ОПИСАНИЕ ОСОБЕННОСТЕЙ РЕАЛИЗАЦИИ КЛИЕНТСКОГО МОДУЛЯ

ИНФОРМАЦИОННОЙ СИСТЕМЫ АПТЕКИ

СОДЕРЖАНИЕ

1. СТРУКТУРА ПРОЕКТА

2. ОСОБЕННОСТИ РЕАЛИЗАЦИИ И ИСПОЛЬЗОВАНИЯ КЛАССОВ

3. ОПИСАНИЕ работы программы

|  |  |
| --- | --- |
| Logger | Логгер (класс, работающий с логом событий и действий пользователя) |
| Settings | Настройки и данные пользователя |

**СТРУКТУРА ПРОЕКТА**

* Логика приложения:
* Окна и GUI-элементы:

|  |  |
| --- | --- |
| Autorization | Окно авторизации пользователя; обращается к свойствам Settings |
| Logger | Окно с логом событий |
| Admin | Категория пользователя |
| User | Категория пользователя |
| Pharmacist | Категория пользователя |
| DefaultOperations | Общие действия, что доступны каждой категории пользователей |

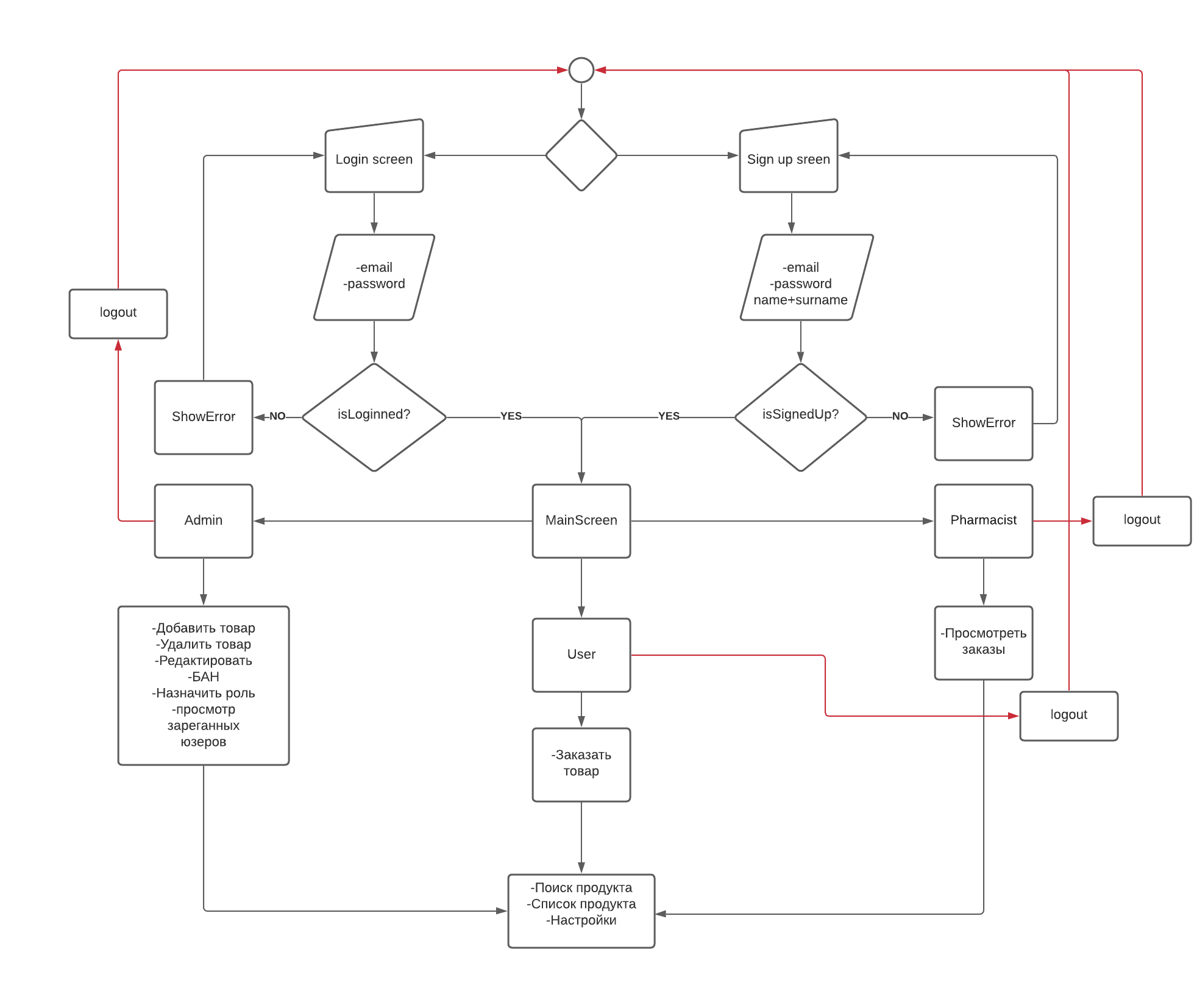


Рисунок - Схема классов

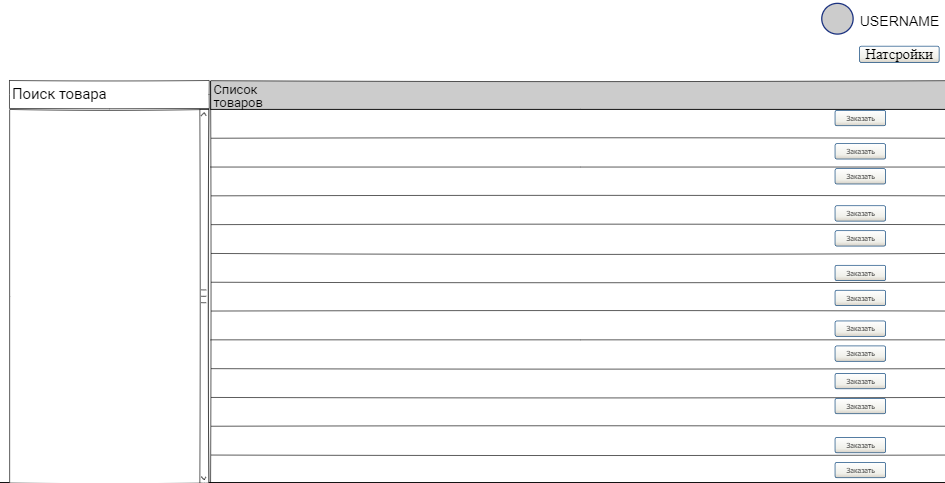


Рисунок - пример гр интерфейса

**ОСОБЕННОСТИ РЕАЛИЗАЦИИ И ИСПОЛЬЗОВАНИЯ КЛАССОВ**

* Класс Autorization

Класс для регистрации/входа в систему

class Autorization

{

private:

std::string email\_;

std::string password\_;

public:

void login();

void resetPassword();

bool checkedSignedUp();

};

* Класс DefaultOperations

Класс содержащий общедоступные методы и свойства для всех категорий пользователей

class DefaultOperations

{

public:

void seeListOfProducts();

void seachProduct();

void logout();

};

* Класс Logger

Класс для логирования всех событий и действий пользователя, для которых разработана приведенная ниже система кодов (от 0 до 4)

//вход/выход

#define User\_LogIn 0

#define User\_LogOut 1

// сканирование

#define Scan\_Unit 2

//Обновление

#define Update\_Guide 3

#define Update\_Tasks 4

class Logger

{

protected:

int logOffset;

void WriteTime();

void WriteString();

public:

void Close();

//пишущие функции логгера

void WriteLog(unsigned char nScanMode, CString cBarCode, short nQty);

void EndLog();

void OpenForWriting();

void StartLog(CString cFile);

//читающие функции логгера

std::string FindLog(std::string logTaskName);

void OpenForReading();

};

Функции логгера можно разделить на:

- пишущие (нужны для записи в лог различных событий и действий пользователя)

- читающие (нужны для просмотра лога).

* Класс Settings

Класс для настройки пользователем своего профиля

class Settings

{

///

public:

void changeUserName();

void changeTheme();

void setAvatar();

///

};

Функция changeUserName() – меняет имя пользователя в системе

Функция changeTheme() – меняет тему цветового оформления(из вариантов темная и светлая)

Функция setAvatar() – ставит аватарку на страницу польхователя

* Класс Admin

Класс для категории пользователя Админ

class Admin: public DefaultOperations

{

protected:

std::string firstname\_;

std::string lastname\_;

std::string category\_;

void editProduct();

void deleteProduct();

void addProduct();

void BAN();

void seachUsersOnSite();

void editRole();

};

Администратор – пользователь с самым большим количеством возможных действий.

Функция editProduct() – редактирует сведения о товаре в списке товаров ИСА

Функция deleteProduct() - удаляет товар из списка товаров ИСА

Функция addProduct() - добавляет товар в список товаров в ИСА

Функция BAN() – блокирует доступ к сайту выбранному администратором пользователю

Функция seachUsersOnSite() – открывает новое окно со всеми зарегистрированными на сайте пользователями всех категорий

Функция editRole() – меняет роль с обычного зарегистрированного пользователя на роль фармацевта и наоборот – с фармацевта на обычного юзера

* Класс User

Класс для всех пользователей, кто не работает в ИСА

class User:public DefaultOperations

{

protected:

std::string firstname\_;

std::string lastname\_;

std::string category\_;

void orderingProduct();

};

Класс Юзер имеет все функции классов DefaultOperations и Settings, а также функцию

orderingProduct() – функция заказа товара в ИСА

* Класс Pharmacist

Класс для категории пользователя Фармацевт

class Pharmacist: public DefaultOperations

{

protected:

std::string firstname\_;

std::string lastname\_;

std::string category\_;

protected:

void realizationOrderingProduct();

};

Класс Pharmacist имеет все функции классов DefaultOperations и Settings, а также функцию

realizationOrderingProduct () – функция обработки заказа товара в ИСА, сделанного Юзером.

**Описание работы программы**

При первом заходе в систему открывается окно Авторизации/Регистрации

После прохождения Авторизации/Регистрации пользователь заходит в свой личный кабинет, где, в зависимости от своей категории, может совершать различные действия.