Final Project

Internet Shop

The main class E-Shop contains data about Seller and Customers accounts and list of all Products. E-Shop has a menu method where user choses options Login to account or create account. After logging in user sees menu but they differ depending on whether it is Seller or Customer.

class *E\_Shop*

{

private:

    map<*string*, *SellerAccount*> seller\_accounts;     // Map of seller accounts with their username as key

    map<*string*, *CustomerAccount*> customer\_accounts; // Map of customer accounts with their username as key

    map<*string*, *Product* \*> name\_products;           // Map of products with their name as key

    map<int, *Product* \*> id\_products;                // Map of products with their id as key

    void login\_seller();                            // Login seller

    void login\_customer();                          // Login customer

    void create\_seller\_account();                   // Create a seller account

    void create\_customer\_account();                 // Create a new customer account

    void print\_products();                          // Prints all products

public:

    E\_Shop();

    void enter\_menu(); // Main menu

    friend class *SellerAccount*;

    friend class *CustomerAccount*;

};

Class Seller contains data about seller and products of this seller. Seller can add new product, remove their product, update their product, view only their products and view all products.

class *SellerAccount*

{

private:

*string* password;                       // Password of the seller account

*string* name;                           // Name of the seller account

*string* email;                          // Email of the seller account

*string* phone;                          // Phone number of the seller account

    map<int, *Product*> products;            // Map of products with their id as key

*E\_Shop* \*e\_shop;                        // Pointer to the e-shop

    void enter\_menu();                     // Menu of the seller account

    void change\_password();                // Change password of the seller account

    void add\_product();                    // Add a new product

    void remove\_product();                 // Remove a product

    void update\_product();                 // Update a product

    void update\_product(*Product* \**product*); // Update a product

    void view\_your\_products();             // View all products of the seller account

    void view\_all\_products();              // View all products

public:

    SellerAccount() = default;

    SellerAccount(*string* *name*, *string* *email*, *string* *phone*, *string* *password*, *E\_Shop* \**e\_shop*);

    bool login(*string* *password*); // Login seller account

};

Class Customer contains data about customer and their cart. Cart contains link to Product and quantity. Customer can view products, view cart, add to cart, remove from cart and buy.

class *CustomerAccount*

{

private:

*string* password;                         // Password of the customer account

*string* name;                             // Name of the customer account

*string* email;                            // Email of the customer account

*string* phone;                            // Phone number of the customer account

    map<*Product* \*, int> cart;                // Map of quantity of products with link to product as key

*E\_Shop* \*e\_shop;                          // Pointer to the e-shop

    void enter\_menu();                       // Menu of the customer account

    void change\_password();                  // Change password of the customer account

    void view\_products();                    // View all products

    void view\_cart();                        // View cart

    void add\_to\_cart();                      // Add product to cart

    void add\_to\_cart(*Product* \**product*);      // Add product to cart

    void remove\_from\_cart();                 // Remove product from cart

    void remove\_from\_cart(*Product* \**product*); // Remove product from cart

    void buy();

public:

    CustomerAccount() = default;

    CustomerAccount(*string* *name*, *string* *email*, *string* *phone*, *string* *password*, *E\_Shop* \**e\_shop*);

    bool login(*string* *password*); // Login customer account

};

Both classes can change password of their account.

Class Product contains data about product, price, quantity, link to seller and id which is taken automatically.

class *Product*

{

private:

    inline static int id\_counter = 0;  // Counter for the id of the product

    inline static std::stack<int> ids; // Stack of free ids for the products which are deleted

    int id;                            // Id of the product

*string* name;                       // Name of the product

*string* description;                // Description of the product

    float price;                       // Price of the product

    int quantity;                      // Quantity of the product

*SellerAccount* \*seller;             // Seller of the product

    friend class *SellerAccount*;

    Product(*string* *name*, *string* *description*, float *price*, int *quantity*, int *id*, *SellerAccount* \**seller*);

    int new\_id(); // Get a new id for the product

public:

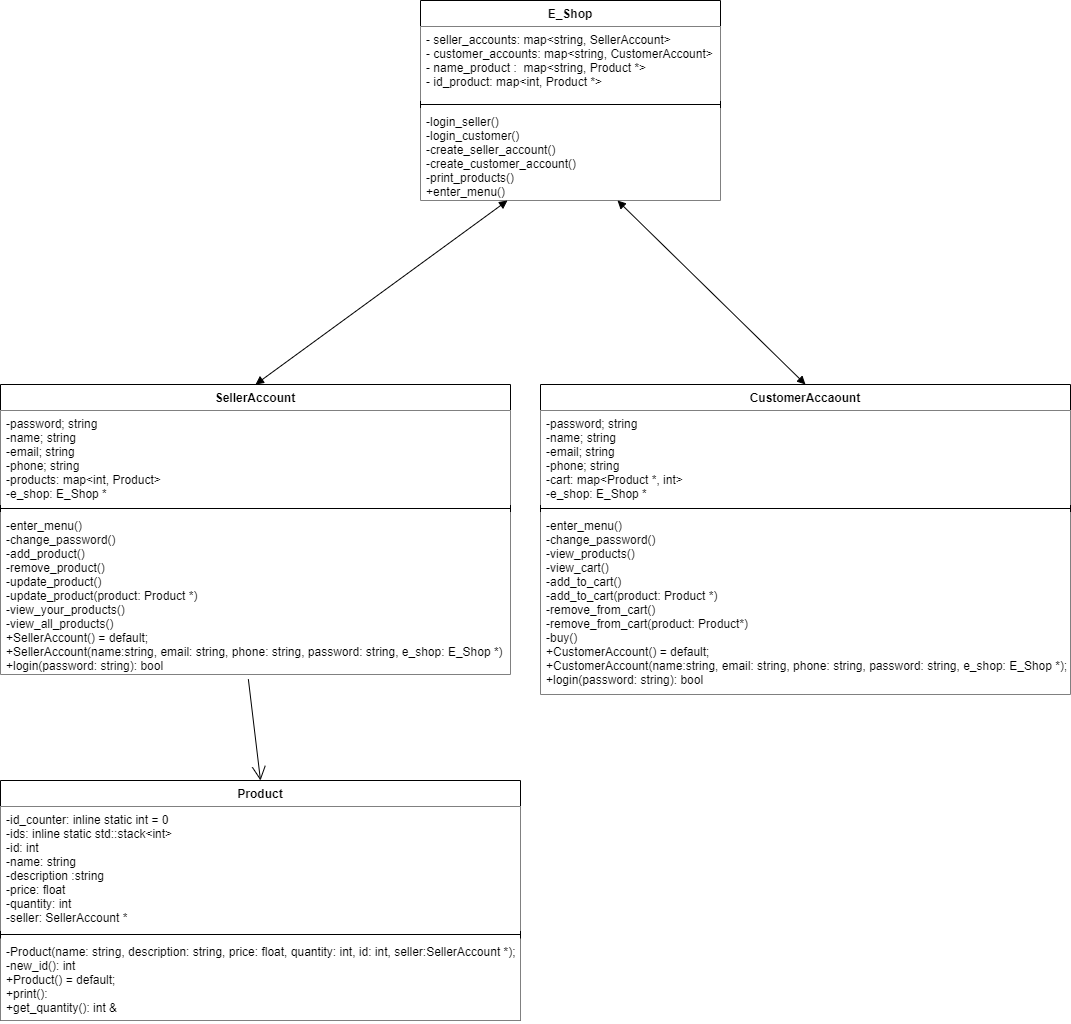
    Product() = default;

    void print();        // Prints the product

    int &get\_quantity(); // Get quantity of the product

};

**Memory map**



**Tests**

1. Try to enter incorrect username and password in functions: E\_Shop::login\_seller and E\_Shop::login\_customer.
2. Try to enter number less than 0 or greater than maximum in menus in functions: E\_Shop::enter\_menu, SelletAccount::enter\_menu, CustomerAccount::enter\_menu, SellerAccount::remove\_product, SellerAccount::update\_product, CustomerAccount::add\_to\_cart, CustomerAccount::remove\_from\_cart.
3. Try to enter incorrect quantity in functions CustomerAccount::add\_to\_cart, CustomerAccount::remove\_from\_cart.
4. Try to enter name of product which is same with another product in functions SellerAccount::add\_product and SellerAccount::update\_product.