

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Г.И. Радченко  
« \_\_\_ » \_\_\_\_\_ 2021 г.

## РАЗРАБОТКА ВЕБ - ПРИЛОЖЕНИЯ ПО ПРОГНОЗИРОВАНИЮ ПОКАЗАТЕЛЕЙ ВИРУСНЫХ ЗАБОЛЕВАНИЙ

### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,  
к.т.н., доцент каф. ЭВМ  
\_\_\_\_\_ В.А. Парасич  
« \_\_\_ » \_\_\_\_\_ 2021 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ И.А. Колесников  
« \_\_\_ » \_\_\_\_\_ 2021 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
« \_\_\_ » \_\_\_\_\_ 2021 г.

Лист задания  
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Г.И. Радченко  
« \_\_\_ » \_\_\_\_\_ 2021 г.

**ЗАДАНИЕ**  
**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Колесникову Ивану Алексеевичу  
обучающемуся по направлению  
09.03.01. «Информатика и вычислительная техника»

**Тема работы:** «Разработка веб-приложения по прогнозированию показателей вирусных заболеваний» утверждена приказом ректора Южно-Уральского государственного университета от 26 апреля 2021 г. №714-13/12

- 1. Срок сдачи студентом законченной работы:** 4 июня 2021 г.
- 2. Исходные данные к работе:** статьи, книги, техническое задание.
  - Гмурман, В.Е. Руководство к решению задач по теории вероятностей и математической статистике / В.Е. Гмурман. – М.: Высшая школа, 2004. – 404с
  - Вентцель, Е.С. Теория вероятностей / Е.С. Вентцель. – М.: Наука, 2006. – 567 с.
  - Андрухаев, Х. М. Теория вероятностей и математическая статистика. Сборник задач : учебное пособие для прикладного бакалавриата / Х. М. Андрухаев. — 3-е изд., испр. и доп. — М.: Изд-во Юрайт, 2018. — 177 с.

### 3. Перечень подлежащих разработке вопросов:

- анализ существующих моделей и статистических методов прогнозирования элементов временных рядов;
- получение вариационных статистических временных рядов с помощью парсинга официальных сайтов;
- построение алгоритма прогнозирования показателей, таких как количество заболевших, умерших при заболевании COVID-19 в Челябинской области;
- разработка веб-приложения для прогнозирования показателей при заболевании COVID-19 в Челябинской области;
- тестирование данного веб-приложения с помощью различных методов тестирования.

4. Дата выдачи задания: 1 декабря 2020 г.

Руководитель работы \_\_\_\_\_ /В.А. Парасич/

Студент \_\_\_\_\_ /И.А. Колесников /

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2021	
Разработка модели, проектирование	01.04.2021	
Реализация системы	01.05.2021	
Тестирование, отладка, эксперименты	15.05.2021	
Компоновка текста работы и сдача на нормоконтроль	20.05.2021	
Подготовка презентации и доклада	24.05.2021	

Руководитель работы \_\_\_\_\_ /В.А. Парасич/

Студент \_\_\_\_\_ /И.А. Колесников /

## **АННОТАЦИЯ**

Колесников И.А. Разработка веб-приложения по прогнозированию показателей вирусных заболеваний. – Челябинск, ЮУрГУ, ЭВМ; 2021, 77 с., 24 ил., библиогр. список – 20 наим.

Выпускная квалификационная работа посвящена разработке веб-приложения по прогнозированию показателей вирусных заболеваний (количество заболевших, умерших) в Челябинской области на примере COVID-19.

В данной работе были проанализированы существующие модели и статистические методы прогнозирования элементов временных рядов, и построен алгоритм прогнозирования показателей, таких как количество заболевших, умерших при заболевании COVID-19 в Челябинской области. Для получения статистических рядов используется парсинг данных со специализированного официального сайта.

В работе изучены методы разработки веб-приложений, выполнен анализ требований, на основе которых было осуществлено проектирование, разработка и тестирование веб-приложения.

Результатом работы является полностью функционирующее веб-приложение «COVID-STATISTIC», которое в дальнейшем будет опубликовано в сети Интернет.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	8
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	10
1.1. ОБЗОР ДРУГИХ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ РАСПРОСТРАНЕНИЯ ВИРУСНЫХ ЗАБОЛЕВАНИЙ.....	10
1.2. ОПРЕДЕЛЕНИЕ ОСНОВНЫХ ПОДСИСТЕМ .....	13
1.3. АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ .....	15
1.3.1. Backend.....	15
1.3.1.1. Выбор языка программирования и фреймворков.....	15
1.3.1.2. Выбор системы управления базой данных .....	16
1.3.2. Frontend .....	18
1.3.2.1. Выбор средств разработки .....	18
1.3.3. Вывод о выборе технологических решений.....	18
2. ОСНОВНЫЕ ТРЕБОВАНИЯ.....	19
2.1. ОСНОВНЫЕ ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ.....	19
2.2. ОСНОВНЫЕ НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ .....	19
3. ПРОЕКТИРОВАНИЕ.....	20
3.1. ПОСТРОЕНИЕ ДОВЕРИТЕЛЬНЫХ ИНТЕРВАЛОВ .....	20
3.1.1. ОСНОВНЫЕ СВЕДЕНИЯ.....	20
3.1.2. ДОВЕРИТЕЛЬНЫЕ ИНТЕРВАЛЫ ПРОГНОЗИРУЕМЫХ ХАРАКТЕРИСТИК.....	21
3.1.3. ЭКСТРАПОЛЯЦИЯ НА ОСНОВЕ СРЕДНЕГО ЗНАЧЕНИЯ ВРЕМЕННОГО РЯДА .....	23
3.1.4. ИСПОЛЬЗОВАНИЕ СКОЛЬЗЯЩИХ СРЕДНИХ ДЛЯ ПРОГНОЗИРОВАНИЯ.....	25

3.2. АРХИТЕКТУРА ПРОГРАММЫ.....	28
3.3. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ .....	29
3.4. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ БАЗЫ ДАННЫХ.....	33
3.5. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ..	35
3.6. ПРОЕКТИРОВАНИЕ АВТОМАТИЧЕСКОГО ОБНОВЛЕНИЯ ДАННЫХ НА СЕРВЕРЕ .....	38
4. РЕАЛИЗАЦИЯ .....	39
4.1. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ.....	39
4.2. КОМПОНЕНТЫ СИСТЕМЫ.....	40
4.3. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА ВЕБ-ПРИЛОЖЕНИЯ .....	42
4.4. РЕАЛИЗАЦИЯ АВТООБНАВЛЕНИЯ ДАННЫХ .....	49
5. ТЕСТИРОВАНИЕ СИСТЕМЫ.....	50
5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ .....	50
5.2. ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ.....	50
ЗАКЛЮЧЕНИЕ .....	53
ПРИЛОЖЕНИЕ А .....	56
ПРИЛОЖЕНИЕ Б.....	63
ПРИЛОЖЕНИЕ В .....	64

## **ВВЕДЕНИЕ**

### **Актуальность**

На протяжении истории человечества и по мере его расселения инфекционные болезни были его неизменным спутником. Вспышки различных болезней происходили и происходят постоянно. В истории человечества можно отметить такие смертоносные пандемии как чума Антонины (165–180 гг), пандемия оспы (XVI век), Великая пандемия бубонной чумы (XVII век), пандемии холеры (1817–1923 гг.), Испанский грипп (1918-1919 гг), пандемия ВИЧ/СПИД (с 1981 г и по настоящее время) и многие другие.

В декабре 2019 года произошла вспышка пневмонии в Ухане, в результате которой был впервые обнаружен штамм COVID-19. К марту 2021 года пандемия охватила 188 стран, в которых было выявлено более 134 миллиона случаев заражения, 2915915 человек из которых погибли. Российская Федерация по числу выявленных случаев находится на 5 месте в рейтинге стран с 4666209 случаями заражения на март 2021 года. Челябинская область также не осталась в стороне от пандемии COVID-19. По информации сайта «СТОПКОРОНАВИРУС.РФ» Челябинская область на протяжении всей пандемии в Российской Федерации занимала по количеству заболевших 12-17 места среди всех регионов страны, уступая «первенство» Москве, Санкт-Петербургу, Екатеринбург и т.п.

В связи с пандемией для населения страны (региона) актуальна любая информация о распространении вируса, количества заболевших, выздоровевших, летальных случаев. Особенно важно иметь представление и о перспективах развития заболеваемости, то есть осуществлять прогноз тенденции изменения показателей пандемии на будущий временной интервал. Такая информация может быть полезна различным категориям пользователей, например, для образовательных учреждений (планирование экскурсий, олимпиад и т.п. школьников); для медицинских учреждений (планирование свободного количества койко-мест); для досуговых учреждений; и для личной безопасности отдельных физических лиц.

Кроме получения информации о заболеваемости COVID-19 немаловажным является её представление в наиболее удобном для пользователей виде (графики, диаграммы и т.п.).

## **Цели исследования**

Выпускная квалификационная работа посвящена разработке веб-приложения для получения пользователем статистических данных об основных показателях пандемии в Челябинской области на текущее время и получение прогноза этих показателей на будущий временной промежуток. Разработанное приложение, используя статистические данные по вирусным заболеваниям, строит прогноз количества заболевших, умерших, выздоровевших в Челябинской области на некоторый будущий временной интервал (на следующий день, на ближайшую неделю и т.п.).

## **Задачи исследования:**

- получение статистических данных о показателях заболеваемости в регионе на основе парсинга данных с официального сайта;
- статистическая обработка полученных данных и построение наиболее подходящего доверительного интервала для прогнозирования распространения заболеваемости COVID-19 в Челябинской области;
- проверка надёжности построенных доверительных интервалов с помощью методов математической статистики;
- разработка веб-приложения «COVID-STATISTIC», с помощью которого пользователь может получить информацию о протекании пандемии в регионе за прошедший период и получить прогноз показателей заболевания на будущий временной промежуток;
- осуществление тестирования веб-приложения.

## **Новизна исследования**

Новизна исследования заключается в том, что для прогнозирования показателей заболеваемости используется метод скользящих средних и построение доверительных интервалов осуществляется на основе среднего значения временного ряда.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

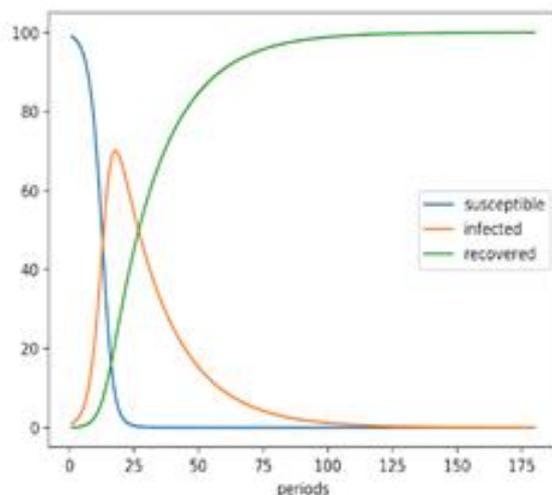
## 1.1. ОБЗОР ДРУГИХ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ РАСПРОСТРАНЕНИЯ ВИРУСНЫХ ЗАБОЛЕВАНИЙ

В настоящее время предлагается множество математических моделей для распространения вирусных заболеваний. Для моделирования распространения инфекционных заболеваний используется классическая модель SIR. Население в модели делится на три части в зависимости от статуса: восприимчивый к заболеванию, заражённый и выздоровевший или умерший (*Susceptible – Infected – Recovered*, первые буквы слов составляют название модели). Люди переходят из состояния в состояние, «перетекая» между накопителями, пока эпидемия не закончится. Чтобы отражать действительность, модель должна основываться на реальных свойствах конкретной болезни и учитывать изменения системы под управленческими воздействиями – например, количество контактов можно снижать карантинными мерами. В настоящее время появилась модификация SIR-модели под названием SEIR (*Susceptible – Exposed – Infected – Recovered*). Здесь к трём состояниям в SIR-модели добавляется еще одно – *Exposed* (заражённый, находящийся в инкубационном периоде).

Распространение любой эпидемии происходит по закону SIR, разработанному в 20-х годах прошлого столетия. Модель SIR предсказывает распространение болезни, общее число инфицированных людей, продолжительность эпидемии, оценивает различные эпидемиологические параметры, например, репродуктивное число  $R_0$ . Число  $R_0$  характеризует заразность инфекционного заболевания в эпидемиологии. Обычно  $R_0$  определяется как количество людей, которые будут заражены типичным заболевшим, попавшим в полностью неиммунизированное окружение при отсутствии специальных эпидемиологических мер, направленных на предотвращение распространения заболевания (например, карантина). Если  $R_0 > 1$ , то на начальном этапе число заболевших будет расти экспоненциально. В начале пандемии по данным ВОЗ предполагалось, что репродуктивное число  $R_0$  для вируса COVID-19 составляет от 2 до 2,5, что гораздо выше, чем для гриппа.

Модель SIR показывает, как различные меры общественного здравоохранения влияют на исход эпидемии, каков наиболее эффективный метод для выпуска ограниченного числа вакцин в данной популяции и т.д. Базовая SIR-модель и её продвинутые версии SEIR — единственные практические модели, способные строить наиболее точные прогнозы[13].

Официальные данные о распространении COVID-19 (источник — *Университет Джона Хопкинса*) обрабатываются по модели SEIR. После чего входные и выходные данные анализируются с использованием методов машинного обучения, минимизируя ошибку между прогнозируемыми данными и фактическими результатами, при этом учитываются десятки поправочных коэффициентов для каждой страны, прогнозируя распространение с максимальной точностью.



SEIR — производная SIR с 4 группами: S, E, I и R.

S — число восприимчивых к инфекции особей  
E — особи в стадии инкубационного периода патогена  
I — инфицированные особи  
R — выздоровевшие или умершие.

Рисунок 1 – SEIR-модель

Для высокой точности прогнозирования требуется учитывать множество изменяющихся параметров:

- скорость инфицирования;
- индекс репродукции;

- скорость выздоровления;
- изменяющиеся параметры карантинных ограничений и т.д.

На данный момент считается, что после перенесенного заболевания COVID-19, у человека вырабатывается иммунитет к повторному заражению. Однако, если устойчивость к вирусу непостоянна, модель SIR преобразуется в SIS-модель. SIS – вариация SIR-модели, позволяющая моделировать эпидемии, при которых узлы могут быть инфицированы несколько раз. В модели SIS узлы переключаются между двумя возможными состояниями: подвержен (S) и инфицирован (I). Прогнозы SIR не учитывают возможность повторного заражения той же особью, и не учитывают то, что на данный момент происходит массовая вакцинация[13].

В данной работе не ставилась цель построения многофакторного прогноза. Построение доверительных интервалов осуществлялось для каждого показателя заболеваемости отдельно, используя метод на основе экстраполяции на основе среднего значения временного ряда, использующий статистику Стьюдента [2].

## 1.2 ОПРЕДЕЛЕНИЕ ОСНОВНЫХ ПОДСИСТЕМ

В состав разработки должны входить следующие подсистемы:

- подсистема получения данных (парсинг специализированных сайтов, выбранных разработчиком, предоставляющих официальную информацию;
- подсистема хранения данных (хранение статистических данных);
- модули статистической обработки данных;
- подсистема анализа и графического изображения полученной информации.

Отметим способы и средства информационного обмена между компонентами в приложении: обмен информацией должен осуществляться из файла в программу и из программы в файл.

*Подсистема получения данных* должна реализоваться с помощью парсинга специализированных сайтов о количестве заболевших, умерших в регионе. Парсинг (Parsing) – это принятое в информатике определение синтаксического анализа. Для этого создается математическая модель сравнения лексем с формальной грамматикой, описанная одним из языков программирования [1].

*Подсистема хранения данных* должна осуществлять хранение статистических данных для последующей их обработки;

*Модули статистической обработки данных* включают в себя подпрограммы построения доверительных интервалов для количества заболевших, умерших. Для этого сначала должны находиться оценки числовых характеристик полученного статистического ряда (оценки математического ожидания, дисперсии, среднего квадратического отклонения).

В случае, если пользователь задает прошедшую дату прогнозирования, приложение должно выдавать реальную статистику на данную дату.

### *Математическое обеспечение*

Для построения прогноза используются элементы теории математической статистики. А именно: построение оценок числовых характеристик статистической

выборки, построение доверительных интервалов с заданной доверительной вероятностью [2], в работе используется значение доверительной вероятности равное 0,95.

Доверительные интервалы являются одним из видов интервальных оценок используемых в математической статистике, которые рассчитываются для заданного уровня значимости. Доверительный интервал гарантирует, что истинное значение неизвестного статистического параметра генеральной совокупности находится в полученном диапазоне значений с вероятностью, которая задана выбранным уровнем статистической значимости [2].

Для построения доверительных интервалов для прогноза на будущий временной промежуток количества заболевших, умерших и т.д. в работе используется метод на основе экстраполяции на основе среднего значения временного ряда, использующий статистику Стьюдента [2].

#### *Информационное обеспечение*

На основе парсинга специализированного сайта будет составлен DATA SET, на основе которого будет строиться статистический прогноз.

Data set (рус. Набор данных, также иногда используется транслитерация «Датасет») — термин, используемый для файловой системы мейнфреймов от IBM; коллекция из логических записей, хранящихся в виде кортежа. Набор данных можно сравнить с файлом, но в отличие от файла набор данных является одновременно и каталогом, и файлом файловой системы, и не может содержать в себе другие наборы [5].

В файловой системе будут три основных компонента: Data Set, код программы на языке Python, разработанное с помощью фреймворка Django, веб-приложение [4].

## 1.3 АНАЛИЗ ОСНОВНЫХ ТЕХНОЛОГИЧЕСКИХ РЕШЕНИЙ

### 1.3.1 Backend

Бэкенд (англ. back-end) — программно-аппаратная часть сервиса. Бэкенд отвечает за осуществление функционирования внутренней части веб-сайта [9].

#### 1.3.1.1 Выбор языка программирования и фреймворков

Для разработки веб-приложения можно использовать следующие языки: C++, C, Java, Ruby, PHP, Python. Проанализируем недостатки перечисленных решений.

Недостатки языка C++ и C:

- очень высокий порог вхождения;
- очень низкий уровень абстракции;
- очень низкая скорость разработки веб-приложений.

Недостатки языка Java:

- высокий порог вхождения;
- более медленная скорость разработки из-за низкого уровня абстракции.

Недостатки Ruby:

- высокий порог вхождения;
- недостатки информационных ресурсов;
- медленно развивается.

Недостатки PHP:

- высокий порог вхождения;
- медленная скорость отладки;
- низкая защищенность;
- неоднозначность в коде.

Для создания веб – приложения выбран язык программирования **Python**, так он является популярным языком программирования, включающим в себя огромное количество фреймворков для веб – разработки.

Преимущества Python[4]:

- простой код;

- широкие возможности;
- кросс-функциональность;
- большое количество фреймворков;

В качестве основного фреймворка для веб – разработки на Python был выбран фреймворк Django.

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation[4].

Преимущества Django[4]:

- безопасность;
- в него легче интегрировать новые решения;
- стандартизированная структура.

Так же, Python широко используется и имеет огромное количество библиотек для работы с данными и графиками, из-за этого он выполняет все требования для данной работы. Для работы с данными (статистическая обработка, построение графиков), были выбраны следующие библиотеки Python:

NumPy — библиотека с открытым исходным кодом для работы с матрицами и массивами, включающая в себя высокоуровневые математические функции [6].

Pandas — это высокоуровневая Python библиотека для анализа данных В экосистеме Python, Pandas является наиболее продвинутой и быстроразвивающейся библиотекой для обработки и анализа данных [7] .

Matplotlib — библиотека на языке программирования Python для визуализации данных двумерной (2D) графикой (3D графика также поддерживается). Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях [8].

### **1.3.1.2 Выбор системы управления базой данных**

Данные можно хранить в файлах и в базах данных (БД). Особенности организации данных в БД по сравнению с файловыми системами:

- базы данных обеспечивают использование одних и тех же данных в различных приложениях;
- БД сводит к минимуму дублирование данных, прибегая к дублированию только для ускорения доступа к данным или для обеспечения восстановления БД при ее разрушении;
- возможность изменения физических особенностей хранения данных без изменения их логической структуры.

Существует множество систем управления базами данных, которые можно использовать для веб – приложений, например PostgreSQL, MongoDB, SQLite.

Недостатки PostgreSQL:

- неразборчивая документация;
- сложная конфигурация;
- скорость работы может падать во время проведения пакетных операция или выполнения запросов чтения.

Недостатки MongoDB:

- SQL не используется в качестве языка запросов;
- программа установки может занять много времени.

Преимущества SQLite:

- распространяется бесплатно;
- простая файловая структура (вся база находится в одном файле);
- отлично подходит для разработки проектов без пользователей;
- удобная в тестирование и разработке.

Таким образом, при разработке - приложения будет использоваться СУБД SQLite.

## **1.3.2 Frontend**

Фронтенд (англ. front-end) — клиентская сторона пользовательского интерфейса программно-аппаратной части сервиса [9].

### **1.3.2.1 Выбор средств разработки клиентской части**

Для разработки клиентской части используются следующие средства.

Язык разметки HTML.

Таблица каскадных стилей CSS.

HTML язык (HyperText Markup Language) – это стандартный язык разметки гипертекста в Интернете. Его основное предназначение – создавать Интернет-страницы и обеспечивать нормальное расположение в документе списков, заголовков, таблиц, картинок и прочих материалов. Гипертекст в данном случае – это текст, связанный указателями-ссылками с другими текстами [9].

Стилем или CSS (Cascading Style Sheets, каскадные таблицы стилей) называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид [9].

### **1.3.3 Вывод о выборе технологических решений**

Для решения данной задачи были выбраны следующие технологические решения: для backend – язык программирования Python, фреймворк Django, библиотеки Numpy, Pandas, Matplotlib, СУБД SQLite; для frontend — язык разметки HTML, таблицы каскадных стилей CSS.

## **2. ОСНОВНЫЕ ТРЕБОВАНИЯ**

### **2.1. ОСНОВНЫЕ ТРЕБОВАНИЯ К ФУНКЦИОНАЛУ СИСТЕМЫ**

Пользователь должен иметь следующие возможности:

- 1) получение информации по количеству заболевших/умерших/выздоровевших в Челябинской области на сегодняшний день и всего за период пандемии;
- 2) возможность получения информации графически;
- 3) прогноз по количеству заболевших/умерших на ближайшее время;
- 4) Состав меню;
- 5) предоставить информацию графически;
- 6) получение прогноза по показателям (на следующие дни);
- 7) обратная связь.

### **2.2. ОСНОВНЫЕ НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

Приложение должно придерживаться определенного стиля:

- 1) хорошо читаемый шрифт;
- 2) отсутствие сильно выделяющихся цветов;
- 3) система должна адаптироваться к различным экранам;
- 4) веб-приложение должно быть работоспособно в любом браузере (актуальной на момент использования версии);
- 5) веб-приложение должно загружать менее чем за 3 секунды (весь контент открытой страницы).

### 3. ПРОЕКТИРОВАНИЕ

#### 3.1. ПОСТРОЕНИЕ ДОВЕРИТЕЛЬНЫХ ИНТЕРВАЛОВ

##### 3.1.1. ОСНОВНЫЕ СВЕДЕНИЯ

В математической статистике при неизвестных законах распределений случайных величин, определяемых выборками, неизвестные параметры распределения заменяются приближенными значениями (оценками). Оценки бывают точечные и интервальные, например, выборочное среднее – точечная оценка математического ожидания. При малом числе наблюдений точечная оценка в значительной степени случайна, и замена истинного значения параметра на оценку может привести к серьезным ошибкам.

Чтобы дать представление о точности и надежности оценки в математической статистике используют так называемые доверительные интервалы и доверительную вероятность.

Пусть найденная по данным выборки величина  $\tilde{a}$  служит оценкой неизвестного параметра  $a$ . Оценка  $\tilde{a}$  определяет параметр  $a$ , тем точнее, чем меньше  $|a - \tilde{a}|$ , то есть чем меньше величина  $\varepsilon$  в неравенстве  $|a - \tilde{a}| < \varepsilon$ ,  $\varepsilon > 0$ . Так как оценка  $\tilde{a}$  – случайная величина, то и разность  $|a - \tilde{a}|$  – случайная величина. Поэтому неравенство  $|a - \tilde{a}| < \varepsilon$ , при заданном  $\varepsilon$ , может выполняться только с некоторой вероятностью.

Доверительной вероятностью (надежностью) оценки  $\tilde{a}$  параметра  $a$  называется вероятность  $\beta$ , с которой оценивается неравенство  $|a - \tilde{a}| < \varepsilon$ .

Доверительную вероятность  $\beta$  выбирают достаточно большой (0,9; 0,95; 0,99), чтобы событие с вероятностью  $\beta$  можно было считать практически достоверным. Затем находят такое значение  $\varepsilon$ , для которого

$$P(|a - \tilde{a}| < \varepsilon) = \beta.$$

В этом случае диапазон возможных значений ошибки, возникающей при замене параметра  $a$  на оценку  $\tilde{a}$ , будет  $\pm\varepsilon$ . Большие по абсолютной величине ошибки будут

появляться только с малой вероятностью  $\alpha = 1 - \beta$ , которую называют *вероятностью риска* или *уровнем значимости*.

Неравенство  $|a - \tilde{a}| < \varepsilon$  можно записать в виде  $\tilde{a} - \varepsilon < a < \tilde{a} + \varepsilon$ .

Доверительным интервалом называется интервал  $(\tilde{a} - \varepsilon; \tilde{a} + \varepsilon)$ , который покрывает неизвестный параметр  $a$  с заданной надежностью  $\beta$ . Доверительный интервал также можно рассматривать как интервал значений параметра  $a$ , совместимых с опытными данными и не противоречащий им.

Точные доверительные интервалы строятся, как правило, в предположении нормальности данных. Следует понимать, что реальные данные, на основании которых мы строим эти интервалы, могут вовсе не выглядеть нормальными (например, это целые положительные числа, в то время как нормальное распределение непрерывно и рассредоточено по всей действительной прямой). Тем не менее, широкое практическое применение описываемых методов дает неплохие результаты. Это объясняется, в частности, асимптотической нормальностью оценок (закон больших чисел).

### **3.1.2. ДОВЕРИТЕЛЬНЫЕ ИНТЕРВАЛЫ ПРОГНОЗИРУЕМЫХ ХАРАКТЕРИСТИК**

Существует несколько классических подходов для построения доверительных интервалов для прогнозируемых значений по данной выборке.

При формировании прогнозов с помощью экстраполяции обычно исходят из статистически складывающихся тенденций изменения тех или иных количественных характеристик исследуемых статистических данных.

Экстраполяционные методы являются одними из самых распространенных и наиболее разработанных среди всей совокупности методов прогнозирования.

С помощью этих методов экстраполируются количественные параметры больших систем, количественные характеристики экономических, научных, производственных и других элементов в сложных системах. В частности, можно использовать

экстраполяционные методы прогнозирования и для построения прогнозов характеристик протекания пандемии COVID-19 в различных странах, регионах и т.п.

Однако степень реальности такого рода прогнозов и соответственно мера доверия к ним в значительной мере обуславливаются аргументированностью выбора пределов экстраполяции. Следует обратить внимание на то, что сложные объекты, как правило, не могут быть охарактеризованы одним параметром. В связи с этим можно определить последовательность действий при статистическом экстраполировании, которая имеет вид:

- во-первых, должно быть чёткое определение задачи, выдвижение гипотез о возможном развитии прогнозируемого объекта;
- во-вторых, выбор оцениваемых характеристик и параметров;
- в-третьих, сбор и систематизация данных. Перед сведением их в соответствующие таблицы, ещё раз проверяется однородность данных и их сопоставимость;
- в-четвертых, когда вышеперечисленные требования выполнены, задача состоит в том, чтобы в ходе статистического анализа и непосредственной экстраполяции данных выявить тенденции или симптомы изменения изучаемых величин.

Для повышения точности экстраполяции используются различные приемы. Один из них состоит, например, в том, чтобы экстраполируемую часть общей кривой развития (тренда) корректировать с учётом реального опыта развития объектов, опережающих в своем развитии прогнозируемый объект. Например, известно, что в России массовая заболеваемость COVID-19 стала проявляться существенно (примерно на месяц) позже, чем в странах Европы. По этому, можно получить прогноз количества заболевших, если определить наиболее похожую на Россию страну, в которой процесс заболеваемости населения протекал аналогично России, но со сдвигом во времени.

**Под трендом** понимается характеристика основной закономерности движения во времени, в некоторой мере свободной от случайных воздействий. Тренд – это длительная тенденция изменения изучаемого показателя. При разработке моделей прогнозирования тренд оказывается основной составляющей прогнозируемого временного ряда, на которую уже накладываются другие составляющие. Результат при этом связывается исключительно с ходом времени.

**Под тенденцией развития** понимают некоторое его общее направление, долговременную эволюцию. Обычно тенденцию стремятся представить в виде более или менее гладкой траектории.

Анализ показывает, что ни один из существующих методов не может дать достаточной точности прогнозов на длительный временной промежуток. Применяемый в прогнозировании метод экстраполяции не дает точных результатов на длительный срок прогноза, потому что данный метод исходит из прошлого и настоящего, и тем самым погрешность накапливается. Этот метод дает положительные результаты на ближайшую перспективу прогнозирования.

### **3.1.3. ЭКСТРАПОЛЯЦИЯ НА ОСНОВЕ СРЕДНЕГО ЗНАЧЕНИЯ ВРЕМЕННОГО РЯДА**

В самом простом случае при предположении о том, что средний уровень ряда не имеет тенденции к изменению или если это изменение незначительно, можно принять  $\hat{y}_{t+l} = \bar{y}$ , то есть прогнозируемый уровень равен среднему значению уровней в прошлом.

Рассмотрим выборку  $x_1, x_2, \dots, x_n$ . Для данной выборки требуется построить доверительный интервал для следующего  $(n+1)$ -го наблюдения (то есть определить границы, в которых оно будет лежать с заданной вероятностью). Для этой цели используется следующий доверительный интервал

$$\bar{x} - st_{\beta} \sqrt{1 + \frac{1}{n}} < x_{n+1} < \bar{x} + st_{\beta} \sqrt{1 + \frac{1}{n}}.$$

Где:

- $n$  – объём выборки;  $\bar{x}$  – выборочное среднее;
- $s$  – исправленная выборочная дисперсия (несмещённая оценка дисперсии),  
 $s = \frac{n}{n-1} D_B$ , где  $D_B$  – выборочная дисперсия;
- $t_\beta$  – критическая точка распределения Стьюдента, определяется по уровню значимости  $\beta$  и по степени свободы  $k = n - 1$  по таблице распределения Стьюдента.

Статистическая надёжность вышеприведенного метода оценивается с помощью коэффициента вариации:

$$V = \frac{\sigma}{\bar{x}} \cdot 100\%,$$

Где:

- $\sigma$  – среднее квадратическое отклонение (или его оценка  $D_B$ );
- $\bar{x}$  – среднее значение временного ряда (выборочное среднее).

Коэффициент вариации показывает степень изменчивости по отношению к среднему показателю выборки. Коэффициент вариации следует вычислять только для данных, измеренных на шкале отношений, то есть шкал, которые имеют значимый нуль и, следовательно, допускают относительное сравнение двух измерений. Метод считается статистически надёжным и может быть использован для прогнозирования, если значение коэффициента вариации не превышает 10%.

В работе для определения промежутка временного ряда, по которому строится прогноз, был просчитан коэффициент вариации (Рисунок 2). Если строить прогноз количества заболевших (умерших) по всей выборке, начиная с апреля 2020 года, то полученный прогноз не будет достоверным, так как коэффициент вариации достигает больших значений (40%). Поэтому возникла задача определения оптимального временного промежутка, по которому и будет осуществляться прогнозирование.

Количество дней	Коэффициент вариации
5 дней	0.91%
10 дней	1.62%
15 дней	2.16%
20 дней	2.92%
25 дней	4.64%
30 дней	6.39%
35 дней	7.43%
40 дней	7.97%
45 дней	8.30%
50 дней	9.62%
55 дней	13.32%
60 дней	19.00%

Рисунок 2 – Таблица с коэффициентами вариации

По результатам вычислений коэффициента вариации было выбрано оптимальное количество дней для построения прогноза – 30-45 дней. Именно такое количество дней и соответственно выборка по этим дням используется для построения доверительного интервала для числа заболевших и т.д.

### **3.1.4. ИСПОЛЬЗОВАНИЕ СКОЛЬЗЯЩИХ СРЕДНИХ ДЛЯ ПРОГНОЗИРОВАНИЯ**

В научно-техническом, экономическом, биологическом прогнозировании в качестве главного фактора-аргумента обычно используют время. Каждому моменту времени соответствуют определённые характеристики признаков, которые со временем в той или иной мере изменяются. Так как прогнозируемый признак – случайная величина, то для прогнозирования актуальным является «сглаживание» полученных значений временного ряда.

Так, например, в июне 2020 года произошла вспышка COVID-19 в Кунашакской районной больнице Челябинской области и, соответственно, резко увеличилось

количество заболевших (Рисунок 3). Если не применять методику сглаживания таких случайных всплесков, то доверительные интервалы будут ошибочными.



Рисунок 3 – Динамика показателей COVID-19

**Скользящее среднее** — общее название для семейства функций, значения которых в каждой точке определения равны некоторому среднему значению исходной функции за предыдущий период. Скользящие средние обычно используются с данными временных рядов для сглаживания краткосрочных колебаний и выделения основных тенденций или циклов. Есть несколько способов построения скользящих средних такие как: простое скользящее среднее, численно равно среднему арифметическому значений исходной функции за установленный период; взвешенное скользящее среднее; экспоненциальное скользящее среднее и т.д.

В работе применяется простое скользящее среднее за 7 дней, дающее хорошее сглаживание с защитой от случайных всплесков (Рисунок 4). Направление движения скользящей средней указывает на превалирующую тенденцию. Метод скользящей средней по сей день является универсальным способом определения тенденции изменения прогнозируемого признака. Простота использования и однозначная интерпретация результата позволяют с высокой долей вероятности определить

господствующую тенденцию. Это минимизирует риск получения недостоверного прогноза.



Рисунок 4 – Скользящее среднее на графике

### 3.2. АРХИТЕКТУРА ПРОГРАММЫ

Для описания функционального состава системы можно представить функциональную схему. Данная схема поясняет отдельные виды процессов, протекающих в целостных функциональных блоках.

Структурная схема – это совокупность элементарных звеньев объекта и связей между ними, один из видов графической модели. Под элементарным звеном подразумевается часть объекта, которая реализует элементарную функцию.

Внутренние компоненты отвечают за логику функционирования и хранение данных, а внешние за визуальное взаимодействие с системой.

Структурная схема представлена на рисунке 5.



Рисунок 5 – Структурная схема

### 3.3. АРХИТЕКТУРА ПРЕДЛАГАЕМОГО РЕШЕНИЯ

Паттерн проектирования – архитектурное решение, предназначенное для проектирования некоторых типовых контекстов.

Подходящими шаблонами для веб-приложений являются MVC. В контексте разработки веб-приложения допустимо использовать классификацию шаблонов проектирования Мартина Фаулера [15].

Существует некоторый набор паттернов, одним из которых является MVC (Model-View – Control) и производные:

1. MVP ( Model – View – Presenter);
2. MVVM ( Model – View – View – Model);
3. HMVC (Hierarchical MVC).

MVC [15] – этот шаблон разделяет работу веб-приложения на три отдельные функциональные роли: модель данных (model), пользовательский интерфейс (view) и управляющую логику (controller). Таким образом, изменения, вносимые в один из компонентов, оказывают минимально возможное воздействие на другие компоненты.

В данном паттерне модель не зависит от представления или управляющей логики, что делает возможным проектирование модели как независимого компонента и, например, создавать несколько представлений для одной модели, представленной на рисунке 6.

Впервые этот шаблон был применён в фреймворке, разрабатываемом для языка Smalltalk в конце 1970-х годов. С этого момента он играет основополагающую роль в большинстве фреймворков с пользовательским интерфейсом. Он в корне изменил взгляд на проектирование приложений.

Большинство фреймворков для веб-программирования сейчас в основе своей содержат именно MVC.

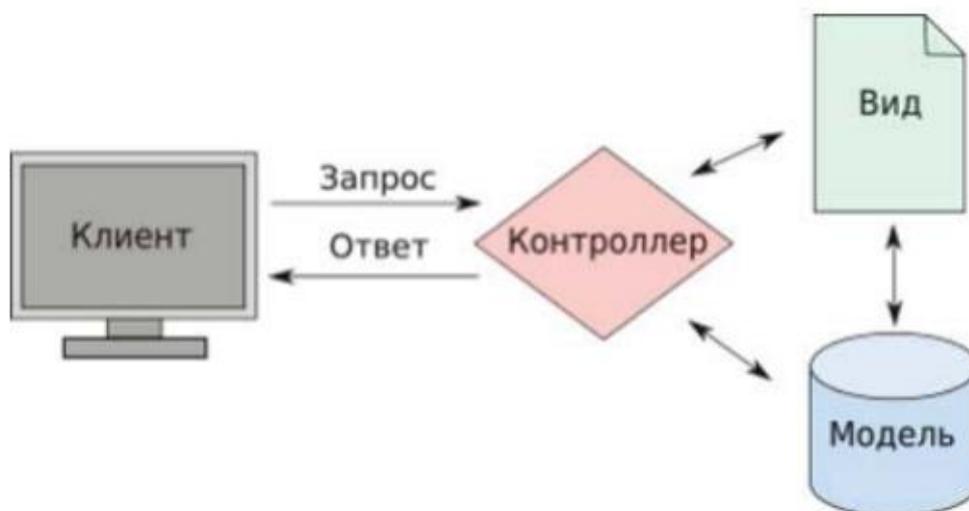


Рисунок 6 – Концепция паттерна MVC

MVVM (Model-View-ViewModel)[16] – это шаблон, который появился для обхода ограничений паттернов MVC и MVP, и объединяющий некоторые их сильные стороны. Эта модель впервые появилась в составе фреймворка Small Talk в 80-х, и была позднее улучшена с учетом обновленной модели презентаций (MVP).

Шаблон MVVM имеет три основных компонента: модель, которая представляет бизнес-логику приложения, представление пользовательского интерфейса XAML, и представление-модель, в котором содержится вся логика построения графического интерфейса и ссылка на модель, поэтому он выступает в качестве модели для представления.

На рисунке 7 представлена диаграмма, которая показывает, как реализовать шаблон MVVM.

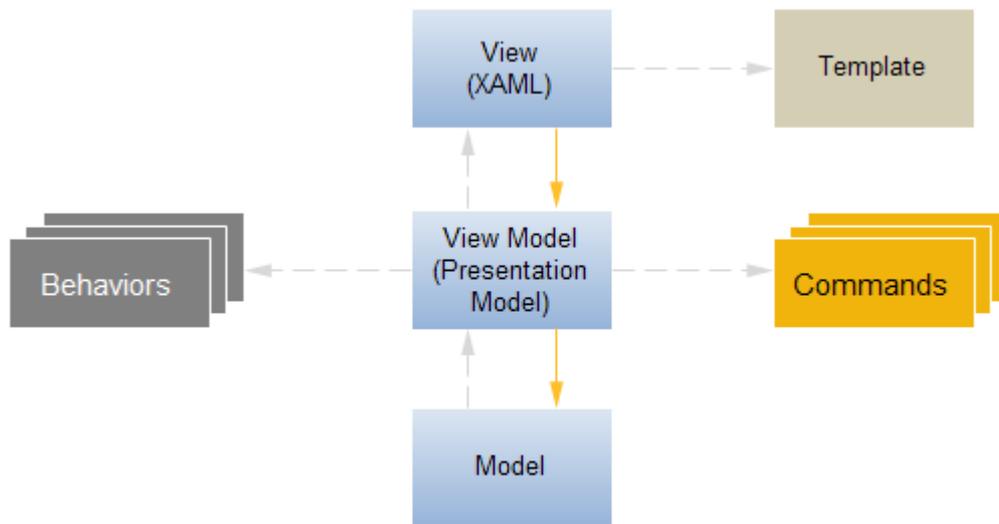


Рисунок 7 – Концепция паттерна MVVM

HMVC[17] – это эволюция концепции MVC, которая используется в многих веб-приложениях. Она появилась как решение некоторых проблем, проявившихся при использовании MVC в веб-приложениях. Решение было представлено на сайте JavaWorld в июле 2000. Предлагалось использовать стандартную триаду Модель-Контроллер-Вид использовать в качестве слоев в «иерархии родитель-потомок». Рисунок 8 отражает принцип работы HMVC.

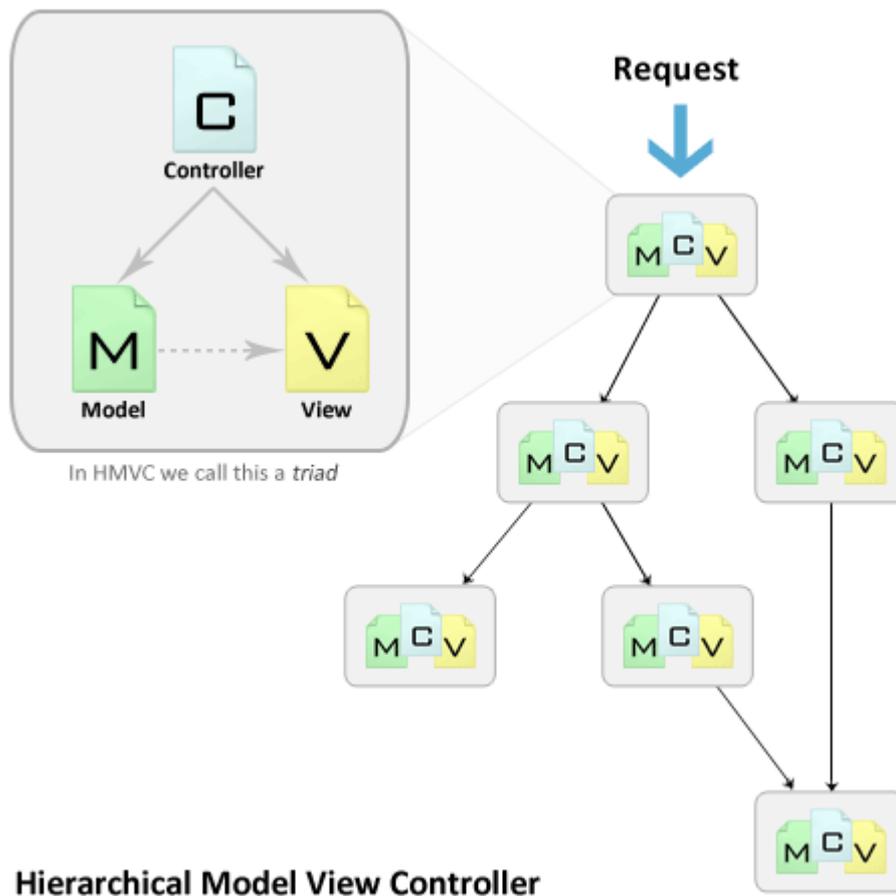


Рисунок 8 – Концепция паттерна HMVC

Для реализации приложения был выбран шаблон MVC, поскольку он обеспечивает простую реализацию бизнес-логики и является типичным решением при разработке веб-приложений.

3.4

### 3.4. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ БАЗЫ ДАННЫХ

Основные задачи проектирования базы данных:

- обеспечение хранения в базе всей необходимой информации;
- обеспечение возможности получения данных по всем необходимым запросам;
- сокращение избыточности данных.

Эскиз спроектированной базы данных представлен на рисунке 9.

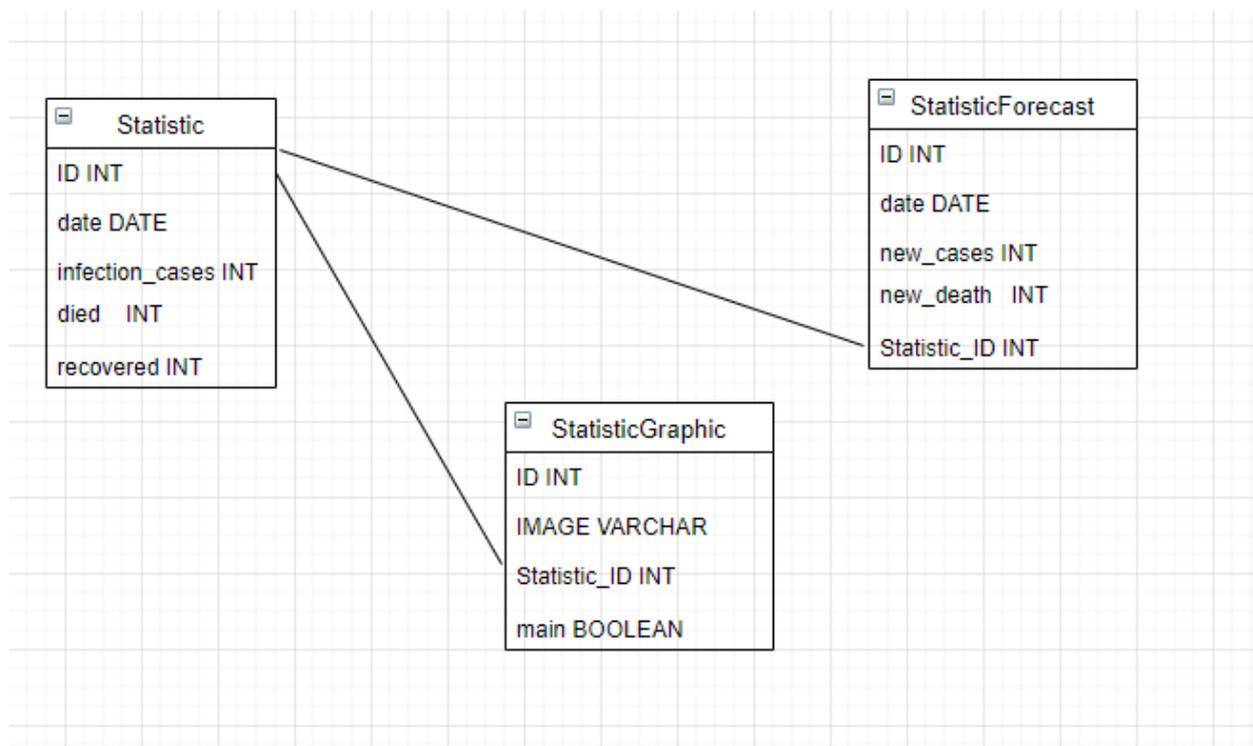


Рисунок 9 – Эскиз базы данных

Проектирование базы данных позволило выделить 3 таблицы:

**Statistic** – таблица в которой хранятся действительные данные о распространении COVID-19 в Челябинской области, включающая в себя следующие поля:

- **ID** - уникальный идентификатор для каждого значения в таблице;
- **Date** – поле, в котором хранятся даты прошедших дней;
- **Infection\_cases** – количество заражений за эту дату;
- **Died** – количество умерших людей в эту дату;
- **Recovered** – количество выздоровевших в эту дату;

**StatisticGraphic** – таблица для реализации вывода графиков, построенных на основе таблицы **Statistic** с помощью специальных фреймворков, включающая в себя следующие поля:

- **ID** - уникальный идентификатор для каждого значения в таблице;
- **IMAGE** – поле, в котором хранится ссылка на таблицу **Statistic**;
- **Main** – поле типа bool, если значение 1, то график выводится на главной странице;
- **Statistic\_ID** – поле типа ForeignKey для связи с таблицей **Statistic**;

**StatisticForecast** – таблица, в которой хранятся данные о прогнозах на ближайшие 3 даты, которые генерируются с помощью статистической обработки с помощью специальных фреймворков, включающая в себя следующие поля:

- **Date** – поле, в котором хранятся даты прогнозируемых дней;
- **new\_cases** – количество прогнозируемых заражений в эту дату;
- **new\_death** – количество прогнозируемых летальных случаев в эту дату;
- **Statistic\_ID** – поле типа ForeignKey для связи с таблицей **Statistic**.

### **3.5. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА**

Интерфейс веб-сервиса состоит из 3 основных страниц: главная страница, страница с информацией о текущем дне и страница «о проекте», на которой находится информация о сервисе и контакты для обратной связи.

В зависимости от того, на какой странице находится пользователь, веб-приложение принимает следующие виды:

- на главной странице выводится прогноз о заболеваемости и количестве летальных случаев на 3 дня вперед, действительная информация о показателях заболеваемости за последние 30 дней и основной график, созданный с помощью специальных фреймворков «Динамика прироста заболеваемости в Челябинской области»
- на странице с информацией о текущем дне находится действительная информация о показателях заболеваемости на сегодняшний день и несколько графиков созданных с помощью специальных фреймворков: «Динамика прироста заболеваемости в Челябинской области», «Прирост заражений в Челябинской области за последние 30 дней + прогноз», «Динамика коронавирусной информации в Челябинской области», «Прирост смертей в Челябинской области за последние 30 дней + прогноз», «Динамика прироста смертей в Челябинской области»
- страница «о проекте» содержит информацию о сервисе и контактные данные для связи с создателем;

Макет графического интерфейса сайта представлен на рисунках 10 – 11.

## Статистическая информация по COVID-19 в Челябинской области

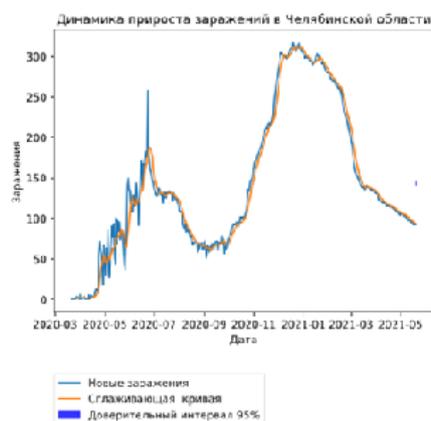
### Прогноз на ближайшие 3 дня

Дата	Ожидается случаев заражения	Ожидается смертей
<input style="width: 80px; height: 40px;" type="text"/>	<input style="width: 80px; height: 40px;" type="text"/>	<input style="width: 80px; height: 40px;" type="text"/>

### Статистика за прошедшие 30 дней

Дата	Случаев заражения	Умерло	Выздоровело
12.05.2021	60 110 (+100)	1 503 (+4)	55 736
11.05.2021	60 010 (+98)	1 499 (+7)	55 639
10.05.2021	59 912 (+98)	1 492 (+8)	55 540
09.05.2021	59 814 (+99)	1 484 (+7)	55 442
08.05.2021	59 715 (+101)	1 477 (+6)	55 342
07.05.2021	59 614 (+103)	1 471 (+8)	55 241
06.05.2021	59 511 (+102)	1 463 (+6)	55 138
05.05.2021	59 409 (+104)	1 457 (+7)	55 038
04.05.2021	59 305 (+105)	1 450 (+5)	54 936
03.05.2021	59 200 (+106)	1 445 (+4)	54 832
02.05.2021	59 094 (+105)	1 441 (+6)	54 731
01.05.2021	58 989 (+106)	1 435 (+5)	54 628
30.04.2021	58 883 (+104)	1 430 (+6)	54 523
29.04.2021	58 779 (+105)	1 424 (+7)	54 416
28.04.2021	58 674 (+107)	1 417 (+8)	54 291

График "Динамика прироста заражений в Челябинской области"



[Переход на страницу с  
подробными данными  
за этот день](#)

[Переход на страницу "о  
проекте"](#)

Переход на  
главную  
страницу

переход на  
страницу "о  
проекте"

Информация о заболевании на сегодняшний день

Дата	Случаев заражения	Умерло	Выздоровело
------	-------------------	--------	-------------

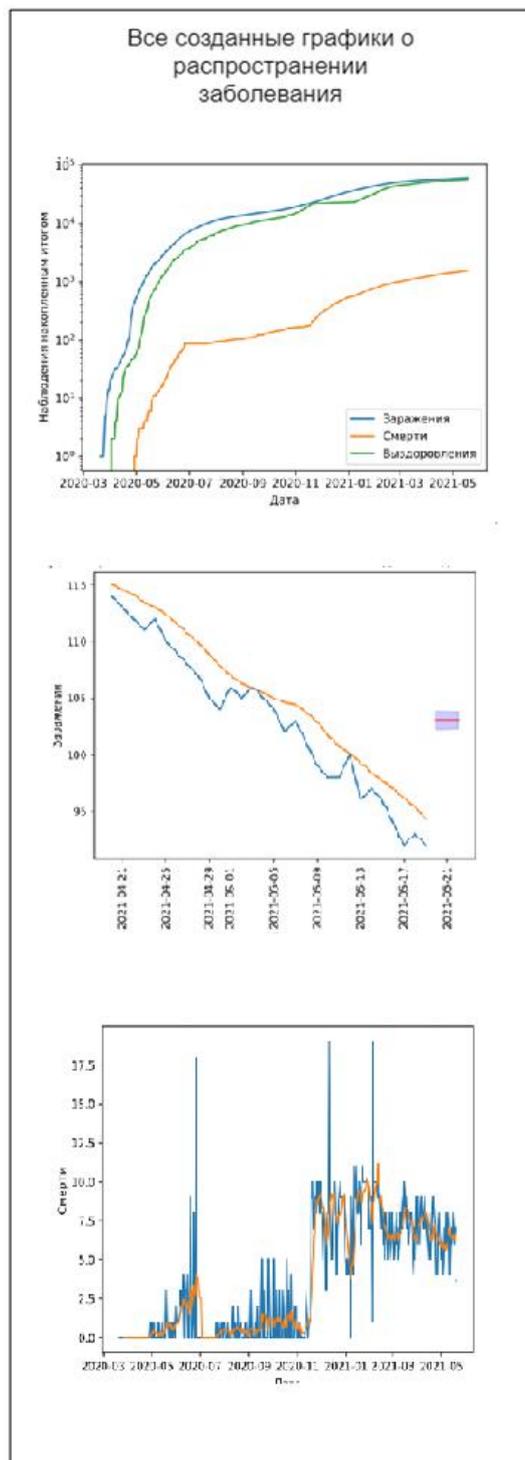


Рисунок 11 – Страница с информацией о текущем дне

### 3.6. ПРОЕКТИРОВАНИЕ АВТОМАТИЧЕСКОГО ОБНОВЛЕНИЯ ДАННЫХ НА СЕРВЕРЕ

Для реализации ежедневного сбора новых данных и ежедневной обработки данных в проекте должна быть реализована команда **management**. Чтобы запустить автоматическое обновление данных, например, в полночь, нужно добавить её в систему в **Cron**, так как это стандартная утилита для планирования задач в веб-приложениях.

Планировщик заданий **Cron** – это один из компонентов операционной системы Linux. Он используется для запуска на хостинге определенных скриптов в нужное время, по расписанию [20].

## 4. РЕАЛИЗАЦИЯ

### 4.1. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ

База данных реализована в СУБД SQLite, с помощью веб-интерфейса Python Django. На рисунке 12 представлена схема базы данных:

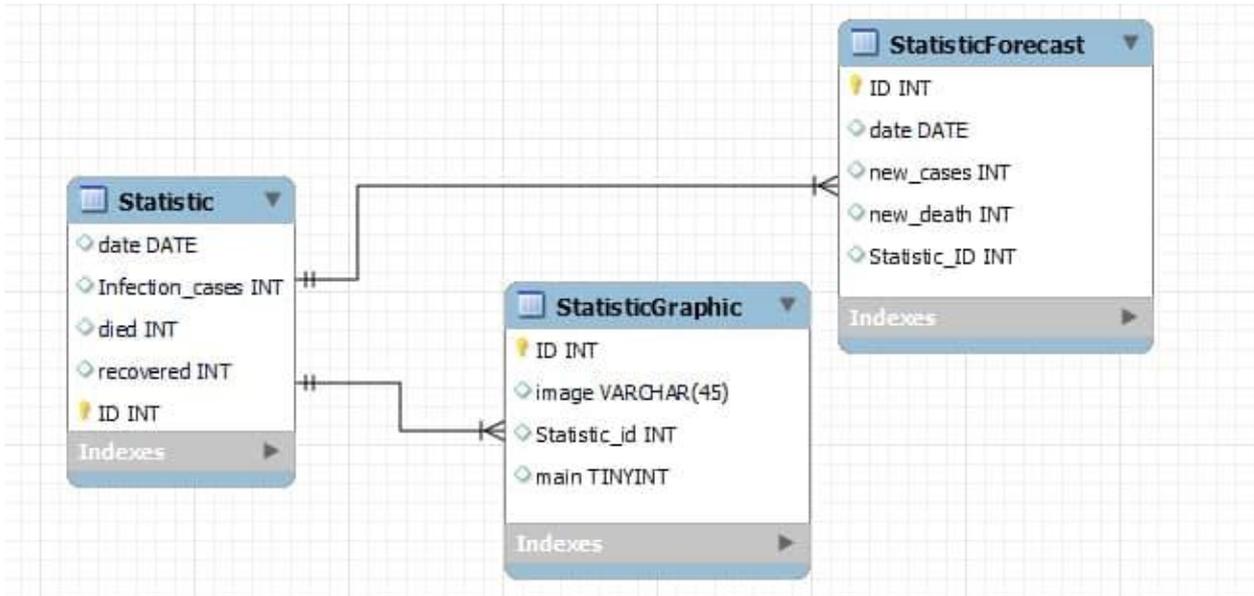


Рисунок 12 – Схема базы данных

## 4.2. КОМПОНЕНТЫ СИСТЕМЫ

На рисунке 13 представлена полная диаграмма компонентов, созданных в ходе реализации веб-приложения в соответствии с паттерном MVC.

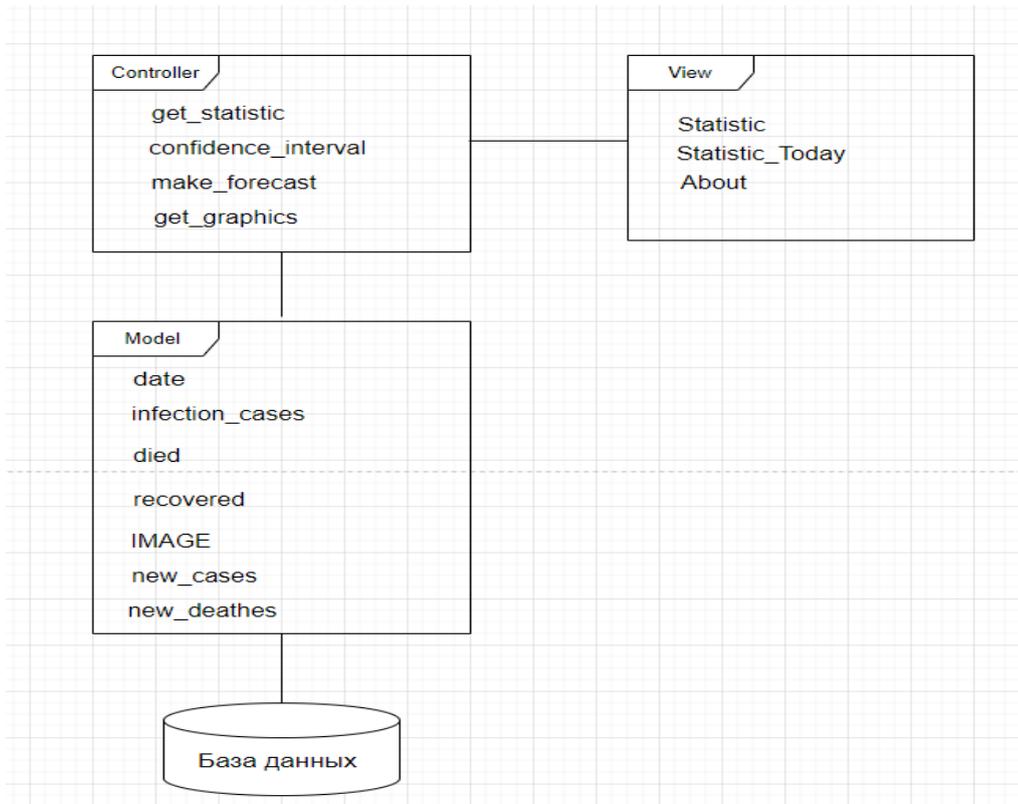


Рисунок 13 – Диаграмма компонентов MVC

Основными компонентами системы являются:

Модели, приведенные в листинге Б.1, приложения Б:

- **Date** – Модель, связанная с таблицами Statistic, StatisticForecast;
- **InfectionCases** – Модель, связанная с таблицей Statistic ( реальные данные по заболевшим);
- **Died** – Модель, связанная с таблицей Statistic ( реальные данные по летальным исходам);
- **Recovered** – Модель, связанная с таблицей Statistic ( реальные данные по выздоровевшим);
- **IMAGE** – Модель, связанная с таблицей StatisticGraphic (полученные графики);

- **New\_cases** – Модель, связанная с таблицей `StatisticForecast` (Прогноз новых случаев заражений);
- **New\_deathes** – Модель, связанная с таблицей `StatisticForecast` (Прогноз новых случаев летальных исходов);

Контроллеры, приведенные в листингах А.1, А.2, А.3, А.4 приложения А:

- **Get\_statistic** – Контроллер, отвечающий за парсинг действительных данных на текущий момент;
- **Confidence\_interval** – Контроллер, отвечающий за вычисление доверительного интервала;
- **Make\_forecast** – Контроллер, отвечающий за прогнозирование заболеваний и смертей на основе доверительного интервала;
- **Get\_graphics** – Контроллер, отвечающий за создание различных графиков, которые потом выводятся в графический интерфейс приложения.

Виды, приведенные в листингах В.1, В.2 приложения В:

- **Statistic** – Вид главной страницы с действительной информацией за последний месяц, прогнозом на 3 дня и главным графиком.
- **Statistic\_Today** – Вид страницы с информацией о текущем дне и всеми остальными графиками;
- **About** – Вид страницы с информацией о системе.

### 4.3. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА ВЕБ-ПРИЛОЖЕНИЯ

На главной странице сайта располагаются результаты прогноза показателей заболеваемости COVID-19 на ближайшие 3 дня, построенного на основе экстраполяции по среднему значению временного ряда; реальная статистика последних 30 дней; основной график «Динамика прироста заболеваемости в Челябинской области». Реализация интерфейса на главной странице представлена на рисунках 14-15.

## Статистическая информация по COVID-19 в Челябинской области

### Прогноз на ближайшие 3 дня

Дата	Ожидается случаев заражения	Ожидается смертей
20.05.2021	104	7
21.05.2021	101	7
22.05.2021	101	7

### Статистика прошедших дней

Дата	Случаев заражения	Умерло	Выздоровело
19.05.2021	60 770 (+92)	1 550 (+7)	56 432
18.05.2021	60 678 (+93)	1 543 (+6)	56 334
17.05.2021	60 585 (+92)	1 537 (+7)	56 245
16.05.2021	60 493 (+94)	1 530 (+8)	56 154
15.05.2021	60 399 (+96)	1 522 (+6)	56 061
14.05.2021	60 303 (+97)	1 516 (+7)	55 966
13.05.2021	60 206 (+96)	1 509 (+6)	55 868
12.05.2021	60 110 (+100)	1 503 (+4)	55 736
11.05.2021	60 010 (+98)	1 499 (+7)	55 639

Рисунок 14 – Реализация главной страницы веб-приложения

24.04.2021	58 240 (+112)	1 393 (+4)	53 828
23.04.2021	58 128 (+111)	1 389 (+7)	53 709
22.04.2021	58 017 (+112)	1 382 (+8)	53 584
21.04.2021	57 905 (+113)	1 374 (+9)	53 455
20.04.2021	57 792 (+114)	1 365 (+8)	53 321



[Подробные данные за 19.05.2021](#)

[О сайте](#)

Рисунок 15 – Реализация главной страницы веб-приложения с графиком

При переходе на страницу «Подробные данные за сегодняшний день» выводится таблица с действительными значениями на сегодня и набор графиков, построенных на основе реальной информации о заражениях в Челябинской области. Реализация

интерфейса на данной странице и набор графиков представлены на рисунках 16-20 . Все графики находятся в БД в таблице **StatisticGraphics**.

[На главную](#) [О сайте](#)

Дата	Случаев заражения	Умерло	Выздоровело
19.05.2021	60 770 (+92)	1 550 (+7)	56 432

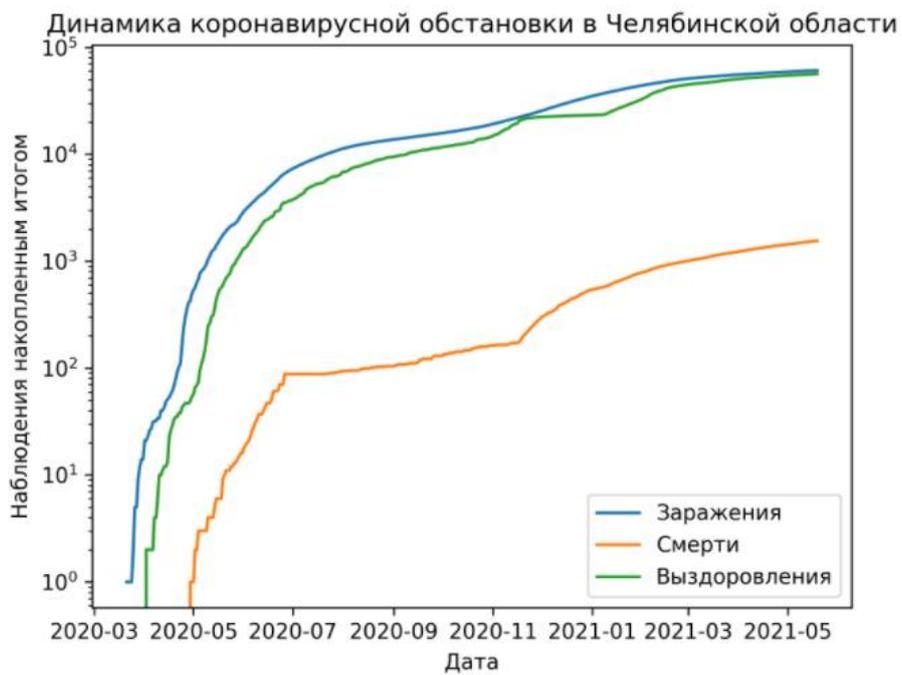


Рисунок 16 – Реализация страницы с действительной информацией за сегодняшний день и графиком «Динамика коронавирусной обстановки в Челябинской области»

Так же на данной странице находятся графики «Динамика прироста заражений в Челябинской области», «Прирост заражений в области за последние 30 дней + прогноз», «Динамика прироста смертей в Челябинской области », «Прирост смертей за последние 30 дней+ прогноз». Данные графики представлены на рисунках 17-20.



Рисунок 17 – График «Динамика прироста заражений в Челябинской области», находящийся на странице с реальной информацией на сегодняшний день

Прирост заражений в Челябинской области за последние 30 дней + прогноз

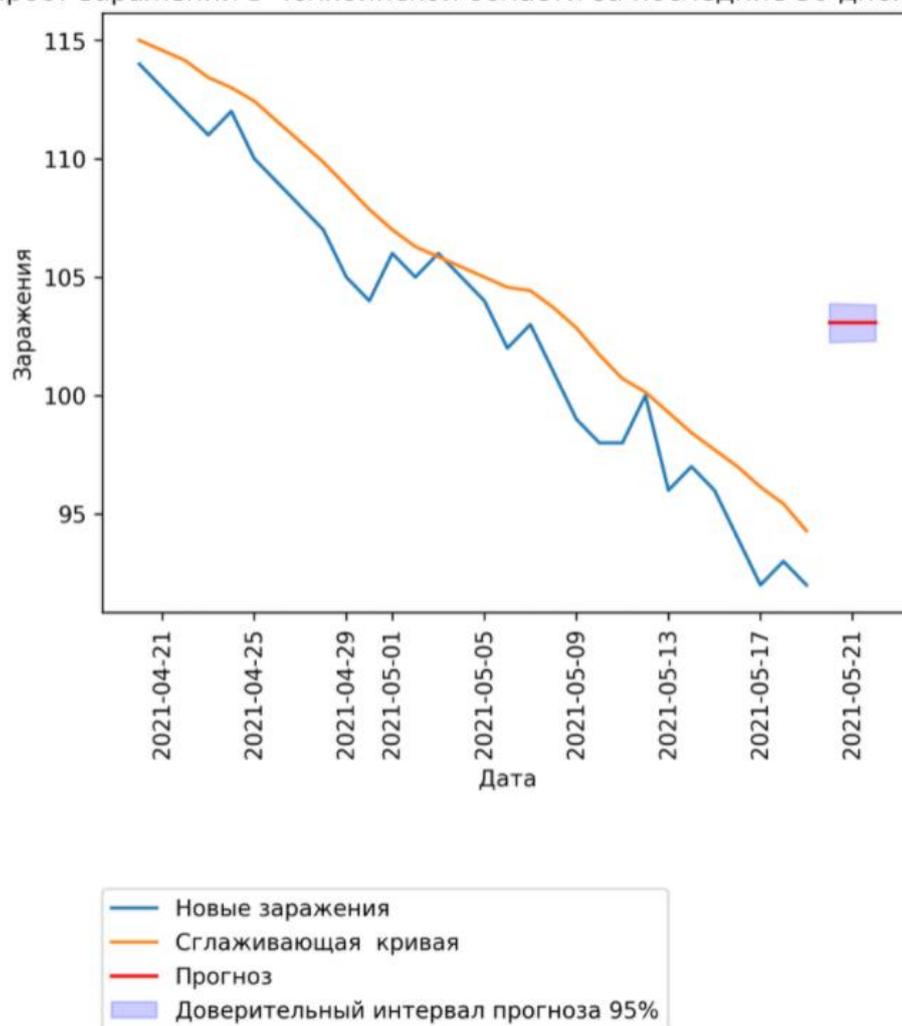


Рисунок 18 – График «Прирост заражений в области за последние 30 дней + прогноз», находящийся на странице с реальной информацией на сегодняшний день

Динамика прироста смертей в Челябинской области

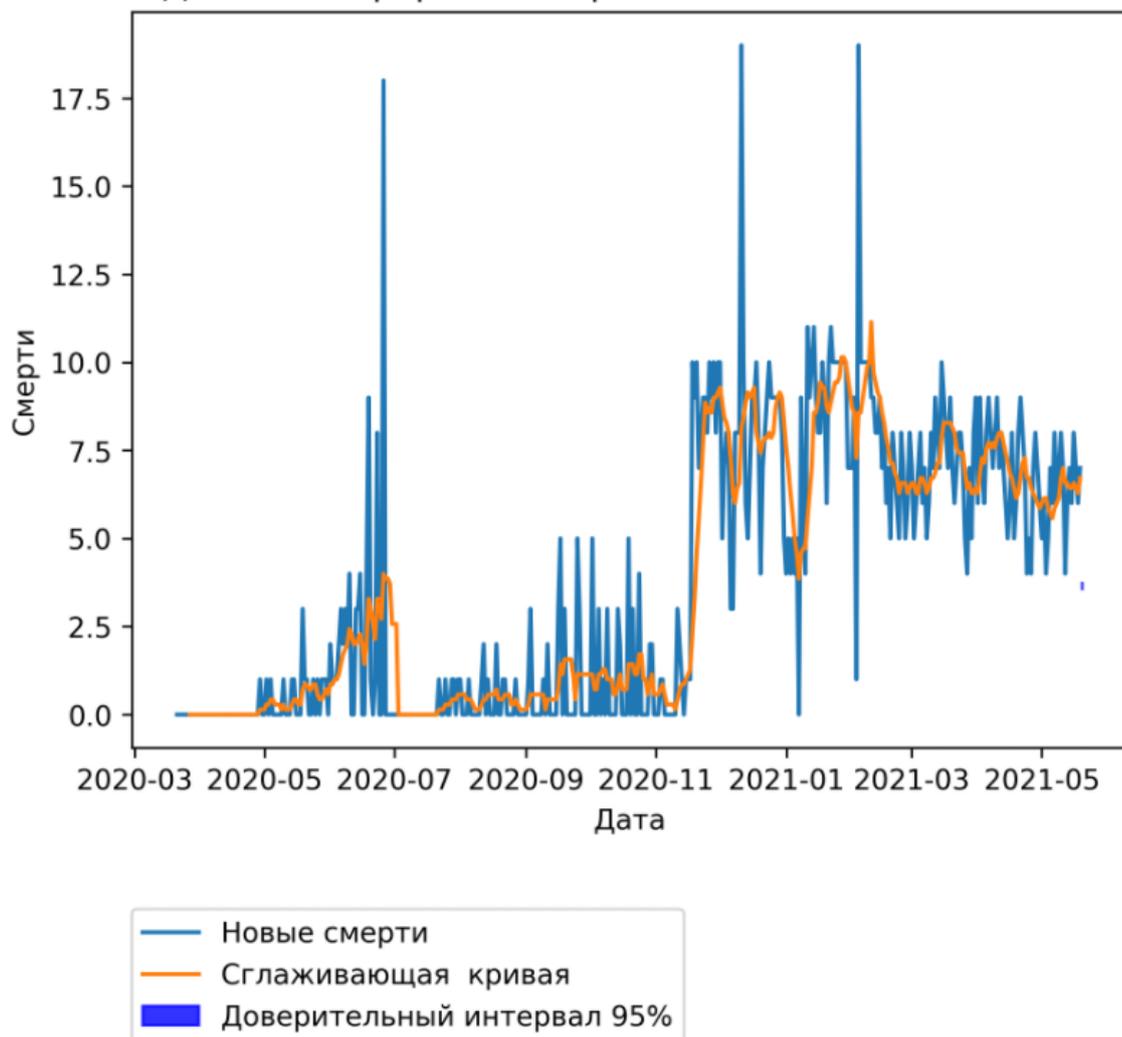
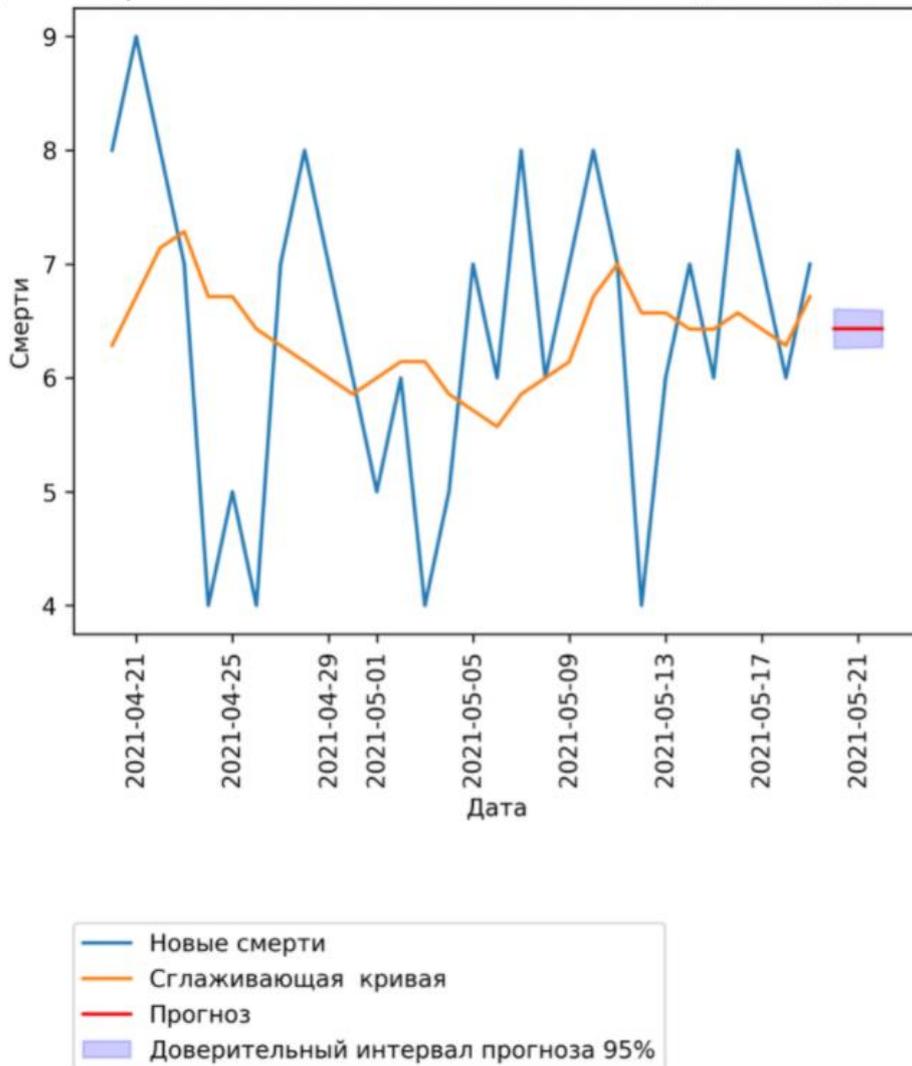


Рисунок 19 – График «Динамика прироста смертей в Челябинской области », находящийся на странице с реальной информацией на сегодняшний день

Прирост смертей в Челябинской области за последние 30 дней + прогноз



[На главную](#)

[О сайте](#)

Рисунок 20 – График «Прирост смертей в области за последние 30 дней + прогноз», находящийся на странице с реальной информацией на сегодняшний день

#### 4.4. РЕАЛИЗАЦИЯ АВТООБНАВЛЕНИЯ ДАННЫХ

В дальнейшем планируется выгрузка данного веб-приложения на хостинг, на данный момент в качестве сервера выступает хостовая машина разработчика, большинство веб-серверов, которые принадлежат хостингам, находятся на ОС Linux/Unix.

Для реализации ежедневного сбора новых данных и ежедневной обработки данных в проекте реализована команда **management**. Чтобы запустить автоматическое обновление данных, например, в полночь, необходимо добавить её в систему в **Cron**.

Путь, который был добавлен в систему **Cron**, представлен на рисунке 21.

```
00 00 * * * /path/to/venv/bin/python /path/to/covid_statistic/manage.py retrieve_new_data
```

Рисунок 21 – Путь, добавленный в систему **Cron** для автоматического обновления данных в 00:00

## **5. ТЕСТИРОВАНИЕ СИСТЕМЫ**

Для тестирования веб-сайта были выбраны следующие виды тестирования:

- приемочное пользовательское тестирование;
- функциональное тестирование;
- интеграционное тестирование.

### **5.1. МЕТОДОЛОГИЯ ТЕСТИРОВАНИЯ**

#### *Юзабилити тестирование*

Юзабилити-тестирование — это метод оценки интерфейса со стороны удобства и эффективности его использования. Чтобы получить его, надо привлечь представителей целевой аудитории программного продукта [18].

#### *Функциональное тестирование*

Функциональное тестирование — это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает [19].

#### *Интеграционное тестирование*

Интеграционное тестирование представляет собой полную проверку готового программного продукта с целью выявления ошибок, возникающих в процессе интеграции программных компонентов.

### **5.2 ПРОВЕДЕНИЕ ПРОЦЕДУРЫ ТЕСТИРОВАНИЯ**

#### *Юзабилити тестирование*

Юзабилити тестирование осуществлялось рядом независимых пользователей, с различным уровнем компьютерной грамотности.

Интерфейс приложения простой, и в ходе тестирования у респондентов не возникло сложностей при работе с сайтом.

#### *Интеграционное тестирование*

Для проведения интеграционного тестирования после окончательной сборки приложение было запущено на реальных экранах с разрешениями:

- 1920x1080 px;

- 1600x900 px;
- 1366x768 px;
- 1024x768 px.

На эмулированных экранах с разрешениями:

- 2560x1600 px;
- 1440x900 px;
- 1280x800 px;
- 1600x1200 px.

На различных по размеру экранам была проверена адаптивность интерфейса приложения.

Веб-приложение было протестировано в браузерах:

- Chrome старше 60 версии;
- Firefox старше 52 версии;
- Opera старше 45 версии;
- Safari старше 10.1 версии;

В различных браузерах проверялась корректная работа функционала сайта.

По результатам тестирования гарантирована корректная работа:

- пользовательского интерфейса веб-сайта у 83% пользователей [18];
- функционала сайта у 95% пользователей[18].

#### *Функциональное тестирование*

Основной функцией веб-приложения является прогнозирование показателей пандемии в Челябинской области, поэтому в качестве функционального тестирования будет проведена проверка на достоверность прогнозируемых показателей по сравнению с реальными показателями на этот день. На представленных рисунках 22-24 верхние значения – прогнозируемые, а нижние действительные.

<b>Дата</b>	<b>Ожидается случаев заражения</b>	<b>Ожидается смертей</b>
02.05.2021	114	7

<b>Дата</b>	<b>Случаев заражения</b>	<b>Умерло</b>
02.05.2021	+105	+6

Рисунок 22 – Прогнозируемые и реальные значения на 02.05.21

<b>Дата</b>	<b>Ожидается случаев заражения</b>	<b>Ожидается смертей</b>
13.05.2021	108	7

<b>Дата</b>	<b>Случаев заражения</b>	<b>Умерло</b>
13.05.2021	+96	+6

Рисунок 23 – Прогнозируемые и реальные значения на 13.05.21

<b>Дата</b>	<b>Ожидается случаев заражения</b>	<b>Ожидается смертей</b>
18.05.2021	106	7

<b>Дата</b>	<b>Случаев заражения</b>	<b>Умерло</b>
18.05.2021	+93	+6

Рисунок 24 – Прогнозируемые и реальные значения на 18.05.21

Для анализа достоверности полученных прогнозируемых показателей определим относительную погрешность этих значений. Так, для прогноза на 02.05.21 (Рис. 22) относительная погрешность количества случаев заражения составила 7,8%, для прогноза на 13.05.2021 (Рис. 23) относительная погрешность этого показателя составила 9,3%, а для 18.05.21 – 11,1%. Для количества умерших по эти датам относительная погрешность составляет 14%. Эти отклонения являются допустимыми для построения прогноза элементов временных рядов.

## ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы было спроектировано и реализовано веб-приложение «COVID - STATISTI» для прогнозирования показателей COVID-19 с графиками.

Для достижения поставленной цели были решены следующие задачи.

1. Выполнен анализ предметной области;
2. Выполнено проектирование веб-приложения с использованием паттерна архитектурного проектирования MVC;
3. Реализация веб-приложения с учётом необходимых функциональных требований;
4. Проведено тестирование веб-приложения с использованием юзабилити-тестирования, интеграционного тестирования, функционального тестирования.

В итоге можно отметить, что получаемые прогнозы достаточно точно определяют показатели заболеваемости COVID-19 в Челябинской области. Гарантируют точность прогноза:

- 1) контроль за коэффициентом вариации, который определяет статистическую надёжность построения доверительных интервалов для элементов временных рядов;
- 2) построение сглаживающей кривой тренда развития показателей заболеваемости, которое позволяет убрать случайные всплески и, тем самым, увеличить надёжность построенных доверительных интервалов.

Отметим, что при функциональном тестировании относительная погрешность прогнозируемых показателей колебалась от 5% до 14% по некоторым показателям. Это является достаточно хорошим результатом в теории прогнозирования.

Реализованное веб-приложение «COVID-STATISTI» может быть полезно многим категориям пользователей, начиная с обычных физических лиц и заканчивая медицинскими учреждениями.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Парсинг [Электронный ресурс]. – Режим доступа :  
<https://ipipe.ru/info/parsing> (Дата обращения: 22.03.2021).
2. Гмурман, В.Е. Руководство к решению задач по теории вероятностей и математической статистике / В.Е Гмурман. – М.:Высшая школа, 2004. – 404с
3. Вентцель, Е.С. Теория вероятностей / Е.С. Вентцель. – М.: Наука, 2006. – 567 с.
4. Что такое Django? [Электронный ресурс]. – Режим доступа :  
<https://habr.com/ru/post/240463> (Дата обращения: 22.03.2021).
5. Data set [Электронный ресурс]. – Режим доступа :  
<https://docs.microsoft.com/ruru/dotnet/api/system.data.dataset?view=netframework-4.8> (Дата обращения: 22.03.2021).
6. NumPy [Электронный ресурс]. – Режим доступа :  
<https://habr.com/ru/post/352678/> (Дата обращения: 22.03.2021).
7. Pandas [Электронный ресурс]. – Режим доступа :  
<https://habr.com/ru/company/ruvds/blog/494720/> (Дата обращения: 22.03.2021).
8. Matplotlib [Электронный ресурс]. – Режим доступа :  
<https://matplotlib.org/> (Дата обращения: 22.03.2021).
9. Froned и Backend [Электронный ресурс]. – Режим доступа :  
<https://tproger.ru/translations/frontend-backend-interaction/> (Дата обращения 23.03.2021).
10. Андрухаев, Х. М. Теория вероятностей и математическая статистика. Сборник задач : учебное пособие для прикладного бакалавриата / Х. М. Андрухаев. — 3-е изд., испр. и доп. — М.: Изд-во Юрайт, 2018. — 177 с.
11. Батраков, П.А., Доверительные интервалы для квантилей распределения / П.А Батраков, А.В. Маер. — Омск: Изд-во Маркет, 2011.- 20с.
12. Симанков, В.С., Определение оптимального сочетания доверительного интервала и доверительной вероятности / В.С. Симанков, В.В. Бучацкая – Спб.: Изд-во Лань. 2019.-101с.
13. Прогнозы распределения covid-19[Электронный ресурс]. – Режим доступа :  
<https://coronavirus-hub.ru/prognozirovanie-covid19/>

(Дата обращения: 22.03.2021).

14. Концепция паттернов [Электронный ресурс]. – Режим доступа :  
<http://design-pattern.ru> (Дата обращения: 22.03.2021).
15. MVC set [Электронный ресурс]. – Режим доступа :  
<http://design-pattern.ru/patterns/mvc.html> (Дата обращения: 22.03.2021).
16. MVVM MVC set [Электронный ресурс]. – Режим доступа :  
<https://professorweb.ru/> (Дата обращения: 22.03.2021).
17. HMVC set [Электронный ресурс]. – Режим доступа :  
<https://ruseller.com/lessons.php?rub=37&id=674> (Дата обращения: 22.03.2021).
18. Юзабилити тестирование [Электронный ресурс]. – Режим доступа :  
<https://habr.com/ru/company/luxoft/blog/508146/> (Дата обращения: 22.03.2021).
19. Функциональное тестирование [Электронный ресурс]. – Режим доступа  
<https://habr.com/ru/post/549054/> (Дата обращения: 22.03.2021).
20. Cron [Электронный ресурс]. – Режим доступа :  
<https://ru.hostings.info/schools/cron.html> (Дата обращения: 22.03.2021).

## ПРИЛОЖЕНИЕ А

### Исходный код контроллеров

#### Листинг 1- Контроллер get\_statistic

```
# считывает таблицу с указанного в настройках сайта, и заносит данные в EXCEL, CSV
def get_statistic():
    tables = pd.read_html(settings.STATISTIC_URL)
    sp500_table = tables[0]
    statistics = []
    for row in sp500_table.to_records().tolist():
        statistics.append(Statistic(
            date=datetime.datetime.strptime(row[1], "%d.%m.%y").date(),
            infection_cases=row[2],
            died=row[3],
            recovered=row[4],
        ))
    Statistic.objects.bulk_create(statistics, ignore_conflicts=True)

    sp500_table.to_csv(settings.STATISTIC_CSV, sep=';', index=False)
    sp500_table.to_excel(settings.STATISTIC_EXCEL, index=False)
```

## Листинг 2- Контроллер confidence\_interval

```
#функция, которая рассчитывает нижнюю и верхнюю границу доверительного интервала
def confidence_interval(x, beta):
    lower = np.mean(x) - t.ppf((1 + beta) / 2, len(x) - 1) * np.std(x, ddof=1) /
    math.sqrt(len(x))
    upper = np.mean(x) + t.ppf((1 + beta) / 2, len(x) - 1) * np.std(x, ddof=1) /
    math.sqrt(len(x))
    return lower, upper
```

## Листинг 3- Контроллер make\_forecast

#функция, которая рассчитывает доверительный интервал на 3 следующих дня

```
def make_forecast(x, dates, forecast_horizon=3):
    forecasts = [np.mean(x)] * forecast_horizon
    forecast_dates = []
    forecast_ci_lower = []
    forecast_ci_upper = []

    for i in range(forecast_horizon):
        if not forecast_dates:

            forecast_dates.append(dates.max() +
pd.Timedelta(days=1))
            else:
                forecast_dates.append(forecast_dates[-1] +
pd.Timedelta(days=1))
                ci_lower, ci_upper = confidence_interval(
                    np.append(x.values, forecasts[:i]),
settings.CONFIDENCE_INTERVALS_SIGNIFICANCE
                )
                forecast_ci_lower.append(ci_lower)
                forecast_ci_upper.append(ci_upper)
    return forecasts, forecast_dates, forecast_ci_lower, forecast_ci_upper
```

## Листинг 4 - Контроллер get\_graphics

```
#функция, которая строит графики с помощью библиотеки matplotlib
def get_graphics():
    df = pd.read_excel(settings.STATISTIC_EXCEL)
    df.info()
    df.head()
    df.columns = ['report_date', 'total_cases', 'total_deaths',
'total_recoveries']
    df[['total_cases', 'new_cases']] = df['total_cases'].str.split(' \(\+', 1,
expand=True)
    df[['total_deaths', 'new_deaths']] = df['total_deaths'].str.split(' \(\+',
1, expand=True)
    df['report_date'] = pd.to_datetime(df['report_date'], format='%d.%m.%y')
    df['total_cases'] = df['total_cases'].str.replace(' ', '').astype(float)
    df['total_deaths'] = df['total_deaths'].str.replace(' ', '').astype(float)
    df['total_recoveries'] = df['total_recoveries'].str.replace(' ',
').astype(float)
    df['new_cases'] = df['new_cases'].str.replace(' |\'', '').astype(float)
    df['new_deaths'] = df['new_deaths'].str.replace(' |\'', '').astype(float)
    df['new_cases'].fillna(0, inplace=True)
    df['new_deaths'].fillna(0, inplace=True)
    df.sort_values('report_date', inplace=True)
    df['rolling_mean_new_cases_7d'] = df['new_cases'].rolling(7).mean()
    df['rolling_mean_new_deaths_7d'] = df['new_deaths'].rolling(7).mean()
    df.index = df['report_date']

    ax = sns.lineplot(data=df, x='report_date', y='total_cases',
label='Заражения')
    sns.lineplot(data=df, x='report_date', y='total_deaths', label='Смерти')
    sns.lineplot(data=df, x='report_date', y='total_recoveries',
label='Выздоровления')

    ax.set_yscale('log')
    #график 'Динамика коронавирусной обстановки в Челябинской области'
    plt.xlabel('Дата')
    plt.ylabel('Наблюдения накопленным итогом')
    plt.title('Динамика коронавирусной обстановки в Челябинской области')
    image = BytesIO()
    plt.savefig(image, dpi=300)
```

```

content_file = ContentFile(image.getvalue())
statistic = Statistic.objects.first()
graphic = StatisticGraphic(statistic=statistic, main=False)
graphic.image.save(settings.STATS_ALL_TIME, content_file)
graphic.save()
plt.clf()

new_cases_sample_ci_lower, new_cases_sample_ci_upper =
confidence_interval(df['new_cases'],

settings.CONFIDENCE_INTERVALS_SIGNIFICANCE)

t.interval(0.95, df['new_cases'].count() - 1, loc=np.mean(df['new_cases']),
           scale=scipy.stats.sem(df['new_cases'], ddof=1))

ax = sns.lineplot(data=df, x='report_date', y='new_cases', label='Новые
заражения')
sns.lineplot(data=df, x='report_date', y='rolling_mean_new_cases_7d',
             label='Сглаживающая кривая')
ax.fill_between(df['report_date'].max() + pd.Timedelta(days=1),
[new_cases_sample_ci_lower],
               [new_cases_sample_ci_upper], color='b', alpha=.8,
label='Доверительный интервал 95%')
#график 'Динамика прироста заражений в Челябинской
plt.xlabel('Дата')
plt.ylabel('Заражения')
plt.title('Динамика прироста заражений в Челябинской области')
lgd = plt.legend(loc=(0, -0.4))
image = BytesIO()
plt.savefig(image, dpi=300, bbox_extra_artists=(lgd,), bbox_inches='tight')
content_file = ContentFile(image.getvalue())
statistic = Statistic.objects.first()
graphic = StatisticGraphic(statistic=statistic, main=True)
graphic.image.save(settings.NEW_CASES_ALL_TIME, content_file)
graphic.save()

```

```

plt.clf()

sample_varition_coefficient = np.std(df['new_cases'] /
np.mean(df['new_cases'])) * 100
sample_varition_coefficient = np.std(df['new_cases'][-30:] /
np.mean(df['new_cases'][-30:])) * 100

new_cases_forecast, new_cases_forecast_dates, new_cases_ci_lower,
new_cases_ci_upper = make_forecast(
    df['new_cases'][-30:], df['report_date'][-30:])
ax = sns.lineplot(data=df[-30:], x='report_date', y='new_cases',
label='Новые заражения')
sns.lineplot(data=df[-30:], x='report_date', y='rolling_mean_new_cases_7d',
label='Сглаживающая кривая')
sns.lineplot(x=new_cases_forecast_dates, y=new_cases_forecast, color='r',
label='Прогноз')
#график 'Прирост заражений в Челябинской области за последние 30 дней +
#прогноз'
plt.fill_between(new_cases_forecast_dates, new_cases_ci_lower,
new_cases_ci_upper, color='b', alpha=0.2,
label='Доверительный интервал прогноза 95%')
plt.xlabel('Дата')
plt.ylabel('Заражения')
plt.title('Прирост заражений в Челябинской области за последние 30 дней +
прогноз')
lgd = plt.legend(loc=(0, -0.7))
plt.xticks(rotation=90)
image = BytesIO()
plt.savefig(image, dpi=300, bbox_extra_artists=(lgd,), bbox_inches='tight')
content_file = ContentFile(image.getvalue())
statistic = Statistic.objects.first()
graphic = StatisticGraphic(statistic=statistic, main=False)
graphic.image.save(settings.NEW_CASES_FORECAST, content_file)
plt.clf()

new_deaths_sample_ci_lower, new_deaths_sample_ci_upper =
confidence_interval(df['new_deaths'],

```

```

settings.CONFIDENCE_INTERVALS_SIGNIFICANCE)

ax = sns.lineplot(data=df, x='report_date', y='new_deaths', label='Новые
смерти')
sns.lineplot(data=df, x='report_date', y='rolling_mean_new_deaths_7d',

label='Сглаживающая кривая')
ax.fill_between(df['report_date'].max() + pd.Timedelta(days=1),
[new_deaths_sample_ci_lower],
[new_deaths_sample_ci_upper], color='b', alpha=.8, label='Доверительный интервал
95%')
# график 'Динамика прироста смертей в Челябинской области'
plt.xlabel('Дата')
plt.ylabel('Смерти')
plt.title('Динамика прироста смертей в Челябинской области')
lgd = plt.legend(loc=(0, -0.4))
image = BytesIO()
plt.savefig(image, dpi=300, bbox_extra_artists=(lgd,), bbox_inches='tight')
content_file = ContentFile(image.getvalue())
statistic = Statistic.objects.first()
graphic = StatisticGraphic(statistic=statistic, main=False)
graphic.image.save(settings.NEW_DEATH_ALL_TIME, content_file)
plt.clf()

sample_varition_coefficient = np.std(df['new_deaths'] /
np.mean(df['new_deaths'])) * 100
sample_varition_coefficient = np.std(df['new_deaths'][-30:] /
np.mean(df['new_deaths'][-30:])) * 100
new_deaths_forecast, new_deaths_forecast_dates, new_deaths_ci_lower,
new_deaths_ci_upper = make_forecast(
df['new_deaths'][-30:], df['report_date'][-30:])

ax = sns.lineplot(data=df[-30:], x='report_date', y='new_deaths',
label='Новые смерти')
sns.lineplot(data=df[-30:], x='report_date', y='rolling_mean_new_deaths_7d',
label='Сглаживающая кривая')

```

```

sns.lineplot(x=new_deaths_forecast_dates, y=new_deaths_forecast, color='r',
label='Прогноз')

plt.fill_between(new_deaths_forecast_dates, new_deaths_ci_lower,
new_deaths_ci_upper, color='b', alpha=0.2,
label='Доверительный интервал прогноза 95%')
#график'Прирост смертей в Челябинской области за последние 30 дней +
#прогноз'
plt.xlabel('Дата')
plt.ylabel('Смерти')
plt.title('Прирост смертей в Челябинской области за последние 30 дней +
прогноз')
lgd = plt.legend(loc=(0, -0.7))
plt.xticks(rotation=90)

image = BytesIO()
plt.savefig(image, dpi=300, bbox_extra_artists=(lgd,), bbox_inches='tight')
content_file = ContentFile(image.getvalue())
statistic = Statistic.objects.first()
graphic = StatisticGraphic(statistic=statistic, main=False)
graphic.image.save(settings.NEW_DEATH_FORECAST, content_file)
plt.clf()

df.to_excel(settings.COVID_XLSX, index=False)
# переводим прогнозы в датафрейм
forecast_df = pd.DataFrame()
forecast_df['report_date'] = pd.Series(new_cases_forecast_dates)
forecast_df['new_cases'] = pd.Series(new_cases_forecast)
forecast_df['new_cases_ci_lower'] = pd.Series(new_cases_ci_lower)
forecast_df['new_cases_ci_upper'] = pd.Series(new_cases_ci_upper)
forecast_df['new_deaths'] = pd.Series(new_deaths_forecast)
forecast_df['new_deaths_ci_lower'] = pd.Series(new_deaths_ci_lower)
forecast_df['new_deaths_ci_upper'] = pd.Series(new_deaths_ci_upper)
forecast_statistics = []
for row in forecast_df.to_records().tolist():
    forecast_statistics.append(
        StatisticForecast(date=timestamp_to_date(int(str(row[1])[:13])),
            new_cases=math.ceil(row[2]),
            new_death=math.ceil(row[5]))
    )
forecast_df.index = forecast_df['report_date']
forecast_df.to_excel(settings.FORECAST_XLSX, index=False)

```

## ПРИЛОЖЕНИЕ Б

### Исходный код моделей

#### Листинг 1- Исходный код моделей

#все таблицы бд

```
class Migration(migrations.Migration):

    initial = True
    dependencies = []
    operations = [
        migrations.CreateModel(
            name='Statistic',
            fields=[
                ('id', models.BigAutoField(auto_created=True,
primary_key=True, serialize=False, verbose_name='ID')),
                ('date', models.DateField(unique=True)),
                ('infection_cases', models.PositiveIntegerField()),
                ('died', models.PositiveIntegerField()),
                ('recovered', models.PositiveIntegerField()),
            ],
        ),
        migrations.CreateModel(
            name='StatisticForecast',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('date', models.DateField(unique=True)),
                ('new_cases', models.PositiveIntegerField()),
                ('new_death', models.PositiveIntegerField()),
            ],
        ),
        migrations.CreateModel(
            name='StatisticGraphic',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('image', models.ImageField(upload_to='')),

                ('main', models.BooleanField()),
                ('statistic', models.ForeignKey(null=True,
on_delete=django.db.models.deletion.SET_NULL, to='statistic.statistic')),
            ],
        ),
    ],
```

## ПРИЛОЖЕНИЕ В

### Исходный код видов

#### Листинг 1- Исходный код statistic\_today

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Сегодня</title>
  <link rel="stylesheet" href="style.css">
  <style>
    @import
url("https://fonts.googleapis.com/css2?family=Rubik:wght@300;500;700&display=swa
p");

    *,
    *::before,
    *::after {
      box-sizing: border-box;
    }

    body,
    html {
      overflow-x: hidden;
    }

    body {
      margin: 50px 0 0 0;
      padding: 0;
      font-family: "Rubik", sans-serif;
    }

    .container {
      width: 840px;
      padding: 0 20px;
      margin: 0 auto;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Сегодня</h1>
  </div>
</body>
</html>
```

```
}  
  
  .titleh1 {  
    font-size: 36px;  
  
text-align: center;  
    margin-bottom: 70px;  
  }  
  
  .titleh3 {  
    font-size: 28px;  
    margin-bottom: 40px;  
  }  
  
/* Таблица */  
.table {  
  
  /* background-image: url('../media/na_fon-1.jpg');  
  background-position: center;  
  background-size: cover; */  
  border: 1px solid #000;  
  border-radius: 15px;  
  overflow: hidden;  
  width: 100%;  
  margin: 0 auto;  
  border-spacing: 0;  
  margin-bottom: 80px;  
  color: rgb(51, 39, 39);  
}  
  
.table__container {  
  max-width: 100%;  
  padding: 0 15px;  
  box-sizing: border-box;  
}
```

```
/* Title */
.table-title {
    font-weight: 700;
    font-size: 20px;
}

.table-title-item {
    word-wrap: break-word;
    width: 33.333%;
    border: 1px solid #000;

    margin: 0;
    padding: 10px 20px;
}

/* info */
.table-info {
    font-weight: 500;
    font-size: 16px;
}

.table-info-item {
    word-wrap: break-word;
    width: 33.333%;
    border: 1px solid #000;
    margin: 0;
    padding: 10px 20px;
}

.table-static-item:nth-child(4) {
    background-color: rgba(21, 241, 21, 0.548);
}

.table-static-item:nth-child(3) {
    background-color: rgba(241, 47, 21, 0.548);
}
```

```
.table-static-item:nth-child(2) {
  background-color: rgba(241, 190, 21, 0.548);
}

.img-grafica {
  width: 100%;
  height: 30%;
  object-fit: contain;
  margin-bottom: 40px;
}

.wrapper-btn {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
margin-bottom: 50px;
}

.btn {
  text-align: center;
  border: 1px solid transparent;
  display: block;
  margin: 0 10px;
  padding: 20px 40px;
  margin-bottom: 20px;
  background-color: #35dbe0;
  color: #fff;
  text-decoration: none;
  border-radius: 30px;
  transition: ease-in 0.3s;
}

.btn:hover {
  background-color: #fff;
  color: #000;
  border: 1px solid #000;
}
```

```
}  
  
@media screen and (max-width: 840px) {  
  .container {  
    width: 540px;  
  }  
  
  /* Title */  
  .table-title {  
    font-weight: 700;  
    font-size: 18px;  
  }  
  
  .table-title-item {  
    width: 33.333%;  
    padding: 15px 10px;  
  }  
  
  /* info */  
  .table-info {  
    font-weight: 500;  
    font-size: 16px;  
  }  
  
  .table-info-item {  
    text-align: center;  
    width: 20%;  
    padding: 15px 10px;  
  }  
  
  .btn {  
    padding: 15px 30px;  
  }  
}  
  
@media screen and (max-width: 540px) {
```

```
.btn {
    font-size: 14px;
}

.titleh1 {
    font-size: 28px;
}

.titleh3 {
    font-size: 20px;
    margin-bottom: 40px;
}

.container {
    width: 300px;
}

/* Title */
.table-title {
    font-weight: 700;
    font-size: 18px;
}

.table__container {
    width: 100%;
    overflow: auto;
    -webkit-overflow-scrolling: touch;
}
}

@media screen and (max-width: 400px) {
    .table__container {
        padding: 0 15px 0 -100px;
    }
}
</style>
```

```

</head>
<body style="text-align: center">
  <div class="container">
    <div class="wrapper-btn">
      <a href="{% url 'statistic' %}" class="btn">На плавную</a>
      <a href="{% url 'about' %}" class="btn">О сайте</a>
    </div>

    <div class="table__container">
      <table class="table">
        <thead>
          <tr class="table-title">
            <td class="table-title-item">Дата</td>
            <td class="table-title-item">Случаев заражения</td>
            <td class="table-title-item">Умерло</td>
            <td class="table-title-item">Выздоровело</td>
          </tr>
        </thead>
        <tr class="table-info ">
          <td class="table-info-item" table-static-item">{{
statistic.date|date:"d.m.Y" }}</td>
          <td class="table-info-item" table-static-item">{{
statistic.infection_cases }}</td>
          <td class="table-info-item" table-static-item">{{
statistic.died }}</td>
          <td class="table-info-item" table-static-item">{{
statistic.recovered }}</td>
        </tr>
      </table>
    </div>

    {% for graphic in statistic.statisticgraphic_set.all %}{% endfor %}

    <div class="wrapper-btn">
      <a href="{% url 'statistic' %}" class="btn">На плавную</a>
      <a href="{% url 'about' %}" class="btn">О сайте</a>
    </div>
  </div>
</body>

</html>

```

## Листинг 2- Исходный код statistics

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Статистика</title>
  <link rel="stylesheet" href="style.css">
  <style>
    @import
url("https://fonts.googleapis.com/css2?family=Rubik:wght@300;500;700&display=swa
p");

    *,
    *::before,
    *::after {
      box-sizing: border-box;
    }
    body, html{
      overflow-x: hidden;
    }
    body {
      margin: 50px 0 0 0;
      padding: 0;
      font-family: "Rubik", sans-serif;
    }
    .container {
      width: 840px;
      padding: 0 20px;
margin: 0 auto;
    }

    .titleh1 {
      font-size: 36px;
      text-align: center;
      margin-bottom: 70px;
    }

    .titleh3 {
      font-size: 28px;
      margin-bottom: 40px;
    }

    /* Таблица */
    .table {

      /* background-image: url('../../media/na_fon-1.jpg');
      background-position: center;
      background-size: cover; */
      border: 1px solid #000;
      border-radius: 15px;
      overflow: hidden;
      width: 100%;
```

```
        margin: 0 auto;
        border-spacing: 0;
        margin-bottom: 80px;
        color: rgb(51, 39, 39);
    }

    .table__container {
        max-width: 100%;
        padding: 0 15px;
        box-sizing: border-box;
    }

    /* Title */
    .table-title {
        font-weight: 700;
        font-size: 20px;
    }

    .table-title-item {
        word-wrap: break-word;
        width: 33.333%;
        border: 1px solid #000;
        margin: 0;
        padding: 10px 20px;
    }

    /* info */
    .table-info {
        font-weight: 500;
        font-size: 16px;
    }

    .table-info-item {
        word-wrap: break-word;
        width: 33.333%;
        border: 1px solid #000;
        margin: 0;
        padding: 10px 20px;
    }

    .table-static-item:nth-child(4) {
        background-color: rgba(21, 241, 21, 0.548);
    }
}
```

```
.table-static-item:nth-child(3) {
    background-color: rgba(241, 47, 21, 0.548);
}

.table-static-item:nth-child(2) {
    background-color: rgba(241, 190, 21, 0.548);
}

.img-grafica {
    width: 100%;
    height: 30%;
    object-fit: contain;
    margin-bottom: 40px;
}

.wrapper-btn {
display: flex;
    flex-wrap: wrap;
    justify-content: center;
    margin-bottom: 50px;
}

.btn {
    text-align: center;
    border: 1px solid transparent;
    display: block;
    margin: 0 10px;
    padding: 20px 40px;
    margin-bottom: 20px;
    background-color: #35dbe0;
    color: #fff;
    text-decoration: none;
    border-radius: 30px;
    transition: ease-in 0.3s;
}
```

```
.btn:hover {
  background-color: #fff;
  color: #000;
  border: 1px solid #000;
}

@media screen and (max-width: 840px) {
  .container {
    width: 540px;
  }

  /* Title */
  .table-title {
    font-weight: 700;
    font-size: 18px;
  }

  .table-title-item {
    width: 33.333%;
    padding: 15px 10px;
  }

  /* info */
  .table-info {
    font-weight: 500;
    font-size: 16px;
  }

  .table-info-item {
    text-align: center;
    width: 20%;
    padding: 15px 10px;
  }

  .btn {
    padding: 15px 30px;
  }
}
```

```
}  
  
@media screen and (max-width: 540px) {  
  .btn{  
    font-size: 14px;  
  }  
  .titleh1 {  
    font-size: 28px;  
  }  
  
  .titleh3 {  
    font-size: 20px;  
    margin-bottom: 40px;  
  }  
  
  .container {  
    width: 300px;  
  }  
  
  /* Title */  
  .table-title {  
    font-weight: 700;  
    font-size: 18px;  
  }  
  
  .table__container {  
    width: 100%;  
    overflow: auto;  
    -webkit-overflow-scrolling: touch;  
  }  
}  
  
@media screen and (max-width: 400px) {  
  .table__container{  
    padding: 0 15px 0 -100px;  
  }  
}
```

```

</style>
</head>

<body>
  <div class="container">
    <h1 class="titleh1">Статистическая информация по COVID-19 в Челябинской
области</h1>

    <h3 class="titleh3">Прогноз на ближайшие 3 дня</h3>
    <div class="table__container">
      <table class="table">
        <thead>
          <tr class="table-title">
            <td class="table-title-item">Дата</td>
            <td class="table-title-item">Ожидается случаев
заражения</td>
            <td class="table-title-item">Ожидается смертей</td>
          </tr>
        </thead>
        <tbody>
          {% for statistic in forecasts reversed%}

          <tr class="table-info">
            <td class="table-info-item" data-label="Дата">{{
statistic.date|date:"d.m.Y" }}</td>
            <td class="table-info-item" data-label="Ожидается случаев
заражения">{{ statistic.new_cases }}
            </td>
            <td class="table-info-item" data-label="Ожидается
смертей">{{ statistic.new_death }}</td>
          </tr>

          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>

```

```

<h3 class="titleh3">Статистика прошедших дней</h3>
  <div class="table__container">
    <table class="table">
      <thead>
        <tr class="table-title">
          <td class="table-title-item">Дата</td>
          <td class="table-title-item">Случаев заражения</td>
          <td class="table-title-item">Умерло</td>
          <td class="table-title-item">Выздоровело</td>
        </tr>
      </thead>
      <tbody>
        {% for statistic in statistics reversed %}
          <tr class="table-info ">
            <td class="table-info-item table-static-item" data-
label="Дата">
              {{ statistic.date|date:"d.m.Y"}}
            </td>
            <td class="table-info-item table-static-item" data-
label="Случаев заражения">
              {{statistic.infection_cases }}
            </td>
            <td class="table-info-item table-static-item" data-
label="Умерло">{{ statistic.died }}</td>
            <td class="table-info-item table-static-item" data-
label="Выздоровело">{{ statistic.recovered }}
            </td>
          </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>

    {% if image %}{% endif %}

    <div class="wrapper-btn">
      <a href="{% url 'statistic_today' %}" class="btn">Подробные данные за
        {{statistics.first.date|date:"d.m.Y"}}
      </a>
      <a href="{% url 'about' %}" class="btn">О сайте</a>
    </div>
  </div>
</body>

</html>

```