Содержание:

**Введение.**

**1. Анализ предметной области**

1.1. Организационная структура

1.2. Функциональная структура

1.3. Информационная структура

**2. Разработка информационной модели**

2.1. Структура информационной базы

2.1.1. Схема базы данных (логическая)

2.1.2. Основные запросы к базе данных

**3. Инфраструктура информационной системы**

3.1. Экранные формы для пользователя

3.2. Отчеты

**Введение.**

В данном проекте объектом исследования является социальная организация. Предметом исследования является структура информационной системы социальных мероприятий и ее состав. Целью является разработка базы данных, которая обеспечивает работу с информацией.

Основная цель разрабатываемой системы в предоставлении актуальной информации о прошедших и планируемых культурно-социальных событий, например, партийных собраниях, а также о действующих членах организации.

Данная система решает проблему структуризации, безопасности и доступности используемых сведений путем построения строгой конфигурации базы данных, правил контроля доступа и прав и открытой публикации в сети интернет.

В информационной системе существуют: пользователи, проверенные пользователи, редакторы и администратор.

Пользователи могут регистрироваться и авторизоваться, просматривать список мероприятий; городов, где могут проходить и происходят мероприятия; произошедших происшествий на событиях и общую сводную статистику.

Проверенные пользователи добавляются администратором после подачи соответствующей заявки и персональных данных, после их проверки. Они имеют те же возможности, что и обычные пользователи, но также могу т добавлять, удалять и редактировать свои события.

Редакторы – сформированная заранее группа, имеющая в дополнение привилегий проверенных пользователей возможность добавлять, удалять и редактировать данные о городах, в которых возможно проведение мероприятий, и происшествиях.

Администратор – контролирует верификацию личных данных и вносит в базу проверенных пользователей.

**Виды запросов в информационной системе:**

* На добавление:
  + Добавление мероприятия в выбранном городе из списка существующих.
  + Добавление города в список доступных.
  + Добавление происшествия на выбранном мероприятии.
  + Добавление пользователя.
  + Добавление проверенного пользователя.
* На изменение:
  + Актуализация информации о выбранном мероприятии.
  + Актуализация информации о выбранном городе.
  + Актуализация информации о выбранном происшествии.
* На поиск:
  + Получение списка доступных городов.
  + Получение списка мероприятий.
  + Получение списка происшествий.
  + Получение списка доступных городов по поисковым параметрам: Название, Заинтересованная публика, Объем дохода – с сортировкой по релевантности.
  + Получение списка мероприятий по поисковым параметрам: название, Город проведения, Количество участников – с сортировкой по релевантности.
* На удаление:
  + Удаление выбранного города – запрещено при наличии событий.
  + Удаление выбранного мероприятия – каскадное удаление связанных происшествий.
  + Удаление выбранного происшествия.

**Определение сущностей:**

Данный проект предусматривает создание базы данных, содержащей информацию о мероприятиях, местах их проведения и структуре взаимодействия с пользователями. Такая база данных должна обеспечить хранение и просмотр информации о мероприятиях, городах, проверенных и рядовых пользователях и происшествиях. Исходя из поставленной задачи, информационная система должна содержать следующие сущности и атрибуты:

1. Города - содержит информацию о доступных городах.
   1. idCity - первичные ключи таблицы.
   2. city – наименование города.
   3. people – заинтересованная публика.
   4. production – объем дохода.
2. Мероприятия - содержит информацию о проведенных и запланированных событиях.
   1. id – первичные ключи таблицы.
   2. name – наименование мероприятия.
   3. date – дата проведения мероприятия.
   4. city – внешний ключ для связи с таблицей Города, определяет город проведения мероприятия.
   5. master – внешний ключ для связи с таблицей Проверенные пользователи, определяет пользователя, который создал мероприятие.
   6. participants – количество участников.
   7. fulltxt – строка, объединяющая наименование связанного города и наименования мероприятия для полнотекстового поиска.
3. Происшествия – содержит информацию о случившихся на мероприятиях ситуациях.
   1. idCrime – первичные ключи таблицы.
   2. crime – наименование происшествия.
   3. witness – внешний ключ для связи с таблицей Проверенные пользователи, определяет подтверждающего событие пользователя.
   4. event – внешний ключ для связи с таблицей Мероприятия, определяет событие, на котором произошла данная ситуация.
4. Проверенные пользователи – содержит информацию о проверенных пользователях (пользователях, что подтвердили личные данные).
   1. idCitizen – первичные ключи таблицы.
   2. login – внешний ключ для связи с таблицей Пользователи, определяет данные регистрации данного пользователя.
   3. city – внешний ключ для связи с таблицей Города, определяет город, за которым закреплен польователь.
   4. pay – членский взнос пользователя.
   5. fio – личные данные пользователя.
   6. passport – паспортные данные пользователя.
   7. rights – права пользователя.
5. Пользователи – прошедшие регистрацию посетители ресурса.
   1. Id– первичные ключи таблицы.
   2. Login – логин для входа в систему.
   3. Password – пароль для входа в систему.

**Анализ предметной области.**

**Организационнаяструктура**

Пользователь

Проверенный пользователь

Администратор

Редактор

**Функциональная структура**

Диаграмма прецедентов

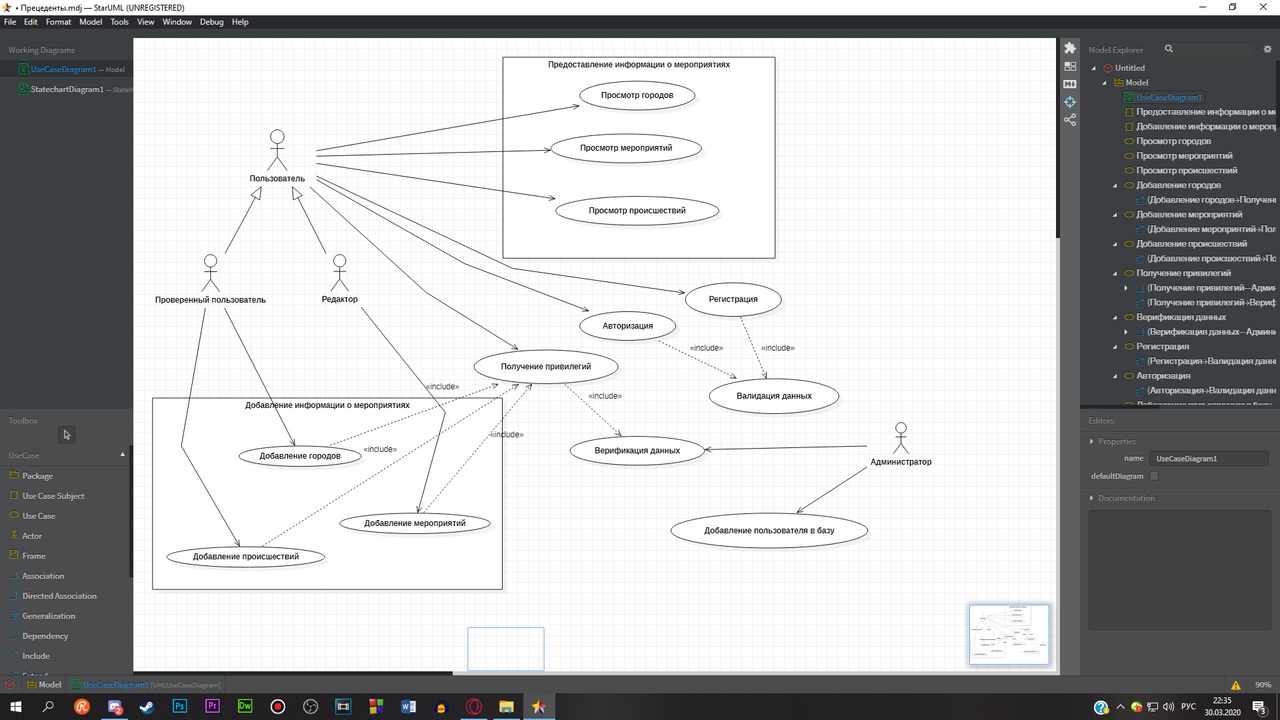
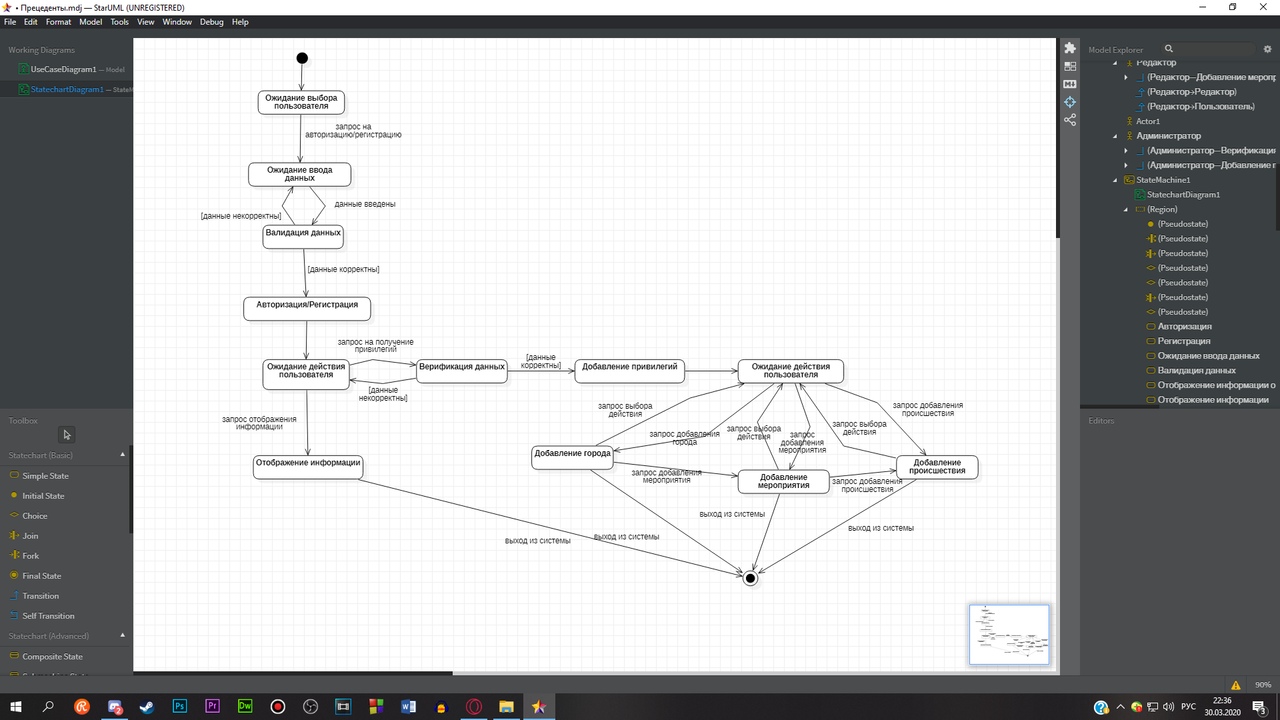
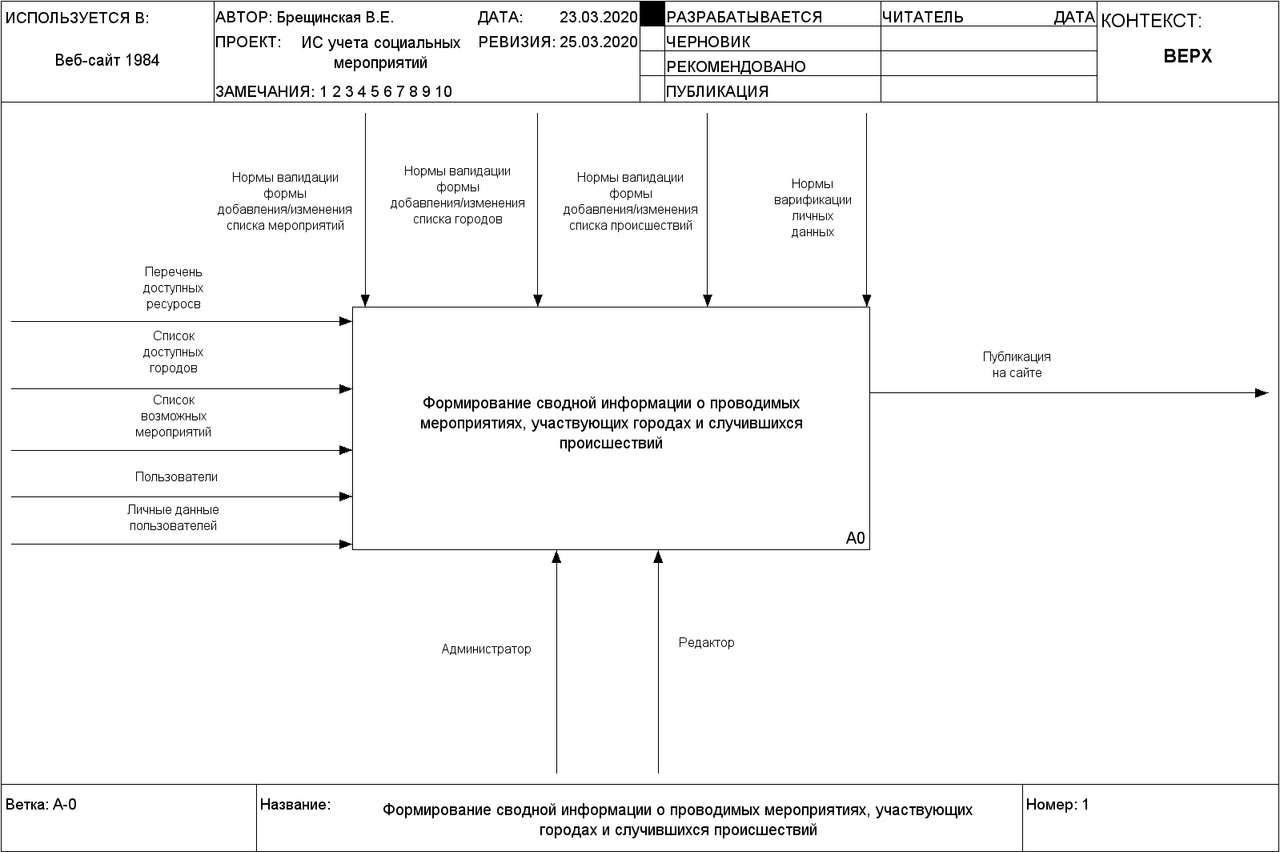
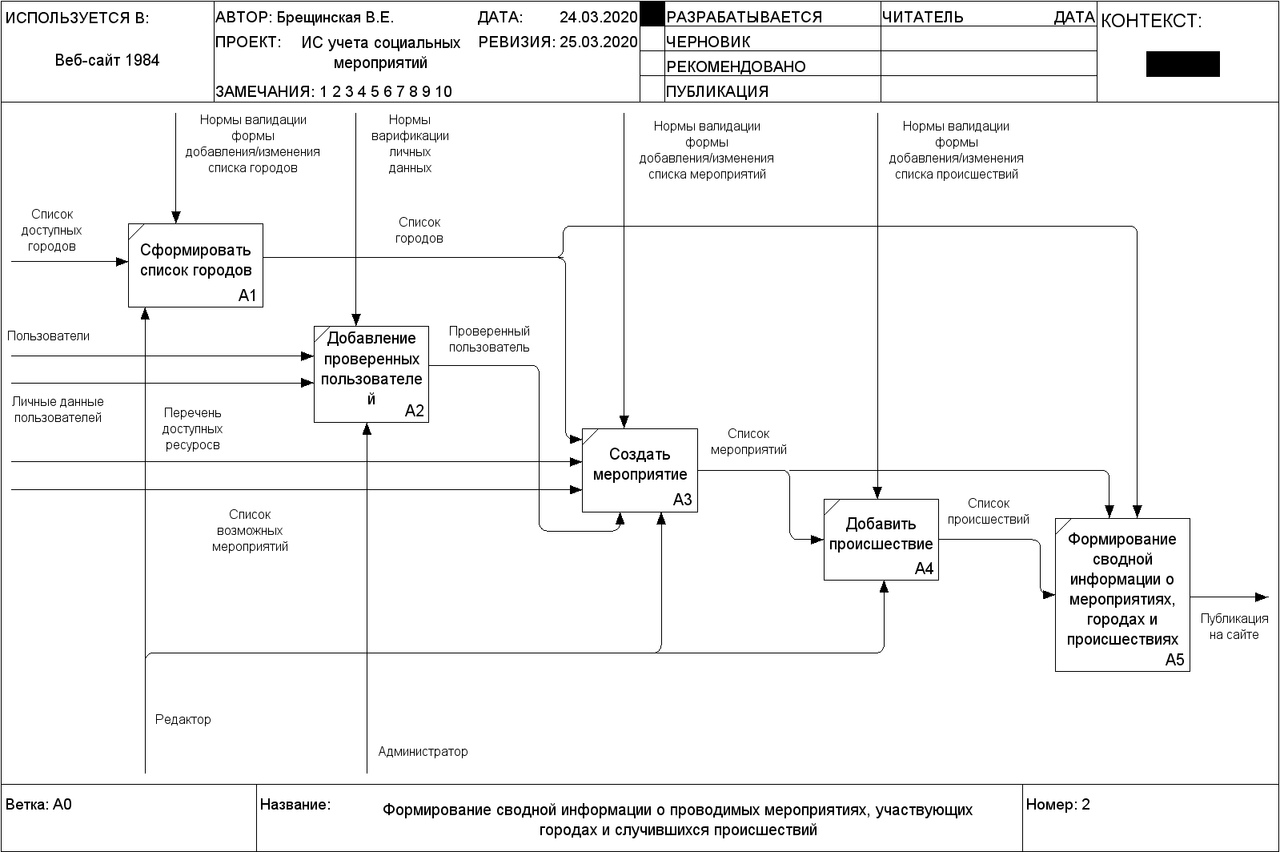


Диаграмма состояний



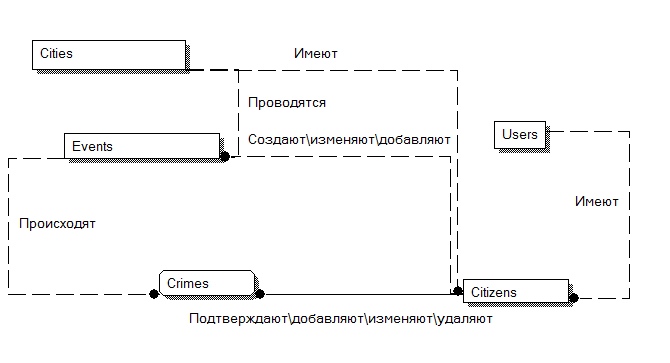
IDEF0 модель



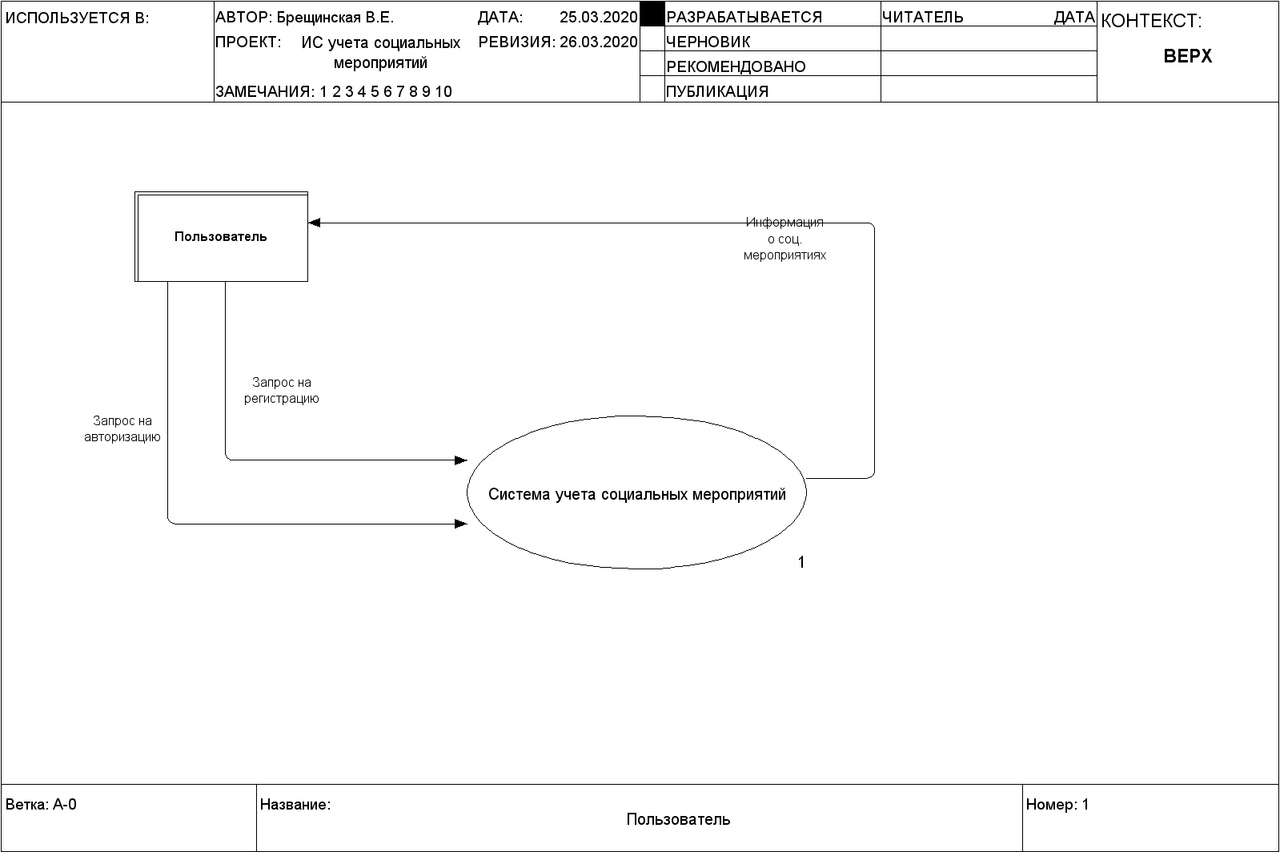


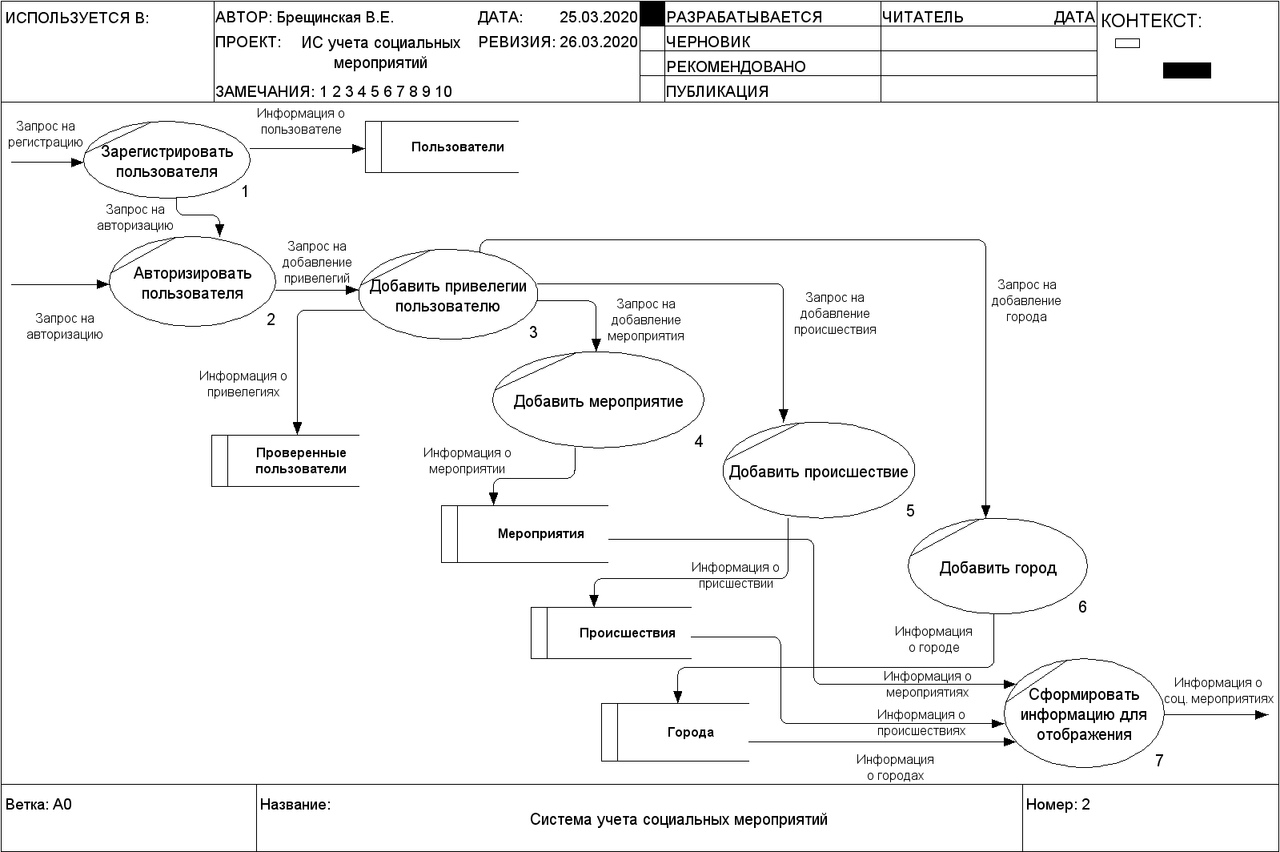
**Информационная система.**

ERдиаграмма



Диаграммы потоков данных



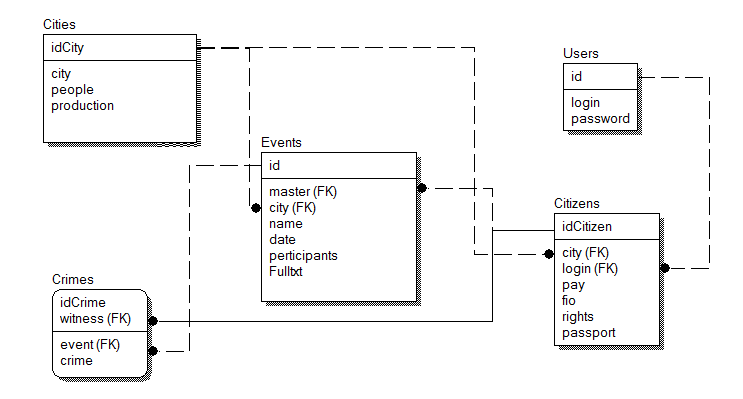


1. **Разработка информационной системы.**

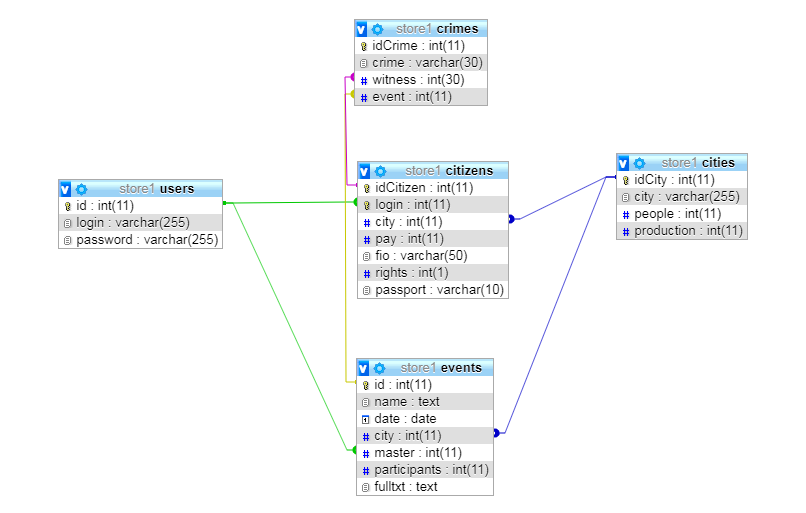
2.1. **Структура информационной системы.**

2.1.1. **Схема базы данных.**

Логическая схема.



Физическая схема.



Описание свойств таблицы «crimes»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Ключевое поле | Тип данных | Обязательное поле |
| idCrime | Да | Числовой | Да |
| crime | Нет | Текстовый | Да |
| witness | Нет | Числовой | Да |
| event | Нет | Числовой | Да |

Описание свойств таблицы «users»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Ключевое поле | Тип данных | Обязательное поле |
| id | Да | Числовой | Да |
| login | Нет | Текстовый | Да |
| password | Нет | Текстовый | Да |

Описание свойств таблицы «citizens»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Ключевое поле | Тип данных | Обязательное поле |
| idCitizen | Да | Числовой | Да |
| login | Нет | Числовой | Да |
| city | Нет | Числовой | Да |
| pay | Нет | Числовой | Да |
| fio | Нет | Текстовый | Да |
| rights | Нет | Числовой | Да |
| passport | Нет | Текстовый | Да |

Описание свойств таблицы «cities»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Ключевое поле | Тип данных | Обязательное поле |
| idCity | Да | Числовой | Да |
| city | Нет | Текстовый | Да |
| people | Нет | Числовой | Да |
| production | Нет | Числовой | Да |

Описание свойств таблицы «events»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Ключевое поле | Тип данных | Обязательное поле |
| id | Да | Числовой | Да |
| name | Нет | Текстовый | Да |
| date | Нет | Дата/время | Да |
| city | Нет | Числовой | Да |
| master | Нет | Числовой | Да |
| participants | Нет | Числовой | Да |
| fulltxt | Нет | Текстовый | Да |

2.1.2 **Запросы.**

Получение списка мероприятий

function askAllEventsFromBD()

"SELECT \* FROM events LEFT JOIN citizens ON events.master=citizens.login LEFT JOIN cities ON events.city=cities.idCity"

Получение списка мероприятий, сортированных по дате \ количеству или обоим полям, dat – нужна сортировка по дате, people – нужна сортировка по кол-ву

function askAllEventsFromBDSort($dat, $people)

"SELECT \* FROM events LEFT JOIN citizens ON events.master=citizens.login LEFT JOIN cities ON events.city=cities.idCity ORDER BY events.date");

"SELECT \* FROM events LEFT JOIN citizens ON events.master=citizens.login LEFT JOIN cities ON events.city=cities.idCity ORDER BY events.participants DESC");

"SELECT \* FROM events LEFT JOIN citizens ON events.master=citizens.login LEFT JOIN cities ON events.city=cities.idCity ORDER BY events.date ASC, events.participants DESC"

Получение данных об конкретном мероприятии по id

function askEventWithIDFromBD($id)

"SELECT \* FROM events WHERE id=?"

Удаление события по id

function deleteEvent($id)

"DELETE FROM events WHERE id = ?")

Добавить запись allowed – поля в которые происходит запись, values – значения полей

function writeNewEvent($allowed, $values)

"INSERT INTO events SET ". $allowed. $values

Изменить запись allowed – поля в которые происходит запись, values – значения полей, по определенному id

function updateEvent($allowed, $values, $id)

"UPDATE events SET ". $allowed. $values." WHERE id=$id"

Релевантный поиск по таблице событий

function findEvents($str)

"SELECT \*, MATCH (fulltxt) AGAINST ('\*$str\*' IN BOOLEAN MODE) as relev FROM events LEFT JOIN citizens ON (events.master=citizens.login and MATCH (fulltxt) AGAINST ('\*$str\*' IN BOOLEAN MODE)>0) LEFT JOIN cities ON events.city=cities.idCity WHERE citizens.fio IS NOT NULL ORDER BY relev DESC"

"SELECT \* FROM events INNER JOIN cities ON events.city=cities.idCity and participants = '$n'"

Выбор списка городов

function askAllCitiesFromBD()

"SELECT \* FROM cities")

Выбор данных о городе по id

function askCityWithIDFromBD($id)

"SELECT \* FROM cities WHERE idCity=?"

Добавить запись allowed – поля в которые происходит запись, values – значения полей

function writeNewCity($allowed, $values)

"INSERT INTO cities SET ".$allowed. $values

Изменить запись allowed – поля в которые происходит запись, values – значения полей, по определенному id

function updateCity($allowed, $values, $id)

"UPDATE cities SET ".$allowed. $values WHERE idCity=$id";

Удаление города по id

function deleteCities($id)

"DELETE FROM cities WHERE idCity = ?"

Релевантный поиск по таблице городов

function findCities($str)

"SELECT \*, MATCH (city) AGAINST ('\*$str\*' IN BOOLEAN MODE) as relev FROM cities WHERE MATCH (city) AGAINST ('\*$str\*' IN BOOLEAN MODE)>0 ORDER BY relev DESC");

"SELECT \* FROM cities WHERE production = '$n' or people = '$n'"

Выбор данных о пользователе по логину

function chekLogin($login)

"SELECT \* FROM users WHERE login = '$login'"

Запись пользователя при регистрации

function writeNewUser($allowed, $values)

"INSERT INTO users SET ".$allowed. $values

Выбор всех данных о происшествиях

function askCrimesFromBD()

"SELECT \* FROM crimes LEFT OUTER JOIN citizens ON crimes.witness=citizens.idCitizen LEFT OUTER JOIN events ON crimes.event=events.id"

Выбор данных о происшествии по id

function askCrimeIDFromBD($id)

"SELECT \* FROM crimes WHERE idCrime=?"

Добавление нового происшествия

function writeNewCrime($allowed, $values)

"INSERT INTO crimes SET ".$allowed. $values

Обновление данных о происшествии

function updateCrime($allowed, $values, $id)

"UPDATE crimes SET ".$allowed. $values." WHERE idCrime=$id"

Удаление происшествия по id

function deleteCrime($id)

"DELETE FROM crimes WHERE idCrime = ?"

Выбор данных обо всех проверенных пользователях по id

function askCitizensFromBD()

"SELECT \* FROM citizens INNER JOIN cities ON citizens.city=cities.idCity INNER JOIN users ON citizens.login=users.id");

Запрос данных о правах проверенного пользователя по id

function askRightsFromBD($id)

"SELECT rights FROM citizens WHERE login = '$id'"

Добавление записи о проверенном пользователе

function writeNewCheckedUser($allowed, $values)

"INSERT INTO citizens SET ".$allowed. $values

Обновление прав проверенного пользователя при подтверждении администратором

function addCheckedUser($allowed, $values, $id)

"UPDATE citizens SET ".pdoSet($allowed)." WHERE idCitizen=$id";

Удаление проверенного пользователя по id

function delCheckedUser($id)

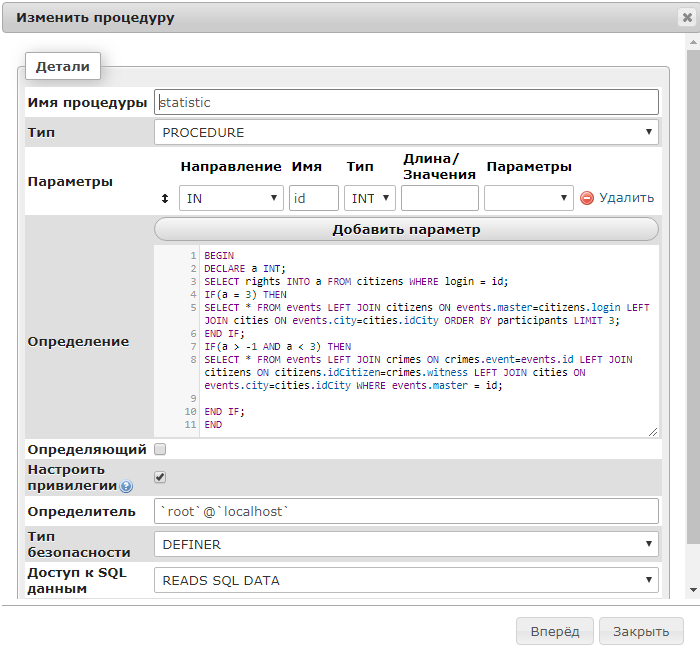
DELETE FROM citizens WHERE idCitizen = ?"

Вызов процедуры по сбору данных о топ-3 мероприятиях или мероприятиях, созданных текущим пользователем, и происшествиях на них.

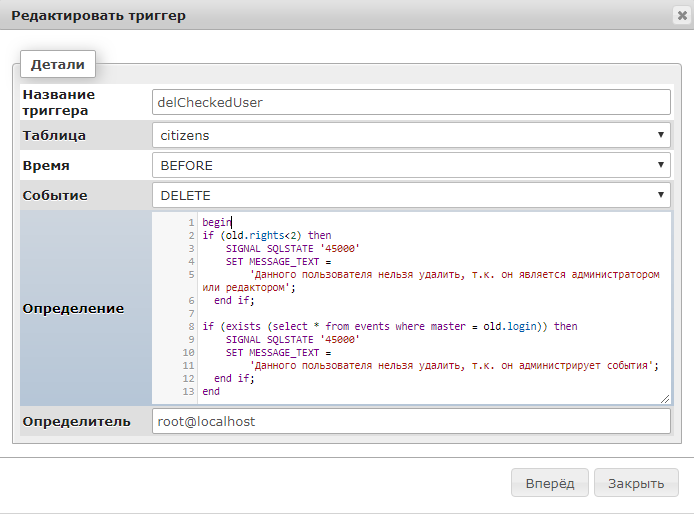
function statistic($id)

"CALL statistic($id)"

Тело:



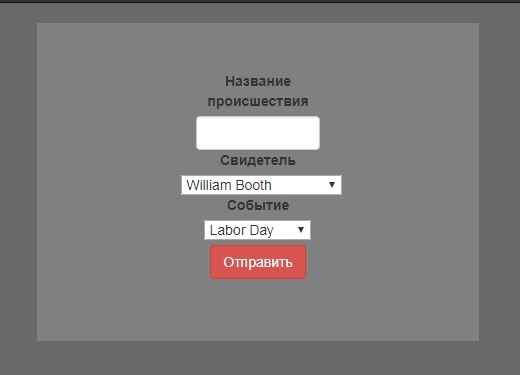
Триггер:



1. **Инфраструктура информационной системы**

3.1**Экранные формы для пользователя**

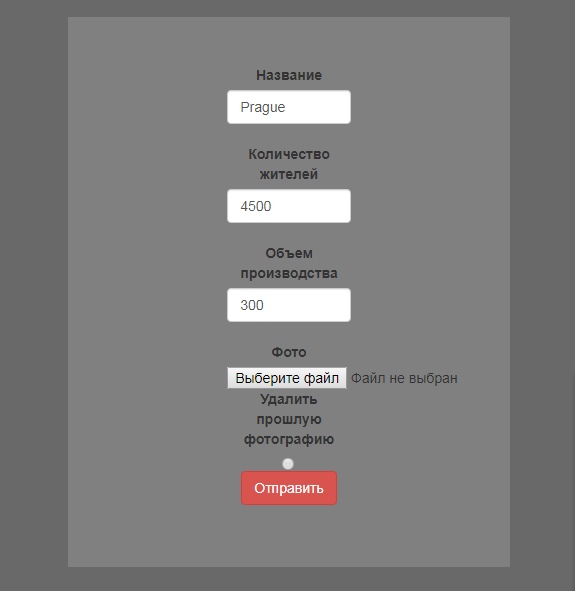
Форма добавления/изменения происшествия



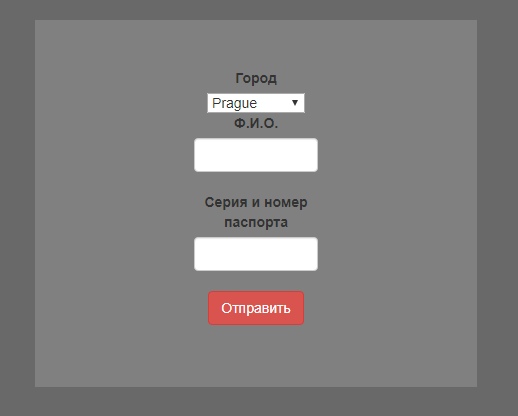
Форма добавления города



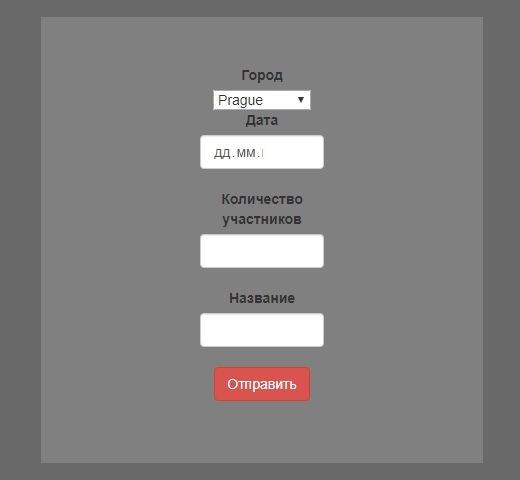
Форма изменения города



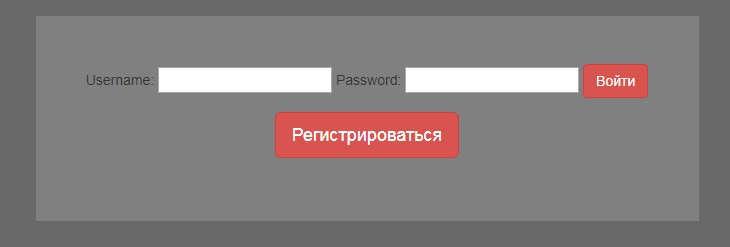
Форма заявки на становление проверенным пользователем



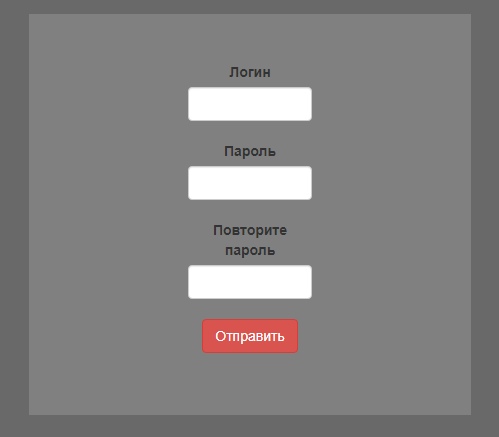
Форма добавления мероприятия



Форма авторизации пользователя

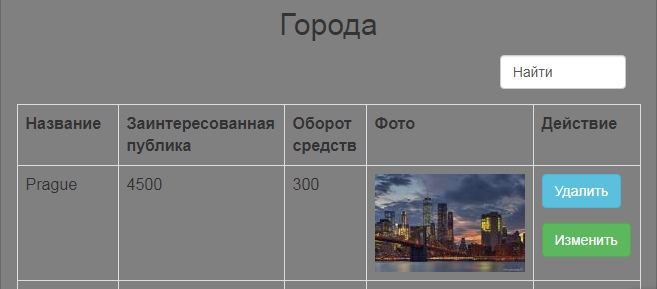


Форма регистрации пользователя



3.2 **Экранные отчеты**

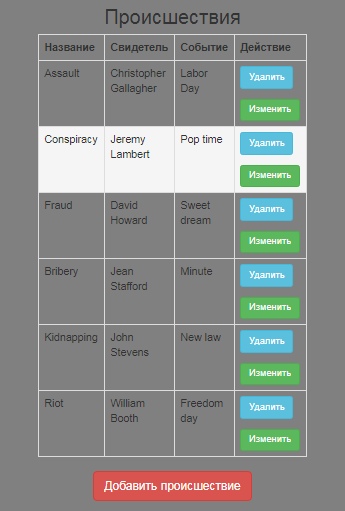
Список городов



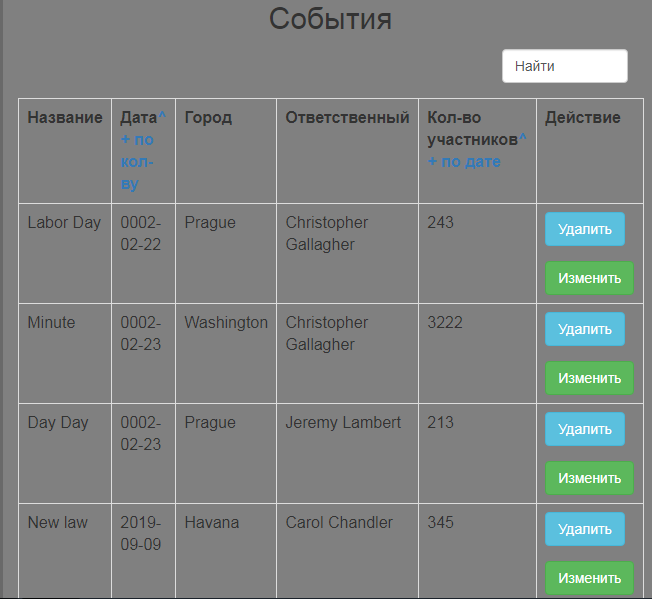
Список проверенных пользователей



Список происшествий



Список событий



Список мероприятий текущего пользователя

