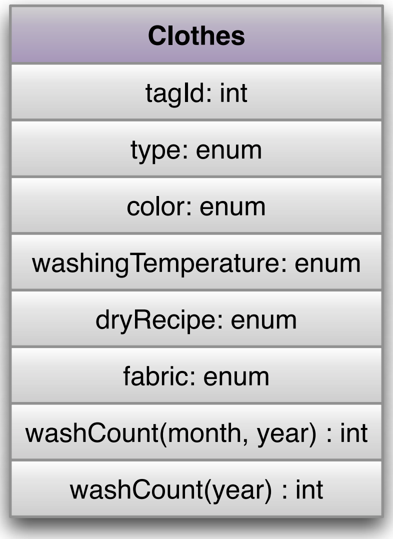
Group Agreement between subtask 5 and 6

This document will describe which agreements and requirements that have been determined by the two groups.

The Clothes object



Clothes

tagId: int

type: enum

color: enum

washingTemperature: enum

dryRecipe: enum

fabric: enum

washCount(month, year) : int

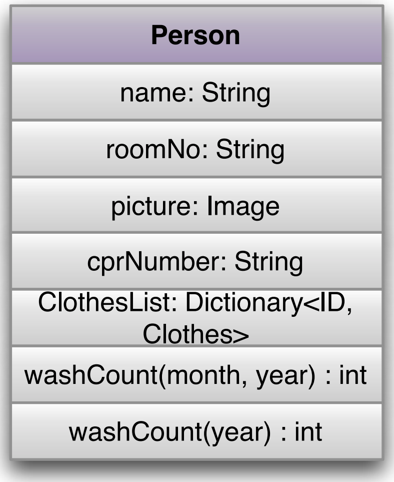
washCount(year) : int

The Cloth obviously has an ID attached, which is given by the tag that has been attached to the cloth. The different properties of the cloth are enums; these can of course be different, but are static at this moment. The enums has the following values:

- Type { Sweater, Shirt, Pants, Underwear, WashingBag, Dress, Shorts, Towel, BedLinen}
- Temperature { Degree30, Degree40, Degree60, Degree95, Handwash}
- Fabric { Cotton, Synthetic, Wool}
- DryRecipe { HangDry, Tumbling }
- Color { Colored, White }

Moreover, the *washCount* method, should use the database to collect data about how many times the specific cloth has been tagged in the washing room, and perhaps attach a timestamp for the tagging in the washing room. In that way, it possible to collect the number of times the cloth has been washed in a given month or a given year. These methods should be supported by the serverFacade/interface. The primary key for the clothes, should be the tagId, since it identifies each piece of clothes. Notice that once a piece of clothes has been registered, it has the same tagId until it will be deleted. The tagId will not change during a wash or other process.

The Person object



Person

name: String

roomNo: String

picture: Image

cprNumber: String

ClothesList: Dictionary<ID, Clothes>

washCount(month, year) : int

washCount(year) : int

Each person that is created by the registering app, has a collection of clothes attached to it. In the same way as the cloth, it is able to ask the person, how many pieces of clothes in a year or a month they wash. This should also be supported by the serverFacade.

When a person picture has to be saved into the DB, the picture should be uploaded to a server, and the url to that picture should be saved with the person, so it is possible to retrieve that picture in any given moment on any given computer.

The primary key for the person should be the cprNumber, since it identifies each person.

The ServerFacade

The serverFacade is the interface between the registering app and the server. The serverFacade interface should be developed by subtask 5.

The methods in the serverFacade only operates on simple data types, and should therefore be simple to save and retrieve from the database. A detailed class diagram of the interface is illustrated in the end of this section. The methods with a green indicator, are the first priority methods that should be implemented in the first version of the serverFacade interface. The blue indicators represents methods that can be implemented in the second version of the serverFacade interface.

The methods in the interface will be described briefly in the following:

getCprNumbers()

This method will return a string array with all the CPR-numbers saved in the database.

getRoomNumber()

This method will return a string with the room numbers saved in the database by the person with the specified CPR-number.

getName()

This method will return the name of the person living in the specified CPR-number.

getPictureUrl()

This method will return the picture url on some FTP server that the person with the specified CPR-number is attached to.

getClothesIds()

This method will return an integer array of all the tag ids the person with the specified CPR-number has.

getClothesType()

This method returns the string representation of the type of the clothes with the specified Id - see section object clothes.

getClothesColor()

This method returns the string representation of the color (colored/white) of the clothes with the specified Id - see section object clothes.

getClothesWashingTemp()

This method returns the string representation of the washing temperature of the clothes with the specified Id - see section object clothes.

getClothesDryRecipe()

This method returns the string representation of the drying recipe of the clothes with the specified Id - see section object clothes.

getClothesFabric()

This method returns the string representation of the fabric of the clothes with the specified Id - see section object clothes.

getWashCount() X 4

This method returns the amount of washes each clothes (tagId) or person (CPR-number) has gone through, in some specified month or year. E.g. getWashCount(7, 2009, “0123456789”) should return the number of pieces of clothes the person with CPR-number “012…” has send washing during July (7) in the year 2009. Likewise getWashCount(2010, 1001295125) should return the number of times clothes with id “1001295125” has been send to be washed during the year 2010.

saveImage()

This method should save or update the specified Bitmap object to a FTP-server, and save the url of the picture to the database on the person with the specified CPR-number.

getImage()

This method should return a Bitmap object with the picture attached to the person with the specified CPR-number, and return null if no image has been saved to the person.

savePerson()

This method should save or update a person with the specified CPR-number, name, room number and picture, and return a boolean indicating whether the process was a success.

deletePerson()

This method should delete the person with the specified CPR-number and all the information that is attached to that person e.g. clothes, and return a boolean indicating whether the process was a success.

saveClothes()

This method should save or update a piece of clothes with the specified information, to the person with the specified CPR-number, and returning a boolean indicating whether the process was a success.

deleteClothes()

This method should delete the piece of clothes with the specified tagId and return a boolean indicating whether the process was a success.

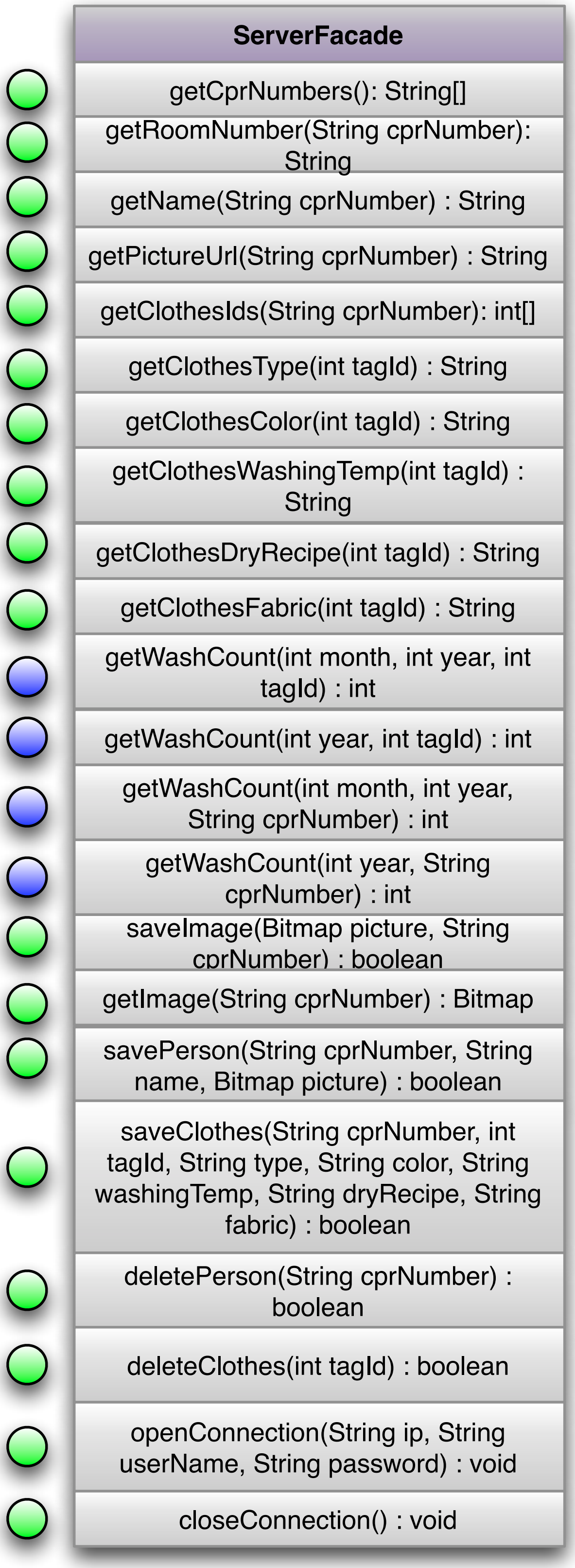
openConnection()

This method should open the database connection to the database with the specified IP-address. It is optional whether username and password should be required.

closeConnection()

This method should close the database connection, and this allows us to control when it is optimal to open and close the connection to the database.

If Subtask 5 have any questions, suggestions, concerns or wishes, please don't hesitate contacting us at dader06@student.sdu.dk



ServerFacade

🟢 getCprNumbers(): String[]

🟢 getRoomNumber(String cprNumber): String

🟢 getName(String cprNumber) : String

🟢 getPictureUrl(String cprNumber) : String

🟢 getClothesIds(String cprNumber): int[]

🟢 getClothesType(int tagId) : String

🟢 getClothesColor(int tagId) : String

🟢 getClothesWashingTemp(int tagId) : String

🟢 getClothesDryRecipe(int tagId) : String

🟢 getClothesFabric(int tagId) : String

🔵 getWashCount(int month, int year, int tagId) : int

🔵 getWashCount(int year, int tagId) : int

🔵 getWashCount(int month, int year, String cprNumber) : int

🔵 getWashCount(int year, String cprNumber) : int

🟢 saveImage(Bitmap picture, String cprNumber) : boolean

🟢 getImage(String cprNumber) : Bitmap

🟢 savePerson(String cprNumber, String name, Bitmap picture) : boolean

🟢 saveClothes(String cprNumber, int tagId, String type, String color, String washingTemp, String dryRecipe, String fabric) : boolean

🟢 deletePerson(String cprNumber) : boolean

🟢 deleteClothes(int tagId) : boolean

🟢 openConnection(String ip, String userName, String password) : void

🟢 closeConnection() : void