FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

# TattooPro

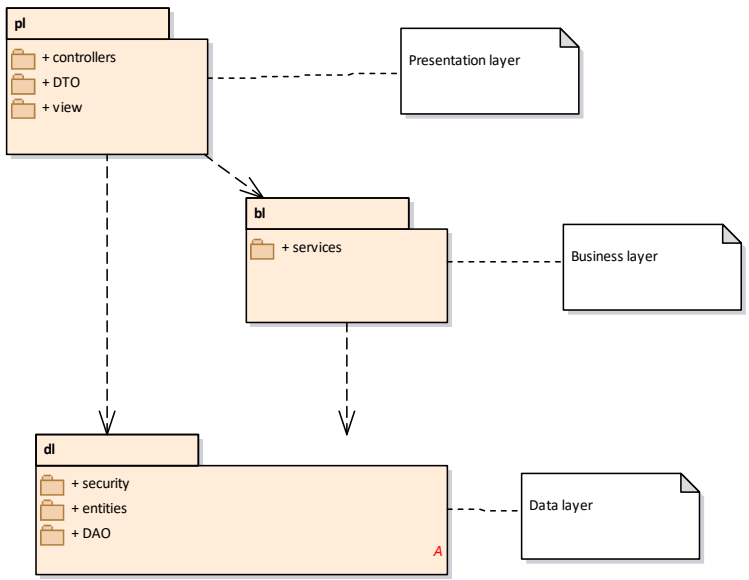
## Návrhová dokumentace (Detailní design)

Dokument vytvořen pro potřeby předmětu BI-SI1

Autoři:
Cihlář Ondřej
Seibert Michal
Patera Michal
Pončák Adam
Šprachta Vít
Zuňiga Jorge


# Obsah

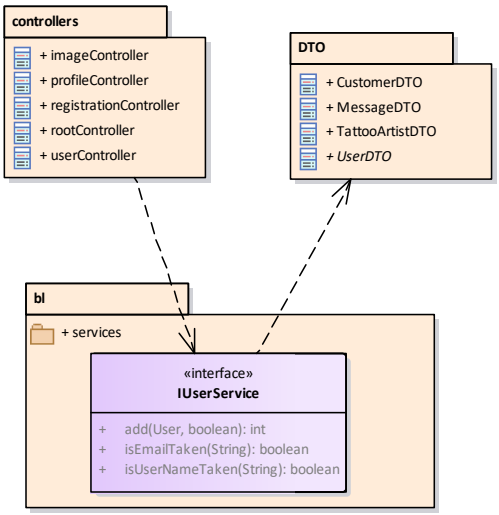
# 1. Model tříd




Obrázek 1 - Model tříd

## 1.1 pl

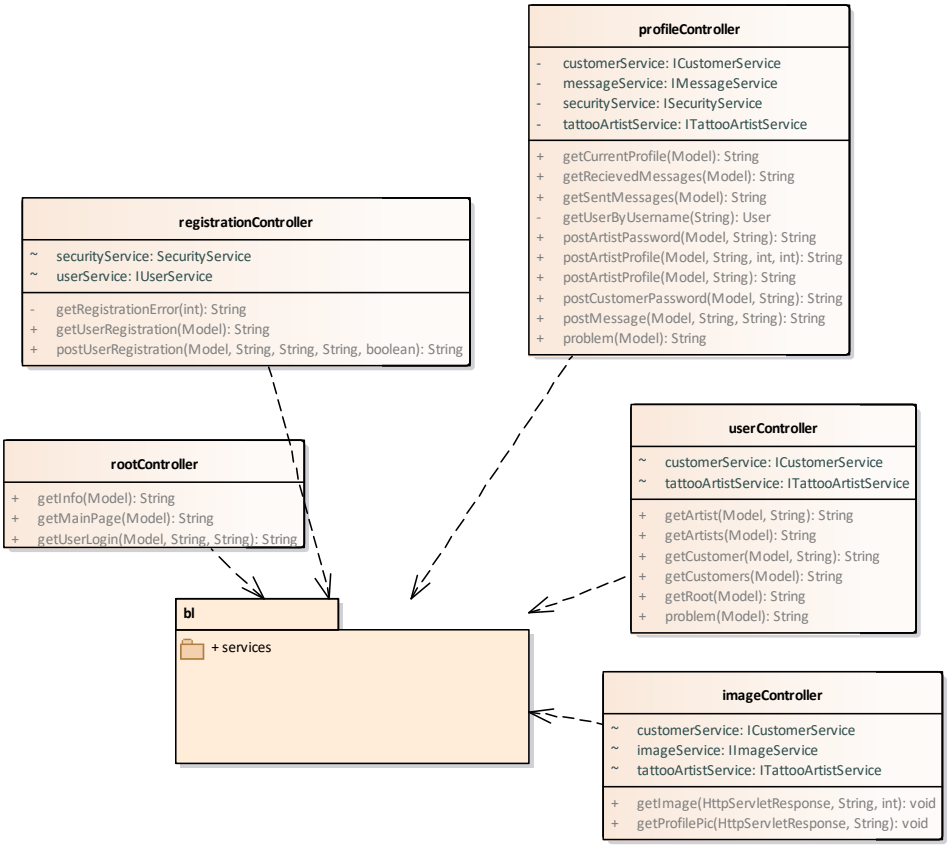
Presentation layer of application.




Obrázek 2 - pl

Graph shows, how user get his view in web browser. HTTP get request calls its controller, controller calls its service. The service makes UserDTO and send to JSP files. JSP makes HTML get request for the user.

### 1.1.1 controllers




Obrázek 3 - controllers

Every controller call his service.

#### 1.1.1.1 Class imageController

Controller for profile images

| Název atributu | Datový typ | Popis |
|---|---|---|
| customerService | ICustomerService | |
| imageService | IImageService | |
| tattooArtistService | ITattooArtistService | |
| **Název metody** | **Návratový typ** | **Popis** |
| getImage | void | Get image on users profile<br>Parametry:<br>**response: HttpServletResponse** - response with the image<br>Parametry:<br>**username: String** - username of the user<br>Parametry:<br>**imgId: int** - ID of the image |
| getProfilePic | void | Get profile pic of selected user |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | Parametry:<br>**response: HttpServletResponse** - response with profile picture<br>Parametry:<br>**username: String** - username of the user |

### 1.1.1.2 Class profileController

Controller for currently logged user

| Název atributu | Datový typ | Popis |
|---|---|---|
| customerService | ICustomerService | |
| messageService | IMessageService | |
| securityService | ISecurityService | |
| tattooArtistService | ITattooArtistService | |
| **Název metody** | **Návratový typ** | **Popis** |
| getCurrentProfile | String | Parametry:<br>**model: Model** - |
| getRecievedMessages | String | Parametry:<br>**model: Model** - |
| getSentMessages | String | Parametry:<br>**model: Model** - |
| getUserByUsername | User | Checks artists as well as customers for given username<br>@return null or found user<br>Parametry:<br>**username: String** - the username to be used |
| postArtistPassword | String | Parametry:<br>**model: Model** -<br>Parametry:<br>**newPassword: String** - |
| postArtistProfile | String | Parametry:<br>**model: Model** -<br>Parametry:<br>**newEmail: String** -<br>Parametry:<br>**newExp: int** -<br>Parametry:<br>**newPrice: int** - |
| postArtistProfile | String | Parametry: |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | **model: Model** -<br>Parametry:<br>**newEmail: String** - |
| postCustomerPassword | String | Parametry:<br>**model: Model** -<br>Parametry:<br>**newPassword: String** - |
| postMessage | String | Parametry:<br>**model: Model** -<br>Parametry:<br>**toUsername: String** -<br>Parametry:<br>**msgText: String** - |
| problem | String | Parametry:<br>**model: Model** - |

### 1.1.1.3 Class registrationController

Controller for registrations

| Název atributu | Datový typ | Popis |
|---|---|---|
| securityService<br>userService | SecurityService<br>IUserService | |
| **Název metody** | **Návratový typ** | **Popis** |
| getRegistrationError | String | Convert error code to message<br>@return the message<br>Parametry:<br>**e: int** - the code |
| getUserRegistration | String | Register form<br>@return corresponding jsp<br>Parametry:<br>**model: Model** - |
| postUserRegistration | String | New profile creation<br>@return redirect to /myprofile on succes, redirect to /register on failure with errorMsg<br>Parametry:<br>**model: Model** -<br>Parametry:<br>**username: String** - username<br>Parametry:<br>**password: String** - password<br>Parametry:<br>**email: String** - email |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | Parametry:<br>**isArtist: boolean** - true if it is an artist |

#### 1.1.1.4 Class rootController

controller for root page - mostly visual things and log in feature

| Název metody | Návratový typ | Popis |
|---|---|---|
| getInfo | String | Page with info about authors<br>@return corresponding jsp<br>Parametry:<br>**model: Model** - |
| getMainPage | String | Root of the webpage<br>@return redirect to /users, whitch is considered to be the main page<br>Parametry:<br>**model: Model** - |
| getUserLogin | String | Login form<br>@return corresponding jsp with optional paramers errorMsg (eg.: wrong passwd) and msg (succesful logout)<br>Parametry:<br>**model: Model** -<br>Parametry:<br>**error: String** -<br>Parametry:<br>**logout: String** - |

#### 1.1.1.5 Class userController

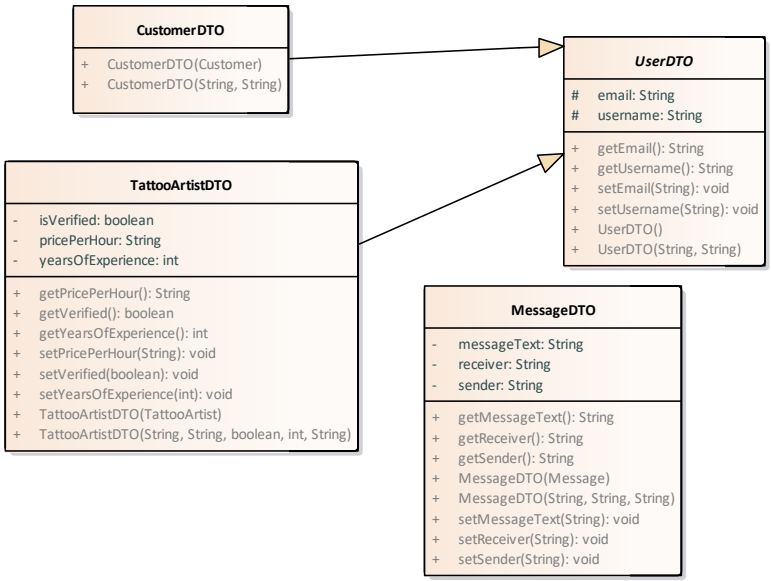
Controller for urls /users/*

| Název atributu | Datový typ | Popis |
|---|---|---|
| customerService | ICustomerService | |
| tattooArtistService | ITattooArtistService | |
| **Název metody** | **Návratový typ** | **Popis** |
| getArtist | String | Specific artist profile<br>@return corresponding jsp with selected artist or redirect to /users/bad if there is no such artist<br>Parametry:<br>**model: Model** -<br>Parametry:<br>**username: String** - username of the artists |
| getArtists | String | Ulr for browsing all artists<br>@return corresponding jsp with all artists<br>Parametry:<br>**model: Model** - |
| getCustomer | String | Specific customer profile |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | @return corresponding jsp with selected customer or redirect to /users/bad if there is no such customer<br>Parametry:<br>**model: Model** -<br>Parametry:<br>**username: String** - username of the customer |
| getCustomers | String | Ulr for browsing all customers<br>@return corresponding jsp with all customers<br>Parametry:<br>**model: Model** - |
| getRoot | String | root page of website<br>@return corresponding jsp<br>Parametry:<br>**model: Model** - |
| problem | String | Error page<br>@return corresponding jsp with errorMsg<br>Parametry:<br>**model: Model** - |

### 1.1.2 DTO

Graf znázorňuje jak funguje zobrazení stránek pro uživatele. HTTP request se pošle do controlleru, který zavolá userService, ten dostane objekt user. Vytvoří se userDTO, které se pošle do JSP, které vytvoří html pro get request pro daného uživatele.




Obrázek 4 - DTO

### 1.1.2.1 Class CustomerDTO

DTO for customer

| Název metody | Návratový typ | Popis |
|---|---|---|
| CustomerDTO | | Constructor with param.<br>Parametry:<br>**customer: Customer** - Customer |
| CustomerDTO | | Constructor with params.<br>Parametry:<br>**username: String** - Username.<br>Parametry:<br>**email: String** - EMail. |

### 1.1.2.2 Class MessageDTO

DTO for Message

| Název atributu | Datový typ | Popis |
|---|---|---|
| messageText | String | |
| receiver | String | |
| sender | String | |
| **Název metody** | **Návratový typ** | **Popis** |
| getMessageText | String | |
| getReceiver | String | |
| getSender | String | |
| MessageDTO | | Constructor with param.<br>Parametry:<br>**message: Message** - Message |
| MessageDTO | | Constructor with params.<br>Parametry:<br>**sender: String** - Sender<br>Parametry:<br>**receiver: String** - Receiver<br>Parametry:<br>**messageText: String** - Message text |
| setMessageText | void | Parametry:<br>**messageText: String** - |
| setReceiver | void | Parametry:<br>**receiver: String** - |
| setSender | void | Parametry:<br>**sender: String** - |

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

### 1.1.2.3 Class TattooArtistDTO

DTO for TattooArtist

| Název atributu | Datový typ | Popis |
|---|---|---|
| isVerified | boolean | |
| pricePerHour | String | |
| yearsOfExperience | int | |

| Název metody | Návratový typ | Popis |
|---|---|---|
| getPricePerHour | String | |
| getVerified | boolean | |
| getYearsOfExperience | int | |
| setPricePerHour | void | Parametry:<br>**pricePerHour: String** - |
| setVerified | void | Parametry:<br>**verified: boolean** - |
| setYearsOfExperience | void | Parametry:<br>**yearsOfExperience: int** - |
| TattooArtistDTO | | Constructor with param.<br>Parametry:<br>**artist: TattooArtist** - Tattoo artist. |
| TattooArtistDTO | | Constructor with params.<br>Parametry:<br>**username: String** - Username<br>Parametry:<br>**email: String** - Email<br>Parametry:<br>**isVerified: boolean** - Is verified<br>Parametry:<br>**yearsOfExperience: int** - Years of experience<br>Parametry:<br>**pricePerHour: String** - Price per hour |

### 1.1.2.4 Class UserDTO

DTO classes as used to view profiles on web - only needed attributes.

| Název atributu | Datový typ | Popis |
|---|---|---|
| email | String | |
| username | String | |

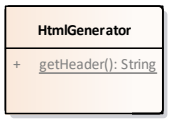
| Název atributu<br>Název metody | Datový typ<br>Návratový typ | Popis<br>Popis |
|---|---|---|
| getEmail | String | |
| getUsername | String | |
| setEmail | void | Parametry:<br>**email: String** - |
| setUsername | void | Parametry:<br>**username: String** - |
| UserDTO | | Empty constructor. |
| UserDTO | | Constructor with params.<br>Parametry:<br>**username: String** - Username<br>Parametry:<br>**email: String** - EMail |

### 1.1.3 view

## 1.1.3.1 html

Obrázek 5 - html

### *1.1.3.1.1 generator*




Obrázek 6 - generator

HTML generator for main page.

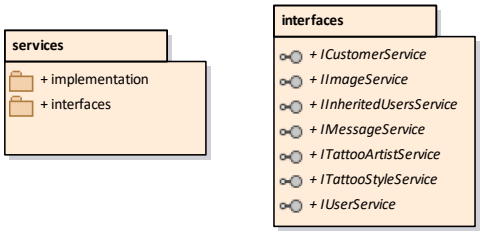
#### 1.1.3.1.1.1 Class HtmlGenerator

Class that stores html that all .jsp files have in common

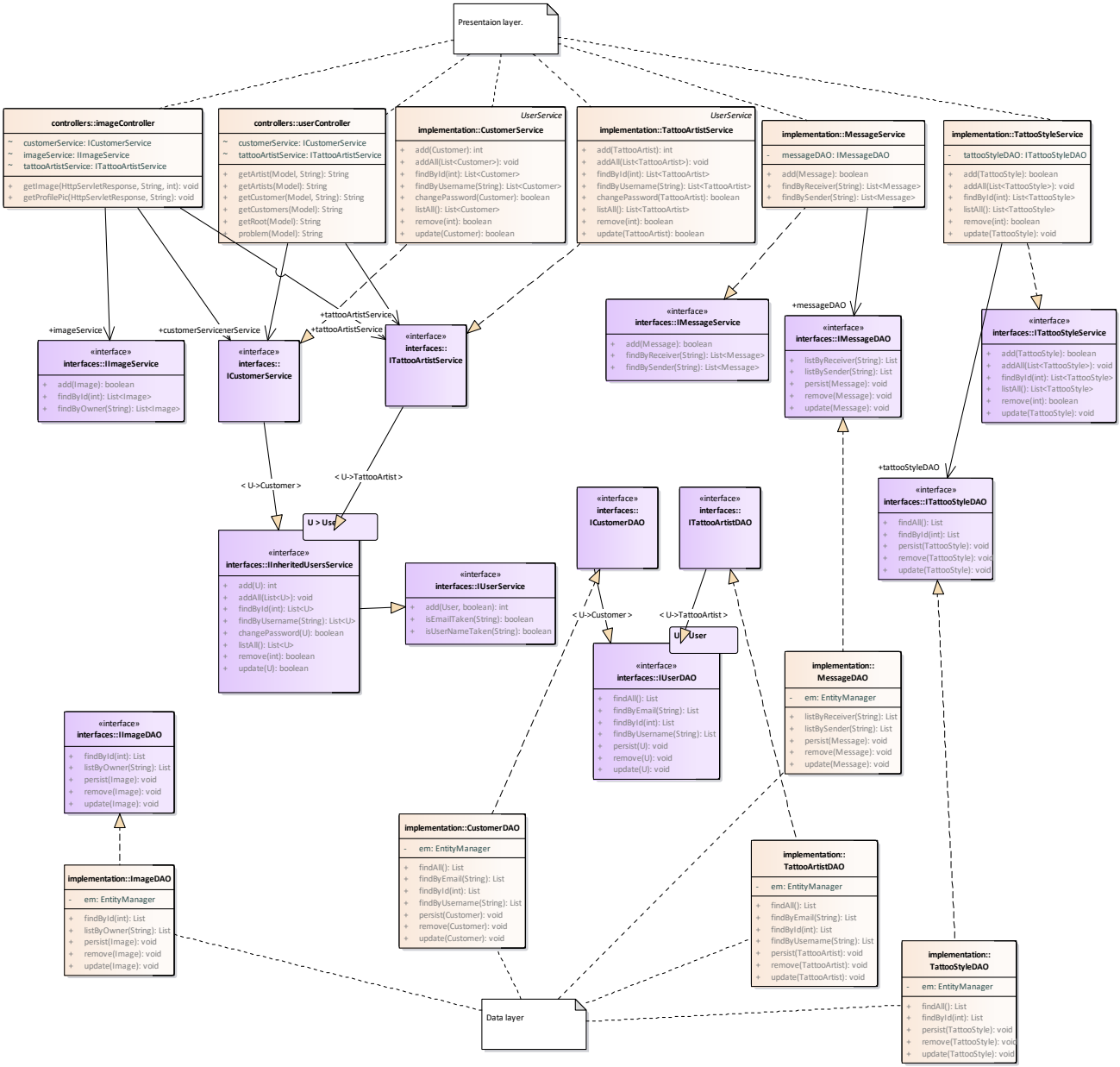
| Název metody | Návratový typ | Popis |
|---|---|---|
| getHeader | String | Page header |

## 1.2 bl

Business layer of application.




Obrázek 7 - bl

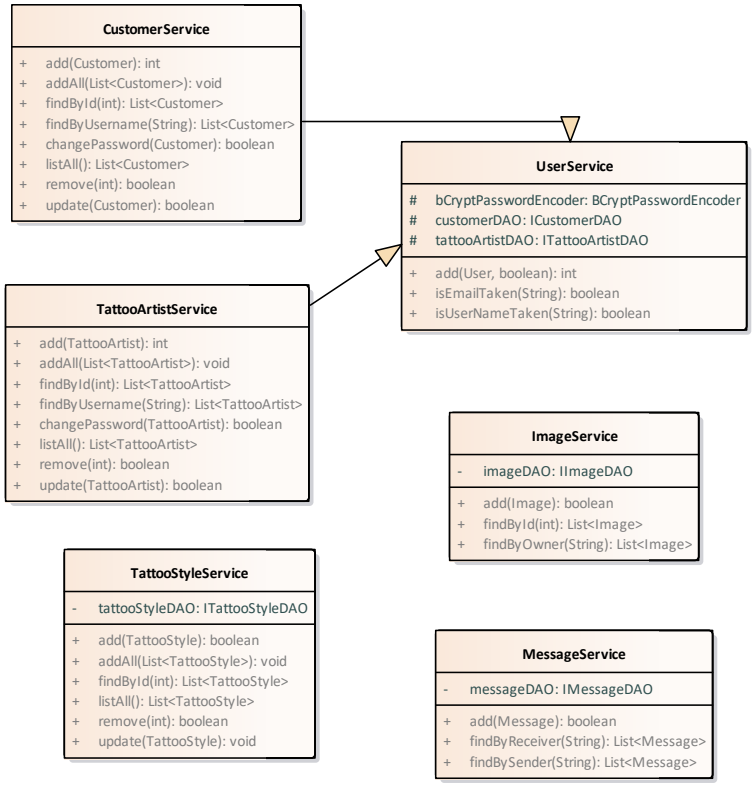



Obrázek 8 - napojení na ostaní vrstvy

Graph shows connections between each layer.

## 1.2.1 services

### 1.2.1.1 implementation




Obrázek 9 - implementation

### 1.2.1.1.1 Class CustomerService

Customer service class.

| Název metody | Návratový typ | Popis |
|---|---|---|
| add | int | Adds customer to database.<br>@return 0 on success, -1 on username taken, -2 on email taken<br>Parametry:<br>**customer: Customer** - Customer to be added |
| addAll | void | Adds a list of customers to the database.<br>Parametry:<br>**customerList: List<Customer>** - List of customers. |
| findById | List<Customer> | Finds a customer by his user ID.<br>@return Customer<br>Parametry:<br>**id: int** - User ID |
| findByUsername | List<Customer> | Finds a customer by his username.<br>@return Customer<br>Parametry: |

| Název metody | Návratový typ | Popis |
|---|---|---|
| | | **username: String** - Customer username. |
| changePassword | boolean | Changes user password<br>@return true if updated, false if user not found<br>Parametry:<br>**customer: Customer** - customer to be updated |
| listAll | List<Customer> | Lists all customers.<br>@return List of all customers |
| remove | boolean | Remove customer.<br>@return True if customer was successfully removed. False otherwise.<br>Parametry:<br>**id: int** - Customer user ID. |
| update | boolean | Updates customer, does not change password<br>@return true if updated, false if user not found<br>Parametry:<br>**customer: Customer** - Customer to be updated. |

#### 1.2.1.1.2 Class ImageService

Image service class

| Název atributu | Datový typ | Popis |
|---|---|---|
| imageDAO | IImageDAO | |
| **Název metody** | **Návratový typ** | **Popis** |
| add | boolean | Adds an image to the database.<br>@return True on success<br>Parametry:<br>**image: Image** - Image to be added |
| findById | List<Image> | Finds all images with id given by param.<br>@return List of all images with equal ID<br>Parametry:<br>**Id: int** - Image ID |
| findByOwner | List<Image> | Lists all images of a user given by his username.<br>@return List of all images where the owners username is equal to the username from the param.<br>Parametry:<br>**username: String** - User username |

#### 1.2.1.1.3 Class MessageService

Message service class

| Název atributu | Datový typ | Popis |
|---|---|---|
| messageDAO | IMessageDAO | |
| **Název metody** | **Návratový typ** | **Popis** |
| add | boolean | Adds a message to the database |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | @return True on success.<br>Parametry:<br>**message: Message** - Message o be added |
| findByReceiver | List<Message> | Lists all messages where receiver is equal to the username from the param.<br>@return List of all messages where receiver is equal to username param.<br>Parametry:<br>**username: String** - Username |
| findBySender | List<Message> | Lists all messages where sender is equal to the username from the param.<br>@return List of all messages where sender is equal to username param.<br>Parametry:<br>**username: String** - Username |

#### 1.2.1.1.4 Class TattooArtistService

Tattoo artist service class.

| Název metody | Návratový typ | Popis |
|---|---|---|
| add | int | Adds artist to the database<br>@return 0 on success, -1 on username taken, -2 on email taken<br>Parametry:<br>**tattooArtist: TattooArtist** - the artist |
| addAll | void | Adds a list of tattoo artists to the database.<br>Parametry:<br>**tattooArtistList: List<TattooArtist>** - List of tattoo artists. |
| findById | List<TattooArtist> | Finds a tattoo artist by its user ID.<br>@return Tattoo artist.<br>Parametry:<br>**id: int** - Tattoo artist user ID. |
| findByUsername | List<TattooArtist> | Finds a tattoo artist by its username.<br>@return Tattoo artist.<br>Parametry:<br>**username: String** - Tattoo artist username. |
| changePassword | boolean | Changes user password<br>@return true if updated, false if there is no such user<br>Parametry:<br>**tattooArtist: TattooArtist** - artist to be updated |
| listAll | List<TattooArtist> | Lists all tattoo artists.<br>@return List of all tattoo artists. |

| Název metody | Návratový typ | Popis |
|---|---|---|
| remove | boolean | Removes a tattoo artist by its ID.<br>@return True if the artist was successfully removed. False otherwise.<br>Parametry:<br>**id: int** - Tattoo artist user ID. |
| update | boolean | Updates a tattoo artist.<br>@return true if updated, false if there is no such user<br>Parametry:<br>**tattooArtist: TattooArtist** - Tattoo artist to be updated. |

#### 1.2.1.1.5 Class TattooStyleService

Tattoo style service class.

| Název atributu | Datový typ | Popis |
|---|---|---|
| tattooStyleDAO | ITattooStyleDAO | Tattoo style data access object. |
| **Název metody** | **Návratový typ** | **Popis** |
| add | boolean | Adds tattoo style to the database.<br>@return True if the style was successfully added.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be added. |
| addAll | void | Adds a list of tattoo styles.<br>Parametry:<br>**tattooStyleList: List<TattooStyle>** - List of tattoo styles. |
| findById | List<TattooStyle> | Finds a tattoo style by its ID.<br>@return Tattoo style.<br>Parametry:<br>**id: int** - Tattoo style ID. |
| listAll | List<TattooStyle> | Returns a list of all tattoo styles.<br>@return List of all tattoo styles. |
| remove | boolean | Removes a tattoo style from the database.<br>@return True if a style with such ID has been found and removed. False otherwise.<br>Parametry:<br>**id: int** - Tattoo style ID. |
| update | void | Updates tattoo style. If no such style is found adds it into the database.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be updated. |

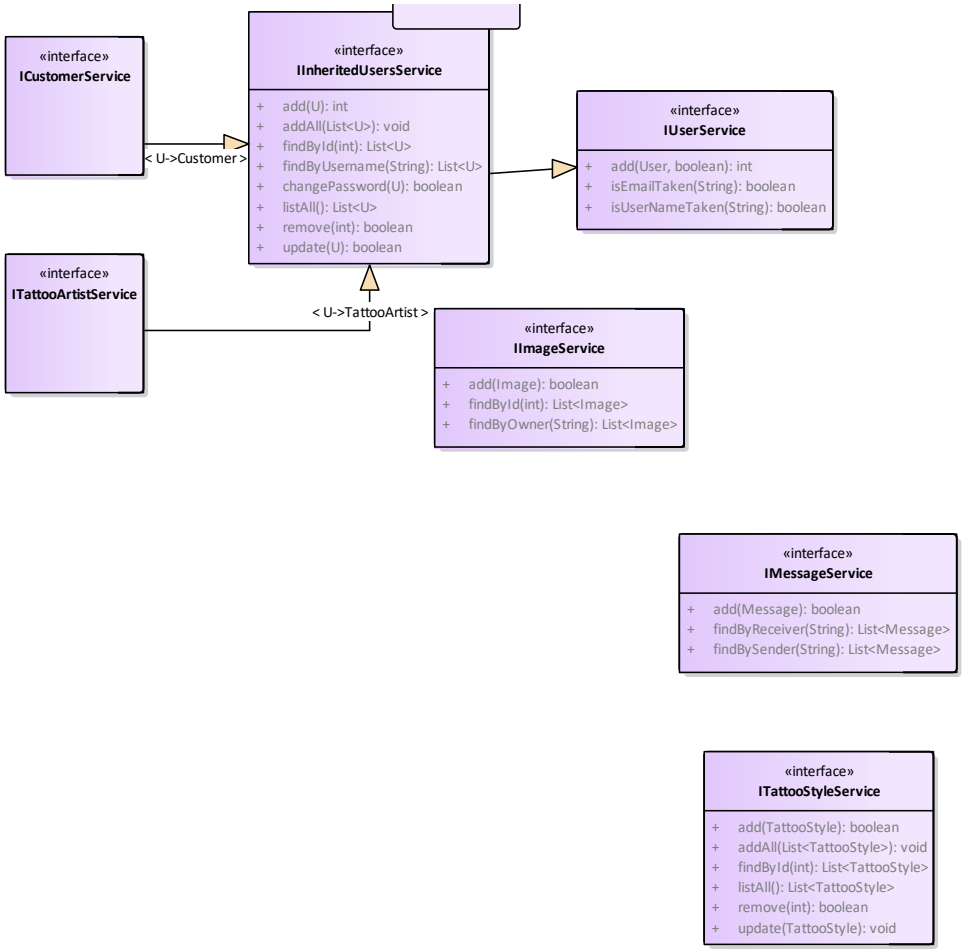
#### 1.2.1.1.6 Class UserService

Service that helps us with finding used usernames, email etc. All users inherits it

| Název atributu | Datový typ | Popis |
|---|---|---|
| bCryptPasswordEncoder | BCryptPasswordEnco | Password encoder. |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | der | |
| customerDAO | ICustomerDAO | Customer data access object. |
| tattooArtistDAO | ITattooArtistDAO | Tattoo artist data access object. |
| **Název metody** | **Návratový typ** | **Popis** |
| add | int | Adds a user to the database.<br>@return 0 on success, -1 on username taken, -2 on email taken<br>Parametry:<br>**user: User** - User to be added<br>Parametry:<br>**isArtist: boolean** - Insert true if the user is artist. |
| isEmailTaken | boolean | Checks whether email is taken.<br>@return True when the email is taken, false otherwise.<br>Parametry:<br>**email: String** - Email |
| isUserNameTaken | boolean | Checks whether username is taken.<br>@return True when the username is taken, false otherwise.<br>Parametry:<br>**username: String** - Username |

#### 1.2.1.2 interfaces




Obrázek 10 - interfaces

### 1.2.1.2.1 Interface ICustomerService

Interface for tattooArtist service class

### 1.2.1.2.2 Interface IImageService

Interface for image service

| Název metody | Návratový typ | Popis |
|---|---|---|
| add | boolean | Adds an image to the database.<br>@return True on success<br>Parametry:<br>**image: Image** - Image to be added |
| findById | List<Image> | Finds all images with id given by param.<br>@param Id Image ID<br>@return List of all images with equal ID<br>Parametry:<br>**id: int** - |

| Název metody | Návratový typ | Popis |
|---|---|---|
| | | |
| findByOwner | List<Image> | Lists all images of a user given by his username.<br>@return List of all images where the owners username is equal to the username from the param.<br>Parametry:<br>**username: String** - User username |

#### 1.2.1.2.3 Interface IInheritedUsersService

Service for User entites, that inherit from user
@param <U> type of subclass (Eg customer)

| Název metody | Návratový typ | Popis |
|---|---|---|
| add | int | Adds user to the database<br>@return 0 on success, -1 on username taken, -2 on email taken<br>Parametry:<br>**user: U** - the user |
| addAll | void | Adds a list of users to the database.<br>Parametry:<br>**userList: List<U>** - List of users. |
| findById | List<U> | Finds a user by its user ID.<br>@return the user or empty list.<br>Parametry:<br>**id: int** - user ID. |
| findByUsername | List<U> | Finds a user by its username.<br>@return the user or empty list.<br>Parametry:<br>**username: String** - the username. |
| changePassword | boolean | Changes user password<br>@return true if updated, false if there is no such user<br>Parametry:<br>**user: U** - user to be updated |
| listAll | List<U> | Lists all users.<br>@return List of all users. |
| remove | boolean | Removes an user by its ID.<br>@return True if the artist was successfully removed. False otherwise.<br>Parametry:<br>**id: int** - user ID. |
| update | boolean | Updates an user.<br>@return true if updated, false if there is no such user<br>Parametry:<br>**user: U** - user to be updated. |

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

#### 1.2.1.2.4 Interface IMessageService

Interface for image service class

| Název metody | Návratový typ | Popis |
|---|---|---|
| add | boolean | Adds a message to the database<br>@return True on success.<br>Parametry:<br>**message: Message** - Message o be added |
| findByReceiver | List<Message> | Lists all messages where receiver is equal to the username from the param.<br>@return List of all messages where receiver is equal to username param.<br>Parametry:<br>**username: String** - Username |
| findBySender | List<Message> | Lists all messages where sender is equal to the username from the param.<br>@return List of all messages where sender is equal to username param.<br>Parametry:<br>**username: String** - Username |

#### 1.2.1.2.5 Interface ITattooArtistService

Interface for tattooArtist service class

#### 1.2.1.2.6 Interface ITattooStyleService

Interface for tattooStule service class

| Název metody | Návratový typ | Popis |
|---|---|---|
| add | boolean | Adds tattoo style to the database.<br>@return True if the style was successfully added.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be added. |
| addAll | void | Adds a list of tattoo styles.<br>Parametry:<br>**tattooStyleList: List<TattooStyle>** - List of tattoo styles. |
| findById | List<TattooStyle> | Finds a tattoo style by its ID.<br>@return Tattoo style.<br>Parametry:<br>**id: int** - Tattoo style ID. |
| listAll | List<TattooStyle> | Returns a list of all tattoo styles.<br>@return List of all tattoo styles. |
| remove | boolean | Removes a tattoo style from the database. |

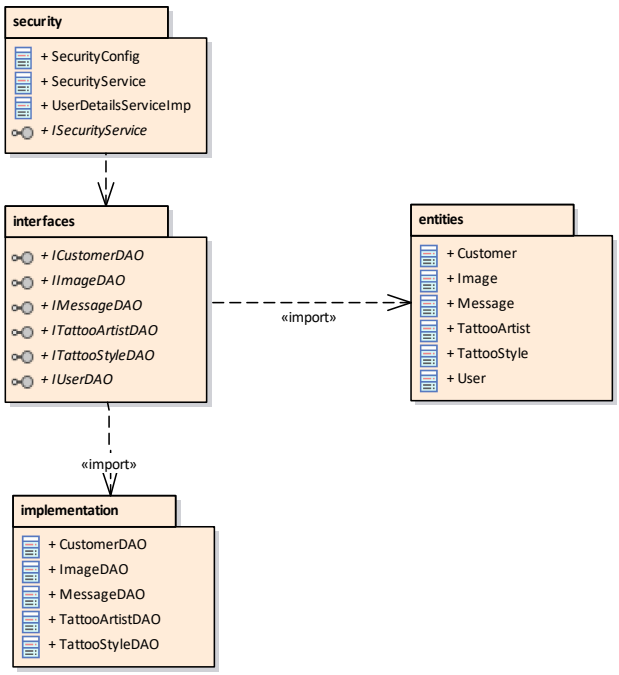
| Název metody | Návratový typ | Popis |
|---|---|---|
| | | @return True if a style with such ID has been found and removed. False otherwise.<br>Parametry:<br>**id: int** - Tattoo style ID. |
| update | void | Updates tattoo style. If no such style is found adds it into the database.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be updated. |

#### *1.2.1.2.7 Interface IUserService*

Interface for user service class

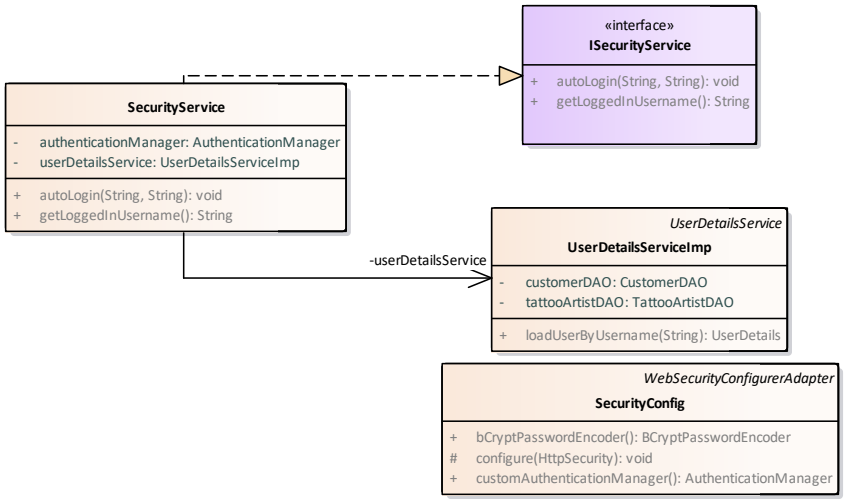
| Název metody | Návratový typ | Popis |
|---|---|---|
| add | int | Adds a user to the database.<br>@return 0 on success, -1 on username taken, -2 on email taken<br>Parametry:<br>**user: User** - User to be added<br>Parametry:<br>**isArtist: boolean** - Insert true if the user is artist. |
| isEmailTaken | boolean | Checks whether email is taken.<br>@return True when the email is taken, false otherwise.<br>Parametry:<br>**email: String** - Email |
| isUserNameTaken | boolean | Checks whether username is taken.<br>@return True when the username is taken, false otherwise.<br>Parametry:<br>**username: String** - Username |

## 1.3 dl




Obrázek 11 - dl

### 1.3.1 security




Obrázek 12 - security

Security package.

#### 1.3.1.1 Class SecurityConfig

Configuration for security - mainly for editing profiles

| Název metody | Návratový typ | Popis |
|---|---|---|
| bCryptPasswordEncoder | BCryptPasswordEncoder | Bean with password encoder |
| configure | void | Overwritten method that defines secured urls Spring security method, spring handles it itself<br>Parametry:<br>**http: HttpSecurity** - |
| customAuthenticationManager | AuthenticationManager | Bean with authentication context |

### 1.3.1.2 Class SecurityService

Security related tasks such as getting logged username or autolog after registration

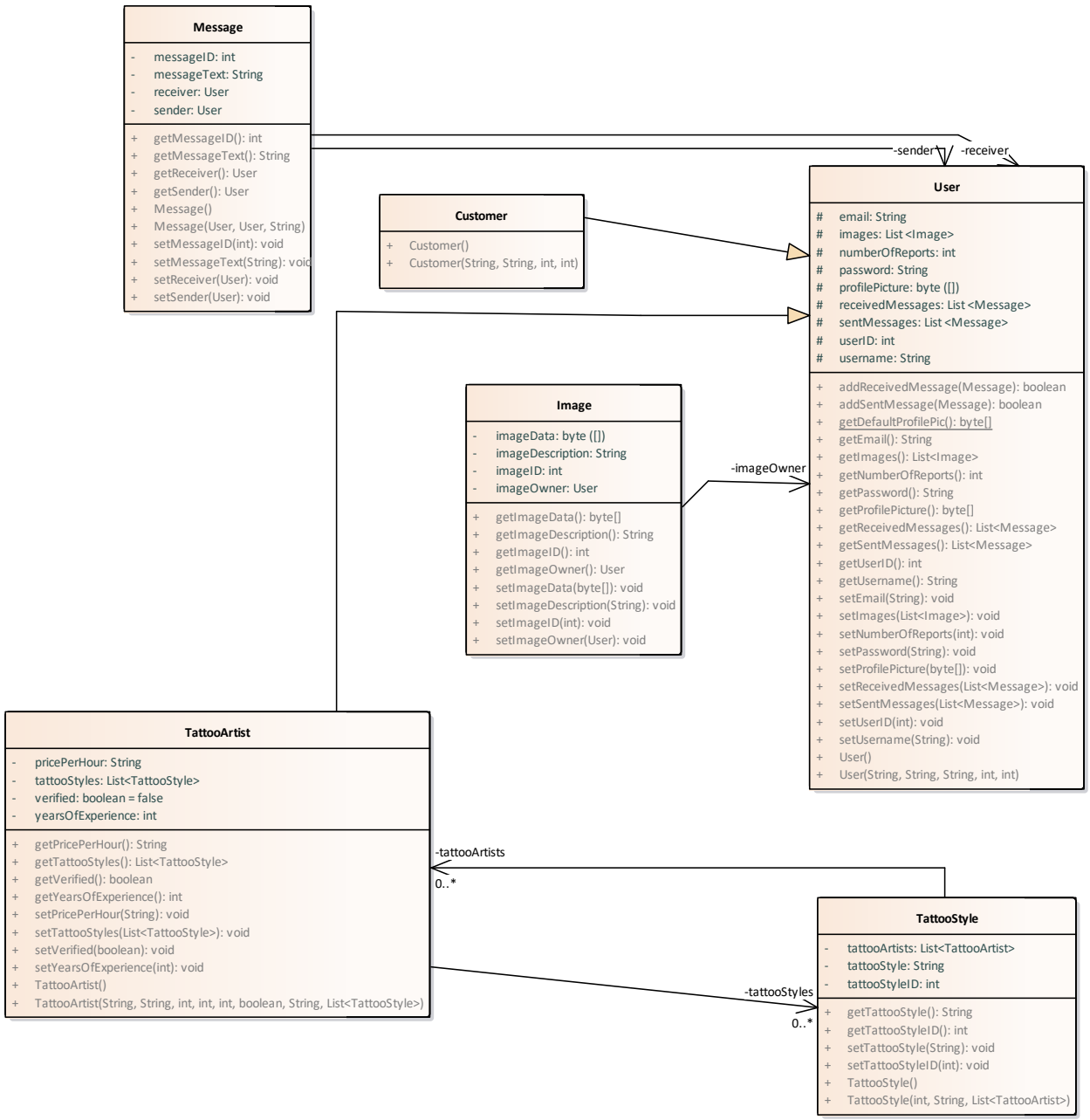
| Název atributu | Datový typ | Popis |
|---|---|---|
| authenticationManager | AuthenticationManager | |
| userDetailsService | UserDetailsServiceImp | |
| **Název metody** | **Návratový typ** | **Popis** |
| autoLogin | void | Auto login user after registration<br>Parametry:<br>**username: String** -<br>Parametry:<br>**password: String** - |
| getLoggedInUsername | String | get username of user that is currently logged in<br>@return the username or null when user is not logged in |

### 1.3.1.3 Class UserDetailsServiceImp

springboot security class for currently logged user

| Název atributu | Datový typ | Popis |
|---|---|---|
| customerDAO | CustomerDAO | |
| tattooArtistDAO | TattooArtistDAO | |
| **Název metody** | **Návratový typ** | **Popis** |
| loadUserByUsername | UserDetails | sets permisions for currently logged user<br>Parametry:<br>**username: String** - |

## 1.3.2 entities




Obrázek 13 - entities

### 1.3.2.1 Class Customer

Customer entity represents a customer. Every customer is a user. Every customer is NOT a tattoo artist.

| Název metody | Návratový typ | Popis |
|---|---|---|
| Customer | | Empty constructor. |

| Název metody | Návratový typ | Popis |
|---|---|---|
| Customer | | Constructr with parameters.<br>Parametry:<br>**username: String** - Username<br>Parametry:<br>**password: String** - Password<br>Parametry:<br>**numberOfReports: int** - Number of reports<br>Parametry:<br>**userID: int** - ID |

### 1.3.2.2 Class Image

Image entity.

| Název atributu | Datový typ | Popis |
|---|---|---|
| imageData | byte | Image data As well as in User - pictures are bigger than 255 bytes |
| imageDescription | String | Image description |
| imageID | int | Autogenerated image ID. |
| imageOwner | User | Image owner. |

| Název metody | Návratový typ | Popis |
|---|---|---|
| getImageData | byte | |
| getImageDescription | String | |
| getImageID<br><br>getImageOwner | int<br><br>User | |
| setImageData | void | Parametry:<br>**imageData: byte[]** - |
| setImageDescription | void | Parametry:<br>**imageDescription: String** - |
| setImageID | void | Parametry:<br>**imageID: int** - |
| setImageOwner | void | Parametry:<br>**imageOwner: User** - |

### 1.3.2.3 Class Message

Represents a message between two users.

| Název atributu | Datový typ | Popis |
|---|---|---|
| messageID | int | Autogenerated ID. |
| messageText | String | Text of the message. |

| Název atributu | Datový typ | Popis |
|---|---|---|
| receiver | User | User that received the message. |
| sender | User | User that sent the message. |
| **Název metody** | **Návratový typ** | **Popis** |
| getMessageID | int | |
| getMessageText | String | |
| getReceiver | User | |
| getSender | User | |
| Message | | Empty message constructor. |
| Message | | Message constructor with params.<br>Parametry:<br>**sender: User** - Sender user<br>Parametry:<br>**receiver: User** - Reciever user<br>Parametry:<br>**messageText: String** - Message text |
| setMessageID | void | Parametry:<br>**messageID: int** - |
| setMessageText | void | Parametry:<br>**messageText: String** - |
| setReceiver | void | Parametry:<br>**receiver: User** - |
| setSender | void | Parametry:<br>**sender: User** - |

### 1.3.2.4 Class TattooArtist

Tattoo artist entity represents a tattoo artist. Every tattoo artist is a user. Every tattoo artist is NOT a customer.

| Název atributu | Datový typ | Popis |
|---|---|---|
| pricePerHour | String | Artist's price per hour. Can be in any currency. Any range. That's why it's a string. |
| tattooStyles | List<TattooStyle> | Styles in which the tattoo artist prefers to work. |
| verified | boolean | Verification status. True = verified. False = not verified. |
| yearsOfExperience | int | Years of experience the tattoo artist has. |
| **Název metody** | **Návratový typ** | **Popis** |
| getPricePerHour | String | |
| getTattooStyles | List<TattooStyle> | |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | |
| getVerified | boolean | |
| getYearsOfExperience | int | |
| setPricePerHour | void | Parametry:<br>**pricePerHour: String** - |
| setTattooStyles | void | Parametry:<br>**tattooStyles: List<TattooStyle>** - |
| setVerified | void | Parametry:<br>**verified: boolean** - |
| setYearsOfExperience | void | Parametry:<br>**yearsOfExperience: int** - |
| TattooArtist | | Empty constructor. |
| TattooArtist | | Constructor with parameters.<br>Parametry:<br>**username: String** - Username<br>Parametry:<br>**password: String** - Password<br>Parametry:<br>**numberOfReports: int** - Number of reports<br>Parametry:<br>**userID: int** - ID<br>Parametry:<br>**yearsOfExperience: int** - Years of experience<br>Parametry:<br>**verified: boolean** - Verified<br>Parametry:<br>**pricePerHour: String** - Price per hour<br>Parametry:<br>**tattooStyles: List<TattooStyle>** - Tattoo styles |

### 1.3.2.5 Class TattooStyle

TattooStyle entity represents the artistic style of a tattoo.

| Název atributu | Datový typ | Popis |
|---|---|---|
| tattooArtists | List<TattooArtist> | Artists which have this style listed on their profile. |
| tattooStyle | String | Tattoo style name. |
| tattooStyleID | int | TattooStyleID |
| **Název metody** | **Návratový typ** | **Popis** |
| getTattooStyle | String | |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | |
| getTattooStyleID | int | |
| setTattooStyle | void | Parametry:<br>**tattooStyle: String** - |
| setTattooStyleID | void | Parametry:<br>**tattooStyleID: int** - |
| TattooStyle | | Empty constructor. |
| TattooStyle | | Constructor with params.<br>Parametry:<br>**tattooStyleID: int** - ID<br>Parametry:<br>**tattooStyle: String** - Tattoo style name<br>Parametry:<br>**tattooArtists: List<TattooArtist>** - Artists with such style |

### 1.3.2.6 Class User

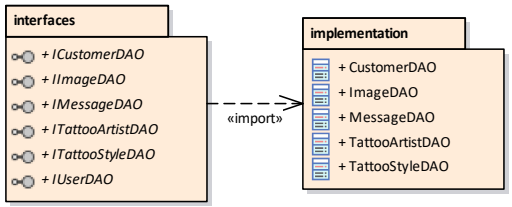
User class represents user. Users can be strictly one of two types. Customer OR TattooArtist

| Název atributu | Datový typ | Popis |
|---|---|---|
| email | String | User email. |
| images | List <Image> | Images. |
| numberOfReports | int | Number of times the user has been reported. |
| password | String | User password. |
| profilePicture | byte | Profile picture as byte array. Needs to have enough length - pictures are way bigger than 255 bytes |
| receivedMessages | List <Message> | Messages received. |
| sentMessages | List <Message> | Messages sent. |
| userID | int | Autogenerated ID |
| username | String | User username. |
| **Název metody** | **Návratový typ** | **Popis** |
| addReceivedMessage | boolean | Add received message.<br>@return List.add<br>Parametry:<br>**message: Message** - Message |
| addSentMessage | boolean | Add sent message.<br>@return List.add<br>Parametry:<br>**message: Message** - Message |
| getDefaultProfilePic | byte | |
| getEmail | String | |

| Název atributu | Datový typ | Popis |
|---|---|---|
| getImages | List<Image> | |
| getNumberOfReports | int | |
| getPassword | String | |
| getProfilePicture | byte | |
| getReceivedMessages | List<Message> | |
| getSentMessages | List<Message> | |
| getUserID | int | |
| getUsername | String | |
| setEmail | void | Parametry:<br>**email: String** - |
| setImages | void | Parametry:<br>**images: List<Image>** - |
| setNumberOfReports | void | Parametry:<br>**numberOfReports: int** - |
| setPassword | void | Parametry:<br>**password: String** - |
| setProfilePicture | void | Parametry:<br>**profilePicture: byte[]** - |
| setReceivedMessages | void | Parametry:<br>**receivedMessages: List<Message>** - |
| setSentMessages | void | Parametry:<br>**sendMessages: List<Message>** - |
| setUserID | void | Parametry:<br>**userID: int** - |
| setUsername | void | Parametry:<br>**username: String** - |

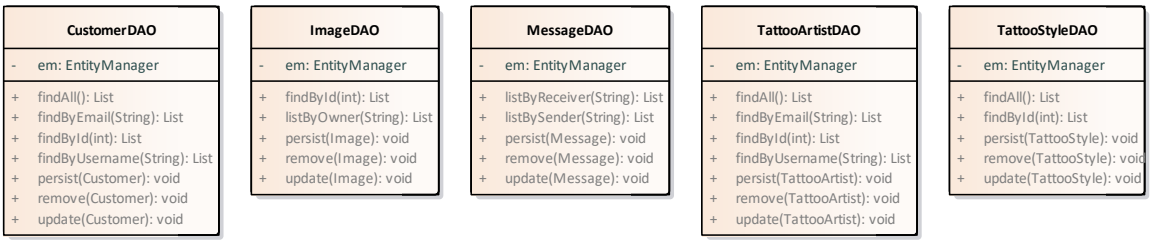
| Název atributu | Datový typ | Popis |
|---|---|---|
| | | |
| User | | Empty constructor. |
| User | | Constructor with params.<br>Parametry:<br>**username: String** - Username<br>Parametry:<br>**email: String** - Email<br>Parametry:<br>**password: String** - Password<br>Parametry:<br>**numberOfReports: int** - Number of reports<br>Parametry:<br>**userID: int** - ID |

### 1.3.3 DAO




Obrázek 14 - DAO

#### 1.3.3.1 implementation




Obrázek 15 - implementation

##### *1.3.3.1.1 Class CustomerDAO*

Customer repository !!! Database constraints are programmers responsibility. (Adding multiple things with the same id makes the database go boom) !!!

| Název atributu | Datový typ | Popis |
|---|---|---|
| em | EntityManager | |
| **Název metody** | **Návratový typ** | **Popis** |

| Název atributu | Datový typ | Popis |
|---|---|---|
| findAll | List | Finds all customers.<br>@return List of all customers. |
| findByEmail | List | Finds a customer with selected email.<br>@param username Customer email.<br>@return List of all customer with such email.<br>Parametry:<br>**email: String** - |
| findById | List | Finds a customer with selected ID.<br>@return List of all customers with such ID.<br>Parametry:<br>**id: int** - Customer ID. |
| findByUsername | List | Finds a customer with selected username.<br>@return List of all customers with such username.<br>Parametry:<br>**username: String** - Customer username. |
| persist | void | Persists a customer.<br>Parametry:<br>**customer: Customer** - Customer to be saved. |
| remove | void | Removes a customer.<br>Parametry:<br>**customer: Customer** - Customer to be removed. |
| update | void | Updates a customer.<br>Parametry:<br>**customer: Customer** - Customer to be updated. |

#### 1.3.3.1.2 Class ImageDAO

Image repository !!! Database constraints are programmers responsibility. (Adding multiple things with the same id makes the database go boom) !!!

| Název atributu | Datový typ | Popis |
|---|---|---|
| em | EntityManager | |
| **Název metody** | **Návratový typ** | **Popis** |
| findById | List | Lists image by ID.<br>@return Image with such ID in a List.<br>Parametry:<br>**Id: int** - Image ID. |
| listByOwner | List | Parametry:<br>**username: String** - |
| persist | void | Persists an image.<br>Parametry:<br>**image: Image** - Image to be saved. |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | |
| remove | void | Removes an image.<br>Parametry:<br>**image: Image** - Image to be removed. |
| update | void | Updates an image<br>Parametry:<br>**image: Image** - Image to be updated. |

#### 1.3.3.1.3 Class MessageDAO

Message repository !!! Database constraints are programmers responsibility. (Adding multiple things with the same id makes the database go boom) !!!

| Název atributu | Datový typ | Popis |
|---|---|---|
| em | EntityManager | |
| **Název metody** | **Návratový typ** | **Popis** |
| listByReceiver | List | Lists messages by receiver.<br>@return List of all messages where receiver == username.<br>Parametry:<br>**username: String** - Owner of messages. |
| listBySender | List | Lists messages by sender.<br>@return List of all messages where sender == username.<br>Parametry:<br>**username: String** - Owner of messages. |
| persist | void | Persists a message.<br>Parametry:<br>**message: Message** - Message to be saved. |
| remove | void | Removes a message.<br>Parametry:<br>**message: Message** - Message to be removed. |
| update | void | Updates a message.<br>Parametry:<br>**message: Message** - Message to be updated. |

#### 1.3.3.1.4 Class TattooArtistDAO

TattooArtist repository !!! Database constraints are programmers responsibility. (Adding multiple things with the same id makes the database go boom) !!!

| Název atributu | Datový typ | Popis |
|---|---|---|
| em | EntityManager | |
| **Název metody** | **Návratový typ** | **Popis** |
| findAll | List | Finds all tattoo artists.<br>@return List of all tattoo artists. |
| findByEmail | List | Finds a tattoo artist with selected email.<br>@param username Tattoo artist email. |

| Název atributu | Datový typ | Popis |
|---|---|---|
| | | @return List of all tattoo artists with such email.<br>Parametry:<br>**email: String** - |
| findById | List | Finds a tattoo artist with selected ID.<br>@return List of all tattoo artists with such ID.<br>Parametry:<br>**id: int** - Tattoo artist ID. |
| findByUsername | List | Finds a tattoo artist with selected username.<br>@return List of all tattoo artists with such username.<br>Parametry:<br>**username: String** - Tattoo artist username. |
| persist | void | Persists a tattoo artist.<br>Parametry:<br>**tattooArtist: TattooArtist** - Tattoo artist to be saved. |
| remove | void | Removes a tattoo artist.<br>Parametry:<br>**tattooArtist: TattooArtist** - tattoo artist to be removed. |
| update | void | Updates a tattoo artist.<br>Parametry:<br>**tattooArtist: TattooArtist** - Tattoo artist to be updated. |

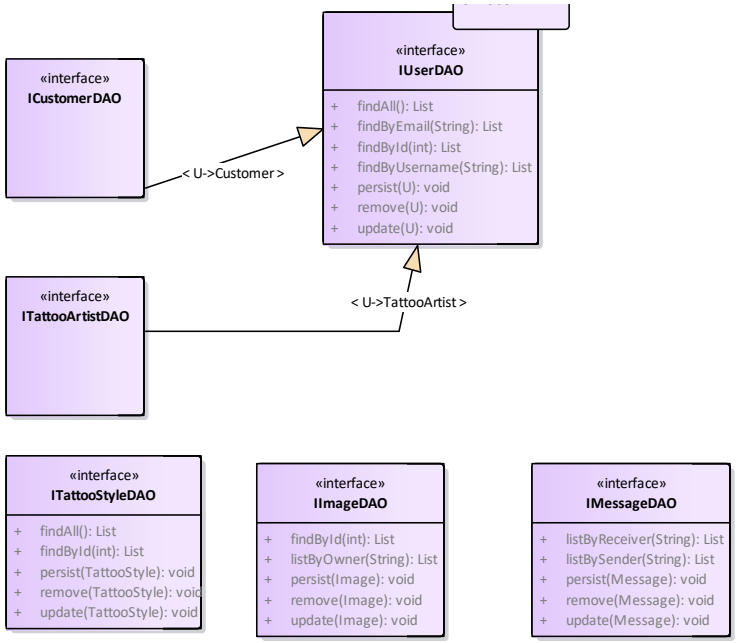
#### 1.3.3.1.5 Class TattooStyleDAO

TattooStyle repository !!! Database constraints are programmers responsibility. (Adding multiple things with the same id makes the database go boom) !!!

| Název atributu | Datový typ | Popis |
|---|---|---|
| em | EntityManager | |
| **Název metody** | **Návratový typ** | **Popis** |
| findAll | List | Lists tattoo style by owner.<br>@return List of all tattoo styles. |
| findById | List | Lists tattoo style by ID.<br>@param Id Tattoo style ID.<br>@return Tattoo style with such ID in a List.<br>Parametry:<br>**id: int** - |
| persist | void | Persists a tattoo style.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be saved. |
| remove | void | Removes a tattoo style.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be removed. |

| Název atributu | Datový typ | Popis |
|---|---|---|
| update | void | Updates a tattoo style.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be updated. |

### 1.3.3.2 interfaces




Obrázek 16 - interfaces

#### *1.3.3.2.1 Interface ICustomerDAO*

Interface defining data layer interfaces for customer management.

#### *1.3.3.2.2 Interface IImageDAO*

Interface defining data layer interfaces for image management.

| Název metody | Návratový typ | Popis |
|---|---|---|
| findById | List | Lists image by ID.<br>@return Image with such ID in a List.<br>Parametry:<br>**Id: int** - Image ID. |
| listByOwner | List | Lists images by owner.<br>@return List of all images<br>Parametry:<br>**username: String** - Owner of images. |

| Název metody | Návratový typ | Popis |
|---|---|---|
| persist | void | Persists an image.<br>Parametry:<br>**image: Image** - Image to be saved. |
| remove | void | Removes an image.<br>Parametry:<br>**image: Image** - Image to be removed. |
| update | void | Updates an image.<br>Parametry:<br>**image: Image** - Image to be updated. |

#### 1.3.3.2.3 Interface IMessageDAO

Interface defining data layer interfaces for message management.

| Název metody | Návratový typ | Popis |
|---|---|---|
| listByReceiver | List | Lists messages by receiver.<br>@return List of all messages where receiver == username.<br>Parametry:<br>**username: String** - Owner of messages. |
| listBySender | List | Lists messages by sender.<br>@return List of all messages where sender == username.<br>Parametry:<br>**username: String** - Owner of messages. |
| persist | void | Persists a message.<br>Parametry:<br>**message: Message** - Message to be saved. |
| remove | void | Removes a message.<br>Parametry:<br>**message: Message** - Message to be removed. |
| update | void | Updates a message.<br>Parametry:<br>**message: Message** - Message to be updated. |

#### 1.3.3.2.4 Interface ITattooArtistDAO

Interface defining data layer interfaces for tattoo artist management.

#### 1.3.3.2.5 Interface ITattooStyleDAO

Interface defining data layer interfaces for tattoo style management.

| Název metody | Návratový typ | Popis |
|---|---|---|
| findAll | List | Lists tattoo style by owner.<br>@return List of all tattoo styles. |
| findById | List | Lists tattoo style by ID. |

| Název metody | Návratový typ | Popis |
|---|---|---|
| | | @param Id Tattoo style ID.<br>@return Tattoo style with such ID in a List.<br>Parametry:<br>**id: int** - |
| persist | void | Persists a tattoo style.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be saved. |
| remove | void | Removes a tattoo style.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be removed. |
| update | void | Updates a tattoo style.<br>Parametry:<br>**tattooStyle: TattooStyle** - Tattoo style to be updated. |

#### *1.3.3.2.6 Interface IUserDAO*

Interface defining data layer interfaces for users. Every other interface for instances of user should inherit from this one.

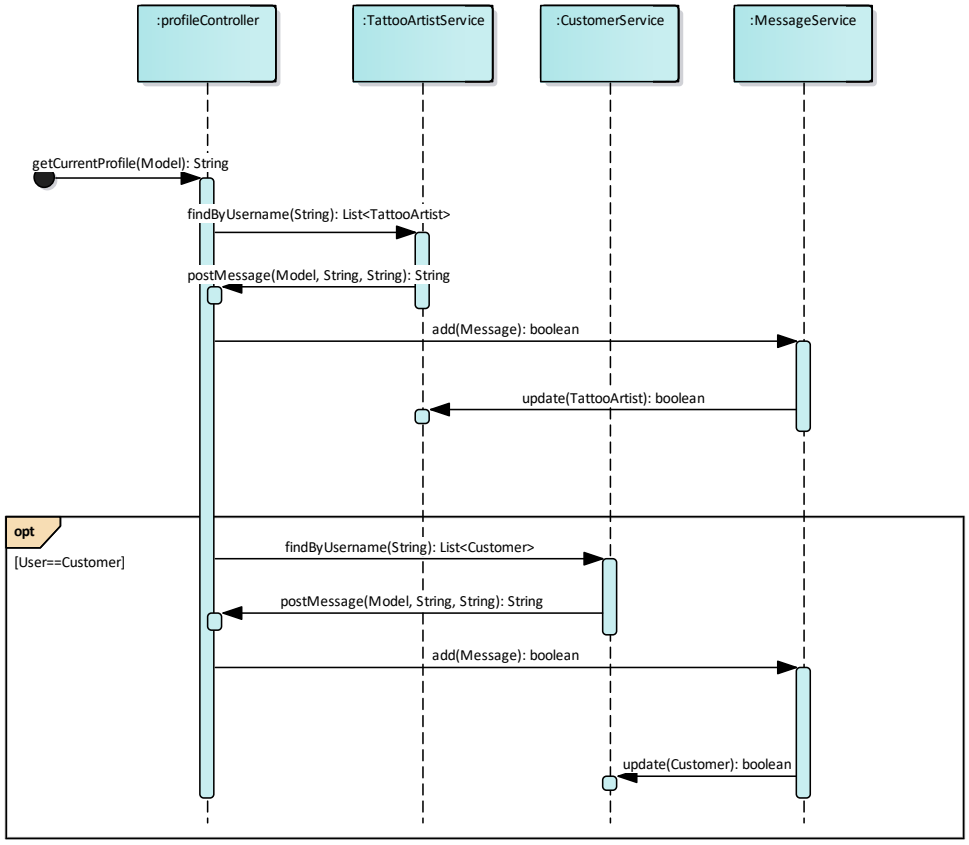
| Název metody | Návratový typ | Popis |
|---|---|---|
| findAll | List | Lists all users.<br>@return |
| findByEmail | List | Finds a user by his email.<br>@return Users with such email in a List.<br>Parametry:<br>**email: String** - User email. |
| findById | List | Finds a user by his ID.<br>@return Users with such ID in a List.<br>Parametry:<br>**id: int** - User ID. |
| findByUsername | List | Finds a user by his username.<br>@return Users with such username in a List.<br>Parametry:<br>**username: String** - User username. |
| persist | void | Persists a user.<br>Parametry:<br>**user: U** - User to be saved. |
| remove | void | Removes a user.<br>Parametry:<br>**user: U** - User to be removed. |
| update | void | Updates a user.<br>Parametry:<br>**user: U** - User to be updated. |

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

# 2. Realizace případů užití

## 2.1 Posílání zpráv

Graf zachycující případ užití pro posílání zpráv.

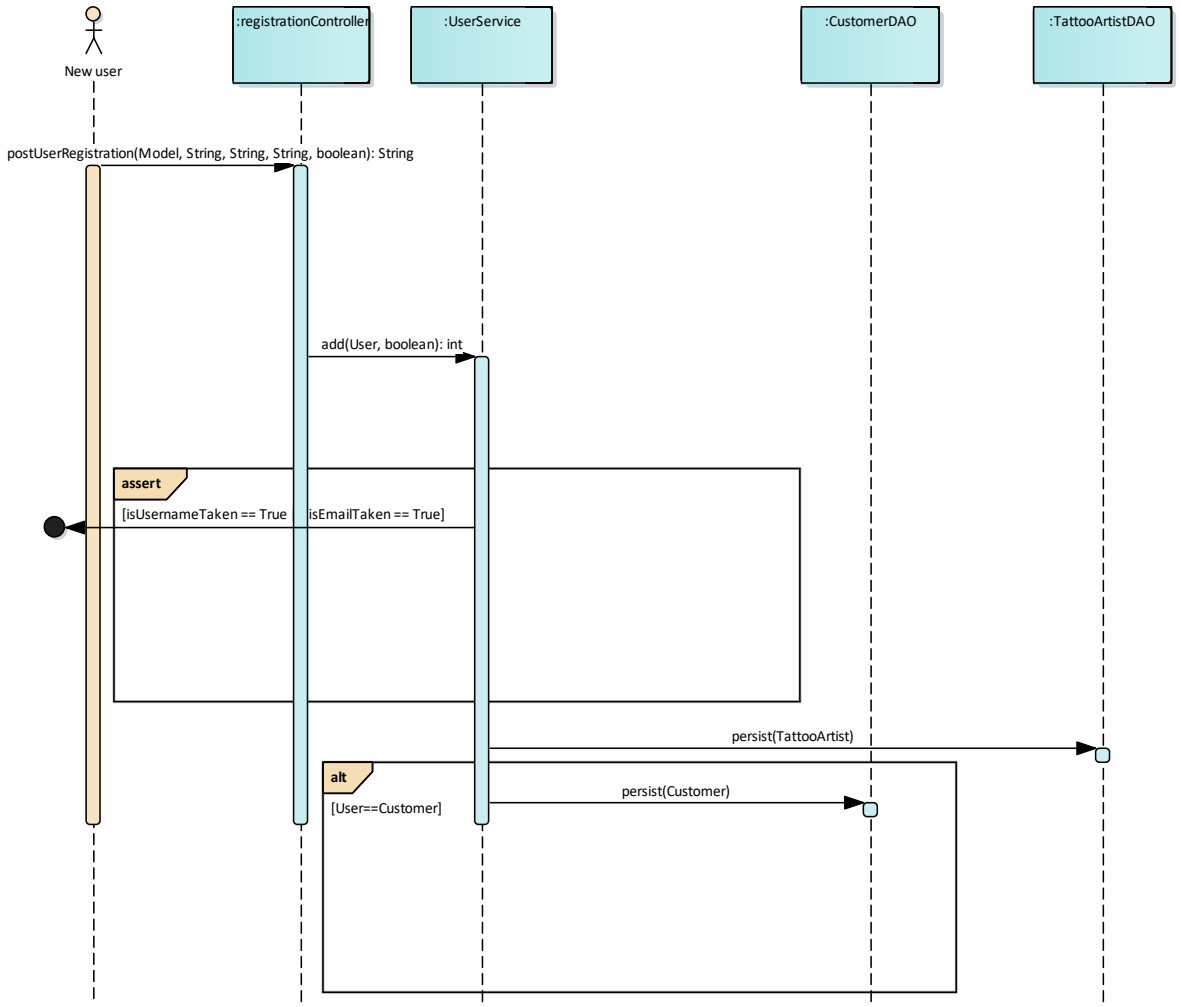



Obrázek 17 - Posílání zpráv

When user wants to send a message, he must sign in his profile and send a message. The message is saved. Each user then can see his incoming messages from these saved messages.

## 2.2 Registrace

Graf zachycující případ užití pro registraci uživatele.




Obrázek 18 - Registrace

Use case registration. Registration Controller calls User Service's add method. User service checks, if user with assigned email or username already exists in database. If not, User service calls CustomerDAO or TattooArtistDAO to add new user to the database.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE