis
closeOrder	OrderInRepresentation	Uzavře logistickou objednávku @return Logistická objednávka Parametry: <b>user: User</b> - Uživatel, který operaci provedl Parametry: <b>id: long</b> - ID Parametry: <b>request: OrderInRepresentation</b> - Uzavírána objednávka
getOrder	OrderInRepresentation	Vrací logistickou objednávku se zadaným ID @return Logistická objednávka Parametry: <b>id: long</b> - ID
OrderInResource		Konstruktor. Parametry: <b>orderInOps: OrderInOperations</b> - OrderInOperations s business logikou logistických objednávek
refuseOrder	OrderInRepresentation	Zamítne logistickou objednávku @return Logistická objednávka Parametry: <b>user: User</b> - Uživatel, který operaci provedl Parametry: <b>id: long</b> - ID
updateOrder	OrderInRepresentation	Upraví logistickou objednávku se zadaným ID podle informací v OrderInChange @return Logistická objednávka Parametry: <b>id: long</b> - ID Parametry: <b>change: OrderInChange</b> - Informace o změně

#### 1.1.4.7.3 Class OrderInResource

Implementuje REST koncové body aplikace, které umožňují uživatelům tvořit a vyhledávat logistické objednávky.

Název atributu	Datový typ	Popis
LOGGER	Logger	
orderInOps	OrderInOperations	
Název metody	Návratový typ	Popis
create	OrderInRepresentation	Vytvoří novou logistickou objednávku. @return Logistická objednávka Parametry: <b>user: User</b> - Uživatel, který operaci provedl Parametry: <b>request: OrderInChange</b> - Informace o nové objednávce



Název atributu	Datový typ	Popis
getAll	List<OrderInRepresentation>	Vrati seznam všech logistických objednavek v systému. @return Seznam logistických objednavek
OrderInsResource		Konstruktor. Parametry: <b>orderInOps: OrderInOperations</b> - OrderInOperations s business logikou logistických objednavek

#### 1.1.4.7.4 Class ProductResource

Implementuje REST koncové body aplikace, které umožňují uživatelům vyhledávat a upravovat produkty.

Název atributu	Datový typ	Popis
error404	WebAppExceptionSupplier	
LOGGER	Logger	
productOps	ProductOperations	
Název metody	Návratový typ	Popis
edit	ProductRepresentation	Upraví produkt se zadaným ID. @return Zmenený produkt Parametry: <b>productId: long</b> - ID Parametry: <b>change: ProductChange</b> - Objekt s informacemi o změně
generateRepresentation	ProductRepresentation	Parametry: <b>p: Product</b> -
get	ProductRepresentation	Vrati produkt se zadaným ID. @return Produkt Parametry: <b>productId: long</b> - ID
ProductResource		Konstruktor. Parametry: <b>productOps: ProductOperations</b> - ProductOperations s business logikou produktu

#### 1.1.4.7.5 Class ProductsResource

Implementuje REST koncové body aplikace, které umožňují uživatelům tvořit a vyhledávat produkty.

Název atributu	Datový typ	Popis
LOGGER	Logger	
productOps	ProductOperations	
Název metody	Návratový typ	Popis
createProduct	ProductRepresentation	Přidá do systému zadaný produkt. @return Přidaný produkt (tak, jak vypadá v systému) Parametry: <b>user: User</b> - Uživatel, který operaci provedl



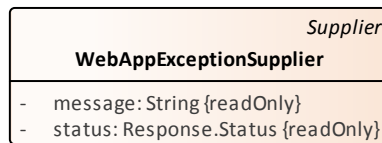
Název atributu	Datový typ	Popis
		Parametry: <b>product: ProductChange</b> - Pridavany produkt
listProducts	List<ProductRepresentation>	Vrati seznam vseh produktu v systemu. @return Seznam produktu
ProductsResource		Konstruktor. Parametry: <b>ops: ProductOperations</b> - ProductOperations s business logikou produktu

#### 1.1.4.7.6 *Class RepresentationConverter*

Obsahuje staticke metody, ktere konvertuji domenove objekty do svych reprezentaci.

Název metody	Návratový typ	Popis
representOrderIn	OrderInRepresentation	Konvertovani pro logisticke objednávky. @return Reprezentace Parametry: <b>order: OrderIn</b> - Logisticka objednávka
representProduct	ProductRepresentation	Konvertovani pro produkty. @return Reprezentace Parametry: <b>p: Product</b> - Produkt
representProduct	ProductRepresentation	Konvertovani pro produkty. @return Reprezentace Parametry: <b>p: Product</b> - Produkt Parametry: <b>positions: List&lt;ProductPosition&gt;</b> - Seznam pozic, na kterych je produkt k nalezeni
representProduct	ProductRepresentation	Konvertovani pro produkty. @return Reprezentace Parametry: <b>li: LinItem</b> - LinItem
representUser	UserRepresentation	Konvertovani pro uzivatele. @return Reprezentace Parametry: <b>u: User</b> - Uzivatel

#### 1.1.4.8 *util*



Obrázek 11 - util

#### 1.1.4.8.1 Class WebAppExceptionSupplier

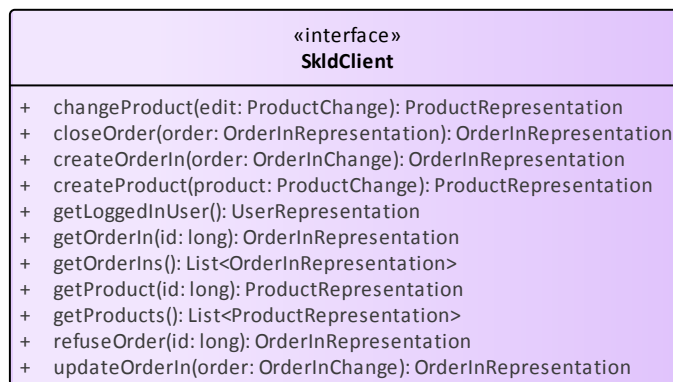
Implementace interfejsu Supplier přes WebAppExceptions.

Název atributu	Datový typ	Popis
message	String	
status	Response.Status	
Název metody	Návratový typ	Popis
get	WebApplicationException	Vrací exception s parametry, které byly specifikovány v konstruktoru.

WebAppExceptionSupplier

Konstruktor. Vytvoří nového Suppliera s přednastavenou WebApplicationException se zadanou zprávou a statusem.  
Parametry:  
**message: String** - Zpráva výjimky  
Parametry:  
**status: Response.Status** - Status výjimky

#### 1.1.5 client



Obrázek 12 - client

#### 1.1.5.1 Interface SkIdClient

Interface pro prezentační vrstvu klienta, který díky němu nemusí komunikovat se serverem přímo. Interface nevystavuje metodu pro přihlášení. Instance datové vrstvy má místo toho být rovnou vytvořena již s přihlášeným uživatelem, díky čemuž si rovnou uchovává přihlašovací tokeny.

Název metody	Návratový typ	Popis
changeProduct	ProductRepresentation	Změní produkt podle vyplněných v ProductChange. V

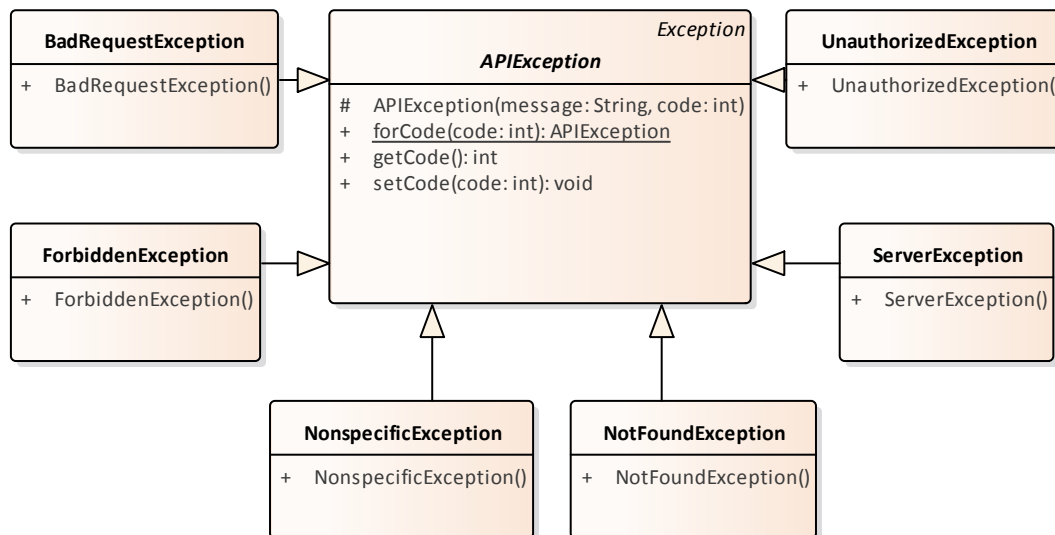


Název metody	Návratový typ	Popis
	n	soucasnosti umoznuje pouze zmenu nazvu produktu. @return Zmeneny produkt Parametry: <b>edit: ProductChange</b> - Objekt reprezentujici parametry zmeneneho produktu (zmenena budou veskera pole, nejen ta vyplnena)
closeOrder	OrderInRepresentation	Nastavi status dane logisticke objednavky jako uzavreny. V budoucnu rovnou zada do systemu pozice predanych produktu. @return Uzavrena logisticka objednavka Parametry: <b>order: OrderInRepresentation</b> - Uzavirana logisticka objednavka
createOrderIn	OrderInRepresentation	Vytvori v systemu novou objednavku podle vyplnenych poli v OrderInChange. Muze byt volana jen uzivatelem s administratorskymi pravy (vedouci smeny). @return Vytvorena logisticka objednavka Parametry: <b>order: OrderInChange</b> - Objekt reprezentujici parametry nove objednavky
createProduct	ProductRepresentation	Prida do systemu nový produkt. @return Vytvoreny produkt Parametry: <b>product: ProductChange</b> - Objekt obsahujici parametry noveho produktu
getLoggedInUser	UserRepresentation	Vraci prave prihlaseneho uzivatele, tedy toho, který provadi veskere operace volane pres tuto instanci. Uzivatel zustava stejny po celou dobu zivota instance. @return Prihlaseny uzivatel
getOrderIn	OrderInRepresentation	Vraci reprezentaci logisticke objednavky s pozadovany ID. @return Logisticka objednavka Parametry: <b>id: long</b> - ID logisticke objednavky
getOrderIns	List<OrderInRepresentation>	Vraci seznam vseh logistickych objednavek v systemu. @return Seznam logistickych objednavek
getProduct	ProductRepresentation	Vraci produkt s pozadovany ID. @return Produkt Parametry: <b>id: long</b> - ID produktu
getProducts	List<ProductRepresentation>	Vraci seznam vseh produktu v systemu. @return Seznam produktu
refuseOrder	OrderInRepresentation	Nastavi status dane logisticke objednavky jako odmitnuty. @return Odmitnuta logisticka objednavka



Název metody	Návratový typ	Popis
		Parametry: <b>id: long</b> - ID odmítané logistické objednávky
		Změní logistickou objednávku podle vyplněných polí v <code>OrderInChange</code> . <b>@return</b> Změněná logistická objednávka
updateOrderIn	OrderInRepresentation	Parametry: <b>order: OrderInChange</b> - Objekt reprezentující parametry změněné objednávky (změněná budou všechna pole, nejen ta vyplněná)

### 1.1.5.2 exceptions



Obrázek 13 - exceptions

#### 1.1.5.2.1 Class APIException

Reprezentace výjimky specifické pro datovou vrstvu klienta.

Název atributu	Datový typ	Popis
code	int	
Název metody	Návratový typ	Popis
APIException		Parametry: <b>message: String</b> - Parametry: <b>code: int</b> -
forCode	APIException	Vytvoří specifický objekt výjimky podle daného chybového kódu. <b>@return</b> Instance výjimky



Název atributu	Datový typ	Popis
		Parametry: <b>code: int</b> - Chybový kod
getCode	int	Vrací chybový kod vyjimky. <b>@return</b> Kod vyjimky
setCode	void	Nastavuje chybový kod vyjimky Parametry: <b>code: int</b> - Kod vyjimky

#### **1.1.5.2.2 Class BadRequestException**

Vyjimka reprezentující nesmyslný požadavek.

Název metody	Návratový typ	Popis
BadRequestException		

#### **1.1.5.2.3 Class ForbiddenException**

Vyjimka reprezentující chybu způsobenou nedostatečnými právy pro provedení operace.

Název metody	Návratový typ	Popis
ForbiddenException		

#### **1.1.5.2.4 Class NonspecificException**

Vyjimka reprezentující neznámou chybu.

Název metody	Návratový typ	Popis
NonspecificException		

#### **1.1.5.2.5 Class NotFoundException**

Vyjimka reprezentující nenalezení požadovaného objektu.

Název metody	Návratový typ	Popis
NotFoundException		

#### **1.1.5.2.6 Class ServerException**

Vyjimka reprezentující chybu na serveru.

Název metody	Návratový typ	Popis
ServerException		

#### **1.1.5.2.7 Class UnauthorizedException**

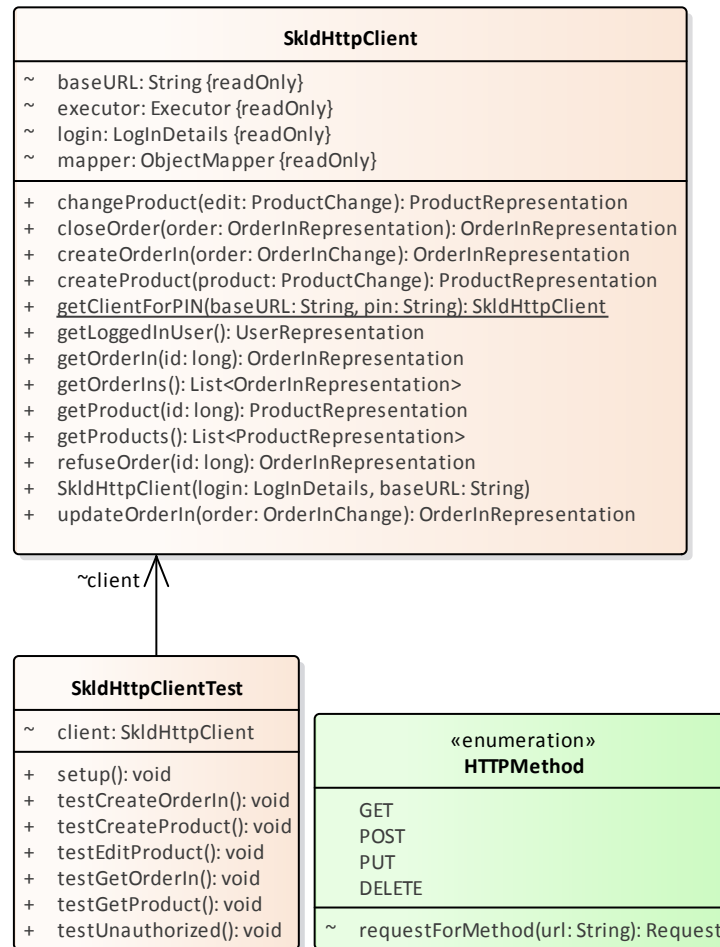
Vyjimka reprezentující chybu, že uživatel není přihlášený.

Název metody	Návratový typ	Popis
UnauthorizedException		





### 1.1.5.3 http



Obrázek 14 - http

#### 1.1.5.3.1 Class SkIdHttpClient

Objekt reprezentující datovou vrstvu aplikace. Založen na komunikaci se serverem přes REST API pomocí HTTP.

Název atributu	Datový typ	Popis
authHeader	String	
baseUrl	String	
executor	Executor	
login	LoginDetails	
mapper	ObjectMapper	
Název metody	Návratový typ	Popis
changeProduct	ProductRepresentation	Změní produkt podle vyplněných v ProductChange. V současnosti umožňuje pouze změnu názvu produktu. @return Změněný produkt Parametry:



Název atributu	Datový typ	Popis
		<b>edit: ProductChange</b> - Objekt reprezentující parametry změněného produktu (změněna budou všechna pole, nejen ta vyplněná)
closeOrder	OrderInRepresentation	Nastaví status dané logistické objednávky jako uzavřenou. V budoucnu rovnou zadá do systému pozici předaných produktů. @return Uzavřená logistická objednávka Parametry: <b>order: OrderInRepresentation</b> - Uzavíraná logistická objednávka
createOrderIn	OrderInRepresentation	Vytvoří v systému novou objednávku podle vyplněných polí v OrderInChange. Může být volána jen uživatelem s administrátorskými právy (vedoucí směny). @return Vytvořená logistická objednávka Parametry: <b>order: OrderInChange</b> - Objekt reprezentující parametry nové objednávky
createProduct	ProductRepresentation	Přidá do systému nový produkt. @return Vytvořený produkt Parametry: <b>product: ProductChange</b> - Objekt obsahující parametry nového produktu
getClientForPIN	SkIdHttpClient	Autentikuje se (se serverem) pomocí předaného PINu a při úspěchu vrátí vytvořenou instanci datové struktury klienta. @return Instance SkIdHttpClienta Parametry: <b>baseUrl: String</b> - URL serveru Parametry: <b>pin: String</b> - PIN kód přihlašovaného uživatele
getLoggedInUser	UserRepresentation	Vrátí právě přihlášeného uživatele, tedy toho, který provádí všechny operace volané přes tuto instanci. Uživatel zůstává stejný po celou dobu života instance. @return Přihlášený uživatel
getOrderIn	OrderInRepresentation	Vrátí reprezentaci logistické objednávky s požadovaným ID. @return Logistická objednávka Parametry: <b>id: long</b> - ID logistické objednávky
getOrderIns	List<OrderInRepresentation>	Vrátí seznam všech logistických objednávek v systému. @return Seznam logistických objednávek
getProduct	ProductRepresentation	Vrátí produkt s požadovaným ID. @return Produkt Parametry: <b>id: long</b> - ID produktu



Název atributu	Datový typ	Popis
getProducts	List<ProductRepresentation>	Vrací seznam všech produktu v systému. @return Seznam produktu
performRequest	Object	Parametry: <b>method: HTTPMethod</b> - Parametry: <b>path: String</b> - Parametry: <b>requestBody: T</b> - Parametry: <b>responseClass: TypeReference</b> -
refuseOrder	OrderInRepresentation	Nastaví status dane logisticke objednávky jako odmítnuty. @return Odmítnuta logistická objednávka Parametry: <b>id: long</b> - ID odmítnuté logisticke objednávky
SkldHttpClient		Vytvoří instanci s přihlášeným uživatelem. Při neplatných přihlasovacích údajích vyvolá UnauthorizedException Parametry: <b>login: LoginDetails</b> - Přihlasovací údaje uživatele Parametry: <b>baseURL: String</b> - URL serveru
updateOrderIn	OrderInRepresentation	Změní logistickou objednávku podle vyplněných polí v OrderInChange. @return Změněná logistická objednávka Parametry: <b>order: OrderInChange</b> - Objekt reprezentující parametry změněné objednávky (změněná budou všechna pole, nejen ta vyplněná)

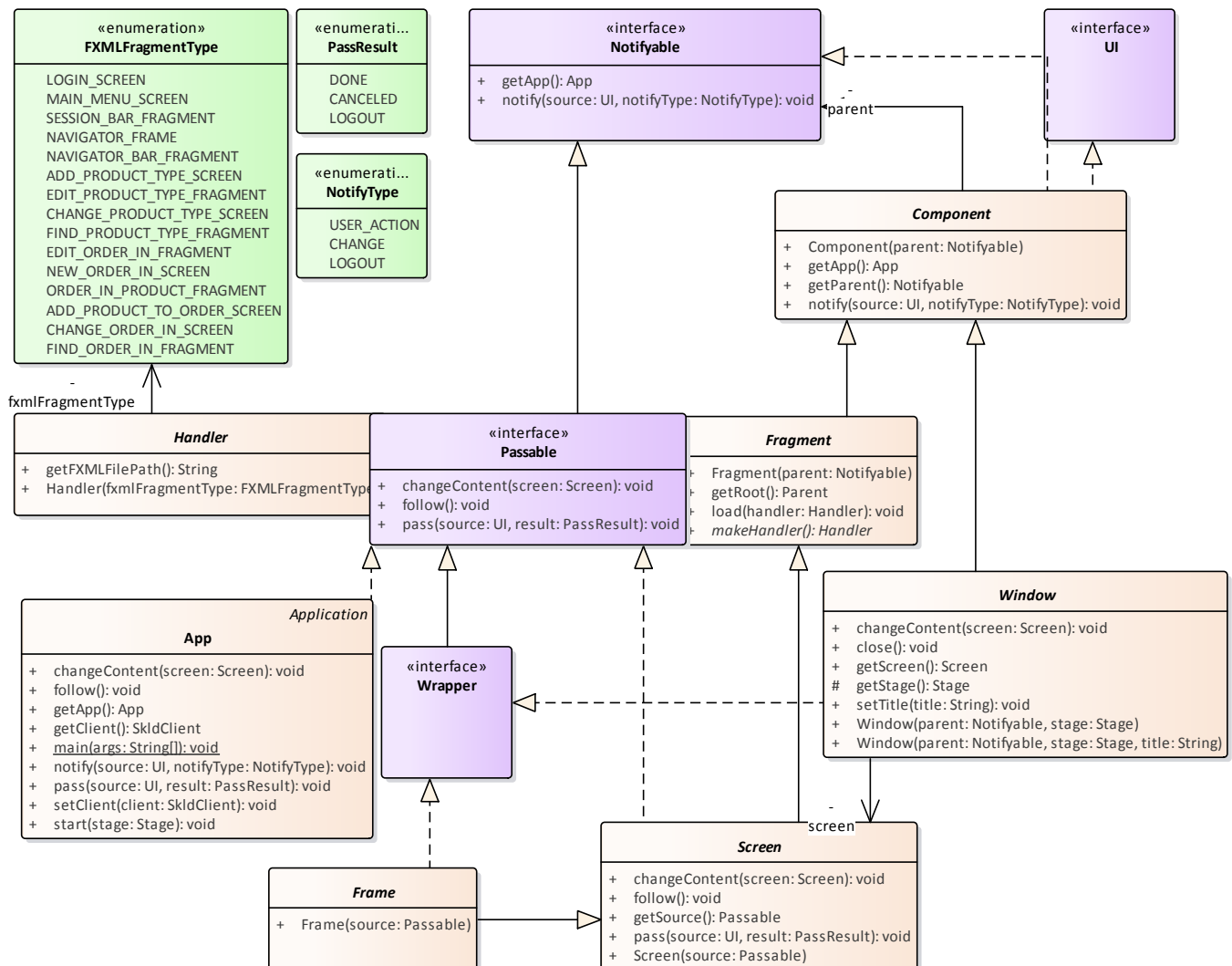
### 1.1.5.3.2 Enumeration HTTPMethod

Rozlišuje HTTP požadavky

Název atributu	Datový typ	Popis
GET		
POST		
PUT		
DELETE		

Název metody	Návratový typ	Popis
requestForMethod	Request	Parametry: <b>url: String</b> -

### 1.1.6 gui



Obrázek 15 - gui

### 1.1.6.1 Class App

Nejvyšší třída a vstupní bod aplikace. Dědí třídu Application definovanou knihovnou JavaFX.

Název atributu	Datový typ	Popis
client	SkldClient	
mainWindow	MainWindow	
Název metody	Návratový typ	Popis
changeContent	void	Parametry: <b>screen: Screen</b> -
follow	void	Prazdna metoda, korenu aplikace nema smysl predat rizeni.
getApp	App	Vraci referenci na celou aplikaci.



Název atributu	Datový typ	Popis
		@return Aplikace
getClient	SkldClient	Vrací SkldClienta aplikace @return SkldClient
main	void	Hlavní metoda. Nacte cestu ke konfiguračnímu souboru a spustí JavaFX aplikaci metodou launch() Parametry: <b>args: String[]</b> - Argumenty předané z příkazové řádky: cesta ke konfiguračnímu souboru
notify	void	App je kořen komponent, notifikace se neposílají dál. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace
pass	void	Předáním řízení kořene komponenty je aplikace ukončena, Parametry: <b>source: UI</b> - Dceřiný objekt Parametry: <b>result: PassResult</b> - Výsledek provedení operací dceřiného objektu
setClient	void	Nastaví SkldClienta, tedy datovou část aplikace Parametry: <b>client: SkldClient</b> - SkldClient
start	void	Vstupní metoda knihovny JavaFX. Vytvoří hlavní okno a předá mu kontrolu. Parametry: <b>stage: Stage</b> - Automaticky vytvořený JavaFX stage

### 1.1.6.2 Class Component

Základní třída všech prvků GUI - oken, obrazovek, tlačítek...

Název atributu	Datový typ	Popis
parent	Notifyable	
Název metody	Návratový typ	Popis
Component		Konstruktor Parametry: <b>parent: Notifyable</b> - Rodič, který obsahuje tuto komponentu (nebo ze kterého byla vytvořena)
getApp	App	Vrací referenci na celou aplikaci. @return Aplikace
getParent	Notifyable	Vrací rodiče, který obsahuje tuto komponentu (nebo ze kterého byla vytvořena) @return Rodič



Název atributu	Datový typ	Popis
notify	void	Preposle notifikaci rodicovskému objektu. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace

### 1.1.6.3 Class Fragment

Fragment uživatelského rozhraní - obaluje GUI prvky obrazovky nebo její části. Ve výsledku zahrnuje tlačítka, textové pole, další fragmenty... Schema každého fragmentu je definováno souborem typu FXML (definováno v JavaFX), který obsahuje grafické schéma zapsané v XML. Každý fragment má vlastní handler (v terminologii JavaFX se jedná o controller), se kterým komunikuje. Handler přímo zpracovává JavaFX objekty a volá jejich metody. Fragment nevolá GUI logiku JavaFX objektu přímo, ale přes handler.

Název atributu	Datový typ	Popis
root	Parent	
Název metody	Návratový typ	Popis
Fragment		Konstruktor Parametry: <b>parent: Notifyable</b> - Rodicovská komponenta, která obsahuje tento fragment
getRoot	Parent	Vrací JavaFX root, kterým si rodič napojí tento fragment @return JavaFX root
load	void	Inicializuje JavaFX GUI a napojuje jej na handler fragmentu. Parametry: <b>handler: Handler</b> - Handler tohoto fragmentu
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvladni tridou dedenou z Fragment) vlastni handler (implementovany zvladni tridou dedenou z Handler). @return Nove vytvoreny handler

### 1.1.6.4 Class Frame

Wrapper, který obsahuje prostor pro obrazovku, která může být nahrazována jinou obrazovkou.

Název metody	Návratový typ	Popis
Frame		Konstruktor Parametry: <b>source: Passable</b> - Rodic, který obsahuje tento ramec

### 1.1.6.5 Class Handler

Trída, která těsně komunikuje s GUI objekty knihovny JavaFX, zpracovává uživatelský vstup a odesílá jej své komponentě. Každý handler (v terminologii JavaFX jde o controller) má vlastní komponentu. Schema GUI je reprezentováno v JavaFX FXML souboru.



Název atributu	Datový typ	Popis
fxmlFragmentType	FXMLFragmentType	
Název metody	Návratový typ	Popis
getFXMLFilePath	String	Vrací cestu k FXML souboru @return Cesta k FXML souboru
Handler		Konstruktor Parametry: <b>fxmlFragmentType: FXMLFragmentType</b> - Typ fragmentu

### 1.1.6.6 Class Screen

Obrazovka je koreňová komponenta Wrapperu (okna, ramu...). Obvykle může být v rámci Wrapperu nahrazena jinou. Může ji být předáno řízení.

Název metody	Návratový typ	Popis
changeContent	void	Prepíše rodičovské komponente požadavek na změnu obrazovky. Parametry: <b>screen: Screen</b> - Nova obrazovka
follow	void	V rodičovském wrapperu zobrazí tuto zobrazení místo libovolně předesele.
getSource	Passable	Vrací rodičovskou komponentu @return Rodičovská komponenta
pass	void	V rodičovském wrapperu zobrazí tuto zobrazení (obvykle místo dcerinnej obrazovky zdroje). Parametry: <b>source: UI</b> - Dcerinny objekt Parametry: <b>result: PassResult</b> - Výsledek provedení operaci dcerinnej objektu
Screen		Konstruktor Parametry: <b>source: Passable</b> - Rodičovská komponenta (obvykle okno), která obsahuje tento fragment

### 1.1.6.7 Class Window

Okno, které může obsahovat nějakou obrazovku.

Název atributu	Datový typ	Popis
screen	Screen	
stage	Stage	
Název metody	Návratový typ	Popis
changeContent	void	Nahradí aktuálně zobrazenou obrazovku za nějakou jinou. Parametry: <b>screen: Screen</b> - Nova obrazovka



Název atributu	Datový typ	Popis
close	void	Zavře okno. Obvykle voláno rodičovským objektem poté, co mu okno předá zpět řízení.
getScreen	Screen	Vrací aktuální obrazovku. @return Aktuální obrazovka
getStage	Stage	Vrací JavaFX stage okna @return JavaFX stage
setTitle	void	Změní text v záhlaví okna. Parametry: <b>title: String</b> - Titulek okna
Window		Konstruktor Parametry: <b>parent: Notifyable</b> - Rodičovský objekt, ze kterého bylo okno vytvořeno Parametry: <b>stage: Stage</b> - JavaFX stage okna
Window		Konstruktor Parametry: <b>parent: Notifyable</b> - Rodičovský objekt, ze kterého bylo okno vytvořeno Parametry: <b>stage: Stage</b> - JavaFX stage okna Parametry: <b>title: String</b> - Text v záhlaví okna

### 1.1.6.8 Interface Notifyable

Rozhraní objektu, kterému může být odesláno oznámení o nějaké události. Obvykle jediné rozhraní, kterým komunikují vestavěné grafické komponenty s rodiči. Vestavěné komponenty pomocí něj obvykle odeslou rodiči oznámení, že u nich došlo ke změně. Ten si další podrobnosti již musí sám načíst z dceřiných komponent.

Název metody	Návratový typ	Popis
getApp	App	Vrací referenci na celou aplikaci. @return Aplikace
notify	void	Oznámí objektu, že došlo k nějaké události. Zavolána na objektu, který je notifikován. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace

### 1.1.6.9 Interface Passable

Rozhraní objektu, kterému může být předáno řízení. Má význam zejména u obrazovek, které se předáním řízení zobrazí místo původní. Takovým objektem je obvykle kontejner primitivních prvků GUI (tlačítek, formulářů), který se stará o komunikaci mezi nimi.





Název metody	Návratový typ	Popis
changeContent	void	Pozaduje po rodicovském objektu změnu obrazovky. Parametry: <b>screen:</b> <b>Screen</b> - Nova obrazovka
follow	void	Predava řízení dcerinnému objektu. Predání řízení znamená, že objekt je plně aktivován a připraven zpracovat uživatelská data. Obvykle to neznamena, že by rodicovský objekt přišel o řízení.
pass	void	Metoda, kterou dcerinny objekt vrácí řízení rodicovskému objektu. Vola ji dcerinny objekt nad rodicovským objektem, obvykle poté, co dokončí požadavek, pro který byl vytvořen. Obvykle to znamená, že rodič může dcerinny objekt zahodit. Parametry: <b>source:</b> <b>UI</b> - Dcerinny objekt Parametry: <b>result:</b> <b>PassResult</b> - Výsledek provedení operaci dcerinného objektu

### 1.1.6.10 Interface UI

Rozhraní objektu reprezentujícího uživatelské rozhraní.

### 1.1.6.11 Interface Wrapper

Rozhraní objektu, které reprezentují grafické komponenty s vymenitelnými částmi (obrazovkami).

### 1.1.6.12 Enumeration FXMLFragmentType

Označuje typ fragmentu, každý fragment má obvykle svůj vlastní. Díky tomuto systému je fragment schopen identifikovat svůj FXML soubor.

Název atributu	Datový typ	Popis
LOGIN_SCREEN		
MAIN_MENU_SCREEN		
SESSION_BAR_FRAGMENT		
NAVIGATOR_FRAME		
NAVIGATOR_BAR_FRAGMENT		
ADD_PRODUCT_TYPE_SCREEN		
EDIT_PRODUCT_TYPE_FRAGMENT		
CHANGE_PRODUCT_TYPE_SCREEN		
FIND_PRODUCT_TYPE_FRAGMENT		
EDIT_ORDER_IN_FRAGMENT		
NEW_ORDER_IN_SCREEN		



Název atributu	Datový typ	Popis
ORDER_IN_PRODUCT_FRAGMENT		
ADD_PRODUCT_TO_ORDER_SCREEN		
CHANGE_ORDER_IN_SCREEN		
FIND_ORDER_IN_FRAGMENT		

### 1.1.6.13 Enumeration NotifyType

Typ notifikace posílané objektu implementujícímu Notifyable. Slouží k odlisění různých notifikací.

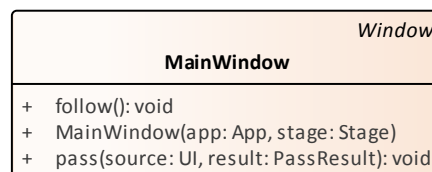
Název atributu	Datový typ	Popis
USER_ACTION		
CHANGE		
LOGOUT		

### 1.1.6.14 Enumeration PassResult

Oznacuje důvod, proč dceřinný objekt předal řízení zpět svému rodiči.

Název atributu	Datový typ	Popis
DONE		
CANCELED		
LOGOUT		

### 1.1.6.15 components



Obrázek 16 - components

#### 1.1.6.15.1 Class MainWindow

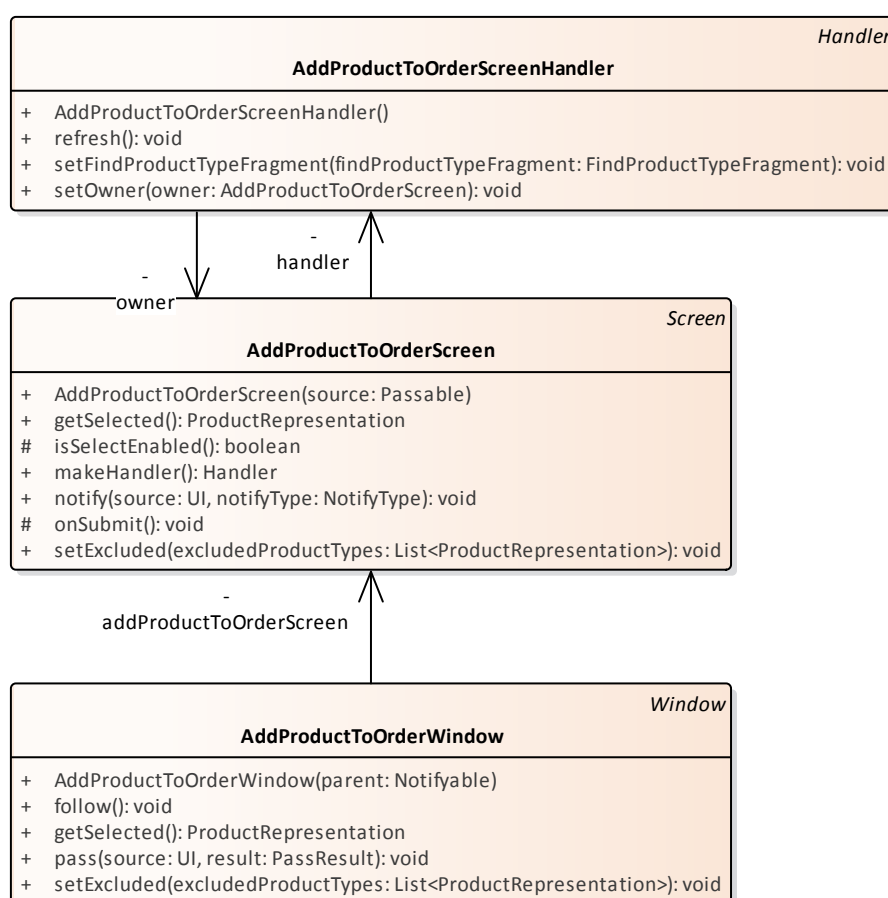
Hlavní okno aplikace. Zobrazeno po jejím spuštění.

Název metody	Návratový typ	Popis
follow	void	Vytvoří přihlasovací obrazovku a předá ji řízení.
MainWindow		Konstruktor Parametry: <b>app: App</b> - Aplikace Parametry: <b>stage: Stage</b> - JavaFX Stage
pass	void	Zavoláno po ukončení činnosti dceřinných komponent. Předá řízení zpět aplikaci.



Název metody	Návratový typ	Popis
		Parametry: <b>source: UI</b> - Dcerinny objekt Parametry: <b>result: PassResult</b> - Vysledek provadeni operaci dcerinneho objektu

### 1.1.6.15.2 *add\_product\_to\_order*



Obrázek 17 - *add\_product\_to\_order*

#### 1.1.6.15.2.1 Class *AddProductToOrderScreen*

Obrazovka pro pro vybrání typu produktu k zarazení do objednávky.

Název atributu	Datový typ	Popis
findProductTypeFragment	FindProductTypeFragment	
handler	AddProductToOrderScreenHandler	



Název atributu Název metody	Datový typ Návratový typ	Popis Popis
AddProductToOrderScreen		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
getSelected	ProductRepresentation	Vrati uzivatelem vybrany Produkt. @return Produkt
isSelectEnabled	boolean	Zjistí zda je uzivateli umozneno vybrat nejaky produkt. @return true vybrani je umozneno
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana během zivota fragmentu pouze jednou. Díky tomu může mít každý fragment (implementovaný zvláštní třídou dedenou z Fragment) vlastní handler (implementovaný zvláštní třídou dedenou z Handler). @return Nove vytvoreny handler
notify	void	Zachyti a zpracuje notifikace o udalostech v dcerinnych objektech. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace
onSubmit	void	Zavolano pri potvrzeni uzivatelem. Informuje rodicovskou komponentu o zmene.
setExcluded	void	Zadane typy produktu nebudou zobrazeny v seznamu. Parametry: <b>excludedProductTypes: List&lt;ProductRepresentation&gt;</b> - Exkludovane typy
setFindProductTypeFragment	void	Nastavi FindProductTypeFragment pro vyhledani typu produktu. Parametry: <b>findProductTypeFragment: FindProductTypeFragment</b> - FindProductTypeFragment

#### 1.1.6.15.2.2 Class AddProductToOrderScreenHandler

Handler pro AddProductToOrderScreen.

Název atributu	Datový typ	Popis
findProductType	AnchorPane	
owner	AddProductToOrderScreen	
submitButton	Button	
Název metody	Návratový typ	Popis
AddProductToOrderScreenHandler		Konstruktor.



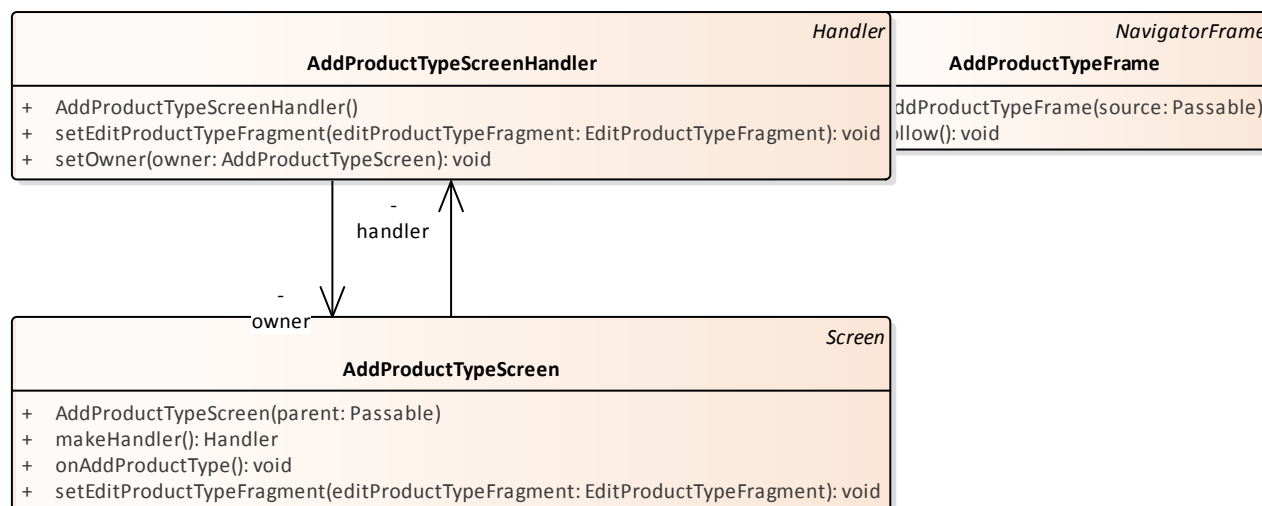
Název atributu	Datový typ	Popis
handleSubmitButtonAction	void	Parametry: <b>event: ActionEvent</b> -
refresh	void	Aktualizuje zobrazené hodnoty.
setFindProductTypeFragment	void	Nastaví FindProductTypeFragment pro vyhledání typu produktu. Parametry: <b>findProductTypeFragment: FindProductTypeFragment</b> - FindProductTypeFragment
setOwner	void	Připojí k handleru příslušný screen. Parametry: <b>owner: AddProductToOrderScreen</b> - Fragment

#### 1.1.6.15.2.3 Class AddProductToOrderWindow

Okno obsahující obrazovku pro vybrání typu produktu k zařazení do objednávky.

Název atributu	Datový typ	Popis
addProductToOrderScreen	AddProductToOrderScreen	
Název metody	Návratový typ	Popis
AddProductToOrderWindow		Konstruktor. Parametry: <b>parent: Notifyable</b> - Rodicovský objekt
follow	void	Předá řízení obrazovce pro přidání produktu.
getSelected	ProductRepresentation	Vrátí uživatelem vybraný Produkt. @return Produkt
pass	void	Předá řízení zpět oknu. Parametry: <b>source: UI</b> - Dcerinny objekt Parametry: <b>result: PassResult</b> - Výsledek provedení operací dcerinného objektu
setExcluded	void	Zadané typy produktu nebudou zobrazeny v seznamu. Parametry: <b>excludedProductTypes: List&lt;ProductRepresentation&gt;</b> - Exkludované typy

#### 1.1.6.15.3 add\_product\_type\_screen



Obrázek 18 - add\_product\_type\_screen

#### 1.1.6.15.3.1 Class AddProductTypeFrame

Wrapper, přidává okolo screenu navigační listu a session bar.

Název metody	Návratový typ	Popis
AddProductTypeFrame		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
follow	void	Vytvoří obrazovku pro přidání nového typu produktu a předá ji řízení.

#### 1.1.6.15.3.2 Class AddProductTypeScreen

Obrazovka pro přidání nového typu produktu.

Název atributu	Datový typ	Popis
editProductTypeFragment	EditProductTypeFragment	
handler	AddProductTypeScreenHandler	
Název metody	Návratový typ	Popis
AddProductTypeScreen		Konstruktor. Parametry: <b>parent: Passable</b> - Rodicovsky objekt
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). @return Nove vytvoreny handler
onAddProductType	void	Zavolano pri potvrzeni odeslani uzivatelem. V pripade zadani



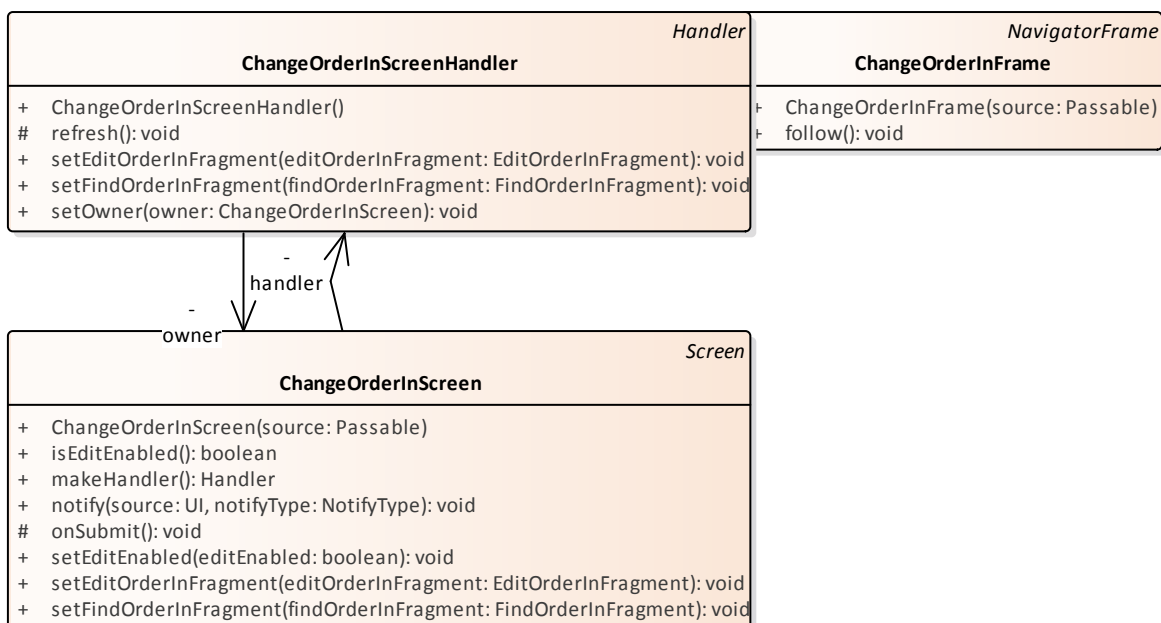
Název atributu	Datový typ	Popis
		nespravného ID zobrazí alert s chybou, jinak data odesle na server a vrátí uživatele do hlavního menu.
setEditProductTypeFragment	void	Nastaví EditProductTypeFragment pro editaci typu produktu. Parametry: <b>editProductTypeFragment:</b> EditProductTypeFragment - EditProductTypeFragment

#### 1.1.6.15.3.3 Class AddProductTypeScreenHandler

Handler pro AddProductTypeScreen.

Název atributu	Datový typ	Popis
editProductType	AnchorPane	
owner	AddProductTypeScreen	
Název metody	Návratový typ	Popis
AddProductTypeScreenHandler		Konstruktor.
handleSubmitButtonAction	void	Parametry: <b>event:</b> ActionEvent -
setEditProductTypeFragment	void	Nastaví EditProductTypeFragment pro editaci typu produktu. Parametry: <b>editProductTypeFragment:</b> EditProductTypeFragment - EditProductTypeFragment
setOwner	void	Připojí k handleru příslušný screen. Parametry: <b>owner:</b> AddProductTypeScreen - Fragment

#### 1.1.6.15.4 change\_order\_in\_screen



Obrázek 19 - change\_order\_in\_screen

#### 1.1.6.15.4.1 Class ChangeOrderInFrame

Wrapper, přidává okolo screenu navigační listu a session bar.

Název metody	Návratový typ	Popis
ChangeOrderInFrame		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
follow	void	Vytvoří obrazovku pro změnu logistických objednávek a předá ji řízení.

#### 1.1.6.15.4.2 Class ChangeOrderInScreen

Obrazovka pro zobrazení existujících typu logistických objednávek a jejich změnu.

Název atributu	Datový typ	Popis
editEnabled	boolean	
editOrderInFragment	EditOrderInFragment	
findOrderInFragment	FindOrderInFragment	
handler	ChangeOrderInScreen Handler	
Název metody	Návratový typ	Popis
ChangeOrderInScreen		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
isEnabled	boolean	Zjistí, zda je uživateli umožněna editace zobrazených hodnot. @return true editace umožněna





Název atributu	Datový typ	Popis
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). @return Nove vytvoreny handler
notify	void	Zachyti a zpracuje notifikace o udalostech v dcerinnych objektech. Parametry: <b>source:</b> <b>UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType:</b> <b>NotifyType</b> - Typ notifikace
onSubmit	void	Zavolano pri odeslani formulare uzivatelem. Pri neplatnem ID uzivatele upozorni na chybu, jinak odesle zmenu na server.
reset	void	Vrati objekt do pocatecniho stavu.
setEditEnabled	void	Nastavi, zda je uzivateli umoznena editace zobrazenych hodnot. Parametry: <b>editEnabled:</b> <b>boolean</b> - true editace bude umoznena
setEditOrderInFragment	void	Nastavi EditOrderInFragment pro editaci existujici logisticke objednavky. Parametry: <b>editOrderInFragment:</b> <b>EditOrderInFragment</b> - EditOrderInFragment
setFindOrderInFragment	void	Nastavi FindOrderInFragment pro vyhledani logisticke objednavky. Parametry: <b>findOrderInFragment:</b> <b>FindOrderInFragment</b> - FindOrderInFragment

#### 1.1.6.15.4.3 Class ChangeOrderInScreenHandler

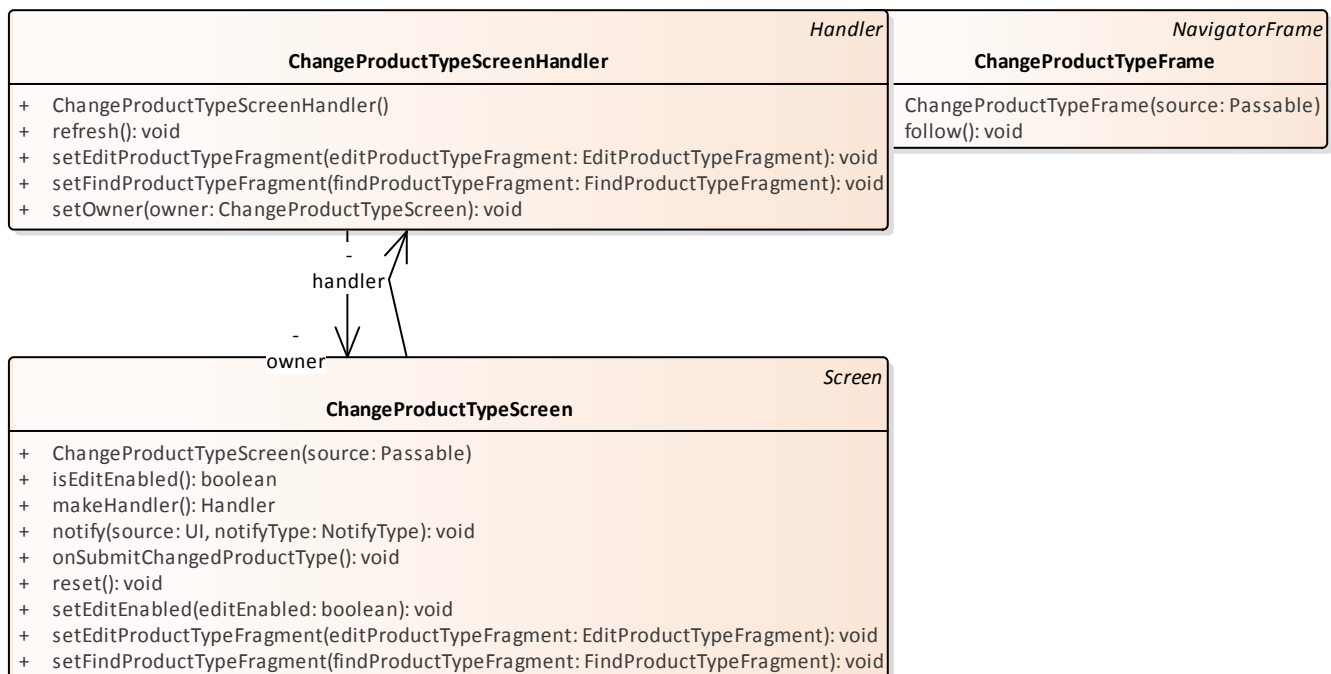
Handler pro ChangeOrderInScreen.

Název atributu	Datový typ	Popis
editOrderIn	AnchorPane	
findOrderIn	AnchorPane	
owner	ChangeOrderInScreen	
submitButton	Button	
Název metody	Návratový typ	Popis
ChangeOrderInScreenHandler		Konstruktor.
handleSubmitButtonAction	void	Parametry: <b>event:</b> <b>ActionEvent</b> -



Název atributu	Datový typ	Popis
refresh	void	Aktualizuje zobrazené hodnoty.
setEditOrderInFragment	void	Nastaví EditOrderInFragment pro editaci existující logistické objednávky. Parametry: <b>editOrderInFragment: EditOrderInFragment</b> - EditOrderInFragment
setFindOrderInFragment	void	Nastaví FindOrderInFragment pro vyhledání logistické objednávky. Parametry: <b>findOrderInFragment: FindOrderInFragment</b> - FindOrderInFragment
setOwner	void	Připojí k handleru příslušný screen. Parametry: <b>owner: ChangeOrderInScreen</b> - Fragment

#### 1.1.6.15.5 change\_product\_type\_screen



Obrázek 20 - change\_product\_type\_screen

##### 1.1.6.15.5.1 Class ChangeProductTypeFrame



Wrapper, přidává okolo screenu navigační listu a session bar.

Název metody	Návratový typ	Popis
ChangeProductTypeFrame		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
follow	void	Vytvoří obrazovku pro změnu existujících typu produktu a předá ji řízení.

#### 1.1.6.15.5.2 Class ChangeProductTypeScreen

Obrazovka pro zobrazení existujících typu produktu a jejich změnu.

Název atributu	Datový typ	Popis
editEnabled	boolean	
editProductTypeFragment	EditProductTypeFragment	
findProductTypeFragment	FindProductTypeFragment	
handler	ChangeProductTypeScreenHandler	
Název metody	Návratový typ	Popis
changeEditingProductType	void	Parametry: <b>product: ProductRepresentation</b> -
ChangeProductTypeScreen		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
isEditEnabled	boolean	Zjistí, zda je uživateli umožněna editace zobrazených hodnot. @return true editace umožněna
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany vlastni tridou dedenou z Fragment) vlastni handler (implementovany vlastni tridou dedenou z Handler). @return Nove vytvoreny handler
notify	void	Zachyti a zpracuje notifikace o udalostech v dcerinnych objektech. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace
onSubmitChangedProductType	void	Zavolano pri odeslani formulare uzivatelem. Odesle zmenu na server.
reset	void	Vraci objekt do pocatecniho stavu.
setEditEnabled	void	Nastavi, zda je uzivateli umoznena editace zobrazenych



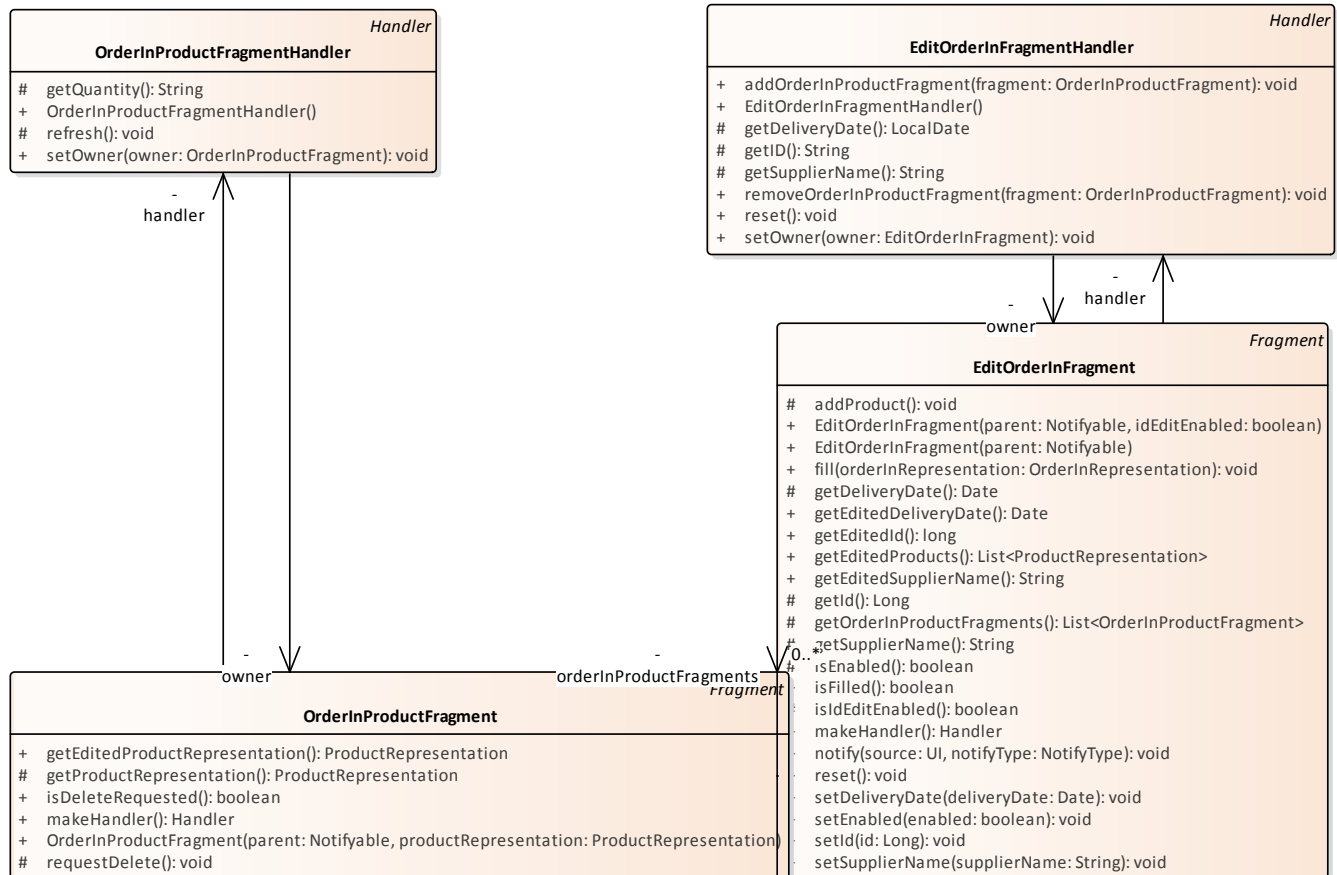
Název atributu	Datový typ	Popis
		hodnot. Parametry: <b>editEnabled: boolean</b> - true editace bude umoznena
setEditProductTypeFragment	void	Nastavi EditProductTypeFragment pro editaci existujiciho typu produktu. Parametry: <b>editProductTypeFragment: EditProductTypeFragment</b> - EditProductTypeFragment
setFindProductTypeFragment	void	Nastavi FindProductTypeFragment pro vyhledani typu produktu. Parametry: <b>findProductTypeFragment: FindProductTypeFragment</b> - FindProductTypeFragment

#### 1.1.6.15.5.3 Class ChangeProductTypeScreenHandler

Handler pro ChangeProductTypeScreen.

Název atributu	Datový typ	Popis
editProductType	AnchorPane	
findProductType	AnchorPane	
owner	ChangeProductTypeScreen	
submitButton	Button	
Název metody	Návratový typ	Popis
ChangeProductTypeScreenHandler		Konstruktor.
handleSubmitButtonAction	void	Parametry: <b>event:(ActionEvent)</b> -
refresh	void	Aktualizuje zobrazene hodnoty.
setEditProductTypeFragment	void	Nastavi EditProductTypeFragment pro editaci existujiciho typu produktu. Parametry: <b>editProductTypeFragment: EditProductTypeFragment</b> - EditProductTypeFragment
setFindProductTypeFragment	void	Nastavi FindProductTypeFragment pro vyhledani typu produktu. Parametry: <b>findProductTypeFragment: FindProductTypeFragment</b> - FindProductTypeFragment
setOwner	void	Pripoji k handleru prislusny screen. Parametry: <b>owner: ChangeProductTypeScreen</b> - Fragment

### 1.1.6.15.6 edit\_order\_in



Obrázek 21 - edit\_order\_in

#### 1.1.6.15.6.1 Class EditOrderInFragment

Fragment pro editaci (existující/nové) logistické objednávky.

Název atributu	Datový typ	Popis
addProductToOrderWin	AddProductToOrderWi	
dow	ndow	
deliveryDate	Date	
enabled	boolean	
filled	boolean	
handler	EditOrderInFragmentH	
id	long	
idEditEnabled	boolean	
orderInProductFragment	List<OrderInProductFr	
s	agment>	
supplierName	String	
Název metody	Návratový typ	Popis



Název atributu	Datový typ	Popis
addOrderInProductFragment	void	Parametry: <b>fragment: OrderInProductFragment</b> -
addProduct	void	Prida typ produktu do seznamu zobrazenych typu produktu.
EditOrderInFragment		Konstruktor. Parametry: <b>parent: Notifiable</b> - Rodicovsky objekt Parametry: <b>idEditEnabled: boolean</b> - true umozni editaci ID typu produktu
EditOrderInFragment		Konstruktor. Parametry: <b>parent: Notifiable</b> - Rodicovsky objekt
fill	void	Vyplni polozky podle hodnot dane logisticke objednávky Parametry: <b>orderInRepresentation: OrderInRepresentation</b> - Logisticka objednávka
getDeliveryDate	Date	Vraci ocekavane datum doruceni puvodni editovane logisticke objednávky. @return Ocekavane datum doruceni
getEditedDeliveryDate	Date	Vraci uzivatelem zadane datum ocekavaneho doruceni logisticke objednávky. @return Datum ocekavaneho doruceni
getEditedId	long	Pokusi se vratit uzivatelem zadane ID, pri nespravnem formatu zobrazi alert a vyhodi vyjimku. @return Zadane ID
getEditedProducts	List<ProductRepresentation>	Vraci uzivatelem zadane produkty. @return Produkty
getEditedSupplierName	String	Vraci uzivatelem zadany nazev dodavatele logisticke objednávky. @return Nazev dodavatele
getId	Long	Vraci puvodni ID editovane logisticke objednávky. @return ID
getOrderInProductFragments	List<OrderInProductFragment>	Vraci fragmenty veskerych produktu obsazenych v aktualne editovane logisticke objednavce. @return Fragmenty produktu
getProducts	List<ProductRepresentation>	
getSupplierName	String	Vraci puvodni jmeno dodavatele editovane objednávky.



Název atributu	Datový typ	Popis
		@return Jmeno dodavatele
isEnabled	boolean	Zjistuje zda je umoznena editace. @return true editace umoznena
isFilled	boolean	Zjistuje zda byly polozky formulare naplneny nejakym existujicim typem produktu (pri jeho zmene). @return true formular byl naplnen
isIdEditEnabled	boolean	Zjistuje zda je umoznena editace ID typu produktu. @return true editace umoznena
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). @return Nove vytvoreny handler
notify	void	Zachyti a zpracuje notifikace o udalostech v dcerinnych objektech. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace
onProductAdded	void	
removeDeletedProducts	void	
reset	void	Vrati objekt do pocatecniho stavu.
setDeliveryDate	void	Nastavi predzobrazene datum doruceni puvodni editovane logisticke objednavky. Parametry: <b>deliveryDate: Date</b> - Predpokladane datum doruceni
setEnabled	void	Umozni/znemozni editaci policek formulare. Parametry: <b>enabled: boolean</b> - true editace bude umoznena
setId	void	Nastavi puvodni ID logisticke objednavky. Parametry: <b>id: Long</b> - ID
setSupplierName	void	Nastavi predzobrazene jmeno dodavatele editovane objednavky. Parametry: <b>supplierName: String</b> - Jmeno dodavatele



#### 1.1.6.15.6.2 Class EditOrderInFragmentHandler

Handler pro EditOrderInFragment.

Název atributu	Datový typ	Popis
addProductButton	Button	
inputDeliveryDate	DatePicker	
inputID	TextField	
inputSupplierName	TextField	
owner	EditOrderInFragment	
productsVBox	VBox	
Název metody	Návratový typ	Popis
addOrderInProductFragment	void	Přidá fragment reprezentující nějaký produkt. Parametry: <b>fragment: OrderInProductFragment</b> - Fragment reprezentující produkt
EditOrderInFragmentHandler		Konstruktor.
getDeliveryDate	LocalDate	
getID	String	
getSupplierName	String	
handleAddProductButtonAction	void	Parametry: <b>event: ActionEvent</b> -
removeOrderInProductFragment	void	Smaze fragment reprezentující dany produkt. Parametry: <b>fragment: OrderInProductFragment</b> - Fragment reprezentující produkt
reset	void	Vrátí objekt do počátečního stavu.
resetInputDeliveryDate	void	
setOwner	void	Připojí k handleru příslušný fragment. Parametry: <b>owner: EditOrderInFragment</b> - Fragment

#### 1.1.6.15.6.3 Class OrderInProductFragment

Fragment reprezentující produkt obsazený v aktuálně editované logistické objednávce.

Název atributu	Datový typ	Popis
deleteRequested	boolean	
handler	OrderInProductFragmentHandler	
productRepresentation	ProductRepresentation	
Název metody	Návratový typ	Popis
getEditedProductRepresentation	ProductRepresentation	Vrátí editovaný produkt s jeho zadaným množstvím.





Název atributu	Datový typ	Popis
ntation	n	@return Produkt
getProductRepresentatio n	ProductRepresentatio n	Vrací původní editovaný produkt @return Produkt
isDeleteRequested	boolean	Zjistí, zda uživatel odstranil produkt ze seznamu. @return true Produkt má být odstraněn
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smí být zavolána během života fragmentu pouze jednou. Díky tomu může mít každý fragment (implementovaný zvláštní třídou dedenou z Fragment) vlastní handler (implementovaný zvláštní třídou dedenou z Handler). @return Nově vytvořený handler
OrderInProductFragment		Konstruktor. Parametry: <b>parent: Notifyable</b> - Rodicovsky objekt Parametry: <b>productRepresentation: ProductRepresentation</b> - Typ produktu
requestDelete	void	Zavolano pri pozadavku uzivatele o odstraneni produkt ze seznamu.
setProductRepresentatio n	void	Parametry: <b>productRepresentation: ProductRepresentation</b> -

#### 1.1.6.15.6.4 Class OrderInProductFragmentHandler

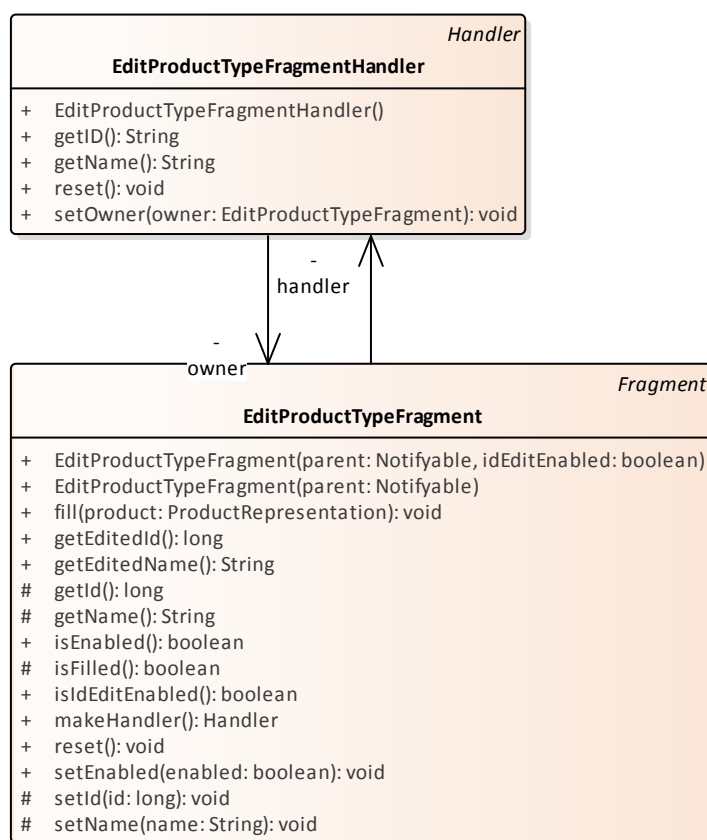
Handler pro OrderInProductFragment.

Název atributu	Datový typ	Popis
owner	OrderInProductFragm ent	
productId	Label	
productName	Label	
productQuantity	TextField	
Název metody	Návratový typ	Popis
getQuantity	String	Vrací uživatelem zadané množství produktu. @return Množství
handleRemoveLinkActio n	void	Parametry: <b>event: ActionEvent</b> -
OrderInProductFragment Handler		Konstruktor.
refresh	void	Aktualizuje zobrazené hodnoty.



Název atributu	Datový typ	Popis
setOwner	void	Připojí k handleru příslušný fragment. Parametry: <b>owner: OrderInProductFragment</b> - Fragment

### 1.1.6.15.7 edit\_product\_type



Obrázek 22 - edit\_product\_type

#### 1.1.6.15.7.1 Class EditProductTypeFragment

Fragment pro editaci (nového/existujícího) typu produktu.

Název atributu	Datový typ	Popis
enabled	boolean	
filled	boolean	
handler	EditProductTypeFragmentHandler	
id	long	
idEditEnabled	boolean	
name	String	
Název metody	Návratový typ	Popis



Název atributu	Datový typ	Popis
EditProductTypeFragment		Konstruktor. Parametry: <b>parent: Notifyable</b> - Rodicovsky objekt Parametry: <b>idEditEnabled: boolean</b> - true umožni editaci ID typu produktu
EditProductTypeFragment		Konstruktor. Parametry: <b>parent: Notifyable</b> - Rodicovsky objekt
fill	void	Vyplni položky podle hodnot daného typu produktu Parametry: <b>product: ProductRepresentation</b> - Typ produktu
getEditedId	long	Pokusí se vrátit uživatelem zadane ID, při nesprávném formátu zobrazí alert a vyhodí výjimku. @return Zadane ID
getEditedName	String	Vrací uživatelem zadane jméno typu produktu. @return Jméno typu produktu
getId	long	Vrací původní ID editovaného typu produktu. @return ID
getName	String	Vrací původní jméno editovaného typu produktu. @return Jméno typu produktu
isEnabled	boolean	Zjistuje zda je umožněna editace. @return true editace umožněna
isFilled	boolean	Zjistuje zda byly položky formuláře naplněny nějakým existujícím typem produktu (při jeho změně). @return true formulář byl naplněn
isIdEditEnabled	boolean	Zjistuje zda je umožněna editace ID typu produktu. @return true editace umožněna
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smí být zavolána během života fragmentu pouze jednou. Díky tomu může mít každý fragment (implementovaný zvláštní třídou dedenou z Fragment) vlastní handler (implementovaný zvláštní třídou dedenou z Handler). @return Nově vytvořený handler
reset	void	Vrátí objekt do počátečního stavu.
setEnabled	void	Umožní/zneumožní editaci políček formuláře. Parametry: <b>enabled: boolean</b> - true editace bude umožněna



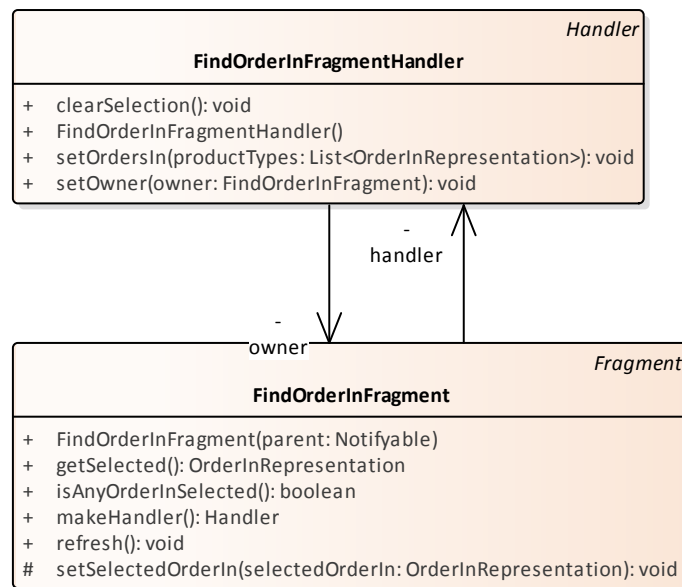
Název atributu	Datový typ	Popis
setId	void	Nastavi puvodni ID editovaneho typu produktu. Parametry: <b>id: long</b> - ID
setName	void	nastavi puvodni jmeno editovaneho typu produktu. Parametry: <b>name: String</b> - Jmeno typu produktu

#### 1.1.6.15.7.2 Class EditProductTypeFragmentHandler

Handler pro EditProductTypeFragment.

Název atributu	Datový typ	Popis
inputID	TextField	
inputName	TextField	
owner	EditProductTypeFragment	
Název metody	Návratový typ	Popis
EditProductTypeFragmentHandler		Konstruktor.
getID	String	Vraci uzivatelem vyplnene ID typu produktu. @return ID typu produktu
getName	String	Vraci uzivatelem zadane jmeno typu produktu. @return Jmeno typu produktu
reset	void	Vraci objekt do pocatecniho stavu.
setOwner	void	Pripoji k handleru prislusny fragment. Parametry: <b>owner: EditProductTypeFragment</b> - Fragment

#### 1.1.6.15.8 *find\_order\_in*



Obrázek 23 - find\_order\_in

#### 1.1.6.15.8.1 Class FindOrderInFragment

Fragment pro vyhledání existujícího typu logistické objednávky.

Název atributu	Datový typ	Popis
handler	FindOrderInFragment Handler	
ordersIn	List<OrderInRepresentation>	
selectedOrderIn	OrderInRepresentation	
Název metody	Návratový typ	Popis
FindOrderInFragment		Konstruktor. Parametry: <b>parent: Notifyable</b> - Rodicovsky objekt
getSelected	OrderInRepresentation	Vrátí uživatelem označenou objednávku. @return Oznacena objednávka
isAnyOrderInSelected	boolean	Zjistí, zda uživatel označil nějakou objednávku ze seznamu @return true pokud existuje označená objednávka
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvladstni tridou dedenou z Fragment) vlastni handler (implementovany zvladstni tridou dedenou z Handler). @return Nove vytvoreny handler
refresh	void	Nacte seznam vseh logistickych objednavek a zobrazi jej.
setSelectedOrderIn	void	Oznaci objednavku jako oznacenu uzivatelem.



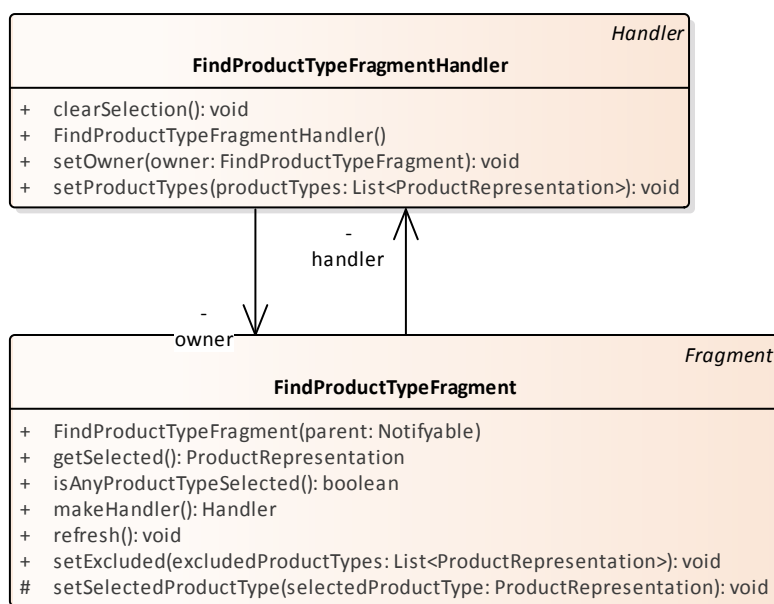
Název atributu	Datový typ	Popis
		Parametry: <b>selectedOrderIn:</b> <b>OrderInRepresentation</b> - Oznacena objednávka

#### 1.1.6.15.8.2 Class FindOrderInFragmentHandler

Handler pro FindOrderInFragment.

Název atributu	Datový typ	Popis
createdByColumn	TableColumn<OrderInRepresentation, String>	
idColumn	TableColumn<OrderInRepresentation, Long>	
orderInTable	TableView<OrderInRepresentation>	
owner	FindOrderInFragment	
stateColumn	TableColumn<OrderInRepresentation, String>	
supplierNameColumn	TableColumn<OrderInRepresentation, String>	
Název metody	Návratový typ	Popis
clearSelection	void	Zadny polozky oznacene uzivatelem jiz nebudou oznaceny.
FindOrderInFragmentHandler		Konstruktor.
handleOrderInClickedAction	void	
initialize	void	Vola JavaFX pri inicializaci handleru.
setOrdersIn	void	Zobrazí seznam logistických objednávek. Parametry: <b>productTypes:</b> <b>List&lt;OrderInRepresentation&gt;</b> - Logistické objednávky ke zobrazení
setOwner	void	Připojí k handleru příslušný fragment. Parametry: <b>owner:</b> <b>FindOrderInFragment</b> - Fragment

#### 1.1.6.15.9 find\_product\_type



Obrázek 24 - find\_product\_type

#### 1.1.6.15.9.1 Class FindProductTypeFragment

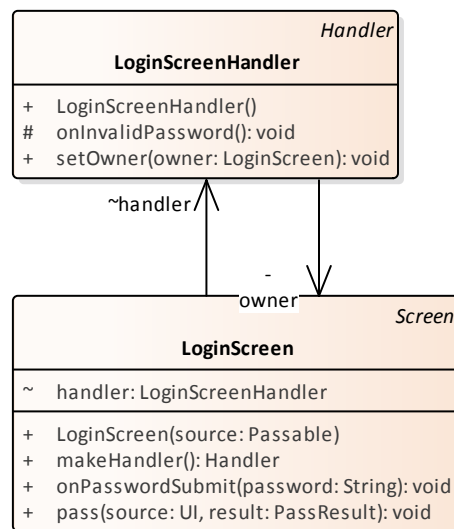
Fragment pro vyhledání existujícího typu produktu.

Název atributu	Datový typ	Popis
excludedProductTypes	List<ProductRepresentation>	
handler	FindProductTypeFragmentHandler	
productTypes	List<ProductRepresentation>	
selectedProductType	ProductRepresentation	
Název metody	Návratový typ	Popis
FindProductTypeFragment		Konstruktor. Parametry: <b>parent: Notifyable</b> - Rodicovsky objekt
getSelected	ProductRepresentation	Vrátí uživatelem označený produkt. <b>@return</b> Označený produkt
isAnyProductTypeSelected	boolean	Zjistí, zda uživatel označil nějaký produkt ze seznamu <b>@return</b> true pokud existuje označený produkt
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). <b>@return</b> Nove vytvoreny handler



Název atributu	Datový typ	Popis
refresh	void	Nacte seznam vseh typu produktu a zobrazi jej (s vynechanim typu oznacenych jako excluded).
setExcluded	void	Zadane produkty nebudou zobrazeny v seznamu. Parametry: <b>excludedProductTypes: List&lt;ProductRepresentation&gt;</b> - Vynechavane zobrazeny
setSelectedProductType	void	Oznaci typ produktu jako oznaceny uzivatelem. Parametry: <b>selectedProductType: ProductRepresentation</b> - Oznaceny typ produktu

### 1.1.6.15.10 login\_screen



Obrázek 25 - login\_screen

#### 1.1.6.15.10.1 Class LoginScreen

Obrazovka pro prihlaseni uzivatele.

Název atributu	Datový typ	Popis
handler	LoginScreenHandler	
Název metody	Návratový typ	Popis
LoginScreen		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy





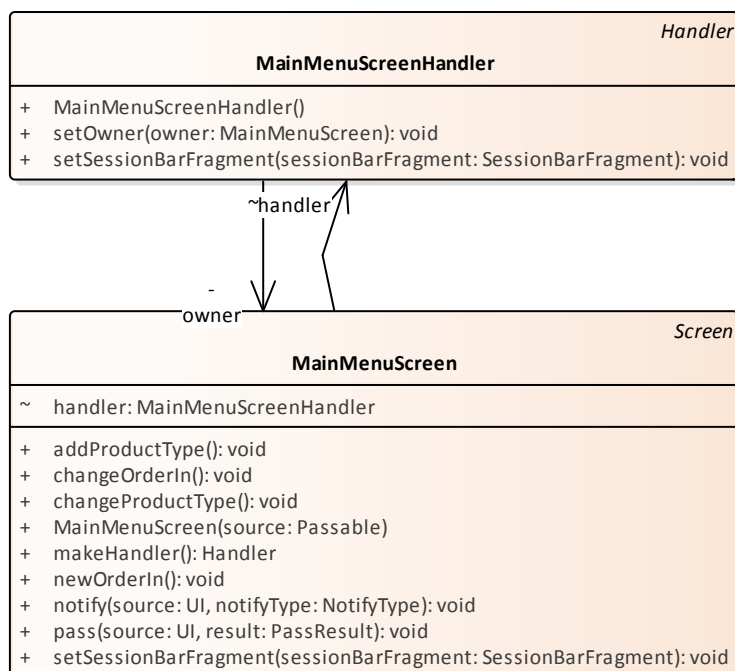
Název atributu	Datový typ	Popis
		fragment (implementovaný zvláštní třídou dedenou z Fragment) vlastní handler (implementovaný zvláštní třídou dedenou z Handler). @return Nove vytvoreny handler
onPasswordSubmit	void	Zavolano po zadani hesla uzivatelem. Pokusi se uzivatele prihlasit a pri neuspechu jej upozorni na nespravne heslo. Po uspesnem prihlaseni uzivatele presune do hlavniho menu. Parametry: <b>password: String</b> - Zadane heslo
pass	void	Vrati rizeni obrazovce. Parametry: <b>source: UI</b> - Dcerinny objekt Parametry: <b>result: PassResult</b> - Vysledek provadeni operaci dcerinneho objektu

#### 1.1.6.15.10.2 Class LoginScreenHandler

Handler pro LoginScreen.

Název atributu	Datový typ	Popis
infoLabel	Label	
owner	LoginScreen	
passwordField	PasswordField	
submitButton	Button	
Název metody	Návratový typ	Popis
handleSubmitButtonAction	void	Parametry: <b>event: ActionEvent</b> -
LoginScreenHandler		Konstruktor.
onInvalidPassword	void	Zablokuje uzivateli na chvili policko pro zadani hesla.
setOwner	void	Pripoji k handleru prislusny screen. Parametry: <b>owner: LoginScreen</b> - Fragment

#### 1.1.6.15.11 main\_menu



Obrázek 26 - main\_menu

#### 1.1.6.15.11.1 Class MainMenuScreen

Obrazovka hlavního menu aplikace. Z ní se uživatel může jedním kliknutím dostat do implementace většiny use cases.

Název atributu	Datový typ	Popis
handler	MainMenuScreenHandler	
sessionBarFragment	SessionBarFragment	
Název metody	Návratový typ	Popis
addProductType	void	Presune uživatele na screen vytvoření nového produktu.
changeOrderIn	void	Presune uživatele na screen změny existující logistické objednávky.
changeProductType	void	Presune uživatele na screen změny existujícího typu produktu.
MainMenuScreen		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovský objekt
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). @return Nove vytvoreny handler
newOrderIn	void	Presune uzivatele na screen vytvoreni nove logisticke



Název atributu	Datový typ	Popis
		objednavky.
notify	void	Zachyti a zpracuje notifikace o udalostech v dcerinnych objektech. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace
pass	void	Vrati rizeni obrazovce. Pokud byl duvodem k predani rizeni pozadavek pro odhlaseni, preda rizeni s timto duvodem rodici. Parametry: <b>source: UI</b> - Dcerinny objekt Parametry: <b>result: PassResult</b> - Vysledek provadeni operaci dcerinneho objektu
setSessionBarFragment	void	Nastavi SessionBarFragment s informacemi o uzivateli. Parametry: <b>sessionBarFragment: SessionBarFragment</b> - SessionBarFragment

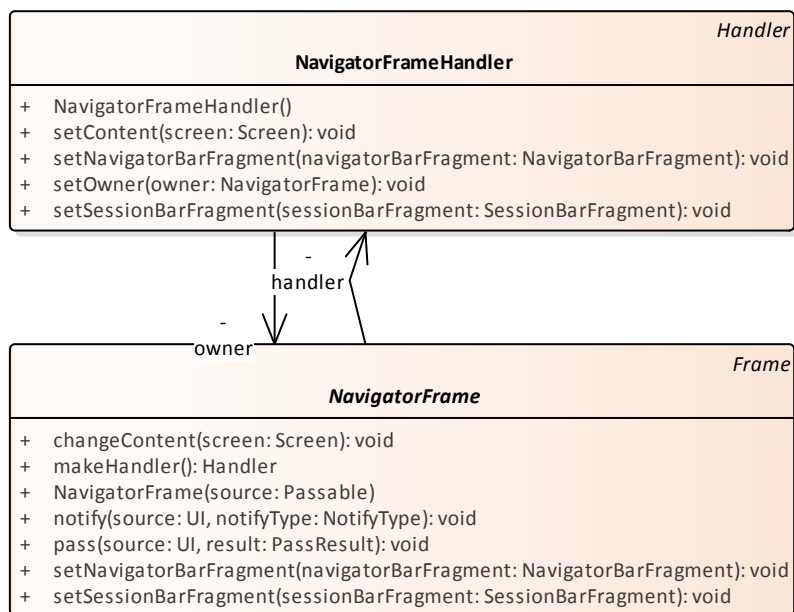
#### 1.1.6.15.11.2 Class MainMenuScreenHandler

Handler pro MainMenuScreen.

Název atributu	Datový typ	Popis
owner	MainMenuScreen	
sessionBar	HBox	
Název metody	Návratový typ	Popis
handleAddProductType ButtonAction	void	Parametry: <b>event: ActionEvent</b> -
handleChangeOrderInBu ttonAction	void	Parametry: <b>event: ActionEvent</b> -
handleChangeProductTy peButtonAction	void	Parametry: <b>event: ActionEvent</b> -
handleNewOrderInButto nAction	void	Parametry: <b>event: ActionEvent</b> -
MainMenuScreenHandle r		Konstruktor.
setOwner	void	Pripoji k handleru prislusny screen. Parametry: <b>owner: MainMenuScreen</b> - Fragment

Název atributu	Datový typ	Popis
setSessionBarFragment	void	Nastavi SessionBarFragment s informacemi o uživateli. Parametry: <b>sessionBarFragment: SessionBarFragment</b> - SessionBarFragment

### 1.1.6.15.12 navigator



Obrázek 27 - navigator

#### 1.1.6.15.12.1 Class NavigatorFrame

Wrapper, který umožňuje navigaci a přidává okolo vestaveného screenu navigační listu a session bar. Navigace probíhá zpravidla o krok o jednu obrazovku dopředu nebo zpět (jako ve webovém prohlížeči).

Název atributu	Datový typ	Popis
content	Screen	
handler	NavigatorFrameHandler	
navigatorBarFragment	NavigatorBarFragment	
sessionBarFragment	SessionBarFragment	
Název metody	Návratový typ	Popis
changeContent	void	Nahradi aktuálně zobrazenou obrazovku za jinou. Parametry: <b>screen: Screen</b> - Nova obrazovka
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment)



Název atributu	Datový typ	Popis
		vlastní handler (implementovány zvláštní třídou dedenou z Handler). @return Nově vytvořený handler
NavigatorFrame		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovský objekt
notify	void	Zachytí a zpracuje notifikace o událostech v dcerinných objektech. Parametry: <b>source: UI</b> - Objekt, který odeslal notifikaci Parametry: <b>notifyType: NotifyType</b> - Typ notifikace
pass	void	Vrátí řízení rodičovskému objektu. Parametry: <b>source: UI</b> - Dcerinný objekt Parametry: <b>result: PassResult</b> - Výsledek provedení operací dcerinného objektu
setNavigatorBarFragment	void	Nastaví fragment s navigační listou. Parametry: <b>navigatorBarFragment: NavigatorBarFragment</b> - Fragment s navigační listou
setSessionBarFragment	void	Nastaví SessionBarFragment s informacemi o uživateli. Parametry: <b>sessionBarFragment: SessionBarFragment</b> - SessionBarFragment

#### 1.1.6.15.12.2 Class NavigatorFrameHandler

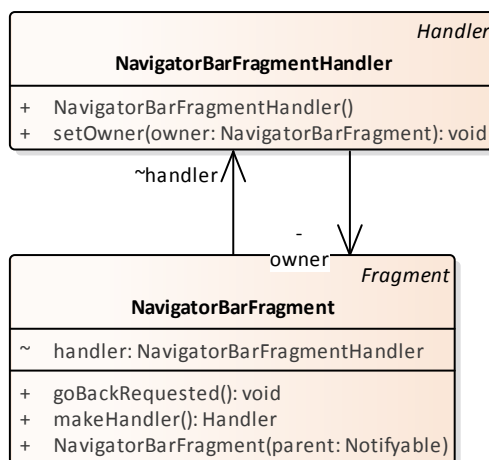
Handler pro NavigatorFrame.

Název atributu	Datový typ	Popis
content	AnchorPane	
navigatorBar	HBox	
owner	NavigatorFrame	
sessionBar	HBox	
Název metody	Návratový typ	Popis
NavigatorFrameHandler		Konstruktor.
setContent	void	Změní aktuálně zobrazený screen. Parametry: <b>screen: Screen</b> - Nový screen
setNavigatorBarFragment	void	Nastaví fragment s navigační listou. Parametry: <b>navigatorBarFragment: NavigatorBarFragment</b> - Fragment



Název atributu	Datový typ	Popis
		s navigacni listou
setOwner	void	Pripoji k handleru prislusny frame. Parametry: <b>owner: NavigatorFrame</b> - Fragment
setSessionBarFragment	void	Nastavi SessionBarFragment s informacemi o uzivateli. Parametry: <b>sessionBarFragment: SessionBarFragment</b> - SessionBarFragment

### 1.1.6.15.12.3 navigator\_bar



Obrázek 28 - navigator\_bar

#### 1.1.6.15.12.3.1 Class NavigatorBarFragment

Fragment umožňující navigaci o krok dopředu nebo zpět.

Název atributu	Datový typ	Popis
handler	NavigatorBarFragment Handler	
Název metody	Návratový typ	Popis
goBackRequested	void	Zpracuje požadavek uživatele o přesun o jednu obrazovku zpět.
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). @return Nove vytvoreny handler
NavigatorBarFragment		Konstruktor.



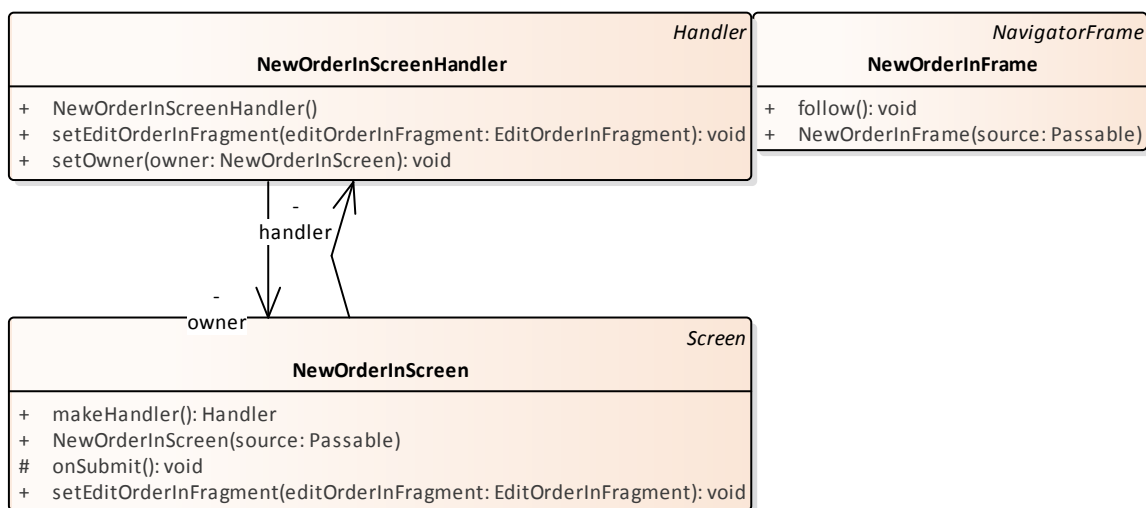
Název atributu	Datový typ	Popis
		Parametry: <b>parent: Notifyable</b> - Rodicovsky objekt

#### 1.1.6.15.12.3.2 Class NavigatorBarFragmentHandler

Handler pro NavigatorBarFragment.

Název atributu	Datový typ	Popis
owner	NavigatorBarFragment	
Název metody	Návratový typ	Popis
handleGoBackLinkAction	void	Parametry: <b>event: ActionEvent</b> -
NavigatorBarFragmentHandler		Konstruktor.
		Připojí k handleru příslušný fragment.
setOwner	void	Parametry: <b>owner: NavigatorBarFragment</b> - Fragment

#### 1.1.6.15.13 new\_order\_in\_screen



Obrázek 29 - new\_order\_in\_screen

#### 1.1.6.15.13.1 Class NewOrderInFrame

Wrapper, přidává okolo screenu navigační listu a session bar.

Název metody	Návratový typ	Popis
follow	void	Vytvoří obrazovku pro vytvoření nové logistické objednávky a předá ji řízení.



Název metody	Návratový typ	Popis
NewOrderInFrame		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt

#### 1.1.6.15.13.2 Class NewOrderInScreen

Obrazovka pro vytvoreni nove logisticke objednávky.

Název atributu	Datový typ	Popis
editOrderInFragment	EditOrderInFragment	
handler	NewOrderInScreenHandler	

Název metody	Návratový typ	Popis
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). @return Nove vytvoreny handler
NewOrderInScreen		Konstruktor. Parametry: <b>source: Passable</b> - Rodicovsky objekt
onSubmit	void	Zavolano pri odeslani objednávky uzivatelem. Provede validaci vstupu a pri uspechu odesle objednávku na server a vrati uzivatele do hlavniho menu, jinak zobrazi uzivateli alert s chybou.
setEditOrderInFragment	void	Nastavi fragment pro editaci logisticke objednávky. Parametry: <b>editOrderInFragment: EditOrderInFragment</b> - Fragment pro editaci logisticke objednávky

#### 1.1.6.15.13.3 Class NewOrderInScreenHandler

Handler NewOrderInScreenu.

Název atributu	Datový typ	Popis
editOrderIn	AnchorPane	
owner	NewOrderInScreen	

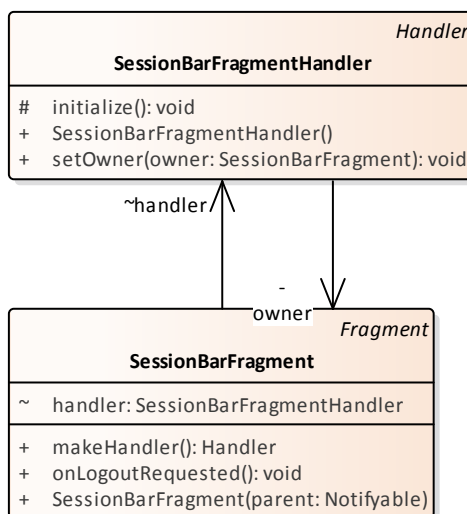
Název metody	Návratový typ	Popis
handleSubmitButtonAction	void	Parametry: <b>event:(ActionEvent)</b> -
NewOrderInScreenHandler		Konstruktor.
setEditOrderInFragment	void	Nastavi fragment pro editaci logisticke objednávky. Parametry: <b>editOrderInFragment: EditOrderInFragment</b> - Fragment pro editaci logisticke objednávky





Název atributu	Datový typ	Popis
setOwner	void	Připojí k handleru příslušný screen. Parametry: <b>owner: NewOrderInScreen</b> - Fragment

#### 1.1.6.15.14 session\_bar



Obrázek 30 - session\_bar

#### 1.1.6.15.14.1 Class SessionBarFragment

Fragment listy, která zobrazuje informace o přihlášeném uživateli a umožňuje mu odhlášení.

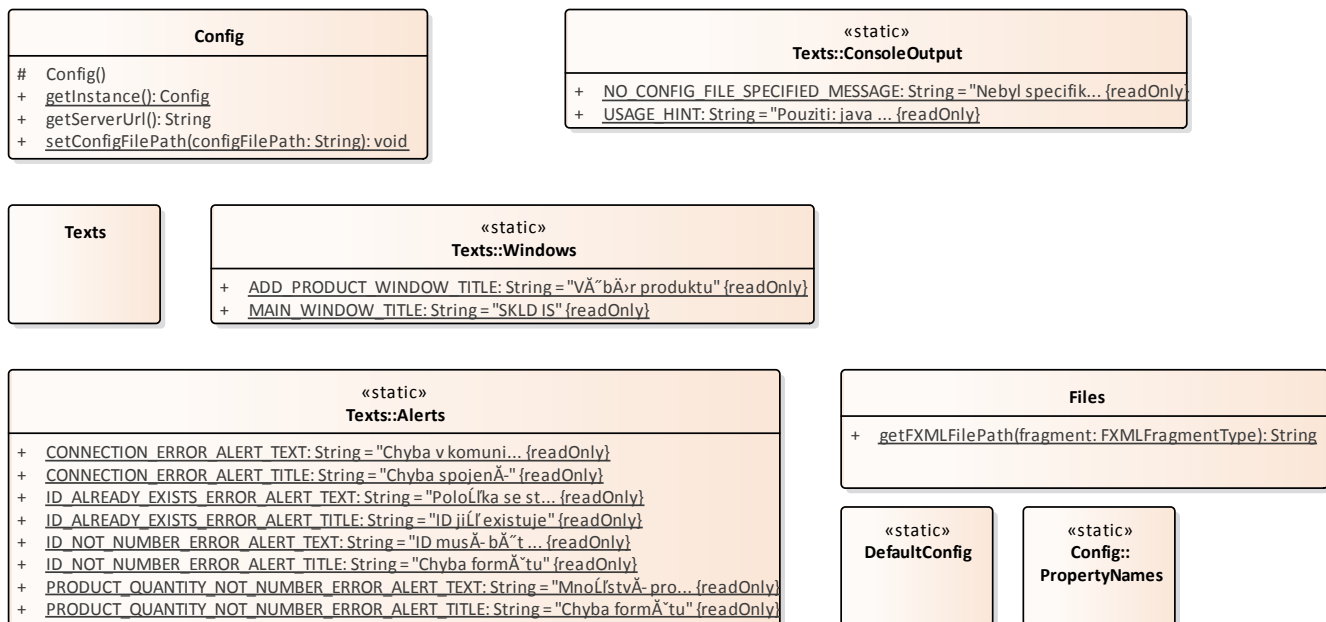
Název atributu	Datový typ	Popis
handler	SessionBarFragmentHandler	
Název metody	Návratový typ	Popis
makeHandler	Handler	Factory metoda pro tvoreni handleru. Smi byt zavolana behem zivota fragmentu pouze jednou. Diky tomu muze mit kazdy fragment (implementovany zvlastni tridou dedenou z Fragment) vlastni handler (implementovany zvlastni tridou dedenou z Handler). @return Nove vytvoreny handler
onLogoutRequested	void	Zpracuje pozadavek uzivatele o odhlaseni
SessionBarFragment		Konstruktor. Parametry: <b>parent: Notifyable</b> - Rodicovsky objekt

#### 1.1.6.15.14.2 Class SessionBarFragmentHandler

Handler SessionBarFragmentu

Název atributu	Datový typ	Popis
owner	SessionBarFragment	
signedUser	Label	
Název metody	Návratový typ	Popis
handleLogoutLinkAction	void	Parametry: <b>event:(ActionEvent)</b> -
initialize	void	Vola JavaFX pri inicializaci handleru.
SessionBarFragmentHandler		Konstruktor.
setOwner	void	Pripoji k handleru prislusny fragment. Parametry: <b>owner: SessionBarFragment</b> - Fragment

### 1.1.6.16 resources



Obrázek 31 - resources

#### 1.1.6.16.1 Class Config

Shromazduje veskerou konfiguraci aplikace. Kazde nastaveni muze byt nastavitelne bud pouze zde ve tride jako konstantni hodnota, nebo muze byt umoznena jeho konfigurace uzivatelem v .properties souboru aplikace. Singleton trida. Pri vytvoreni instance se pokusi nacist uzivatelem specifikovany soubor s konfiguraci, jinak budou pouzity defaultni hodnoty.

Název atributu	Datový typ	Popis
configFilePath	String	



Název atributu	Datový typ	Popis
instance	Config	
properties	Properties	
Název metody	Návratový typ	Popis
Config		Singleton konstruktor. Pokusi se nacist uživatelem specifikovaný soubor s konfigurací.
getInstance	Config	Vrací (singleton) instanci @return Instance
getServerUrl	String	Vrací URL serveru SKLD systému. Konfigurovatelné uživatelem v config souboru. @return URL serveru
setConfigFilePath	void	Nastaví cestu k souboru s konfigurací. Musí být nastaveno před vytvořením singleton instance. Parametry: <b>configFilePath: String</b> - Cesta k souboru s konfigurací

#### 1.1.6.16.1.1 Class PropertyNames

Obsahuje názvy atributu v souboru s konfigurací.

Název atributu	Datový typ	Popis
SERVER_URL	String	

#### 1.1.6.16.2 Class DefaultConfig

Obsahuje defaultní hodnoty atributu v souboru s konfigurací. Jsou použity, pokud uživatel nezadá příslušnou hodnotu do konfigurace.

Název atributu	Datový typ	Popis
SERVER_URL	String	

#### 1.1.6.16.3 Class Files

Obsahuje názvy (případně cesty) k souborům, typicky resource souborům

Název metody	Návratový typ	Popis
getFXMLFilePath	String	Vrací FXML soubor definující GUI pro příslušný fragment @return FXML soubor pro příslušný typ fragmentu Parametry: <b>fragment: FXMLElementType</b> - Typ fragmentu

#### 1.1.6.16.4 Class Texts

Shromažďuje texty používané v kódu aplikace. Výjimkou jsou zpravidla texty ve FXML souborech, ty jsou v nich umístěny samostatně (hard-coded).

##### 1.1.6.16.4.1 Class Alerts

Texty související s alerty (pop-up okna)

Název atributu	Datový typ	Popis
CONNECTION_ERROR_ALERT_TEXT	String	



Název atributu	Datový typ	Popis
CONNECTION_ERROR_ALERT_TITLE	String	
ID_ALREADY_EXISTS_ERROR_ALERT_TITLE	String	
ID_ALREADY_EXISTS_ERROR_ALERT_TITLE	String	
ID_NOT_NUMBER_ERROR_ALERT_TITLE	String	
ID_NOT_NUMBER_ERROR_ALERT_TITLE	String	
PRODUCT_QUANTITY_NOT_NUMBER_ERROR_ALERT_TITLE	String	
PRODUCT_QUANTITY_NOT_NUMBER_ERROR_ALERT_TITLE	String	

#### 1.1.6.16.4.2 Class ConsoleOutput

Texty, které jsou vypsané na konzoli

Název atributu	Datový typ	Popis
NO_CONFIG_FILE_SPECIFIED_MESSAGE	String	
USAGE_HINT	String	

#### 1.1.6.16.4.3 Class Windows

Texty související s okny

Název atributu	Datový typ	Popis
ADD_PRODUCT_WINDOW_TITLE	String	
MAIN_WINDOW_TITLE	String	

### 1.1.6.17 util

FXUtil
+ <code>displayAlert(title: String, text: String): void</code>
+ <code>setAnchor(node: Node, top: double, right: double, bottom: double, left: double): void</code>

Obrázek 32 - util

#### 1.1.6.17.1 Class FXUtil

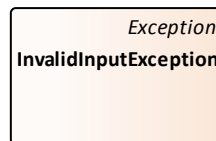
Třída obsahující pomocné metody pro práci s knihovnou JavaFX.

Název metody	Návratový typ	Popis
<code>displayAlert</code>	<code>void</code>	Zobrazí uživateli okno se zprávou/chybou.



Název metody	Návratový typ	Popis
		Parametry: <b>title: String</b> - Text v zhlavi okna Parametry: <b>text: String</b> - Zprava uzivateli
		Ukotvi node na rozmery sveho rodice. Parametry: <b>node: Node</b> - Ukotvovany objekt (obvykle AnchorPane) Parametry: <b>top: double</b> - Margin odshora Parametry: <b>right: double</b> - Margin zprava Parametry: <b>bottom: double</b> - Margin odspodu Parametry: <b>left: double</b> - Margin zleva
setAnchor	void	

## 1.1.6.17.2 exceptions

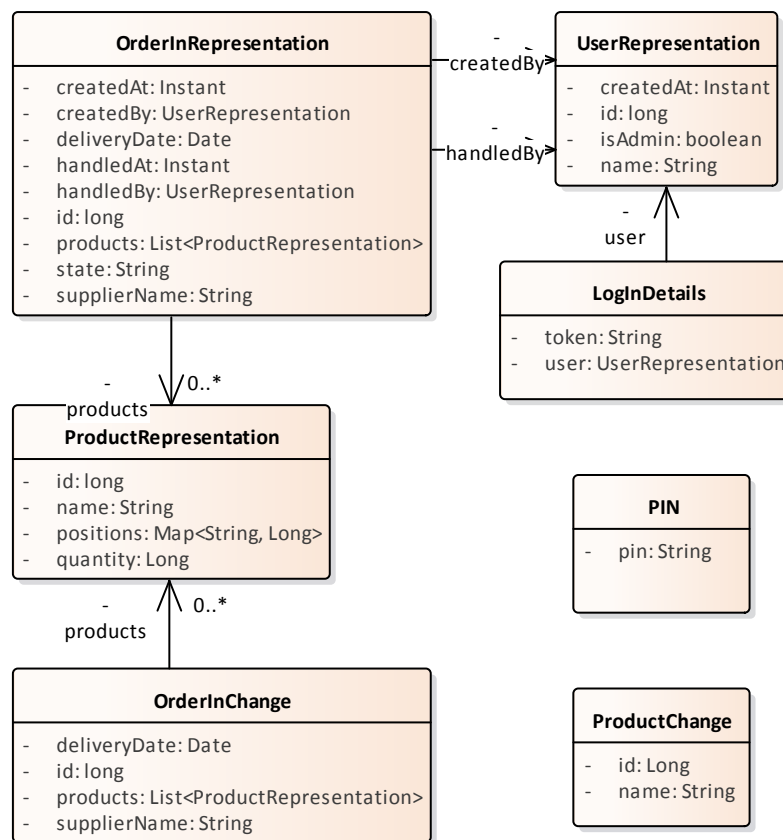


Obrázek 33 - exceptions

### 1.1.6.17.2.1 Class InvalidInputException

Vyjimka vyvolana pri zadani neplatneho vstupu uzivatelem.

## 1.1.7 representations



Obrázek 34 - representations

### 1.1.7.1 Class LoginDetails

Odpověď na požadavek k přihlášení. Obsahuje informace o přihlášeném uživateli a bezpečnostní token (může být použit v Authorization HTTP headeru).

Název atributu	Datový typ	Popis
token	String	
user	UserRepresentation	
Název metody	Návratový typ	Popis
getToken	String	
getUser	UserRepresentation	
LoginDetails		Konstruktor.
		Konstruktor.
		Parametry:
LoginDetails		<b>user: UserRepresentation</b> - Přihlasovaný uživatel
		Parametry:
		<b>token: String</b> - Bezpečnostní token
setToken	void	



Název atributu	Datový typ	Popis
		Parametry: <b>token: String</b> -
setUser	void	Parametry: <b>user: UserRepresentation</b> -

### 1.1.7.2 Class OrderInChange

Reprezentuje změnu provedenou v nějaké logistické objednávce.

Název atributu	Datový typ	Popis
deliveryDate	Date	
id	long	
products	List<ProductRepresentation>	
supplierName	String	
Název metody	Návratový typ	Popis
getDeliveryDate	Date	
getId	long	
getProducts	List<ProductRepresentation>	
getSupplierName	String	
OrderInChange		Konstruktor.
OrderInChange		Konstruktor. Parametry: <b>id: long</b> - ID Parametry: <b>supplierName: String</b> - Jméno dodavatele
setDeliveryDate	void	Parametry: <b>deliveryDate: Date</b> -
setId	void	Parametry: <b>id: long</b> -
setProducts	void	Parametry: <b>products: List&lt;ProductRepresentation&gt;</b> -
setSupplierName	void	Parametry: <b>supplierName: String</b> -



### 1.1.7.3 Class OrderInRepresentation

Obsahuje veskere detaily nejake logisticke objednavky.

Název atributu	Datový typ	Popis
createdAt	Instant	
createdBy	UserRepresentation	
deliveryDate	Date	
handledAt	Instant	
handledBy	UserRepresentation	
id	long	
products	List<ProductRepresenten	
state	String	
supplierName	String	

Název metody	Návratový typ	Popis
getCreatedAt	Instant	
getCreatedBy	UserRepresentation	
getDeliveryDate	Date	
getHandledAt	Instant	
getHandledBy	UserRepresentation	
getId	long	
getProducts	List<ProductRepresenten	
getState	String	
getSupplierName	String	
OrderInRepresentation		Konstruktor.
OrderInRepresentation		Konstruktor. Parametry: <b>id: long</b> - ID logisticke objednavky Parametry: <b>supplierName: String</b> - Jmeno dodavatele
setCreatedAt	void	Parametry: <b>createdAt: Instant</b> -
setCreatedBy	void	Parametry: <b>createdBy: UserRepresentation</b> -
setDeliveryDate	void	Parametry: <b>deliveryDate: Date</b> -





Název atributu	Datový typ	Popis
setHandledAt	void	Parametry: <b>handledAt: Instant</b> -
setHandledBy	void	Parametry: <b>handledBy: UserRepresentation</b> -
setId	void	Parametry: <b>id: long</b> -
setProducts	void	Parametry: <b>products: List&lt;ProductRepresentation&gt;</b> -
setState	void	Parametry: <b>state: String</b> -
setSupplierName	void	Parametry: <b>supplierName: String</b> -

#### 1.1.7.4 Class PIN

Pozadavek na přihlášení, obsahující uživatelský PIN.

Název atributu	Datový typ	Popis
pin	String	
Název metody	Návratový typ	Popis
getPin	String	Vrací PIN. @return PIN
PIN		Konstruktor.
PIN		Konstruktor. Parametry: <b>pin: String</b> - PIN přihlasovaného uživatele
setPin	void	Nastaví PIN. Parametry: <b>pin: String</b> - PIN

#### 1.1.7.5 Class ProductChange

Reprezentuje změnu provedenou v nějakém produktu.

Název atributu	Datový typ	Popis
id	Long	



Název atributu	Datový typ	Popis
name	String	
Název metody	Návratový typ	Popis
getId	Long	
getName	String	
ProductChange		Konstruktor.
ProductChange		Konstruktor. Parametry: <b>id: long</b> - ID Parametry: <b>name: String</b> - Název produktu
setId	void	Parametry: <b>id: Long</b> -
setName	void	Parametry: <b>name: String</b> -

### 1.1.7.6 Class ProductRepresentation

Obsahuje veskere detaily nejakého produktu.

Název atributu	Datový typ	Popis
id	long	
name	String	
positions	Map<String, Long>	
quantity	Long	
Název metody	Návratový typ	Popis
equals	boolean	Parametry: <b>o: Object</b> -
getId	long	
getName	String	
getPositions	Map<String, Long>	
getQuantity	Long	
hashCode	int	
ProductRepresentation		Konstruktor.
ProductRepresentation		Konstruktor. Parametry: <b>id: long</b> - ID



Název atributu	Datový typ	Popis
		Parametry: <b>name: String</b> - Nazev Parametry: <b>positions: Map&lt;String, Long&gt;</b> - Mapa obsahující nazvy pozic a mnozstvi produktu ulozenych v techto pozicich  Konstruktor. Parametry: <b>id: long</b> - ID Parametry: <b>name: String</b> - Nazev Parametry: <b>quantity: long</b> - Mnozstvi produktu
ProductRepresentation		
ProductRepresentation		Konstruktor. Parametry: <b>id: long</b> - ID Parametry: <b>quantity: long</b> - Mnozstvi produktu
setId	void	Parametry: <b>id: long</b> -
setName	void	Parametry: <b>name: String</b> -
setPositions	void	Parametry: <b>positions: Map&lt;String, Long&gt;</b> -
setQuantity	void	Parametry: <b>quantity: Long</b> -

### 1.1.7.7 Class UserRepresentation

Reprezentace nejakeho uzivatele.

Název atributu	Datový typ	Popis
createdAt	Instant	
id	long	
isAdmin	boolean	
name	String	
Název metody	Návratový typ	Popis
equals	boolean	Parametry: <b>o: Object</b> -
getCreatedAt	Instant	

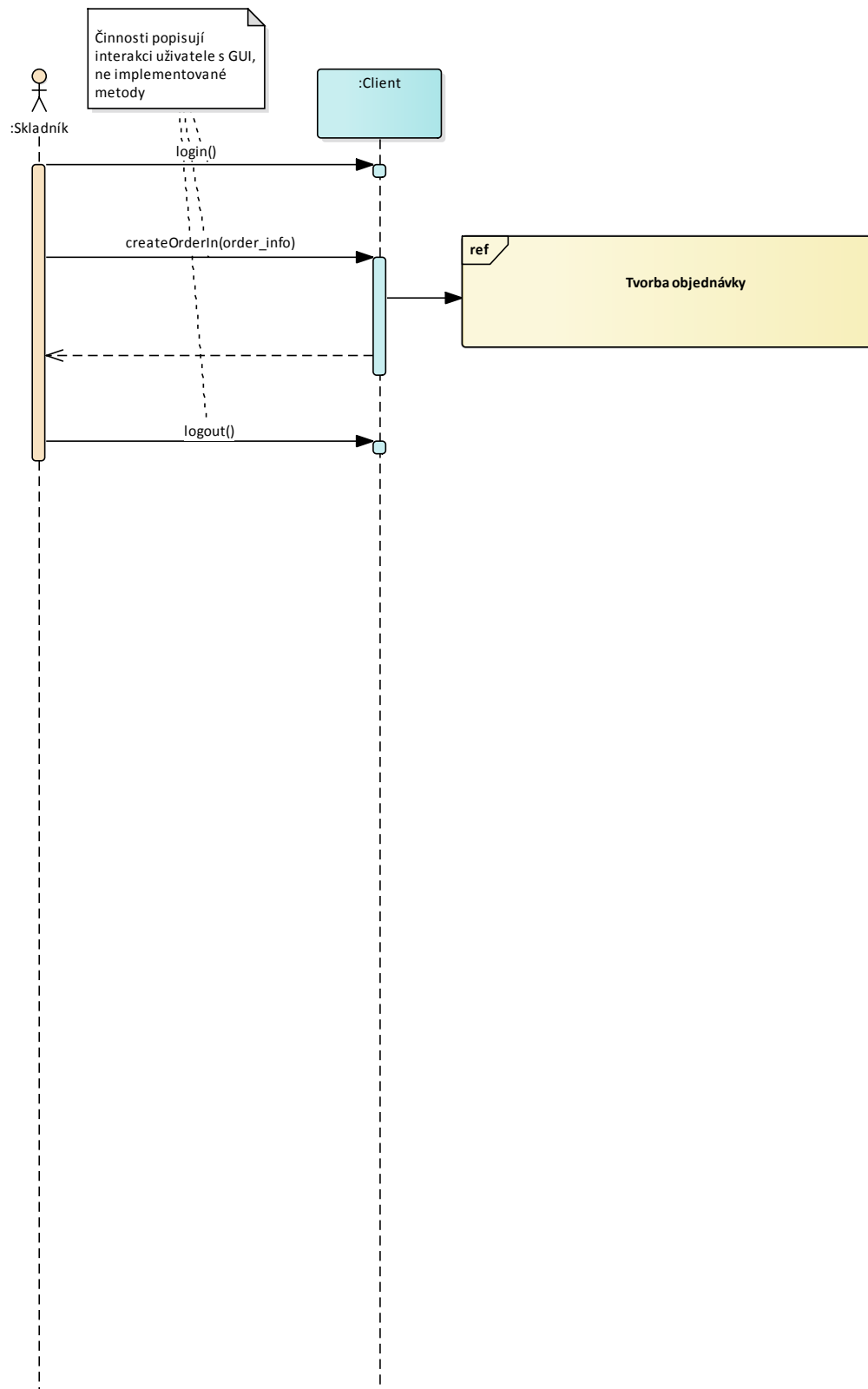


Název atributu	Datový typ	Popis
getId	long	
getName	String	
hashCode	int	
isAdmin	boolean	
setAdmin	void	Parametry: <b>admin: boolean</b> -
setCreatedAt	void	Parametry: <b>createdAt: Instant</b> -
setId	void	Parametry: <b>id: long</b> -
setName	void	Parametry: <b>name: String</b> -
UserRepresentation		Konstruktor. Parametry: <b>id: long</b> - ID Parametry: <b>name: String</b> - Jmeno uzivatele Parametry: <b>createdAt: Instant</b> - Cas, kdy byl uzivatel pridan do systemu Parametry: <b>isAdmin: boolean</b> - True uzivatel ma administratorska prava
UserRepresentation		



## **2. Model komunikace**

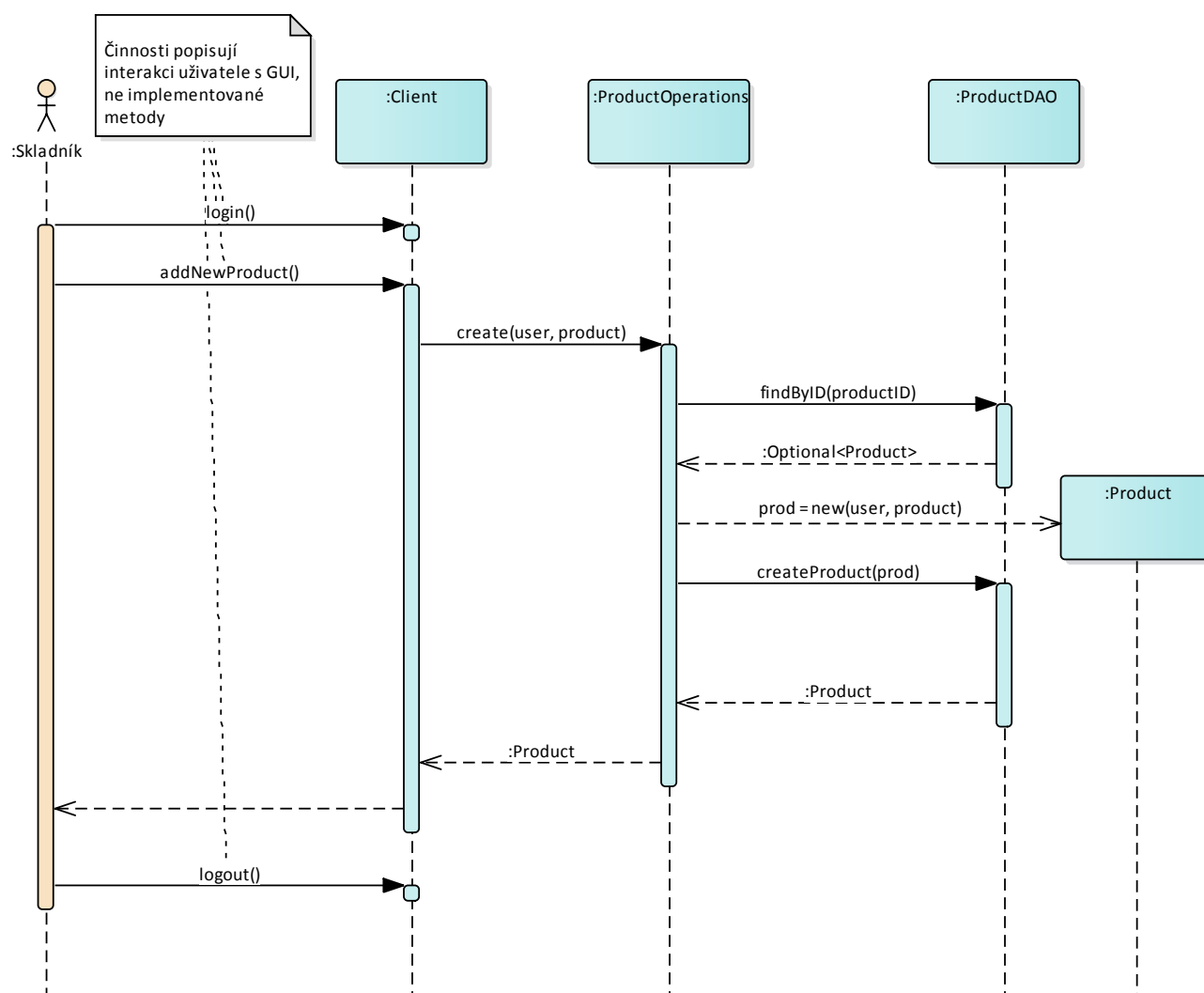
### **2.1 Realizace UC2 - Vytvoření logistické objednávky**





Obrázek 35 - Realizace UC2 - Vytvoření logistické objednávky

## 2.2 Realizace UC8 - Přidání nového typu zboží



Obrázek 36 - Realizace UC8 - Přidání nového typu zboží



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**