The headquarters.

The system is a dedicated PHP server that interacts with MongoDB database. It allows the authorized headquarters user to perform the account and class related operations:

**Main Functions:**

**-Creating a class:**

● The system asks for the class name

○ limited to 50 characters

● The system asks the commander’s personal number

○ limited to 7 numbers

● The system asks for the number of soldiers

○ from 1 to 65 soldiers

● The system asks for the soldier’s personal number - which will generate all the soldier’s data (image, role, exceptions, etc..)

○ if the soldier number doesn’t exist in the system, then the system will give an error that the soldier doesn’t exist in the system

○ else the system will show an option of adding this soldier to the class

● The system shows a “save” option which will add the class with it’s specification to the system’s DB

**-Remove class:**

● The system asks for the class name

○ If the class name is not the system asks for the input again

● The system asks for commander’s personal number

○ If the number isn't verified the system asks for the input again

● The system checks that all class’s soldiers’ numbers are valid

○ if it finds a number that isn't verified, it pushes an error stating the invalid

number.

● The system pushes “Are you sure want to remove class ‘class name’?“

○ if the input is true, the system shall change the class to none in its current soldiers’ data, and then remove the class from the DB

**-Creating account**

● The system asks for the soldier’s personal number

○ limited to 7 numbers

● The system checks with the MongoDB database the soldier’s information

○ if personal number wasn’t found, the system will push an error that the soldier doesn’t exist in the system

○ if exists, show the soldier’s personal info

■ if the class doesn’t exist in the system, add class as none

■ the user can edit the personal number to show another soldier info and can add the account to the system as well

● After creating an account, the system will generate a one-time password

**-Updating accounts**

● The system asks the soldier’s personal number that the user want to update or delete

○ If exist:

■ The system will show the soldier’s information with two buttons: update and delete

○ Else:

■ show error that the personal number doesn’t exist within the squad

● If the user chooses to delete the system asks to enter the password

○ If the password is valid then the system proceeds to the delete function.

○ If not then the system alerts that the password is incorrect and there are 3 more tries, after those tries, the action is invalid for an hour.

○ In the delete function there is an “are you sure” safety measure to prevent unwanted deletion.

-If the user presses “yes” the system deletes account and notifies the user of successful deletion

-Else the system notifies the user that the account was not deleted.

● If the user chooses to update the system will ask for the password to permit the change

○ If the password is valid then the system will ask for the new fullName of the soldier.

-If the new fullName is valid, it updates it, shows a success notification and updates the authorities.

-Else the system notifies that the name was not updated.

○ If not then the system alerts that the password is incorrect and there are 3 more tries, after those tries, the action is invalid for an hour.

**-Help recovering account:**

● The system will ask for the soldier’s personal number

○ If the number isn't verified the system asks for the input again

● The system will generate and update the new password

**-Show Main Page:**

● The system will have a link to select a class

● If a class is selected, the system shows in a table all soldiers information in the selected class

○ Show the personal number, name, and class, if the soldier not in a class it will show none.

● The system will have a link to create a class

● The system will have a link to create a new account

● The system will have a link to update a class

**Service Functions:**

**-soldierExists:**

● Check that soldier actually exists in the system

**-encrypt:**

● encrypt a password (may be used for other strings as well)

**-verifyPassword:**

● verify the password. Encrypts before sending.

**-updateSoldierClass**

● update class for a soldier

**-deleteSoldierFromCollection**

●delete the soldier from connected MongoDB collection.

**Notes:**

-Password is encrypted before being sent, it is possible to choose encryption method.

-Some link functions are written in separate PHP documents, which reference the main one and are called upon pressing link.

-The headquarters object utilizes Session system to make sure it connects properly to PHP files in links (that session remains the same and no security issues appear).