Документация

# Описание программы - ГОСТ 19.402-79

1) Общие сведения:

Именование программы: Очередь пациентов

Для разработки приложения необходимо следующее стороннее программное обеспечение (ПО):

* Компилятор G++
* CMake
* Visual Studio
* Postgresql

Для функционирования программы используются следующие библиотеки:

* SFML
* Libpq-fe.h

Языки программирования на которых написана программа: с++ 20 standart

2) Функциональное назначение:

классы решаемых задач:

1. задача хранения пациентов в базе
2. задача передачи их пользователю
3. задача предоставления пользователю интерфейса работы с пациентом
4. задача записи обработанных пациентов в базу

Функциональное назначение:

1. Составление очереди из пациентов на сервере
2. Передача пациента пользователю
3. Обработка этого пациента пользователем
4. Запись обработанного пациента обратно в базу

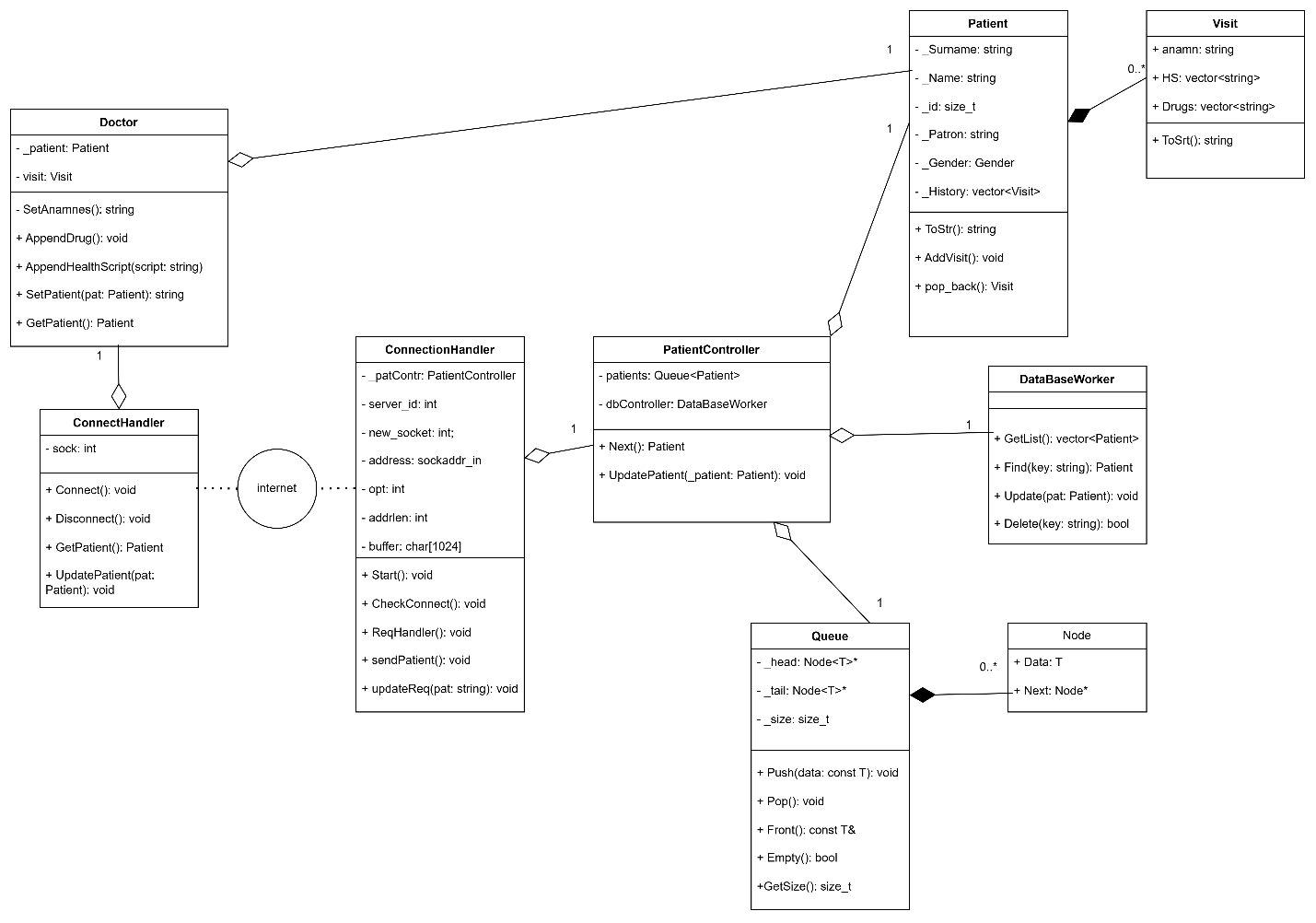
функциональные ограничения отсутствуют

3) Логическая структура:

Структура программы:

1. Server – блок, отвечающий за хранение и обработку внесённых изменений
2. Client – блок, отвечающий за обработку введённых пользователем данных и предоставлением ему ответа
3. GUI – блок, отвечающий за вывод информации на экран

структура программы с описанием функций составных частей и связи между ними:



Описание классов и методов:

Блок Server

Структура Node: шаблонная структура. Звено списка, хранящая данные шаблонного типа и указатель на следующий элемент.

Класс Queue: шаблонный класс очередь. Реализует структуру данных очередь на основе списка. В программе используется для хранения пациентов

Методы:

Push: добавляет элемент в конец очереди

Pop: удаляет элемент из начала очереди

Front: возвращает элемент из начала очереди

Empty: проверяет очередь на пустоту

GetSize: возвращает текущий размер очереди

Структура Visit: структура, хранящая предыдущие визиты пациента к врачу(его анамнез, выписанные лекарства и предписания)

Методы:

ToStr: конвертирует визит пациента в строку

Класс Patient: класс, хранящий информацию о пациенте

Методы:

ToStr: конвертирует пациента в строку

AddVisit: добавляет визит в историю болезни пациента

Pop\_back: возвращает последний визит пациента

Класс PatientController: Класс, отвечающий за управление очередью пациентов. Её заполнением и выдачей пациентов. Заполнение происходит в конструкторе при создании экземпляра, получая пациентов из dbController

Методы:

Next: выдаёт следующего пациента из очереди

UpdatePatient: принимает пациента и передают его в dbController для дальнейшего его обновления в бд

Класс ConnectionHandler : класс отвечающий за отправку данных с сервера клиенту через интернет.

Методы:

Start: запускает сервер

CheckConnect: Слушает входящие соединения

ReqHandler: Определяет тип запроса и вызывает соответствующие функцию

sendPatient: отправляет пациента клиенту

updateReq: отправляет пациента PatientController для дальнейшего обновления его в бд

Класс DataBaseWorker: класс, предназначенный для работы с базой данных(Извлечением и обновлением данных). Связывает бд с PatientController.

Методы:

GetList: получение всех пациентов из бд

Find: находит конкретного пациента в бд по ключу

Update: обновляет данные о пациенте в бд

Delete: Удаляет пациента из бд по ключу

Блок Client

Класс ConnectHandler: класс отвечающий за подключение клиента к серверу через интернет.

Методы:

Connect: подключается к серверу

Disconnect: отключается от клиента

GetPatient: Отправляет серверу запрос на получение следующего пациента

UpdatePatient: отправляет серверу пациента для дальнейшего обновления его в бд

Класс Doctor: класс, предназначенный для работы с самим пациентом. Созданием нового визита и его заполнением

Методы:

SetAnamnes: получение анамнеза пациента

AppendDrug: Добаление лекарств пациенту в текущий визит

AppendHealthScript: добавление предписаний врача в текущий визит

Блок GUI

Класс App: Основной класс в клиентской части, отвечающий за объединение всех имеющихся компонент.

Методы:

Start: запускает приложение

updateMain: обновление окна

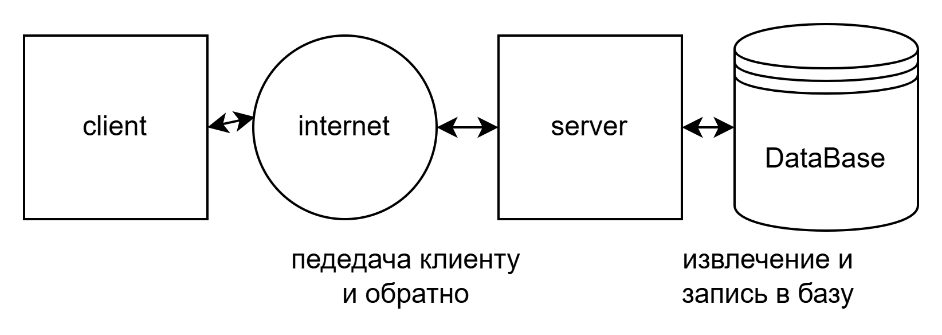
Класс Button: класс, реализующий произвольную кнопку

Класс Historyes: класс, реализующий пространство на экране выводящие прошлые визиты пациента и позволяющий посмотреть подробную информацию о каждом визите

Класс Input: класс, реализующий поле для ввода предписаний врача.

Класс InputDrugs: класс, реализующий выбор из предложенного списка лекарст

Класс MessageWindow: класс, реализующий создание диалогового окна и вывод туда произвольного сообщения(например ошибки)



Алгоритм Программы:

1. Начало работы алгоритма;

a. Запуск приложения при открытии файла Client.exe.

2. Работа программы

a. Программа подключается к серверу

б. Программа считывает действия пользователя

в. Если пользователь взаимодействовал с приложением будут предприняты соответствующие действие, иначе ничего

г. Отключение пользователя от сервера

д. Заново пункт а

3. Завершение работы алгоритма

а. При нажатии на крестик

4) используемые технические средства:

Для выполнения программы необходимо устройство, в качестве которого может выступать персональный компьютер на базе Window( не ниже 7 )или Linux.

процессор с частотой не ниже 1.8 ГГерц

количество ядер – минимум 1

графический процессор не слабее NVIDIA RTX 5060

оперативная память не ниже – 1гб

наличие WiFi

5) вызов и загрузка:

способ вызова программы с соответствующего носителя данных:

После скачивания программы, открыть файл Client.exe.

6) входные данные:

При запуске программы не требуется никаких входных данных.

При обработке каждого из пациентов требуются входные в виде:

Списка лекарств. Требуется(необязательно) выбрать из предоставленного списка лекарств.

Лечение: Требуется(необязательно) ввести рецепт лечение в специально поле ввода

7) выходные данные:

Выходными данными для пользователя являются данные о пациентах: их имя, фамилия, отчество, их история болезни включающая в себя их прошлые визиты в врачу. Пользователю они предоставляются в виде графического интерфейса.

Руководство оператора - ГОСТ 19.505-79

1) Назначение программы:

Программа предназначена для хранения, изменения и отображению пользователю пациентов.

2) Условия Выполнения программы:

Для выполнения программы необходимо устройство, в качестве которого может выступать персональный компьютер на базе Window( не ниже 7 )или Linux.

процессор с частотой не ниже 1.8 ГГерц

количество ядер – минимум 1

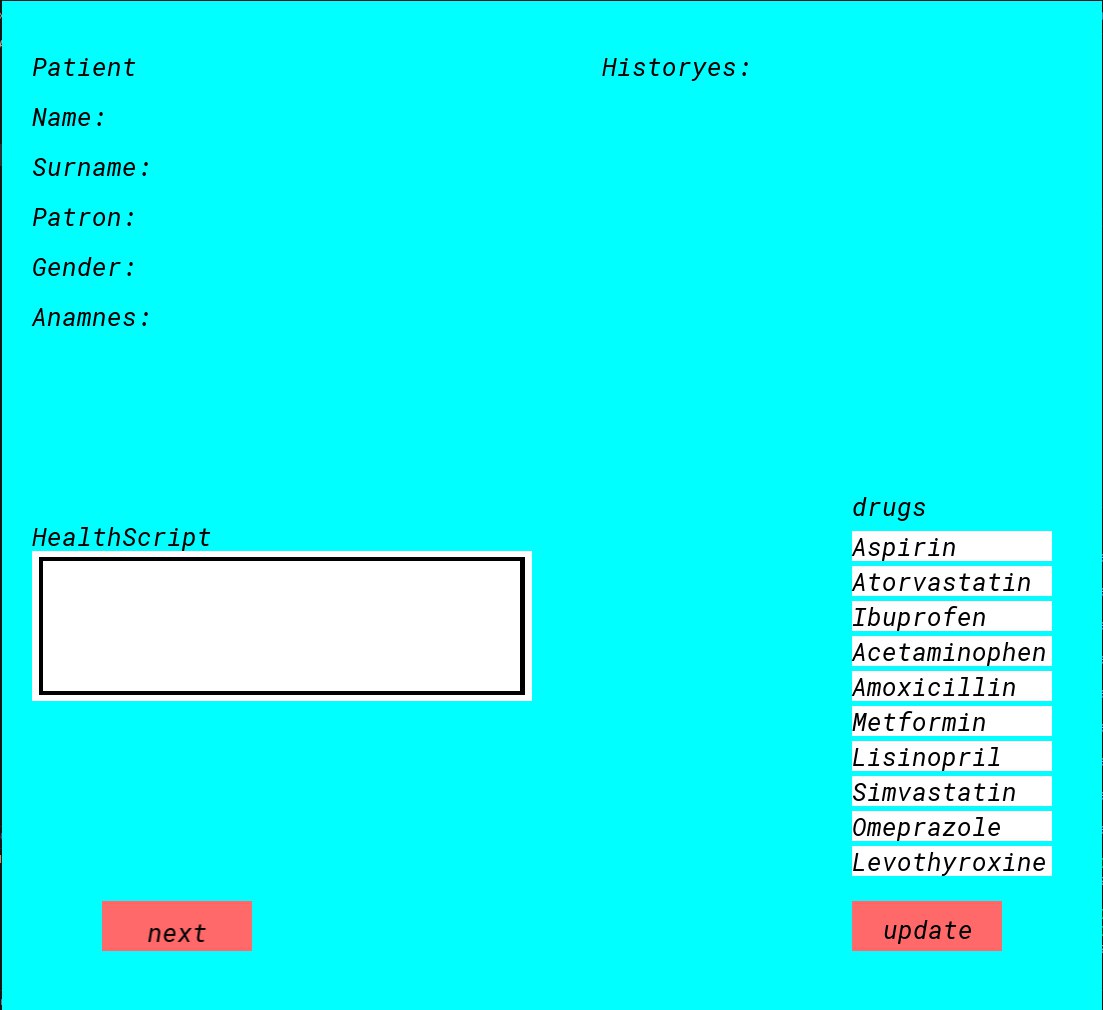
графический процессор не слабее NVIDIA RTX 5060

оперативная память не ниже – 1гб

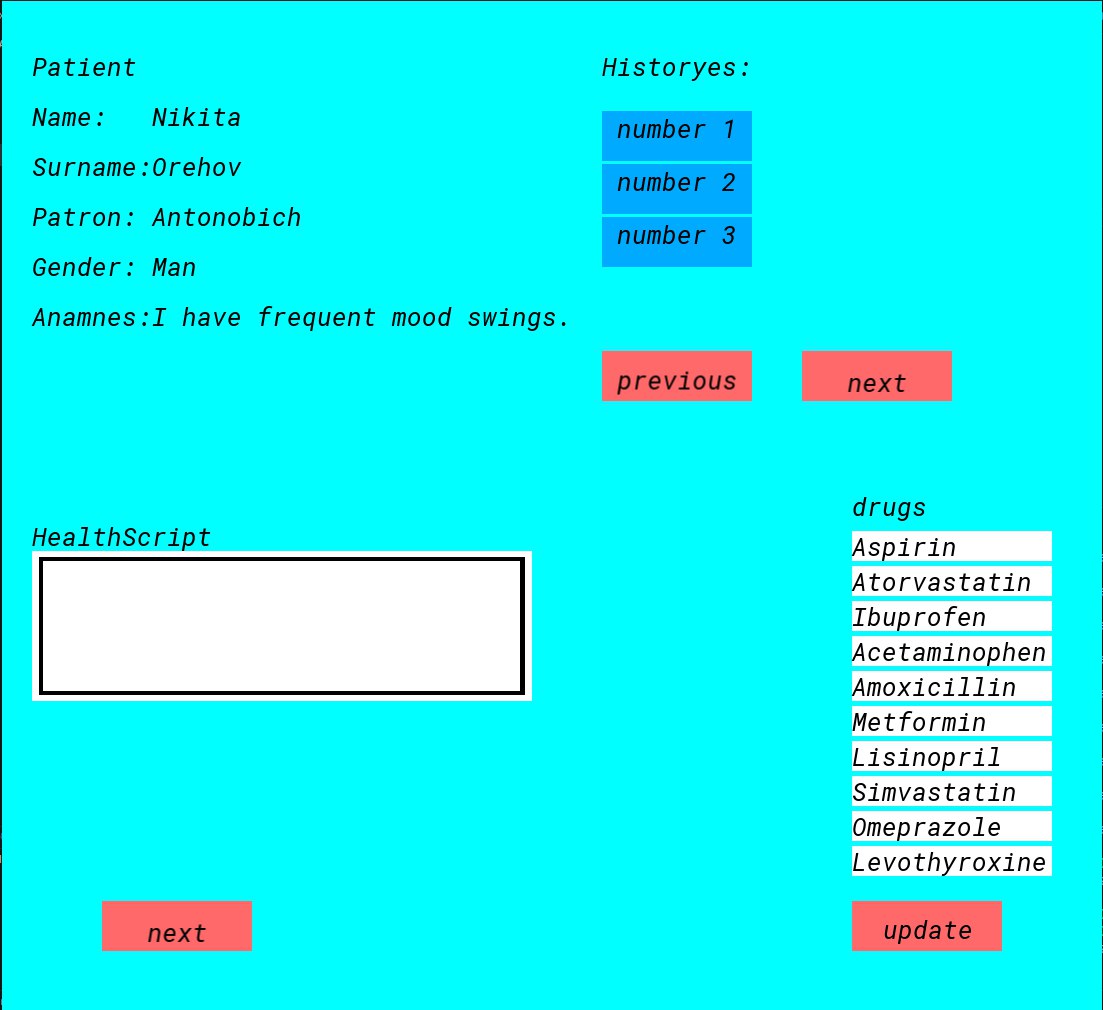
наличие WiFi

3) Выполнение программы:

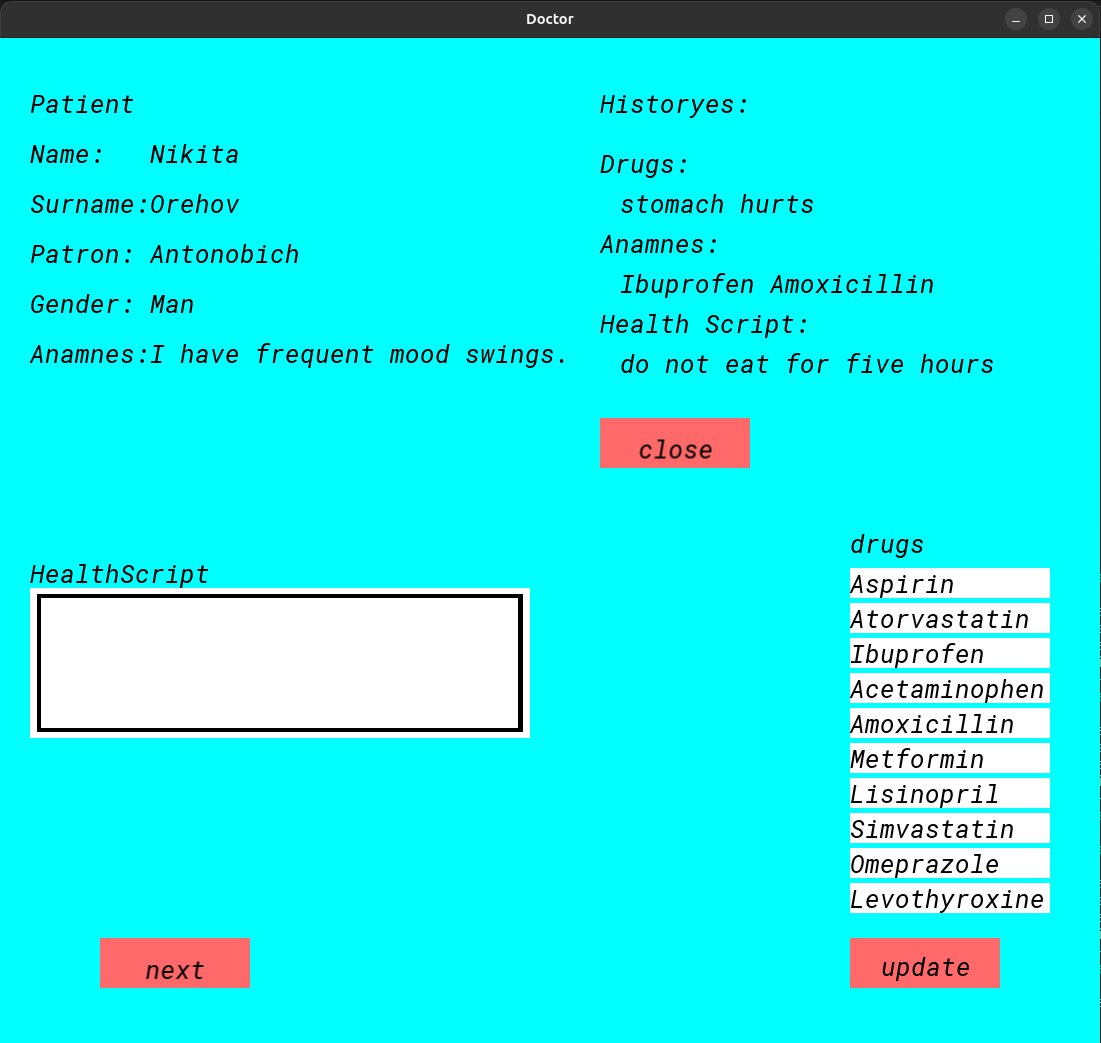
Запуск программы: Для начала работы скачайте программу в удобное для вас место на PC и откройте файл Client.exe. Откроется окно программы. Содержание кнопку Next и Update, прямоугольное поле для ввода лечение в левом нижнем углу и список лекарств в правой части экрана.



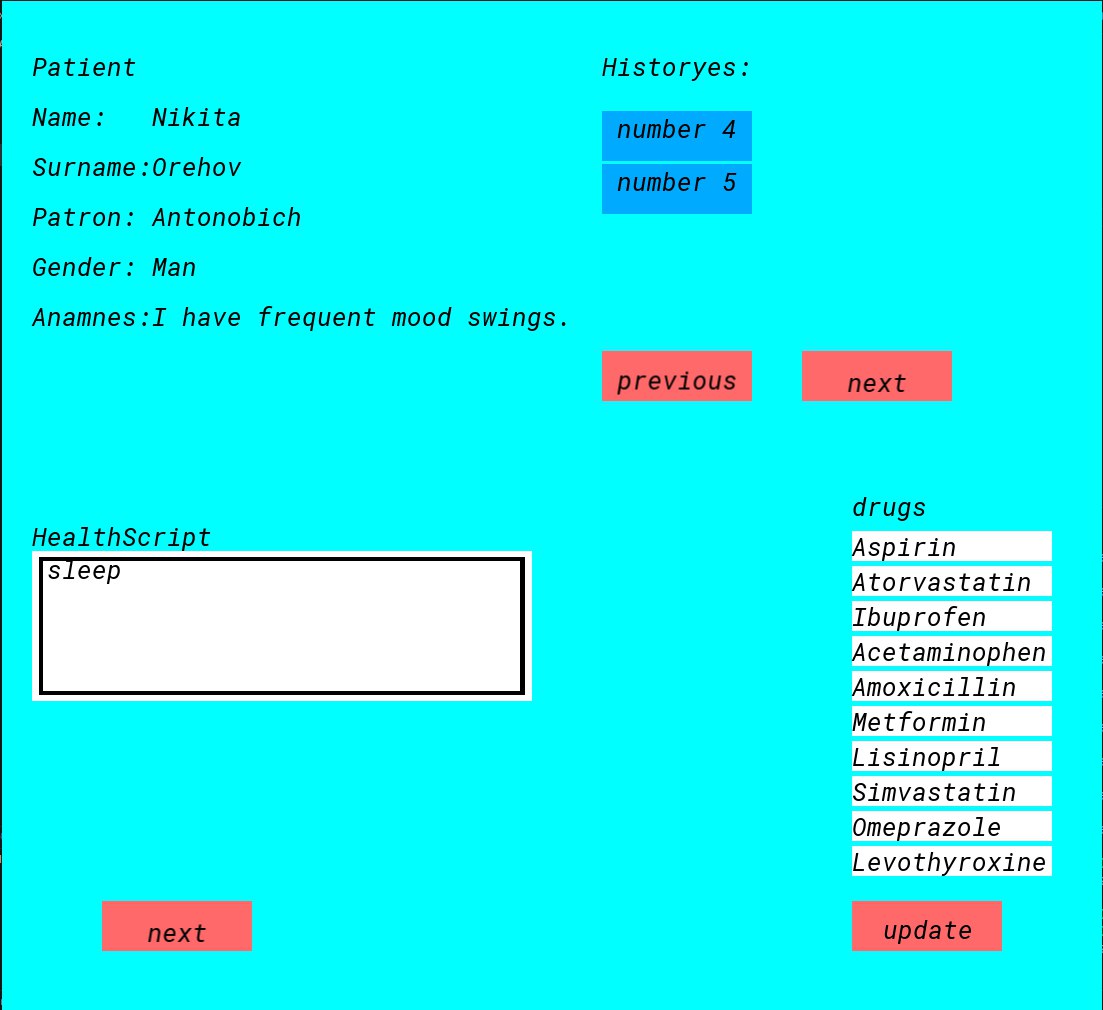
Выполнение программы: Для начала работы нажмите “Next”, чтобы получить данные о пациенте. В верхней части экрана появится личная информация о пациенте (Patient), а справа его история болезни(Historyes).



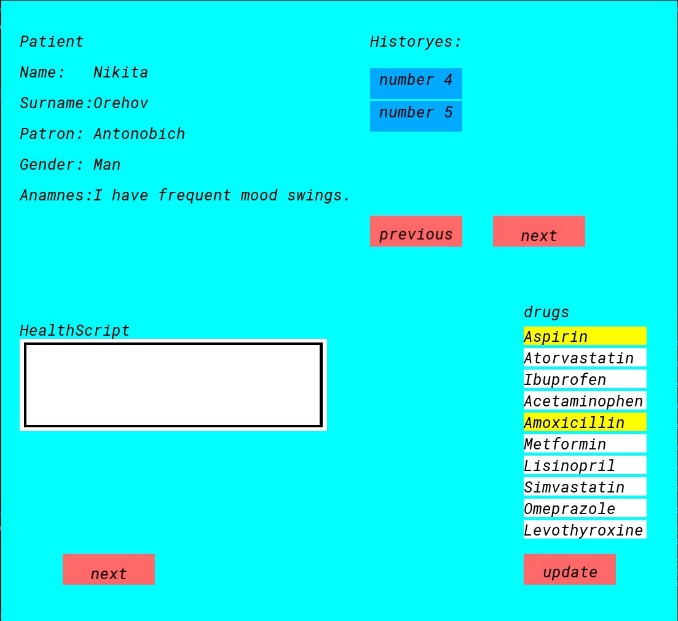
Нажав на любою из записей истории болезни (number1, number2, number3 и т. д.) вместо историй появится информация об этом визите пациента, а также внизу появится кнопка “Close”, для закрытия подробной информации о визите и возврате к историям болезни.



Введите в окно для ввода (поле можно оставить пустым) лечения пациенту,



а в правой части экрана выберите список лекарств (можно не выбирать), которые необходимо назначить пациенту. Выбрать можно нажав правой кнопкой мыши по соответсвующему лекарству (к примеру Aspirin и Amoxicillin). После этого кнопка соответствующего лекарства станет жёлтым.

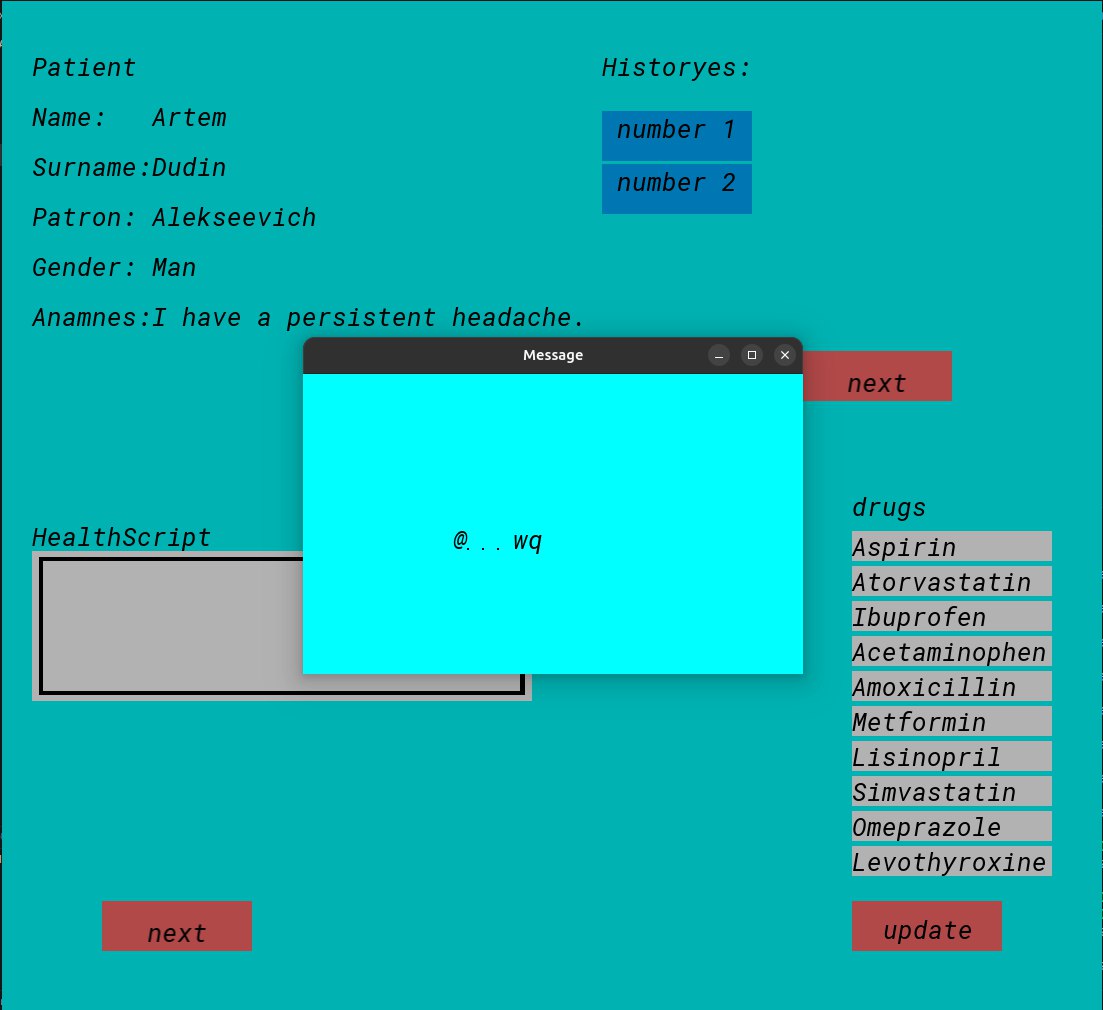


После всех этих операций нажмите кнопку “Update” для того чтобы сохранить введённые данные в базу. При нажатии на кнопку вся информация о пациенту пропадёт и экран вернётся в начальное состояние. Данные о пациенте отправятся на сервер, а в его истории болезни появится новый висит, с введёнными вами данными. После этого можете снова нажать кнопку “Next” для вывода на экран следующего пациента.

Завершение программы: Перед закрытием программы нажмите кнопку “Update” для сохранения введённых вами данных, если вы ещё этого не сделали. Перед закрытием экран должен находиться в начальном состоянии, чтобы никакие данные введённые вами не были утрачены. После этого нажмите на крестик в правом верхнем углу экрана для закрытия программы.

4) Сообщения оператору:

При нажатии кнопки “Next” может появится маленькое диалоговое окно с надписью “Queue is empty”.



Это означает, что очередь пациентов пуста и можно завершать программу. Нажмите на крестик у диалогового окна, а следом на крестик закрытия основной программы. Кнопку “Update” в этом случае перед закрытием нажимать необязательно. Все данные на момент истощения очереди уже сохранены.