



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ТЕОРЕТИЧЕСКАЯ ИНФОРМАТИКА И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ ПО КУРСУ
«Базы данных»

НА ТЕМУ:

Разработка базы данных лекарственных препаратов

Студент группы ИУ9-61Б

(Бакланова А. Д.)

(Подпись, дата)

Руководитель курсового проекта

(Домрачева А. Б.)

(Подпись, дата)

Москва, 2020 г.

СОДЕРЖАНИЕ

Введение.....	3
1. Описание предметной области.....	5
2. Структура базы данных.....	7
2.1 Используемые сущности и их взаимосвязь.....	7
2.2. Преобразование модели «сущность-связь» в реляционную.....	8
2.3. Связи сущностей.....	10
2.4 Используемые сущности.....	11
3. Реализация базы данных.....	16
4. Реализация веб-приложения.....	19
5. Тестирования.....	31
Заключение.....	32
Список использованной литературы.....	33

Введение

В наши дни каждому приложению необходимо хранить, накапливать, обрабатывать и представлять в удобном для пользователей виде разнообразную информацию.

К примеру, в медицинской сфере существует колоссальное количество информации, которую требуется хранить, обрабатывать, представлять в удобном виде и по-разному систематизировать. И чем оперативнее будет получена такая информация, тем лучше, ведь зачастую случаются экстренные ситуации, когда срочно нужна информация о пациенте, или сжатые и самые важные данные о каком-либо заболевании или лекарственном препарате.

В наши дни существует невероятное многообразие различных лекарственных препаратов, имеющих различные противопоказания. Также, много лекарств имеют весьма созвучные названия. Всё это в совокупности достаточно сильно усложняет подбор лекарств каждому пациенту, и поэтому появляется потребность в приложении, которое могло бы на основе введенных данных о пациенте, производить анализ и исключать лекарства, которые являются противопоказанными данному пациенту на основе особенностей его организма. Благодаря такому приложению внимание врача будет сконцентрировано только на самых главных данных о заболеваниях, лекарственных препаратах и никакая лишняя информация не отвлечет, не сбьет с толку. Следовательно, тема курсовой работы актуальна в наши дни.

Таким образом, целью данной курсовой работы является: создание базы данных для хранения и обработки информации о болезнях и лекарственных препаратах, способных вылечить болезни, учитывая индивидуальные особенности организма пациента, а также представление информации в удобном и доступном виде веб-приложения.

Постановку задачи можно сформулировать следующим образом:
требуется разработать веб-приложение, которое собирает информацию о пациенте (о его заболевании, которое нужно вылечить, о положении, сопутствующих заболеваниях, особенностях организма, возрасте и так далее) и выводит необходимую информацию о классификации болезни и лекарственных препаратах, которые необходимы пациенту. Для выполнения задачи планируется:

- 1) Собрать информацию о типах хранимой информации
- 2) Спроектировать реляционную базу данных;
- 3) Программно реализовать базу данных и веб-приложение;
- 4) Протестировать работоспособность.

1. Описание предметной области

Врачу, при назначении лекарств, нужно учитывать множество различных особенностей организма своего пациента. Тщательный анализ и подбор нужны для того, чтобы избежать назначения лекарства, противопоказанного пациенту.

Таким образом, врач может воспользоваться веб-приложением, чтобы получить необходимую ему информацию о лекарственных препаратах, которые безопасны для пациента, полную информацию о котором врач отметит при поиске. Для занесения такой информации не нужно тратить много времени, ведь в данном веб-приложении не нужно заводить карточку пациента, достаточно просто указать самое главное при подборе лекарства — пол пациента, возраст, положение, сопутствующие заболевания, особенности организма и болезнь, к которой требуется найти лечение. Также по данной информации врач быстро сможет узнать, нужно ли выписывать рецепт, какое точное международное название имеет лекарство (так как есть много созвучных названий, которые легко спутать), способ применения, а также будет дана подробная информация о болезни, по которой шел поиск, в частности: формальное название болезни, её иные названия, возможные причины болезни, симптомы, осложнения.

Также, классификации болезней, их названия и идентификаторы согласованы с МКБ-10. Десятый пересмотр Международной статистической классификации болезней и проблем, связанных со здоровьем — это документ, используемый как ведущая статистическая и классификационная основа в здравоохранении. Раз в десять лет пересматривается под руководством Всемирной Организации Здравоохранения (ВОЗ). Международная классификация болезней (МКБ) является нормативным документом, обеспечивающим единство методических подходов и международную сопоставимость материалов.

МКБ была создана для систематизированной регистрации, анализа, интерпретации и сравнения данных о смертности и заболеваемости, полученных в разных странах или регионах и в разное время. МКБ используется для преобразования словесной формулировки диагнозов болезней и других проблем, связанных со здоровьем, в буквенно-цифровые коды, которые обеспечивают удобство хранения, извлечения и анализа данных. [1]

Существует множество баз данных лекарственных препаратов, но ни одна из них не привязана ни к МКБ, ни к заболеваниям. На платформе «Яндекс.Здоровье» можно найти интересующее пользователя лекарство либо поиском по названию, либо по нескольким указанным на странице категориям. Но полноценного поиска лекарственных препаратов по болезням не имеется. Также не имеется привязки к терминологии МКБ и предназначена в первую очередь для обычных пользователей, но не для врачей.

2. Структура базы данных

2.1 Используемые сущности и их взаимосвязь

После изучения предметной области можно выделить сущности для базы данных.

Сперва нужна сущность, отвечающая за информацию о классификации болезней, которая будет хранить данные о формальных названиях групп заболеваний и описании данных групп в соответствии с МКБ-10.

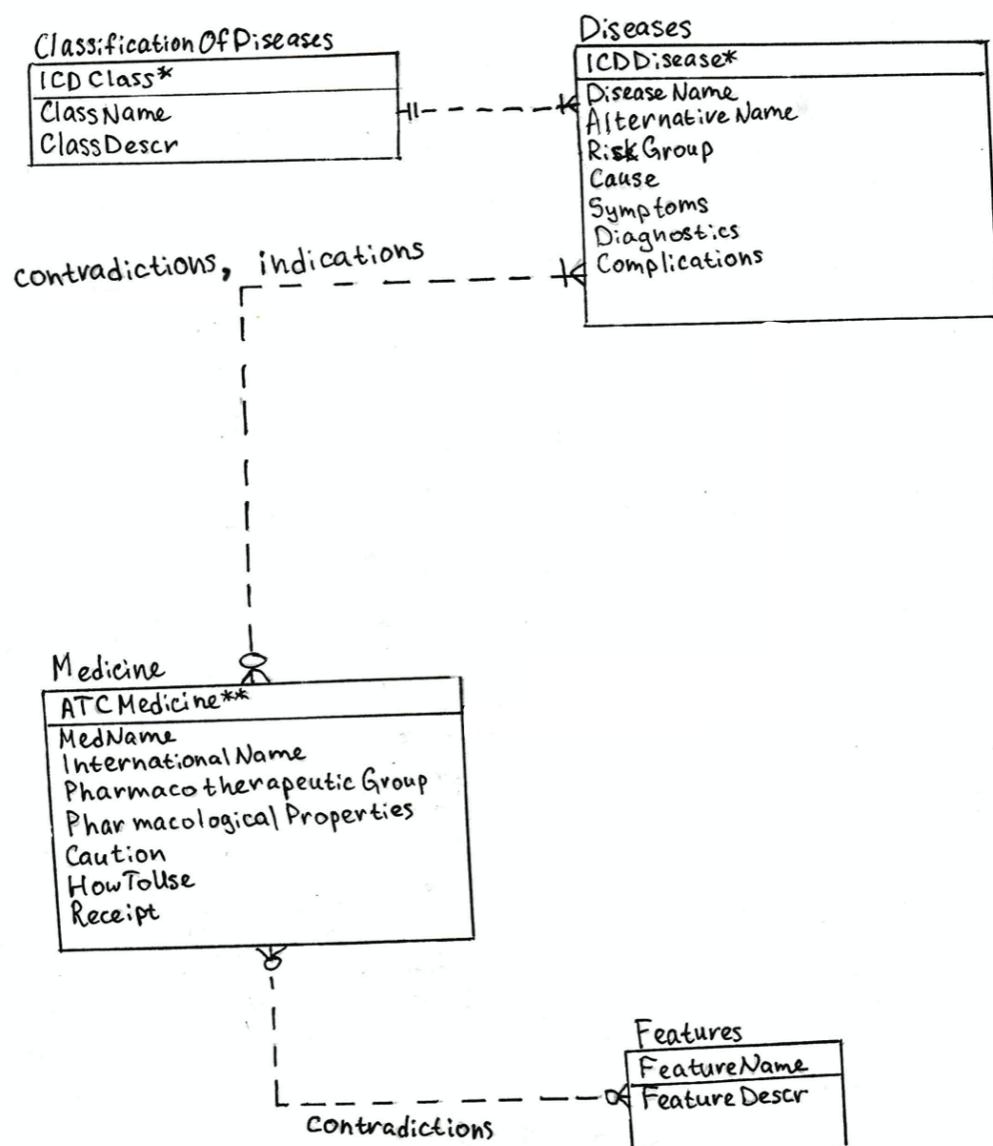
Также нужна сущность заболеваний, где будет вся необходимая информация о конкретной болезни в соответствии с МКБ-10 для удобства пользователя. Эта сущность будет связана с классификациями болезней, то есть каждое заболевание принадлежит какому-либо специальному классу.

Далее сущность с информацией о лекарственных препаратах. Данные, принадлежащие этой сущности, соответствуют официальной инструкции лекарств. Эта сущность связана заболеваниями, так как у каждого лекарства есть показания, а также могут иметься противопоказания в виде иных заболеваний.

Последней сущностью стали особенности человеческого организма, где будет информация с описанием возможных особенностей организма. Данная сущность будет связана с лекарственными препаратами, так как у лекарственных препаратов имеются некоторые противопоказания, не относящиеся к заболеваниям.

2.2 Преобразование модели «сущность-связь» в реляционную

Итого, на основе требований и ограничений к разрабатываемой базе данных, была создана логическая модель данных — модель сущность-связь (Рисунок 1), по которой нужно будет разработать реляционную базу данных.



* ICD – International Statistical Classification of Diseases and Related Health Problems (МКБ)

** ATC – Anatomical Therapeutic Chemical (ATX)

Рисунок 1 — модуль сущность-связь

Далее нужно преобразовать модель «сущность-связь» в реляционную модель. После двух итераций проектирования получилась реляционную модель, которую будем реализовывать с помощью MSSQL (Рисунок 2). Из-за связей **многие-ко-многим** появляются две вспомогательные сущности — DiseaseMedicine_INT и MedicineFeatures_INT.

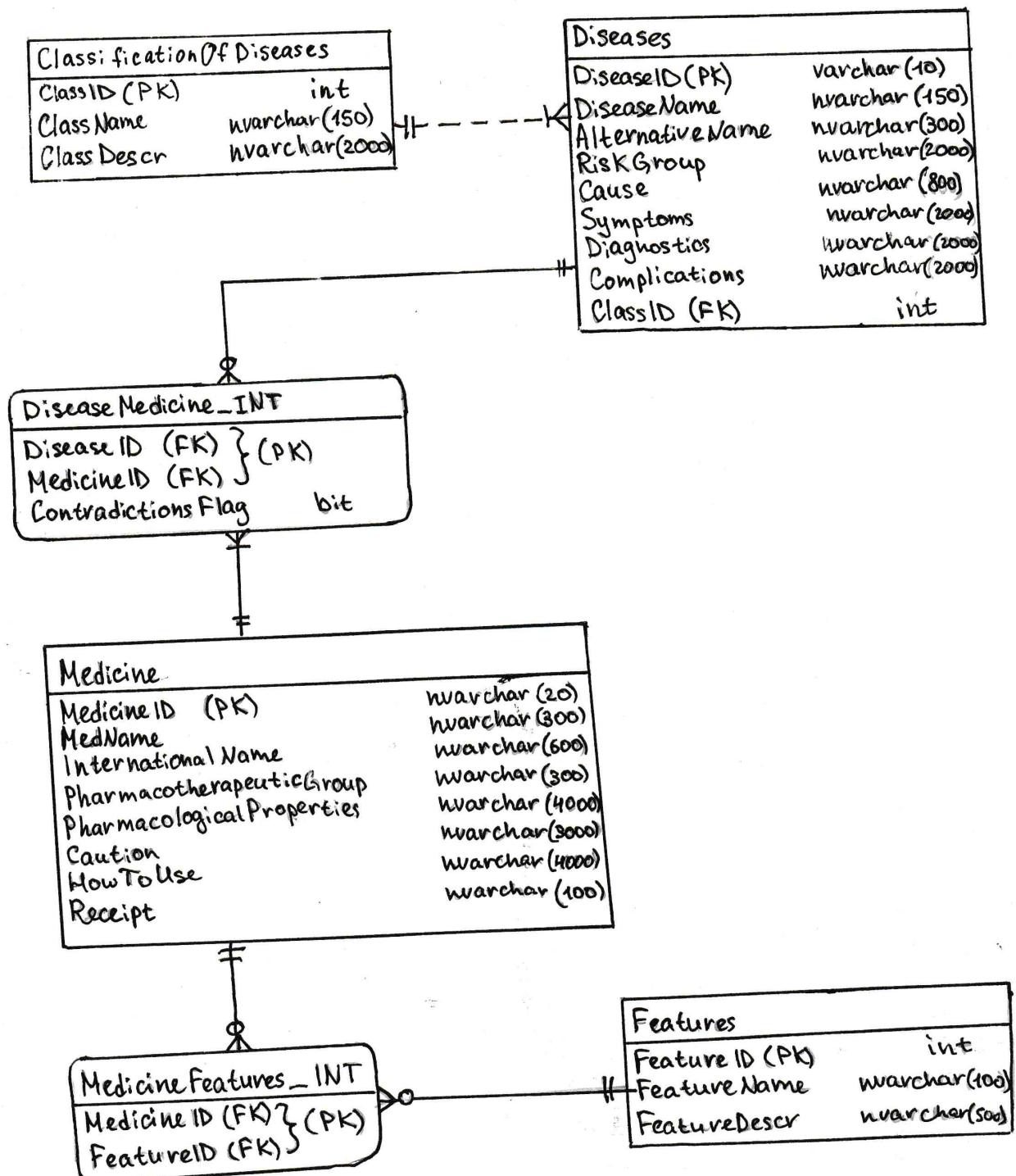


Рисунок 2 — реляционная модель

2.3 Связи сущностей

Связи сущностей (Таблица 1):

- 1) Классификация болезней и Заболевания. Болезнь обязана принадлежать только одной группе болезней, а в группе болезней может быть много заболеваний;
- 2) Заболевание и Показания. У заболевания может не быть лекарства, а также несколько лекарств могут лечить одну болезнь.
- 3) Лекарства могут иметь много показаний, но обязаны иметь хоть одно показание, иначе нет смысла хранить лекарство в базе данных.
- 4) Лекарства могут быть противопоказаны при многих различных особенностях организма, а также могут не иметь противопоказаний.
- 5) Особенности организма могут быть противопоказаны многим лекарствам, а могут не являться противопоказанием вовсе.

Relationships		Cardinality
Parent	Child	
ClassificationOfDiseases	Diseases	1:N
Diseases	DiseaseMedicine_INT	0:N
Medicine	DiseaseMedicine_INT	1:N
Medicine	MedicineFeatures_INT	0:N
Features	MedicineFeatures_INT	0:N

Таблица 1 — Отношения между сущностями

2.4 Используемые сущности

После анализа предметной области можно выделить сущности базы данных, связи между ними, а также их атрибуты.

Итого, база данных имеет 6 сущностей:

- 1) Классификация болезней (в соответствии с МКБ-10) (**ClassificationOfDiseases**);
- 2) Заболевания (**Diseases**);
- 3) Показания (**DiseaseMedicine_INT**)
- 4) Лекарства (**Medicine**)
- 5) Противопоказания у лекарств, относящиеся к особенностям организма (**MedicineFeatures_INT**)
- 6) Особенности организма (**Features**)

Сущность **ClassificationOfDiseases** (Таблица 2) содержит в себе информацию о группах болезней, содержит следующие атрибуты:

- Идентификатор (**ClassID**);
- Название класса (**ClassName**) в соответствии с МКБ-10;
- Описание класса (**ClassDescr**) в соответствии с МКБ-10;

Column Name	Type	Key	Null Status
ClassID	Int	Primary Key	NOT NULL
ClassName	nvarchar(150)	No	NOT NULL
ClassDescr	nvarchar(2000)	No	NOT NULL

Таблица 2 — Сущность *ClassificationOfDiseases*

Сущность **Diseases** (Таблица 3) содержит в себе информацию о заболеваниях, содержит следующие атрибуты:

- Идентификатор (DiseaseID) в соответствии с МКБ-10;
- Название заболевания (DiseaseName) в соответствии с МКБ-10;
- Альтернативные названия (AlternativeNames);
- Группа риска (RiskGroup) — категории людей, у которых риск заболеть определенный заболеванием выше;
- Причины заболевания (Cause) — информация о возможных причинах возникновения болезни;
- Симптомы (Symptoms) — самые распространенные и явные симптомы, по которым определяется данное заболевание;
- Диагностика заболевания (Diagnostics) — способы диагностирования врачом данного заболевания, включая описание проведения первоначального осмотра;
- Осложнения (Complications) — информация о возможных осложнениях данного заболевания;
- Идентификатор класса заболевания (ClassID)

Column Name	Type	Key	Null Status
DiseaseID	varchar(10)	Primary Key	NOT NULL
DiseaseName	nvarchar(150)	No	NOT NULL
AlternativeNames	nvarchar(300)	No	
RiskGroup	nvarchar(2000)	No	NOT NULL
Cause	nvarchar(800)	No	NOT NULL
Symptoms	nvarchar(2000)	No	NOT NULL
Diagnostics	nvarchar(2000)	No	NOT NULL
Complications	nvarchar(2000)	No	NOT NULL
ClassID	int	Foreign key	NOT NULL

Таблица 3 — Сущность Diseases

Сущность **DiseaseMedicine_INT** (Таблица 4) содержит в себе информацию о том, какие болезни лечатся какими лекарственными препаратами и какие лекарственные препараты противопоказаны при каких заболеваниях содержит 3 атрибута:

- Идентификатор заболевания (DiseaseID)
- Идентификатор лекарства (MedicineID)

Column Name	Type	Key	Null Status
DiseaseID	varchar(10)	Primary Key, Foreign Key	NOT NULL
MedicineID	nvarchar(20)	Primary Key, Foreign Key	NOT NULL
ContradictionsFlag	bit		NOT NULL

Таблица 4 — Сущность DiseaseMedicine_indications

Сущность **Medicine** (Таблица 5) содержит в себе информацию о лекарственных препаратах, включает в себя следующие атрибуты:

- Идентификатор (MedicineID) в соответствии с регистрационным номером;
- Торговое название препарата (MedName);
- Международное непатентованное название препарата (InternationalName);
- Фармакотерапевтическая группа (PharmacotherapeuticGroup);
- Фармакологические свойства (PharmacologicalProperties);
- С осторожностью (Caution) — информация, требующая внимания врача при выписывании данного лекарства;
- Способ применения (HowToUse);
- Условия отпуска из аптек (Receipt);

Column Name	Type	Key	Null Status
MedName	nvarchar(300)	No	NOT NULL
InternationalName	nvarchar(600)	No	NOT NULL
PharmacotherapeuticGroup	nvarchar(300)	No	NOT NULL
PharmacologicalProperties	nvarchar(4000)	No	NOT NULL
Caution	nvarchar(3000)	No	NOT NULL
HowToUse	nvarchar(4000)	No	NOT NULL
Receipt	nvarchar(100)	No	NOT NULL

Таблица 5 — Сущность Medicine

Сущность MedicineFeatures_INT (Таблица 6) содержит в себе информацию о возможных особенностях организма человека, при которых лекарство противопоказано. Содержит в себе 2 атрибута:

- Идентификатор лекарства (MedicineID);
- Идентификатор особенности организма (FeatureID);

Column Name	Type	Key	Null Status
DiseaseID	varchar(10)	Primary Key, Foreign Key	NOT NULL
MedicineID	nvarchar(20)	Primary Key, Foreign Key	NOT NULL

Таблица 6 — Сущность MedicineFeatures_INT

Сущность Features (Таблица 7) содержит в себе информацию об разнообразных особенностях организма человека, которые могут быть противопоказаниями к приему лекарств. Содержит в себе следующие атрибуты:

- Идентификатор особенности организма (FeatureID);
- Название особенности (FeatureName);
- Описание особенности (FeatureDescr);

Column Name	Type	Key	Null Status
FeatureID	Int	Primary Key	NOT NULL
FeatureName	nvarchar(100)	No	NOT NULL
FeatureDescr	nvarchar(500)	No	NOT NULL

Таблица 7 — Сущность *Features*

3. Реализация базы данных

Для создания базы данных была выбрана система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft — Microsoft SQL Server. Используется для работы с базами данных различных размеров: от персональных до крупных баз данных масштаба предприятия [2].

В SQL Server применяется одна из разновидностей языка SQL — Transact-SQL (T-SQL). В зависимости от задачи, которую будет выполнять команда T-SQL, язык может принадлежать одному из следующих типов:

- 1) Data Definition Language — язык определения данных. Это команды, которые создают/изменяют базу данных, таблицы и т.д.
- 2) Data Manipulation Language — язык манипуляции данными. Это команды, которые отвечают за выбор, обновление, добавление, удаление данных и т.д.
- 3) Data Control Language — язык управления доступа к данным. Это команды, которые управляют правами по доступу к данным.

Microsoft Azure SQL Databases — это созданная для облака интеллектуальная масштабируемая служба реляционных баз данных. Она обновляется автоматически, и пользователи всегда имеют в своем распоряжении самую актуальную версию. Microsoft Azure SQL Databases основан на Microsoft SQL Server. Преимущества этой службы в том, что это полностью управляемая база данных, в которой установка обновлений, подготовка к работе и резервное копирование выполняются автоматически. Azure автоматически обрабатывает исправления, резервное копирование, репликацию, выявление сбоев, потенциальные сбои основного оборудования, программного обеспечения или сети, исправления ошибок при развертывании, отработку отказа, обновления баз данных и другие задачи обслуживания. [3]

```

Cause          nvarchar(800)      not null,      /* Причина заболевания */
Symptoms       nvarchar(2000)      not null,      /* Симптомы */
Diagnostics    nvarchar(2000)      not null,      /* Диагностика */
Complications  nvarchar(2000)      not null,      /* Осложнения */

ClassID int not NULL
constraint FK_DiseaseClass
foreign key references ClassificationOfDiseases (ClassID)
on update cascade
)

GO

/* Представление Класс болезни–Болезнь */
IF OBJECT_ID (N'DiseaseAndClassView') IS NOT NULL
DROP VIEW DiseaseAndClassView;
GO

CREATE VIEW DiseaseAndClassView AS
SELECT
ClassificationOfDiseases.ClassID, ClassificationOfDiseases.ClassName,
Diseases.DiseaseID, Diseases.DiseaseName

FROM ClassificationOfDiseases INNER JOIN Diseases
ON ClassificationOfDiseases.ClassID = Diseases.ClassID
GO

```

Для удобного отображения данных пользователю используются представления.

Например, сперва на странице веб-приложения должна отображаться краткая информация о введенном заболевании, а именно — ID класса болезни, класс болезни, ID болезни и название самой болезни (в соответствии с МКБ-10). Таким образом, было реализовано следующее представление (Листинг 1, 2):

Листинг 1 — сущности Группы болезней и Болезни

```

/*----- ГРУППЫ БОЛЕЗНЕЙ -----*/
IF OBJECT_ID (N'ClassificationOfDiseases') IS NOT NULL
DROP TABLE ClassificationOfDiseases;
GO

create table ClassificationOfDiseases (
    ClassID      int PRIMARY KEY          not null,
    ClassName    nvarchar(150)             not null,
    ClassDescr   nvarchar(2000)            not null,
)
GO

/*----- БОЛЕЗНИ -----*/
IF OBJECT_ID (N'Diseases') IS NOT NULL
DROP TABLE Diseases;
GO

create table Diseases (
    DiseaseID    varchar(10) PRIMARY KEY    not null,
    DiseaseName  nvarchar(150)                not null, /* Название болезни */
    AlternativeNames nvarchar(300)            not null, /* Иные названия */
    RiskGroup     nvarchar(2000)               not null, /* Риск заболеть выше у след лиц */
)

```

Листинг 2 — сущность Болезнь и представление связи класса болезней с болезнями

4. Реализация веб-приложения

Для разработки веб-приложения использовались HTML, CSS, JavaScript и Golang.

Сформулируем основные требования к разрабатываемому веб-приложению:

- простые и интуитивно понятные строки поиска для быстрого нахождения искомого заболевания;
- поля для быстрого и удобного ввода данных о пациенте;
- поля для указания сопутствующих заболеваний и прочих особенностей организма;
- удобный вывод нужной информации, включающий: данные о заболевании и его классификации по МКБ-10, данные о лекарственных препаратах, дающие врачу исчерпывающую информацию;

Серверная часть веб-приложения была реализована с помощью языка Golang.

Go — относительно молодой, но очень популярный язык программирования, у которого есть огромное количество плюсов. Прежде всего — простой синтаксис, поэтому приложения можно разрабатывать быстрее, чем на некоторых других языках. Также у Golang огромное количество вспомогательных библиотек [4].

Чтобы подключиться к базе данных, необходимо выбрать вид SQL, указать название базы данных, хост, порт, имя пользователя и пароль (Листинг 3)

```

var server = "127.0.0.1"
var port = 1433
var user = "SA"
var password = "PtktysqXfq231Poz"
var d = "MEDICINE_HELPER"

var db *sql.DB

func main() {
    //...
    connString := fmt.Sprintf("server=%s;user id=%s;password=%s;port=%d;database=%s",
        server, user, password, port, d)

    var err error
    db, err = sql.Open("sqlserver", connString)
    //...
    defer db.Close()
    //...
}

```

Листинг 3 — Подключение к базе данных

HTML — язык разметки гипертекста, позволяющий пользователю создавать и структурировать разделы , параграфы, заголовки, ссылки и блоги для веб-страниц и приложений. То есть именно с помощью HTML создается структура веб-страниц [5].

Для создания веб-страницы и некоторых её форм был написан следующий код (Листинг 4 и Листинг 5):

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Medicine helper</title>
        <link rel="stylesheet" type="text/css" href="/static/style.css">
        <script src="/static/script.js"></script>
    </head>

    <body>
        <div class="header">
            <h1>Medicine Helper</h1>
            <p>Введите данные о пациенте чтобы подобрать нужное лекарство</p>
        </div>

```

Листинг 4 — пример работы с HTML

```

<form method="POST">
    <div class="box">
        <div class="parent">
            <div class="child_div">
                <p>Укажите заболевание</p>
                <input type="text" name="disease" list="l1"/><br><br>
                <datalist id="l1">
                    <option value="Инсульт, не уточненный как кровоизлияние или инфаркт"></option>
                    <!-- ... -->
                </datalist>
                <input type="submit" value="Send" />
            </div>

            <div class="child_div">
                <p align="center">Возраст:</p>
                <hr width="95%" class="hr_age" size="2" color="#596868" align="center" />
                <label>Детский возраст до 12 лет</label><br>
                <input type="radio" name="age" value="Детский возраст до 12 лет"/><br><br>
                <!-- ... -->
            </div>
        </div>
    </div>

```

Листинг 5 — пример работы с HTML, создание поля для выбора заболевания и поля для выбора возраста пациента

CSS — каскадные таблицы стилей (Cascading Style Sheets) — язык, отвечающий за визуальное представление набора информации о структуре страницы пользователю. CSS нужен для того, чтобы отделять то, что задает внешний вид страницы, от ее содержания. В HTML достаточно прописать класс, не перечисляя все стили каждый раз. Благодаря CSS разработка становится быстрее и приятнее так как однажды созданный стиль оформления можно применять ко многим страницам, не прописывая одно и то же по несколько раз.

Таким образом, сначала нужно указать путь файла со стилями в HTML:

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Medicine helper</title>
        <link rel="stylesheet" type="text/css" href="/static/style.css">

```

Листинг 6 — подключение файла со стилями

Далее были указаны следующие стили для таблиц, ячеек и строк (Листинг 7):

```
table {  
    border-spacing: 0px 0px;  
    width: 100%;  
}  
  
th{  
    background: #AFCDE7;  
    color: white;  
    padding: 10px 20px;  
}  
  
th, td {  
    border: 2px solid #4682B4;  
    padding: 15px;  
}  
  
td {  
    background: #D8E6F3;  
}
```

Листинг 7 — пример стиля таблицы

JavaScript — язык программирования, который применяется к HTML документу и может обеспечить интерактивность на веб-сайтах. Программы на этом языке называются скриптами и могут быть встроены в HTML и выполняться автоматически при загрузке веб-страницы.

Язык JavaScript использовался для написания некоторых скриптов, например, для блокировки кнопок поля «Положение» при выборе мужского пола (Листинг 8).

```
function disablePregnantButtons() {  
    document.getElementById("notPregnant").disabled = true;  
    document.getElementById("pregnant").disabled = true;  
    document.getElementById("feeding").disabled = true;  
}  
  
function availablePregnantButtons() {  
    document.getElementById("notPregnant").disabled = false;  
    document.getElementById("pregnant").disabled = false;  
    document.getElementById("feeding").disabled = false;  
}
```

Листинг 8 — пример кода, блокирующего кнопки

Для начала работы с веб-приложением, необходимо перейти на <http://localhost:8182>. (Рисунок 3) Далее, нужно в специальных полях указать данные о пациенте: сперва указать заболевание, от которого требуется найти лекарственный препарат. При нажатии на данную поисковую строку, появится всплывающий список (Рисунок 4), в котором можно выбрать нужный вариант или же начать вбивать название заболевания, и список сократится до нужного запроса (Рисунок 5).

The screenshot shows the 'Medicine Helper' application interface. At the top, there is a header bar with the title 'Medicine Helper' and a sub-instruction 'Введите данные о пациенте чтобы проверить корректность схемы лечения'. Below the header, there are five input fields arranged horizontally:

- Укажите заболевание:** A text input field with a placeholder and a 'Send' button below it.
- Возраст:** A dropdown menu showing age ranges: 'Детский возраст до 12 лет', 'Подростковый возраст от 12 лет до 18 лет', 'Взрослый возраст от 19 лет до 60 лет', and 'Старческий возраст от 61 лет'.
- Пол:** A dropdown menu showing gender: 'Мужчина' and 'Женщина'.
- Положение:** A dropdown menu showing pregnancy status: 'Не беременна', 'Беременна', and 'Кормящая'.
- Укажите сопутствующие заболевания:** A dropdown menu listing coexisting conditions: 'Сахарный диабет', 'Инсульт, не уточненный как кровоизлияние или инфаркт', and 'Туберкулез органов дыхания, подтвержденный бактериологически и гистологически'.

Рисунок 3 — внешний вид сайта

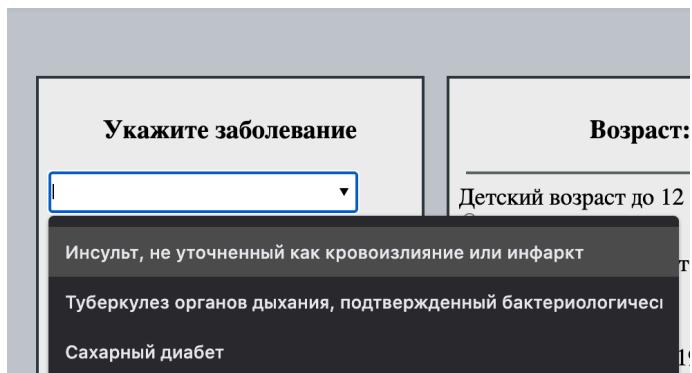


Рисунок 4 — всплывающий список

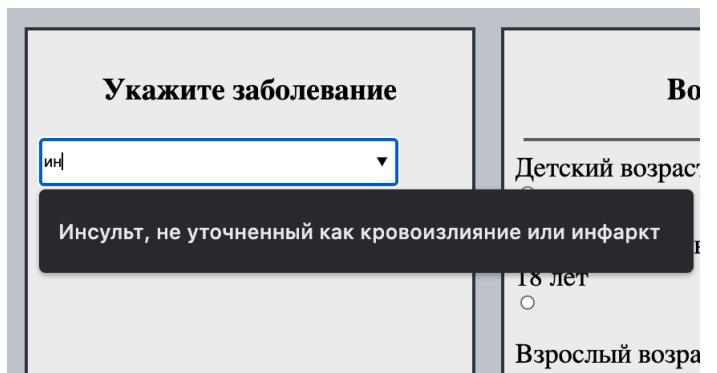


Рисунок 5 — ввод сокращает варианты

Далее требуется указать пол пациента и, в зависимости от данного выбора, будет активно или же не активно поле о положении пациента (Рисунок 6).

Возраст:	Пол:	Положение:	Укажите сопутствующие заболевания
Детский возраст до 12 лет <input type="radio"/> Подростковый возраст от 12 лет до 18 лет <input type="radio"/> Взрослый возраст от 19 лет до 60 лет <input checked="" type="radio"/> Старческий возраст от 61 лет <input type="radio"/>	Мужчина <input checked="" type="radio"/> Женщина <input type="radio"/>	Не беременна <input type="checkbox"/> Беременна <input type="checkbox"/> Кормящая <input type="checkbox"/>	Сахарный диабет <input type="checkbox"/> Инсульт, не уточненный как кровоизлияние или инфаркт <input type="checkbox"/> Туберкулез органов дыхания, подтвержденный бактериологически и гистологически <input type="checkbox"/>

Рисунок 6 — неактивное поле указания положения

Затем остаётся только указать сопутствующие заболевания пациента, особенности организма и предполагаемую схему лечения (Рисунок 7), ведь все эти факторы могут являться противопоказанием к приему некоторых лекарственных препаратов.

Medicine Helper			
Введите данные о пациенте чтобы проверить корректность схемы лечения			
Укажите заболевание	Возраст:	Пол:	Укажите сопутствующие заболевания
Инсульт, не уточненный как кровоизлияние <input type="checkbox"/>	Детский возраст до 12 лет <input type="radio"/> Подростковый возраст от 12 лет до 18 лет <input type="radio"/> Взрослый возраст от 19 лет до 60 лет <input checked="" type="radio"/> Старческий возраст от 61 лет <input type="radio"/>	Мужчина <input checked="" type="radio"/> Женщина <input type="radio"/>	Сахарный диабет <input type="checkbox"/> Инсульт, не уточненный как кровоизлияние или инфаркт <input type="checkbox"/> Туберкулез органов дыхания, подтвержденный бактериологически и гистологически <input type="checkbox"/>
Укажите особенности организма	Укажите выписанные лекарства		
Почечная недостаточность <input type="checkbox"/> Печеночная недостаточность <input type="checkbox"/> Наличие в анамнезе ангионевротического отека <input checked="" type="checkbox"/>	Мексидол - N07XX <input type="checkbox"/> Аскорил - R05C <input type="checkbox"/> Ринсуллин® Р - A10AB01 <input type="checkbox"/>		

Рисунок 7 — вариант заполнения информации о пациенте

Далее, после нажатия кнопки «Send», отправляется запрос в базу данных и, в результате работы программы, пользователь получает подробную информацию: сперва о классификации болезни и её формальном названии и идентификаторе в соответствии с МКБ-10 (Рисунок 8).

Укажите особенности организма	Укажите выписанные лекарства								
Почечная недостаточность Печеночная недостаточность Наличие в анамнезе ангионевротического отека	Мексидол - N07XX Аскорил - R05C Ринсуллин® Р - A10AB01								
<table border="1"><thead><tr><th>ID класса болезни</th><th>Класс болезни</th><th>ID болезни</th><th>Название болезни</th></tr></thead><tbody><tr><td>9</td><td>Болезни системы кровообращения</td><td>I64</td><td>Инсульт, не уточненный как кровоизлияние или инфаркт</td></tr></tbody></table>		ID класса болезни	Класс болезни	ID болезни	Название болезни	9	Болезни системы кровообращения	I64	Инсульт, не уточненный как кровоизлияние или инфаркт
ID класса болезни	Класс болезни	ID болезни	Название болезни						
9	Болезни системы кровообращения	I64	Инсульт, не уточненный как кровоизлияние или инфаркт						

Рисунок 8 — вариант заполнения информации о пациенте и вывод данных о классификации болезни, названии и идентификаторе в соответствии с МКБ-10

Ниже на странице будет наглядно представлена необходимая информация о лекарственных препаратах (Рисунок 9), которая может требовать внимания врача. Для набора данных, приведенном в примере выше, одно из лекарственных препаратов, а именно — Аскорил — не подходит. Следовательно, на основе анализа, Аскорил был удален из схемы лечения и врачу демонстрируется только информация о подходящих лекарствах. Там же можно будет посмотреть точное название препарата, чтобы не спутать его ни с каким другим звучным. Также есть информация об активных веществах, чтобы убедиться, что у пациента нет аллергии или повышенной чувствительности на них. Особое внимание врача должно быть на пункте «с осторожностью» так как есть категория людей, кому лекарственный препарат принимать можно, но

только в дозах, установленных лечащим врачом, ведь прием лекарства по обычным дозировкам, указанным для всех, может быть опасен.



Рисунок 9 — демонстрация информации о лекарственном препарате

Чтобы посмотреть интересующую пользователя информацию, нужно нажать на соответствующее поле и откроется его описание (Рисунок 10):



Рисунок 10 — демонстрация отображения информации о лекарстве

Далее, если промотать страницу в самый низ, будет видна информация о введенном заболевании (Рисунок 11), которое лечится найденным лекарственным препаратом. Здесь вынесена самая важная информация о заболевании, которая может помочь врачу — идентификатор болезни в соответствии с МКБ-10, возможные причины заболевания, самые часто возникающие симптомы, краткое руководство о том, как следует проводить диагностику данного заболевания, включая разные её методы: общий первоначальный внешний осмотр, биопсию, рентгенографию, ультразвуковые исследования, лабораторную диагностику и так далее.

Инсульт, не уточненный как кровоизлияние или инфаркт

ID заболевания

Название заболевания

Инсульт, не уточненный как кровоизлияние или инфаркт

Иные названия

Риск заболеть выше у лиц

Причины заболевания

Симптомы

Диагностика

Пациента просят произвести следующие действия: 1)Выговорить фразу — оценивают четкость произношения отдельных звуков и слов, правильный порядок построение фразы. 2)Поднять руки перед собой — смотрят, может ли человек поднять и держать руки перед собой, одинакова ли сила в конечностях. 3)Попробовать улыбнуться — проверяют, симметрично ли лицо при мимических сокращениях, нет ли провисаний одной половины лица, затруднено ли движение мышц лица при улыбке.

Рисунок 11 — демонстрация информации о введенном заболевании

Если пользователю веб-приложения потребуется срочная информация о заболевании, можно пропустить поля с вводом информации о пациенте, и просто указать интересующее заболевание. Тогда будет доступна информация о классе заболевания, о самом заболевании и всех лекарственных препаратах, имеющихся в базе данных.

Таким образом, сначала считывается информация о пациенте (Листинг 9), затем сервер формирует и отсылает запросы базе данных на основе анализа всей введенной информации (Листинг 10).

```

if r.Method == "POST" {
    err:=r.ParseForm()
    if err != nil {
        log.Println(err)
    }

    //disease
    diseaseName := r.FormValue("disease")
    //age
    age := r.FormValue("age")
    //state
    pregnant := r.FormValue("pregnant")
    feeding := r.FormValue("feeding")
    contra := r.Form["contra"]
    features := r.Form["features"]
    //...
}

```

Листинг 9 — считывание информации о пациенте

```

1. getMedicine := ""
2.     getClass := "Select * From DiseaseAndClassView where DiseaseName=N'" + diseaseName + "'"
3.
4.     if len(contra) != 0 {
5.         getMedicine = "Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties,
Caution, HowToUse, Receipt from DiseaseAndMedicineView where (DiseaseName=N'" + diseaseName + "' and ContradictionsFlag=1) except
Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties, Caution, HowToUse, Receipt
from MedicineAndFeaturesAndContradictionsView where FeatureName = N'" + age + "'"
6.     } else {
7.         getMedicine = "Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties,
Caution, HowToUse, Receipt from DiseaseAndMedicineView where (DiseaseName=N'" + diseaseName + "' and ContradictionsFlag=1) except
Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties, Caution, HowToUse, Receipt
from MedicineAndFeaturesView where FeatureName = N'" + age + "'"
8.     }
9.     getDisease := "Select DiseaseID, DiseaseName, AlternativeNames, RiskGroup, Cause, Symptoms, Diagnostics, Complications
from Diseases where DiseaseName = N'" + diseaseName + "'"
10.
11.    //state
12.
13.    getMedicine += "or FeatureName=N'" + pregnant + "'"
14.
15.    for i:=0; i<len(contra); i++ {
16.        getMedicine += "or DiseaseName=N'" + contra[i] + "'"
17.    }
18.
19.    for i:=0; i<len(features); i++ {
20.        getMedicine += "or FeatureName=N'" + features[i] + "'"
21.    }

```

Листинг 10 — формирование строки запроса

Далее сервер считывает результат запроса в соответствующее хранилище данных и выводит данную информацию в нужном виде в HTML документе (Листинг 11).

```
<!-- ... -->
</head>
<body>
  <div class="tables" height="50px">
    <p class="table1"> </p>
  </div>
  <table>
    <thead><th>ID класса болезни</th><th>Класс болезни</th><th>ID болезни</th><th>Название болезни</th></thead>
    {{range .}}
      <tr>
        <td>{{.ClassID}}</td>
        <td>{{.ClassName}}</td>
        <td>{{.DiseaseID}}</td>
        <td>{{.DiseaseName}}</td>
      </tr>
    {{end}}
  </table>
<!-- ... -->
</body>
</html>
```

Листинг 11 — вывод информации в HTML

Итого, в результате работы программы, по введенным пользователем данным (включая схему лечения) приложение выводит скорректированную схему лечения и информацию о: заболевании пациента, классификации этого заболевания, а также о всех подходящих лекарствах для конкретного пациента. Все неподходящие лекарства будут удалены из финальной демонстрации данных врачу.

К примеру, пользователь введет следующий набор данных о пациенте (Рисунок 12) и решит выписать ему 3 лекарства, 2 из которых являются неподходящими и их прием может привести к ухудшению состояния здоровья пациента.

Medicine Helper

Введите данные о пациенте чтобы проверить корректность схемы лечения

Укажите заболевание Инсульт, не уточненный как кровоизлияние i Send	Возраст: Детский возраст до 12 лет <input type="radio"/> Подростковый возраст от 12 лет до 18 лет <input type="radio"/> Взрослый возраст от 19 лет до 60 лет <input checked="" type="radio"/> Старческий возраст от 61 лет <input type="radio"/>	Пол: Мужчина <input checked="" type="radio"/> Женщина <input type="radio"/>	Положение: Беременна <input type="radio"/> Кормящая <input type="radio"/> Отмена выбора <input type="radio"/>	Укажите сопутствующие заболевания Сахарный диабет <input type="checkbox"/> Инсульт, не уточненный как кровоизлияние или инфаркт <input type="checkbox"/> Туберкулез органов дыхания, подтвержденный бактериологически и гистологически <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Укажите особенности организма Почекная недостаточность <input type="checkbox"/> Печеночная недостаточность <input type="checkbox"/> Наличие в анамнезе ангиневротического отека <input type="checkbox"/> <input type="checkbox"/>
Укажите выписанные лекарства Мексидол - N07XX Аскорил - R05C Ринсулин® Р - A10AB01 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					

Рисунок 12 — ввод данных о пациенте с указанием неподходящих лекарственных препаратов

В результате программа выведет только информацию об одном лекарственном препарате (Мексидоле) вместо трех (Рисунок 13), так как остальные лекарства являются неподходящими на основе анализа подбора лекарств.

ID класса болезни	Класс болезни	ID болезни	Название болезни
9	Болезни системы кровообращения	164	Инсульт, не уточненный как кровоизлияние или инфаркт

Мексидол
<u>ID лекарства</u>
Торговое наименование препарата
Международное наименование
Фармакотерапевтическая группа
Фармакологические свойства
С осторожностью
Способ применения
Условия отпуска из аптек
Инсульт, не уточненный как кровоизлияние или инфаркт

Рисунок 13 — выведено только одно лекарство

5. Тестирование

Во время разработки веб-приложения проводилось тестирование на проверку работоспособности приложения. Многочисленно проверялись и модифицировались запросы к базе данных с целью выборки подходящего лекарственного препарата. Для этого были написаны вспомогательные представления, оптимизирован анализ введенной информации для формирования запроса.

В основном, тестирование опиралось на:

- правильность построения строки запроса к базе данных (Рисунок 14);
- корректность найденой по запросу информации;
- удобство отображения информации;
- правильность построения связей в базе данных.

На этап финальной версии веб-приложения все тесты пройдены, все поставленные задачи выполнены.

```
447 --Select * From DiseaseAndClassView where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'
448 /*Select MedName From DiseaseAndMedicineView where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'
449 EXCEPT (Select MedName from MedicineAndFeaturesView where FeatureName = N'Беременность' AND FeatureName=N'Детский возраст до 12 лет')*/
450
451 --SELECT * from ClassificationOfDiseases where ClassID = (Select ClassID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт')
452
453 --Select * From Medicine where MedicineID=(Select MedicineID from DiseaseMedicine_Indications where DiseaseID =(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'
454 --And (select DiseaseID from Diseases)=(select DiseaseID from DiseaseMedicine_Indications) AND (select DiseaseName from Diseases)=N'Инсульт, не уточненный как кровоизлияние или инфаркт')
455
456 --Select * From Medicine where MedicineID=(Select MedicineID from DiseaseMedicine_Indications where DiseaseID =(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'
457 --EXCEPT Select * From Medicine where MedicineID=(select MedicineID from MedicineFeatures_INT where FeatureID=(select FeatureID from Features where ((FeatureName=N'Взрослый возраст от 19 лет до 60 лет'))))
458
459 --Select * From Medicine where MedicineID=(Select MedicineID from DiseaseMedicine_Indications where DiseaseID =(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'
460 --EXCEPT Select * From Medicine where MedicineID=(select MedicineID from MedicineFeatures_INT where FeatureID=(select FeatureID from Features where ((FeatureName=N'Беременность'))))
461 --GO
462
463 --Select * From Medicine where MedicineID=(Select MedicineID from DiseaseMedicine_Indications where DiseaseID =(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'
464 --EXCEPT select MedicineID from Features join MedicineFeatures_INT on (Features.FeatureID=MedicineFeatures_INT.FeatureID) where FeatureName=N'Беременность' or FeatureName=N'Взрослый возраст от 19 лет до 60 лет'
465 --GO
466
467 --Select * From Medicine where MedicineID=(select MedicineID from MedicineFeatures_INT where FeatureID=(select FeatureID from Features where FeatureName=N'Беременность'))
468 --select FeatureID from Features where ((FeatureName=N'Взрослый возраст от 19 лет до 60 лет') or (FeatureName=N'Беременность') )
469
470 --select * from Features
471
472 --select MedicineID from MedicineFeatures_INT where (FeatureID=(select FeatureID from Features where ((FeatureName=N'Беременность'))) and (FeatureID=(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'))
473
474 --Select * From Medicine where MedicineID=(Select MedicineID from DiseaseMedicine_Indications where DiseaseID =(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт')
475
476 --Select * From Medicine where MedicineID=(Select MedicineID from MedicineFeatures_INT where FeatureID=(select FeatureID from Features where FeatureName=N'Беременность'))
477
478 --select MedicineID from Features join MedicineFeatures_INT on (Features.FeatureID=MedicineFeatures_INT.FeatureID) where FeatureName=N'Беременность' or FeatureName=N'Взрослый возраст от 19 лет до 60 лет'
479
480 --select * from Medicine inner join MedicineFeatures_INT using (MedicineID)
481
482
483 --Select * From Medicine where MedicineID=(Select MedicineID from DiseaseMedicine_Indications where DiseaseID =(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт')
484 --EXCEPT Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties, Caution, HowToUse, Receipt from MedicineAndFeaturesView where FeatureName = N'Беременность'
485 --EXCEPT Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties, Caution, HowToUse, Receipt from MedicineAndFeaturesView where FeatureName = N'Взрослый возраст от 19 лет до 60 лет'
486
487 --Select * from MedicineAndFeaturesView except (select * from MedicineAndFeaturesView where MedName=N'')
488
489
490 --Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties, Caution, HowToUse, Receipt from DiseaseAndMedicineView where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт'
491 --except Select MedicineID, MedName, InternationalName, PharmacotherapeuticGroup, PharmacologicalProperties, Caution, HowToUse, Receipt from MedicineAndFeaturesView where FeatureName = N'' or FeatureName=N'Детский возраст до 12 лет'
492
493 --Select * From Medicine where MedicineID=(Select MedicineID from DiseaseMedicine_Indications where DiseaseID =(select DiseaseID from Diseases where DiseaseName=N'Инсульт, не уточненный как кровоизлияние или инфаркт')
```

Рисунок 14 — демонстрация тестирования запросов к базе данных с

целью проверки правильности работы алгоритма подбора лекарств

Заключение

В итоге выполнения данной курсовой работы были изучены инструменты для работы с Microsoft SQL Server, а также были получены навыки создания веб-приложений на HTML, CSS, JavaScript. Также были приобретены навыки написания сервера на языке Golang, который подключается к базе данных и составляет запросы к базе данных.

Было разработано веб-приложение для удобного и быстрого получения информации о лекарственном препарате по введенным данным о пациенте, причем отсутствует необходимость вводить все имеющиеся на пациента данные, вместо этого достаточно будет одного небольшого набора данных.

Разработанное веб-приложение позволяет пользователю выполнять поиск нужной информации о лекарственных препаратах исходя из данных конкретного пациента. Также приложение дает возможность найти классификацию заболевания в соответствии с МКБ-10 в считанные секунды.

Список использованной литературы

- 1) Международная статистическая классификация болезней и проблем, связанных со здоровьем, 10-го пересмотра. Принята 43-ей Всемирной Ассамблеей Здравоохранения с изменениями и дополнениями Всемирной Организации Здравоохранения 1990-2020 гг. [Электронный ресурс] — <https://mkb-10.com>
- 2) Бондарь А. Г. Microsoft SQL Server 2012 — СПб.: БХВ-Петербург, 2013. — 608с.
- 3) Microsoft Azure SQL Databases Documentation [Электронный ресурс] — <https://azure.microsoft.com/ru-ru/services/sql-database/>
- 4) Golang Documentation [Электронный ресурс] — <https://golang.org/doc/>
- 5) HTML Documentation [Электронный ресурс] — <https://devdocs.io/html/>