2021

Проект «Праска»

СИСТЕМНА ІНЖЕНЕРІЯ

ЕСМІРА АБДУЛЛАЄВА, ГРУПА ФБ-93

Завдання до лабораторної роботи №1

1. Розбити систему на функціональні підсистеми та описати функції підсистем;
2. Розробити структурну діаграми на мові SysML згідно описаних підсистем.

Завдання до лабораторної роботи №2

Побудувати наступні діаграми:

- use cases diagram
- requirements diagram
- state machine diagram

Завдання до лабораторної роботи №3

Побудувати наступні діаграми:

- sequences diagram
- activity diagram
- class diagram

Проект

- побудова об’єктно-орієнтованої моделі системи

Завдання №1

Підсистеми та опис їх функцій

I. Система живлення:

- забезпечує живлення праски

II. Система збереження води:

- забезпечує збереження води для подальшого перероблення її у пару

III. Система подачі пару:

- забезпечує подачу пару для відпарювання/розгладжування тканини

IV. Система регулювання температури:

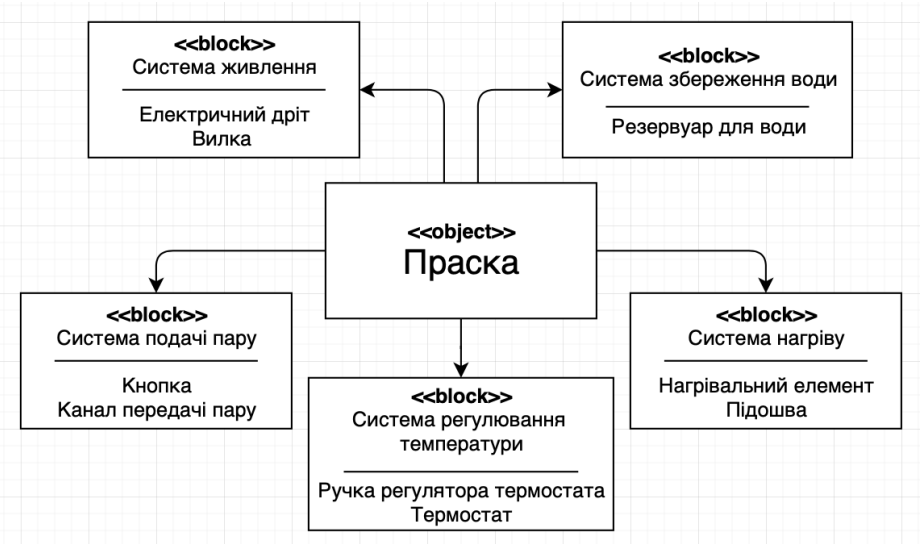
- надає змогу встановлення температури для подальшого використання праски

V. Система нагріву:

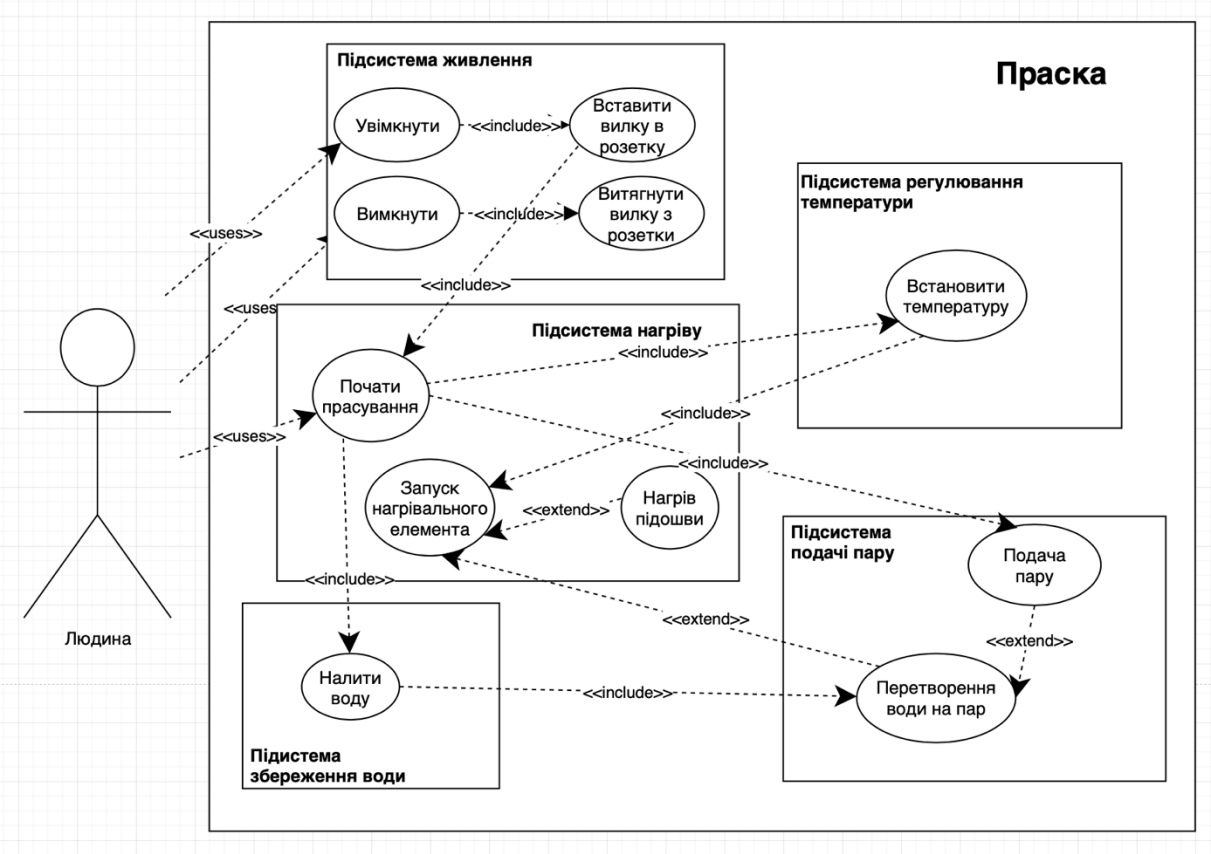
- відповідає за рівномірний розподіл температури
- перетворення води на пар

Коментар: було змінено назви підсистем, але їх функціонал залишився тим самим.

Завдання №2

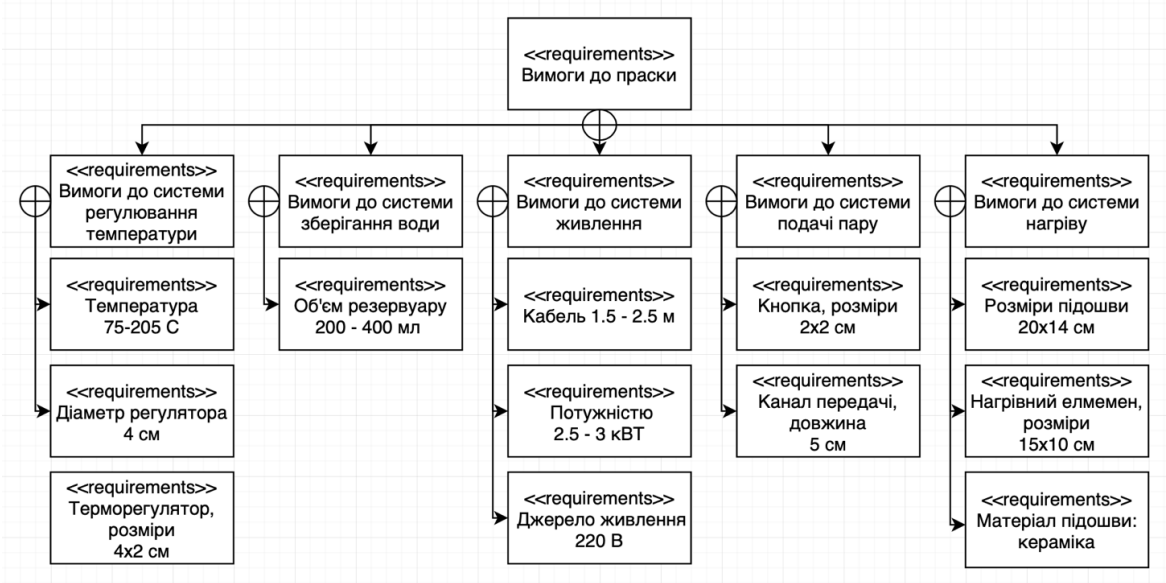


Use cases diagram

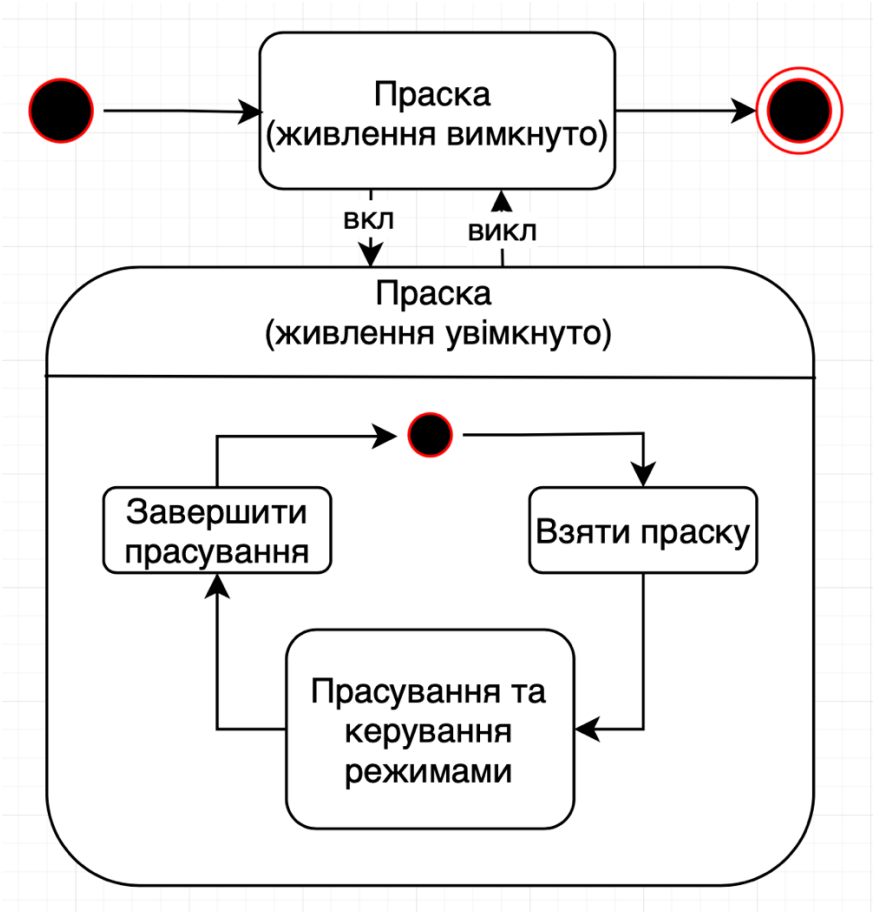


Коментар: у блок живлення додано вилку. Початок прасування включає в себе регулювання температури, наливання води та подачу пару. Запуск нагрівального елемента впливає на перетворення води на пару та на нагрів підошви.

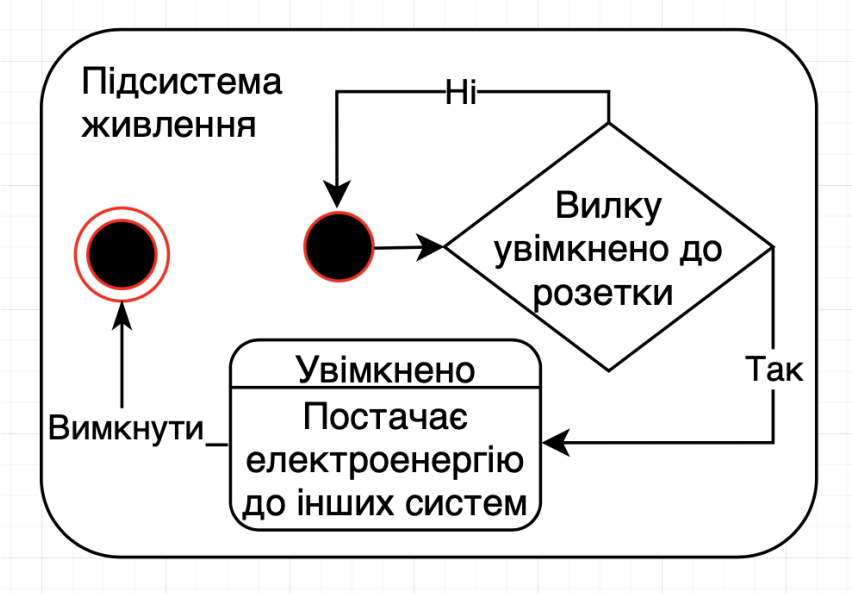
Requirements diagram

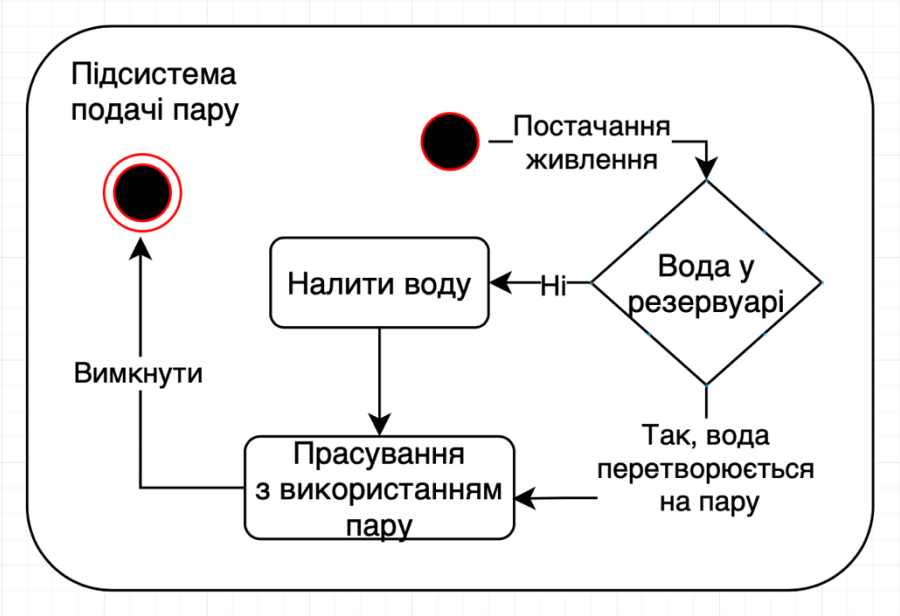


State machine diagram

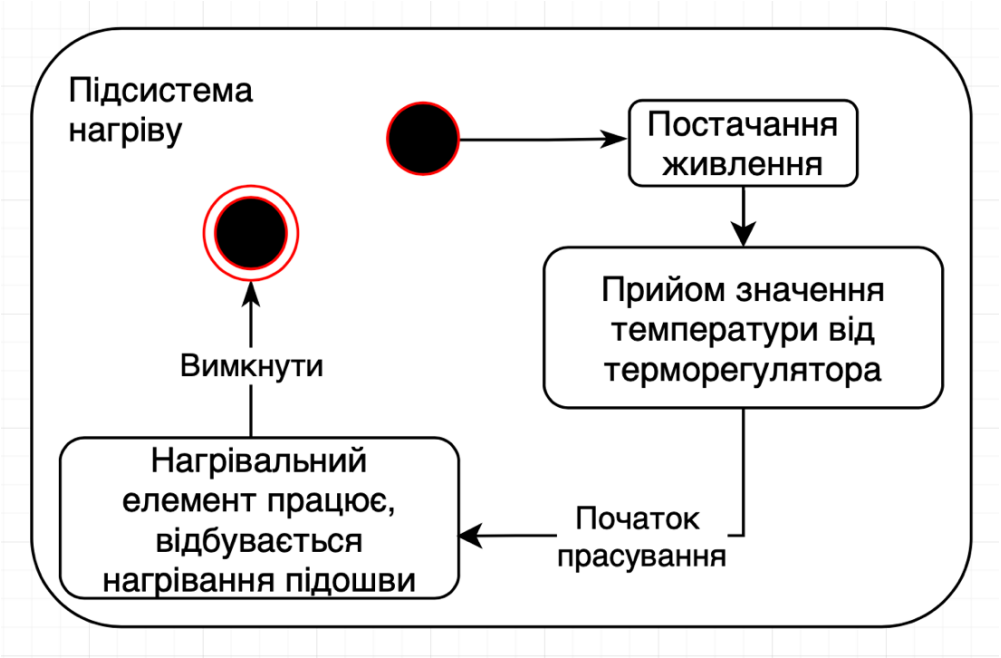


Коментар: діаграма зазнала змін, щоб в подальшому реалізувати спочатку увімкнення праски, а потім вже всі подальші дії керування (зміна температури, додавання поди, прасування з використанням пару).

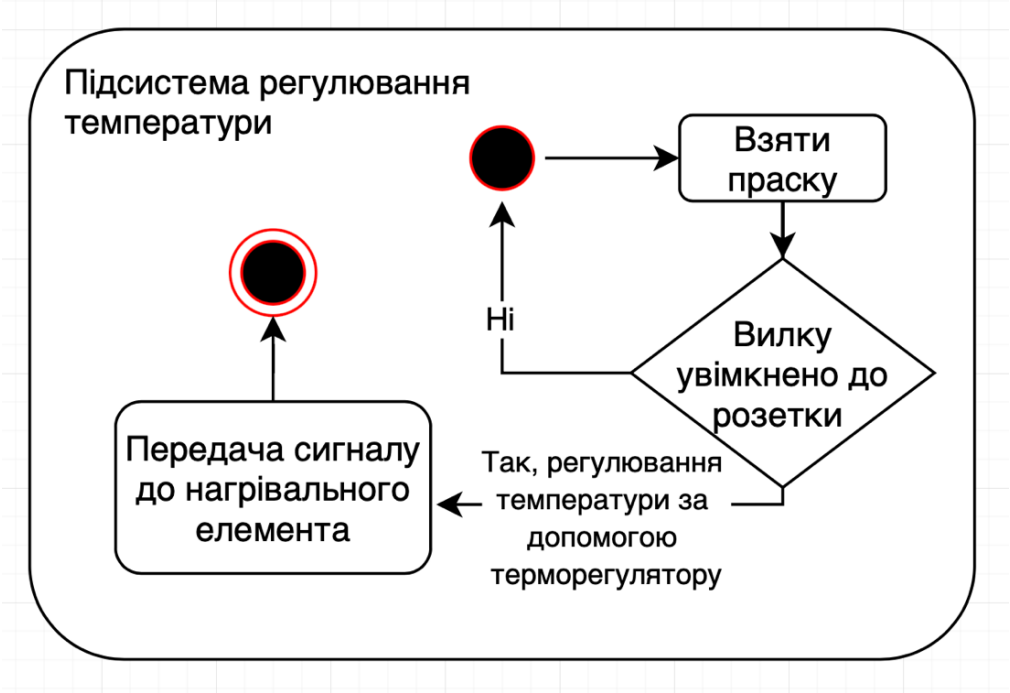




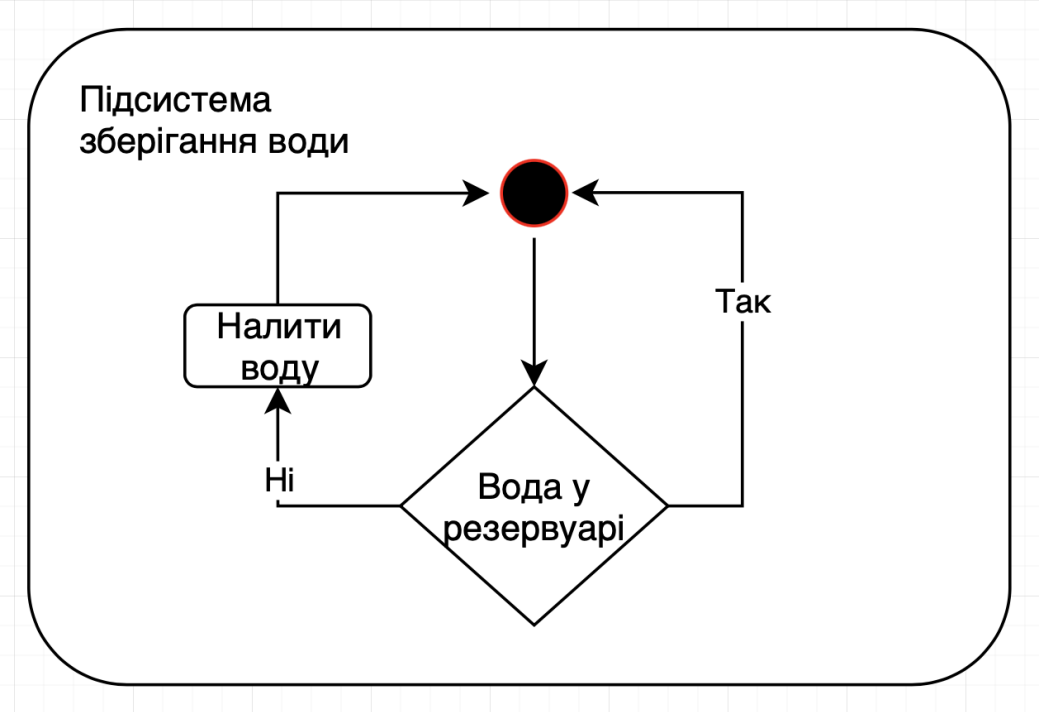
Коментар: додано блок перевірки на наявність води в резервуарі.



Коментар: додано передачу значення температури від терморегулятора (підсистему регулювання температури).

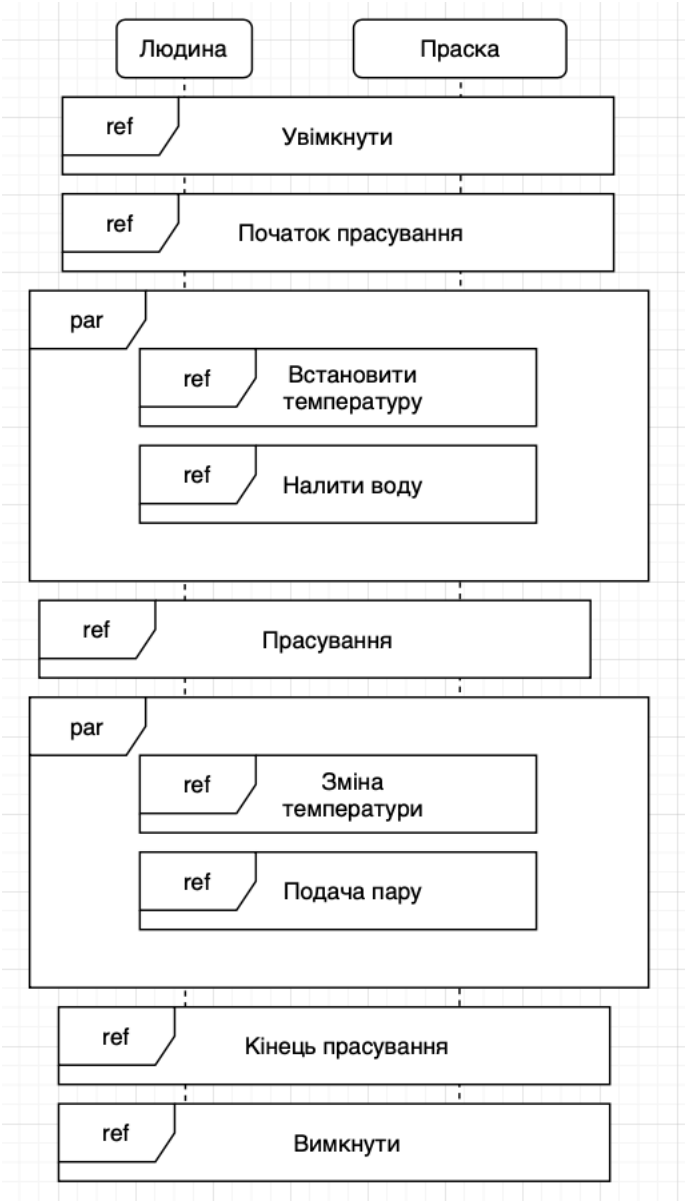


Коментар: додано передачу температури до нагрівального елемента для остаточної зміни температури.

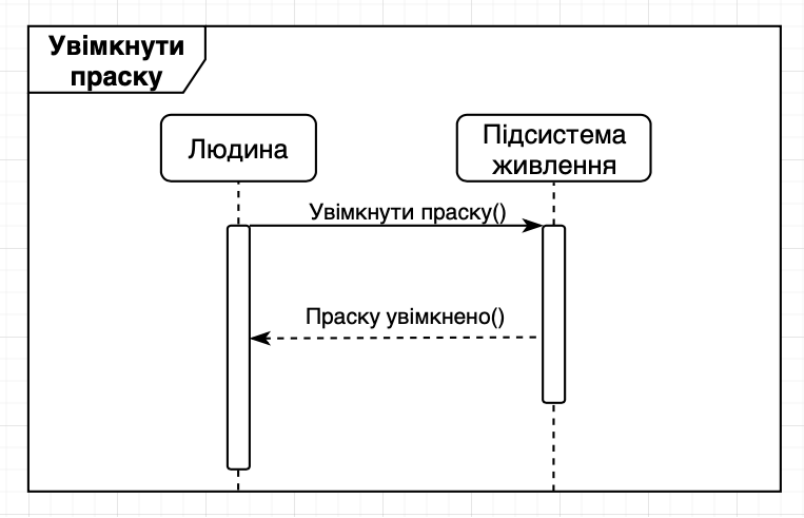


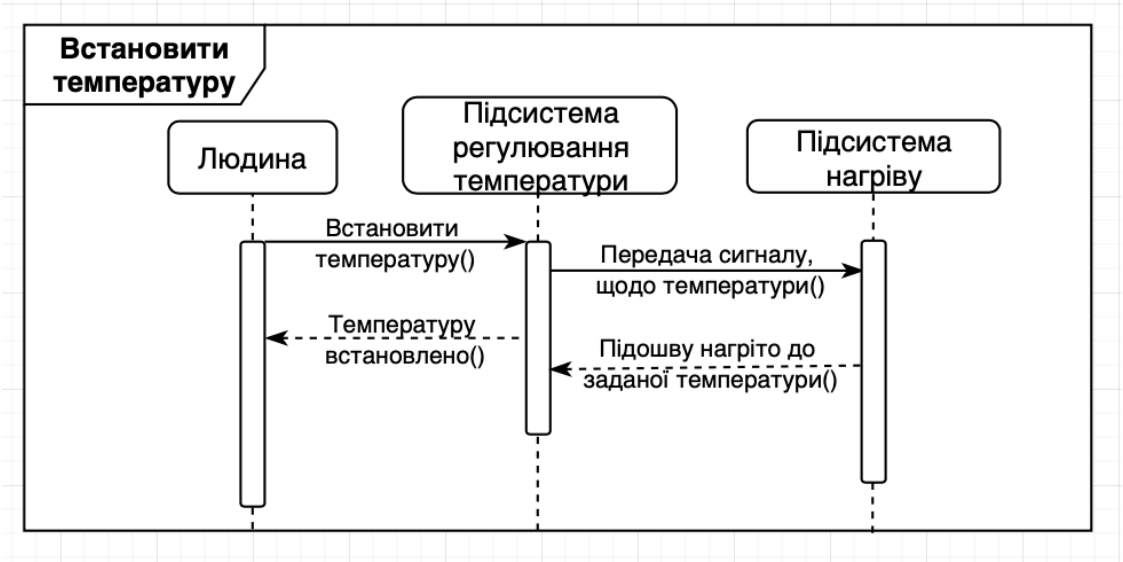
Коментар: спрощення схеми (відсутність моменту відкривання камери)

Sequences diagram

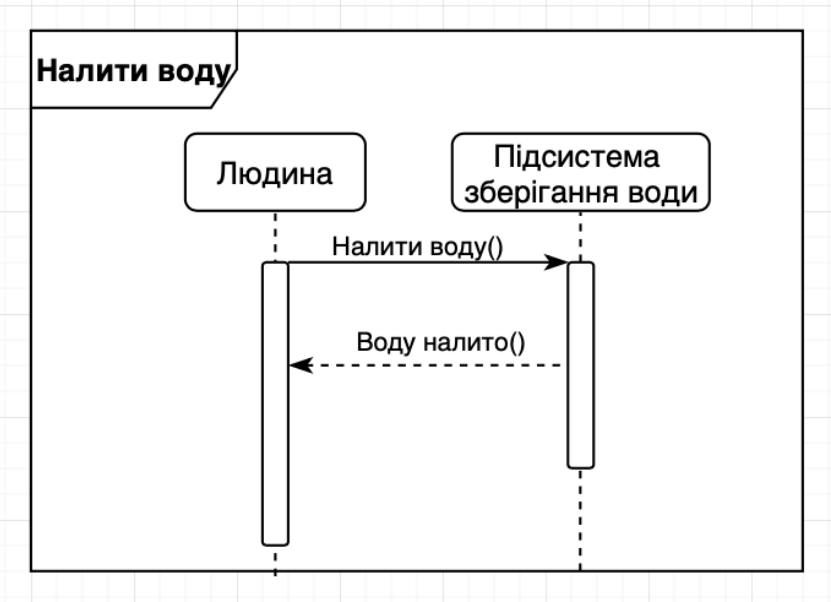


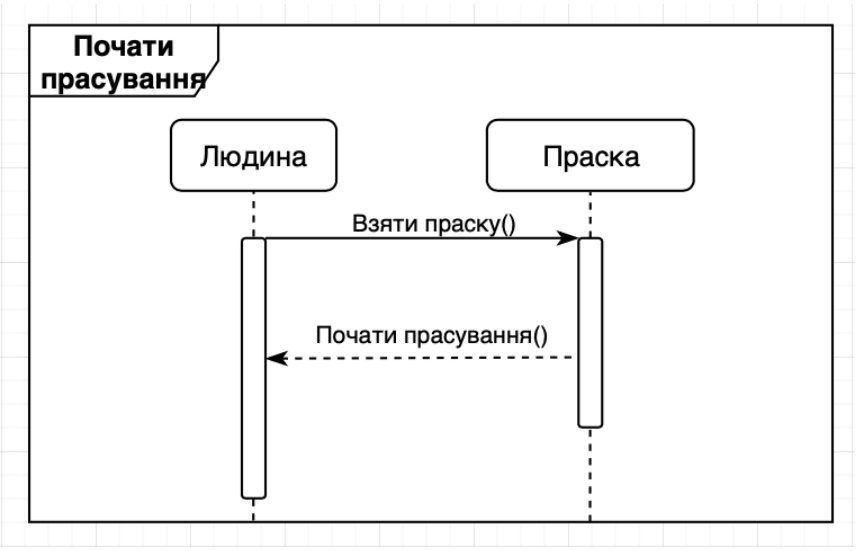
Коментар: першою дією стало вмикання праски, а вже потім встановлення температури та додавання води й початок прасування. Бо навіть не наливши воду та не встановивши температуру (на прасці й так є значення температури, на яку вказує колесо терморегулятора) можна користуватися праскою.

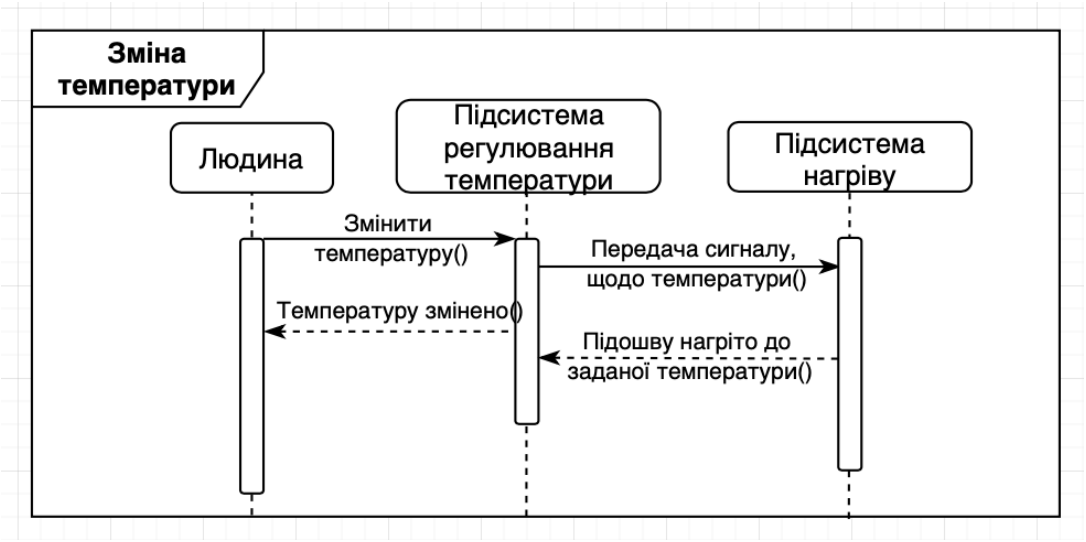




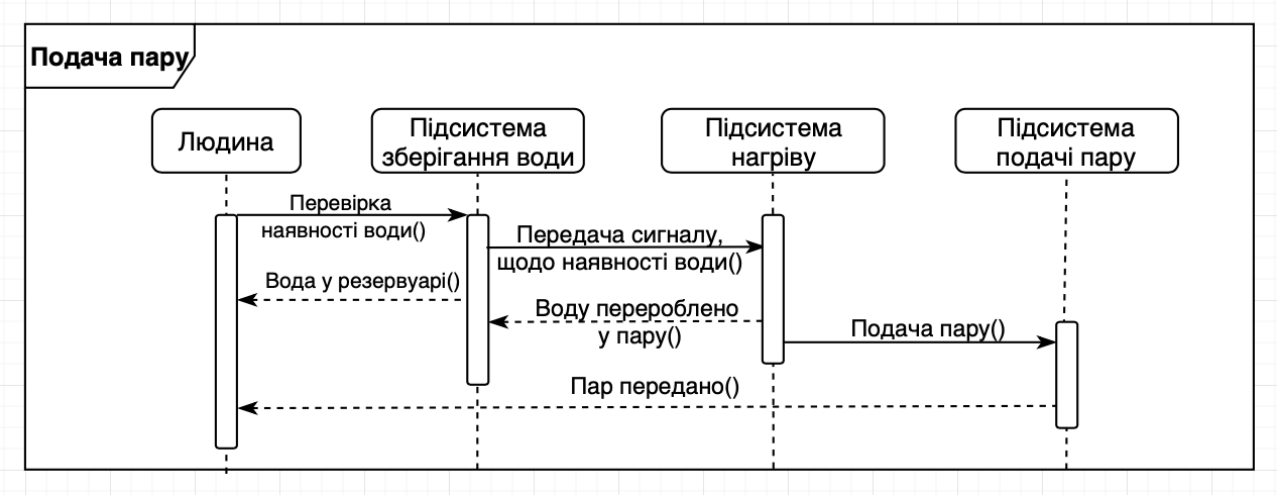
Коментар: встановлення температури відбувається безпосередньо після нагріву підошви, тож було додано взаємодію з системою нагрівання.



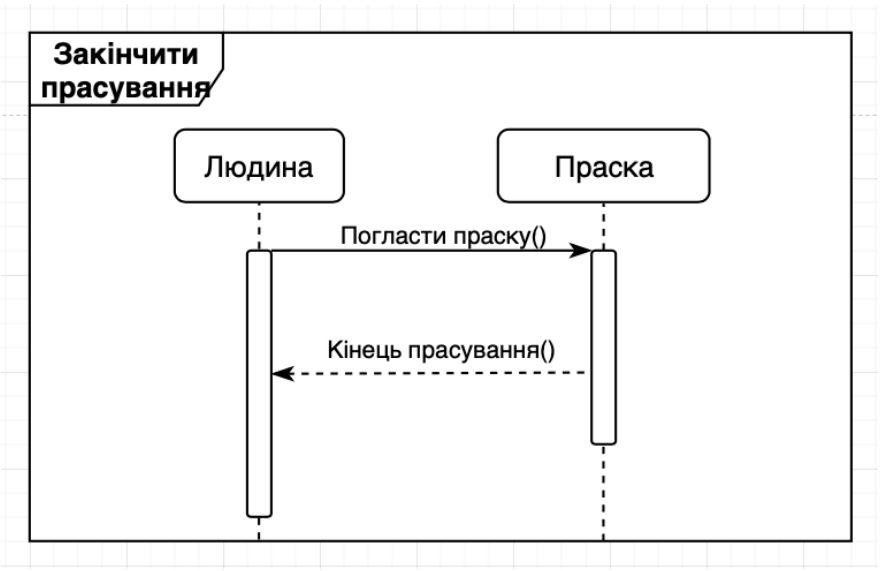


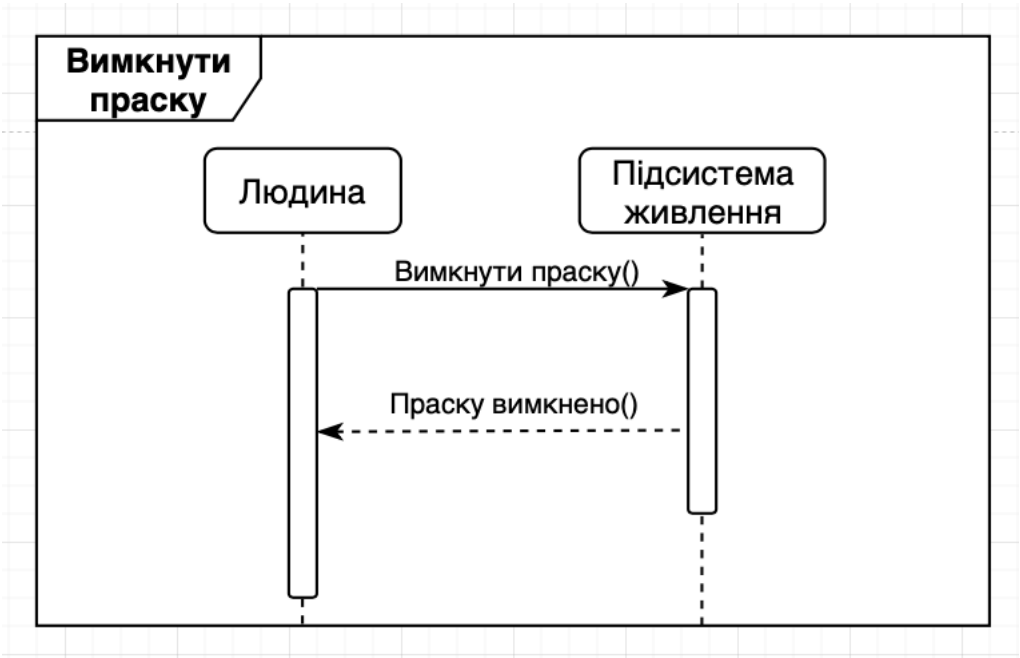


Коментар: остаточна зміна температури відбувається безпосередньо після нагріву підошви, тож було додано взаємодію з системою нагрівання.

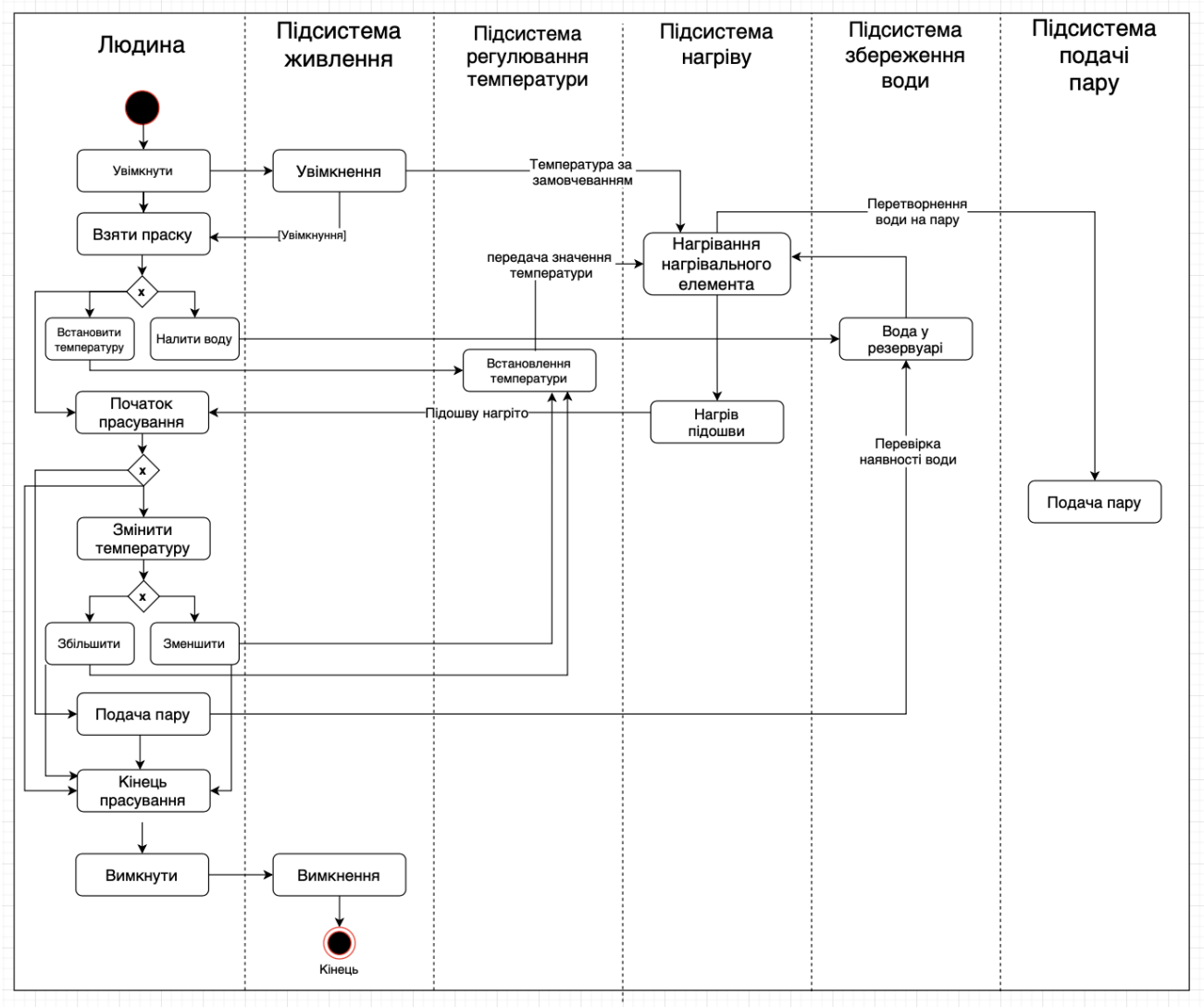


Коментар: вода перетворюється на пару після взаємодії з нагрівальним елементом, тому було додано систему нагрівання.


Закінчити прасування
Людина
Праска
Погласти праску()
Кінець прасування()


Вимкнути праску
Людина
Підсистема живлення
Вимкнути праску()
Праску вимкнено()

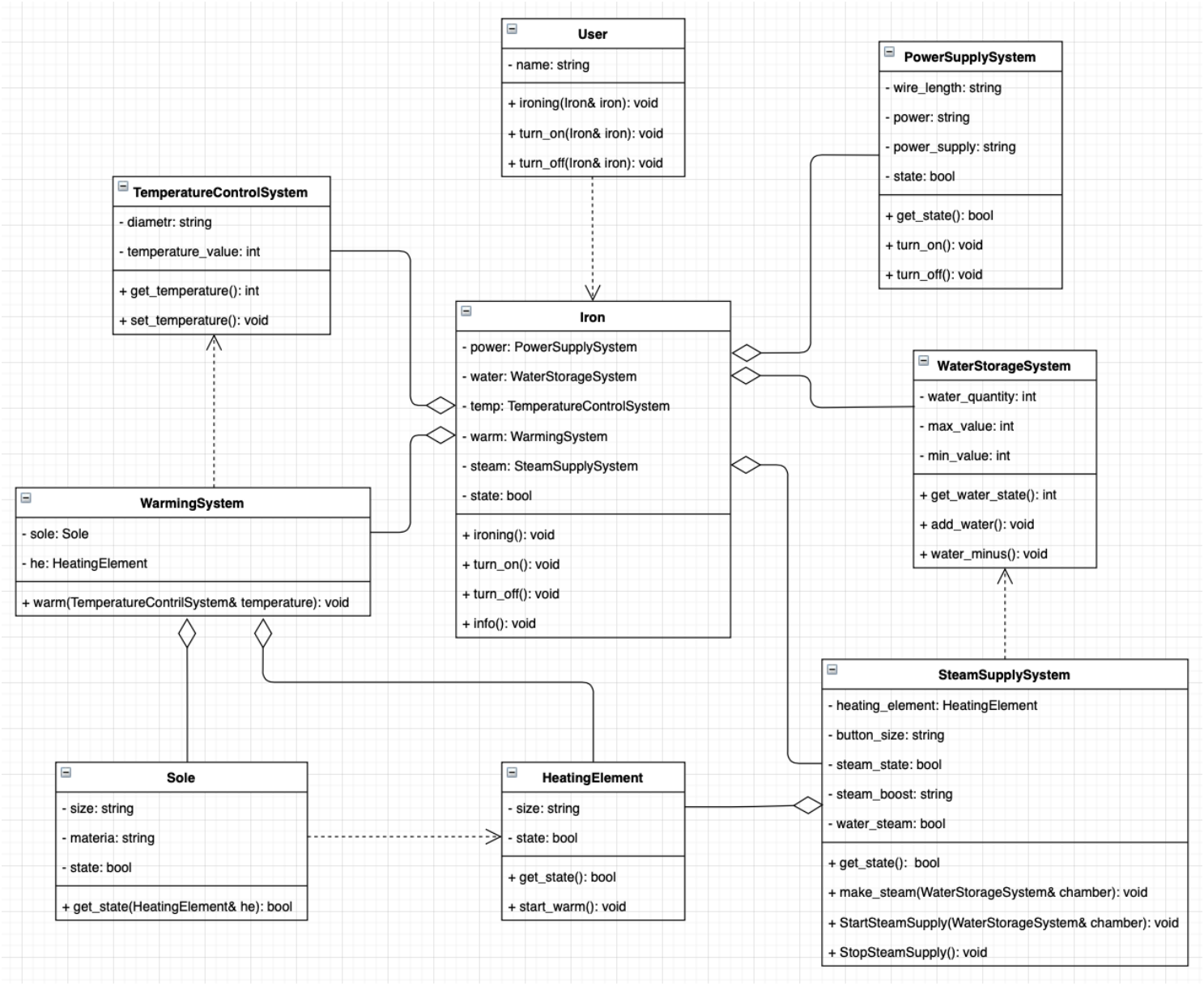
Activity diagram



Коментар: діаграму було розширено з урахуванням змін виконаних у попередніх таблицях, а саме:

- прасувати можна з температурою, яка була на прасці в момент увімкнення;
- після взяття праски можна почати прасування або змінити температуру або налити воду, для подальшого використання пару;
- вода перетворюється на пару після нагрівання;
- температуру можна регулювати у дві сторони: збільшення та зменшення.

Class diagram



Проект

```
#include <iostream>
#include <string>
#include <unistd.h>

using namespace std;

// система живлення
class PowerSupplySystem
{
private:
    string wire_length = "1.5 meters";
    string power = "2.4-3 kW";
    string power_supply = "230 V";
    bool state = false; // вимкнуто за замовчуванням
public:

    bool get_state() {
        return this->state;
    }

    void turn_on() {
        cout << "# IRON ON #" << endl;
        this->state = true;
    }

    void turn_off(){
        cout << "# IRON OFF #" << endl;
        this->state = false;
    }
};

// система регулювання температури
class TemperatureControlSystem
{
private:
    string diametr = "4 cm";
    int temperature_value = 90;
public:

    int get_temperature() {
        return this->temperature_value;
    }

    void set_temperature() {
        int min_temp = 75;
        int max_temp = 205;
        char ans;
        cout << "* set temperature (u)p or (d)own:" << endl;
        cin >> ans;
        switch (ans) {
            case 'u':
                if (temperature_value <= max_temp)
                    this->temperature_value+=10;
                else
                    cout << "x it is maximum value of temperature x" << endl;
                break;
            case 'd':
                if (temperature_value >= min_temp)
                    this->temperature_value-=10;
```

```
            else
                cout << "x it is minimum value of temperature x" << endl;
            break;
        default:
            cout << "Wrong input" <<endl;
            break;
    }
  }
};

// система збереження води
class WaterStorageSystem
{
private:
  int water_quantity = 90;
  int max_volume = 400;
  int min_volume = 100;
public:

  int get_water_state() {
    return this->water_quantity;
  }

  void add_water() {
    char ans;
    cout << "* minimun water quantity: " << this->min_volume << endl;
    cout << "* add (y)es/(n)o:" << endl;
    cout << ">> ";
    cin >> ans;
    switch (ans) {
      case 'y':
        if (water_quantity < min_volume || water_quantity < max_volume) {
          this->water_quantity+=(max_volume-water_quantity);
          cout << "! water added !" << endl;
        }
        else
          cout << "! you have enough water !" << endl;
        break;
      case 'n':
        if(water_quantity <= max_volume && water_quantity >= min_volume)
          cout << "! okay !" << endl;
        else {
          cout << "! you need water !" << endl;
        }
        break;
      default:
        cout << "Wrong input" <<endl;
        break;
    }
  }

  void water_minus() {
        this->water_quantity-=5;
  }
};

// нагривний елемент
class HeatingElement
{
private:
  string size = "17 cm";
  bool state = false; // за замовчуванням нагрівальний елемент холодний
public:

  bool get_state() {
```

```
        return this->state;
    }
    void start_warm(TemperatureControlSystem& temperature) {
        cout << "* heating element -> " << temperature.get_temperature() << "*C" << endl;
        this->state = true;
    }
};

// підошва
class Sole
{
private:
    string size = "20 cm";
    string material = "ceramics";
    bool state = false;
public:
    bool get_state(HeatingElement& he) {
        cout << "* sole -> warmed up" << endl;
        return this->state=he.get_state();
    }
};

class WarmingSystem
{
    Sole sole;
    HeatingElement he;
public:
    void warm(TemperatureControlSystem& temperature) {
        he.start_warm(temperature);
        sleep(2);
        sole.get_state(he);
    }
};

// система подачі пару
class SteamSupplySystem
{
private:
    string button_size = "1 cm";
    bool steam_state = false; // за замовчування кнопку не натиснуто
    string steam_boost = "90 - 210 g/min";
    bool water_steam = false;
    HeatingElement heating_element;
public:
    bool get_state() {
        return this->steam_state;
    }
    void make_steam(WaterStorageSystem& chamber) {
        if (chamber.get_water_state() >= 100){
            cout << "* water -> steam" << endl;
            water_steam = true;
        }
        else
            cout << "x not enough water x" << endl;
    }
    // настиснути кнопку для подачі пару
    void StartSteamSupply(WaterStorageSystem& chamber) {
        this->make_steam(chamber);
        if (water_steam == true){
            cout << "! steam in progress !" << endl;
            chamber.water_minus();
            this->steam_state = true;
        }
        else {}
    }
```

```
    // настиснути кнопку для завершення подачі пару
    void StopSteamSupply() {
        cout << "! steam finished !" << endl;
        this->steam_state = false;
    }
};
// праска
class Iron
{
    PowerSupplySystem power;
    SteamSupplySystem steam;
    TemperatureControlSystem temp;
    WarmingSystem warming;
    WaterStorageSystem water;
    bool state = false;
public:
        void turn_on() {
        this->power.turn_on();
        this->warming.warm(temp);
    }

    void turn_off() {
        this->power.turn_off();
    }

    void info() {
        cout << "------------------------" << endl;
        cout << "|" << " Ironing process   " << "|" << this->state << "|" << endl;
        cout << "|" << " Temperature       " << "|" << this->temp.get_temperature() << "|" << endl;
        cout << "|" << " Water quanity     " << "|" << this->water.get_water_state() << "|" << endl;
        cout << "|" << " Steam state       " << "|" << this->steam.get_state() << "|" << endl;
        cout << "------------------------" << endl;
    }
    void ironing() {
        int flag = 1;
        while (flag)
        {
            int ans;
            cout << "\n----------Iron----------" << endl;
            info();
            cout << "0 - Start ironing" << endl;
            cout << "1 - Set temperature" << endl;
            cout << "2 - Add water" << endl;
            cout << "3 - Use steam" << endl;
            cout << "4 - Stop use steam" << endl;
            cout << "5 - The end" << endl;
            cout << ">> ";
            cin >> ans;
            switch (ans) {
                case 0:
                    if (state == false){
                        this->state = true;
                        cout << "...ironing..." << endl;
                    }
                    else
                        cout << "! already ironing !" << endl;
                    break;
                case 1:
                    this->temp.set_temperature();
                    this->warming.warm(temp);
                    break;
                case 2:
                    if (state == false){
                        this->water.add_water();
                        this->water.get_water_state();
```

```
                }
                else
                    cout << "! stop ironing first !" << endl;
                break;
            case 3:
                if (state == true){
                    if (steam.get_state() == false) {
                        this->steam.StartSteamSupply(water);
                    }
                    else
                        cout << "! steam already in progress !" << endl;
                }
                    else
                        cout << "! start ironing first !" << endl;
                break;
            case 4:
                if(steam.get_state() == true) {
                    this->steam.StopSteamSupply();
                }
                else
                    cout << "! no steam used !" << endl;
                break;
            case 5:
                if (state == true) {
                char enter;
                cout << "* (f)inish or (c)hange parameters?" << endl;
                cout << ">> ";
                cin >> enter;
                switch (enter) {
                        case 'f':
                        if (steam.get_state() == true) {
                                this->steam.StopSteamSupply();
                        }
                            cout << "! stop ironing !" << endl;
                            state = false;
                            flag = 0;
                            break;
                        case 'c':
                        if (steam.get_state() == true){
                                this->steam.StopSteamSupply();
                        }
                            cout << "! stop ironing !" << endl;
                            state = false;
                            break;
                        default:
                            cout << "Wrong input!" << endl;
                            break;

                        }
                }
                else {
                    state = false;
                    flag = 0;
                }
              break;
            default:
                cout << "Wrong input!" << endl;
            break;
        }
    }
  }

};

class User
```

```
{
private:
    string name = "Esmira";
public:
    void turn_on(Iron& iron) {
        iron.turn_on();
    }
    void turn_off(Iron& iron) {
        iron.turn_off();
    }
    void Ironing(Iron& iron) {
        cout << "            Iron" << endl;
        int choice;
        bool flag = 1;
        cout << "* insert plug into the socket" << endl;
        cout << "      enter '1' to do it!" << endl;
        cout << ">> ";
        cin >> choice;
        if (choice != 1) {
            cout << "Try one more time!" << endl;
            return;
        }
        else {
            turn_on(iron);
        }
        while(flag) {
            iron.info();
            cout << endl;
            cout << "* what's next?" << endl;
            cout << "1 - Take Iron" << endl;
            cout << "2 - Switch OFF" << endl;
            cout << ">> ";
            cin >> choice;
            switch (choice) {
                case 1:
                    iron.ironing();
                    break;
                case 2:
                    turn_off(iron);
                    flag = 0;
                    break;
                default:
                    cout << "Wrong input" << endl;
            }
        }
    }
};
int main()
{
    Iron iron;
    User Me;
    Me.Ironing(iron);
    return 0;
}
```

Приклад виконання

1. Увімкнули вилку в розетку, взяли праску та можемо продовжити працювати з праскою:

```
            Iron
* insert plug into the socket
      enter '1' to do it!
>> 1
# IRON ON #
* heating element -> 90*C
* sole -> warmed up
-----------------------
| Ironing process   |0|
| Temperature       |90|
| Water quanity     |90|
| Steam state       |0|
-----------------------

* what's next?
1 - Take Iron
2 - Switch OFF
>>
```

2. Взявши праску, можна одразу почати прасування. Бо температура була встановлена за замовчуванням:

```
* what's next?
1 - Take Iron
2 - Switch OFF
>> 1

----------Iron----------
------------------------
| Ironing process   |0|
| Temperature       |90|
| Water quanity     |90|
| Steam state       |0|
------------------------
0 - Start ironing
1 - Set temperature
2 - Add water
3 - Use steam
4 - Stop use steam
5 - The end
>> 0
...ironing...
```

3. Налити воду під час прасування не можливо, спочатку треба поставити праску:

```
----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |90|
| Water quanity      |90|
| Steam state        |0|
------------------------
0 - Start ironing
1 - Set temperature
2 - Add water
3 - Use steam
4 - Stop use steam
5 - The end
>> 2
! stop ironing first !
```

4. Після завершення прасування, можна долити воду:

```
5 - The end
>> 5
* (f)inish or (c)hange parameters?
>> c
! stop ironing !

----------Iron----------
------------------------
| Ironing process    |0|
| Temperature        |90|
| Water quanity      |90|
| Steam state        |0|
------------------------
0 - Start ironing
1 - Set temperature
2 - Add water
3 - Use steam
4 - Stop use steam
5 - The end
>> 2
* minimun water quantity: 100
* add (y)es/(n)o:
>> y
! water added !

----------Iron----------
------------------------
| Ironing process    |0|
| Temperature        |90|
| Water quanity      |400|
| Steam state        |0|
------------------------
```

5. Тепер можна прасувати з використанням пару (кількість води зменшується):

```
...ironing...

----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |90|
| Water quanity      |400|
| Steam state        |0|
------------------------
0 - Start ironing
1 - Set temperature
2 - Add water
3 - Use steam
4 - Stop use steam
5 - The end
>> 3
* water -> steam
! steam in progress !

----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |90|
| Water quanity      |395|
| Steam state        |1|
------------------------
```

6. Також, під час прасування можна одразу змінити температуру:

```
----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |90|
| Water quanity      |395|
| Steam state        |1|
------------------------
0 - Start ironing
1 - Set temperature
2 - Add water
3 - Use steam
4 - Stop use steam
5 - The end
>> 1
* set temperature (u)p or (d)own:
u
* heating element -> 100*C
* sole -> warmed up

----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |100|
| Water quanity      |395|
| Steam state        |1|
------------------------
```

7. Зупинка використання пару:

```
----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |100|
| Water quanity      |395|
| Steam state        |1|
------------------------
0 - Start ironing
1 - Set temperature
2 - Add water
3 - Use steam
4 - Stop use steam
5 - The end
>> 4
! steam finished !

----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |100|
| Water quanity      |395|
| Steam state        |0|
------------------------
```

8. Якщо пару немає то й не можна зупинити його подачу:

```
5 - The end
>> 5
* (f)inish or (c)hange parameters?
>> f
! stop ironing !
------------------------
| Ironing process    |0|
| Temperature        |90|
| Water quanity      |90|
| Steam state        |0|
------------------------

* what's next?
1 - Take Iron
2 - Switch OFF
>> 2
# IRON OFF #
```

9. Завершення прасування та вимкнення:

```
----------Iron----------
------------------------
| Ironing process    |1|
| Temperature        |100|
| Water quanity      |395|
| Steam state        |0|
------------------------
0 - Start ironing
1 - Set temperature
2 - Add water
3 - Use steam
4 - Stop use steam
5 - The end
>> 4
! no steam used !
```