

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
Тема: Сервис для Тайного Санты

Студентка гр. 1381	_____	Васильева О.М.
Студентка гр. 1381	_____	Денисова О.К.
Студентка гр. 1381	_____	Манцева Т.К.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург

2024

ЗАДАНИЕ

Студентка Васильева О.М.

Студентка Денисова О.К.

Студентка Манцева Т.К.

Группа 1381

Тема: Сервис для Тайного Санты

Исходные данные:

Необходимо реализовать веб-приложение сервиса для Тайного Санты с использованием СУБД MongoDB

Содержание пояснительной записки:

«Содержание»

«Введение»

«Сценарии использования»

«Модель данных»

«Разработанное приложение»

«Выводы»

«Приложение»

«Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 05.09.2024

Дата сдачи реферата: 24.12.2024

Дата защиты реферата: 24.12.2024

Студентка	_____	Васильева О.М.
Студентка	_____	Денисова О.К.
Студентка	_____	Манцева Т.К.
Преподаватель	_____	Заславский М.М.

АННОТАЦИЯ

В рамках ИДЗ разработано веб-приложение, представляющее сервис для Тайного Санты. Приложение включает функционал для регистрации команды, просмотра статистики сервиса по командам и игрокам, загрузку данных для игры (чек оплаты подарка, номер трек-кода), редактирования анкеты игроком, осуществления процесса игры.

Для разработки использованы технологии JavaScript, Python, HTML, CSS, СУБД MongoDB.

Найти исходный код можно по ссылке: [secret-santa](#)

SUMMARY

As part of the IDZ, a web application has been developed that provides a service for Secret Santa. The application includes functionality for registering a team, viewing service statistics on teams and players, downloading data for the game (gift payment receipt, track code number), editing the questionnaire by the player, and implementing the game process.

JavaScript, Python, HTML, CSS, and MongoDB DBMS technologies are used for development.

You can find the source code at the link: [secret-santa](#)

СОДЕРЖАНИЕ

1.	ВВЕДЕНИЕ	7
1.1.	Актуальность проблемы	7
1.2.	Постановка задачи	7
1.3.	Предлагаемое решение.....	7
1.4.	Качественные требования к решению	7
2.	СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ	8
2.1.	Макет UI	8
2.2.	Сценарии использования для импорта	15
2.2.1.	Сценарий использования для импорта данных	15
2.2.2.	Сценарий использования для регистрации команды	16
2.2.3.	Сценарий использования «Заполнение анкеты пользователя»	17
2.3.	Сценарии использования для предоставления данных	18
2.3.1.	Сценарий использования «Просмотр действий в системе»	18
2.3.2.	Сценарий использования «Просмотр команд в системе»	19
2.3.3.	Сценарий использования «Просмотр игроков в системе»	19
2.3.4.	Сценарий использования «Просмотр пожеланий получателя»	20
2.3.5.	Сценарий использования «Поиск по пользователям»	20
2.3.6.	Сценарий использования «Поиск по командам»	21
2.4.	Сценарии использования для анализа данных	22
2.4.1.	Сценарий использования «Просмотр статистики сайта»	22
2.4.2.	Сценарий использования «Статистика команды»	22
2.5.	Сценарий использования для экспорта данных	23
2.6.	Вывод	23
3.	МОДЕЛЬ ДАННЫХ	25
3.1.	Нереляционная модель данных	25
3.1.1.	Графическое представление	25
3.1.2.	Описание назначения коллекций, типов данных и сущностей	25
3.1.3.	Оценка объема информации, хранимой в модели	27
3.1.4.	Примеры запросов для совершения сценариев использования	28
3.2.	Реляционная модель данных	30
3.2.1.	Графическое представление	30
3.2.2.	Описание назначения коллекций, типов данных и сущностей	30
3.2.3.	Оценка объема информации, хранимой в модели	32
3.2.4.	Примеры запросов для совершения сценариев использования	33
3.3.	Сравнение моделей	34
3.3.1.	Объем информации	34
3.3.2.	Количество запросов по юзкейсам	34
3.3.3.	Вывод	34
4.	РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ	35
4.1.	Краткое описание	35
4.2.	Использованные технологии	35
4.3.	Схема экранов приложения	35
5.	ВЫВОДЫ	37
5.1.	Достигнутые результаты	37
5.2.	Недостатки и пути для улучшения	37

5.3.	Будущее развитие решения	37
6.	ПРИЛОЖЕНИЕ	39
6.1.	Документация по сборке и развертыванию приложения	39
6.2.	Инструкция для пользователя	39
7.	ЛИТЕРАТУРА	41

1. ВВЕДЕНИЕ

1.1. Актуальность проблемы

Мероприятия вроде «Тайного Санты» становятся всё популярнее, но требуют значительных усилий и времени. Специализированный онлайн-сервис мог бы автоматизировать этот процесс, упрощая координацию участников и управление списками пожеланий. Такое решение позволило бы сэкономить время и силы организаторов, обеспечивая при этом безопасность данных и удобство для всех участников.

1.2. Постановка задачи

Задача проекта заключается в:

- 1) Реализовать сервис Тайного Санты.
- 2) Сделать возможность регистрации команды и осуществления независимого процесса игры для каждой из них.
- 3) Сделать страницу статистики для организатора по своей команде.
- 4) Добавить возможность импорта и экспорта данных.
- 5) Сделать панель администратора, где будет возможность просматривать список команд, список людей, статистику по событиям и командам, список событий в системе.

1.3. Предлагаемое решение

Приложение реализовано с помощью технологий JavaScript, Python, HTML, CSS, СУБД MongoDB.

1.4. Качественные требования к решению

Приложение должно быть простым в использовании, масштабируемым, функциональным (включать базовые функции, такие как регистрация команды, осуществления процесса игры).

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. Макет UI

Ниже представлены страницы макета приложения (рис. 1-12)



Рис. 1 – Главная страница



Регистрация команды

Введите имя организатора	Введите адрес эл. почты организатора
Введите имя игрока	Введите адрес эл. почты игрока

+

Продолжить >>

Регистрация команды



Регистрация команды

Дедлайн заполнения анкеты	Введите дату
Дедлайн отправки чека	Введите дату
Дедлайн отправки подарка	Введите дату

От

Нижний предел

 100 рублей

Верхний предел

 До 1000 рублей

Зарегистрировать

Рис.2 – Страницы регистрации команды



Введите адрес эл. почты

Войти

Импортировать данные

Рис. 3 – Страница входа в профиль

Действия в системе

Ваня заполнил анкету
Костя заполнил анкету
Артем подтвердил чек оплаты игрока Ваня
Маша заполнила анкету
Петя заполнил анкету
Саша отправил чек оплаты на проверку организатору Лена
Лера подтвердила чек оплаты игрока Сеня

....

Вернуться назад

Список команд

1 команда	Организатор Артем
2 команда	Организатор Лера
3 команда	Организатор Лена

....

Вернуться назад

Панель администратора

Список людей

Ваня	1 команда
Костя	1 команда
Лена	3 команда
Маша	2 команда
Петя	2 команда
Саша	3 команда
Света	1 команда

....

Вернуться назад

Список людей

Список команд

Действия в системе

Статистика

Вернуться назад

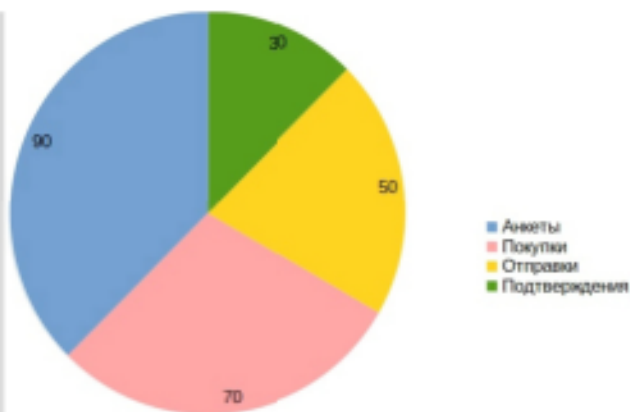
Рис. 4 – Группа страниц панели администратора (список команд, список людей, действия в системе)

Статистика сайта

Тип графика:

- ☐ График по событиям
- ☐ График по командам

Здесь будет располагаться список параметров в зависимости от типа графика (список График по событиям или список График по командам). Для примера выбран график по событиям с параметром 1



[Вернуться назад](#)

График по событиям...

Параметры:

- ☐ Общее кол-во всех событий
- ☐ Кол-во анкет
- ☐ Кол-во отправок
- ☐ Кол-во покупок
- ☐ Кол-во подтверждений

График по командам...

Параметры:

- ☐ Общее кол-во команд - проблемных и нет
- ☐ Команды с различными проблемами
- ☐ Игроки (кол-во отправивших анкету, купивших подарок, отправивших его, подтвердивших получение)

Рис. 5 – Страница статистика сайта (страница для панели администратора)



Экспортировать
данные

Статистика
команды

Данные об игроках

Возможность редактирования столбца "Кому дарит"

Имя игрока	Кому дарит	Анкета	Чек покупки	Отправка подарка
test_user2	3	Анкета заполнена	Загружен	Не отправлен
test_user3	2	Анкета не заполнена	Не загружен	Не отправлен

Сохранить

Рис. 6 – Страница организатора (данные об игроках) с возможностью изменения поля «Кому дарит»



Экспортировать
данные

Статистика
команды

Данные об игроках

Полная
статистика

По анкетам

По чеку
оплаты

По отправке
подарков

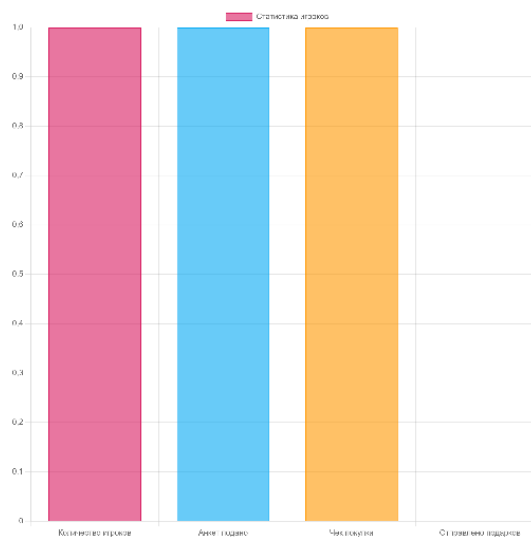


Рис. 7 – Страница статистики команды (для организатора)



Экспортировать данные

Анкета

Текущее состояние

Добро пожаловать в игру "Тайный Санта"!



Дорогой игрок!

Сегодня мы объединились, чтобы создать атмосферу волшебства и подарков. В этом увлекательном приключении ты станешь частью великолепной традиции обмена подарками. Твоя задача — выбрать и подготовить сюрприз для своего "Тайного Санты".

Как это работает:

1. Регистрация: Если ты еще не заполнил анкету, не забудь заполнить её, чтобы Санта знал, как тебя удивить.
2. Выбор подарка: Ознакомься с предпочтениями своего "Клиента" и подготовь настоящий подарок, который принесёт ему радость. Сумма подарка начинается от 100 до 1000 рублей.
3. Обмен: После дня отправки твоего подарка ты получишь трек-код и после уведомления о прибытии подарка в отделение Почты ты сможешь его забрать!



Ты готов испытать волшебство и радость подарков? Давай создадим незабываемые моменты вместе!

Начни своё приключение прямо сейчас!

Счастливого Рождества и веселых праздников!

С любовью,
Команда "Тайный Санта"

Рис. 8 – Страница пользователя

Текущее состояние

Operation SECRET SANTA

Экспортировать данные

Анкета

Текущее состояние

Подтвердить получение подарка

Пожелание получателя

Заполните анкету до 01.11.2024 23:59

Прикрепите чек оплаты подарка для подтверждения до 26.11.2024 23:59

Загрузить

Отправка подарка до 10.12. 23:59

Подтвердить вручение подарка

Ожидайте ваш подарок - Счастливого Нового года!

Ваш код отслеживания: подарок еще не отправлен

Доставка подарка по номеру телефона получателя

Пожелания получателя

Мне хотелось бы получить на Новый год много интересных и вкусных сладостей!

✕

Загрузка трек кода

Выберите тип подарка:

☐ билет
 ☐ сертификат
 ☐ подарок по почте
 ☐ подарок другими сервисами

Введите трек-код посылки

Отправить

Доставка подарка по номеру...

Номер телефона получателя +7(9**)-***-**-**


✕

Алгоритм действий:

1. Нажмите на странице Текущее состояние "Подтверждение вручения подарка".
2. Выберите тип подарка.
3. Если тип подарка - подарок по почте, введите трек-код.
4. Нажмите "Отправить".

Рис. 9 – Страница текущего состояния для пользователя и группа модульных окон данной страницы

13



Адрес электронной почты

Имя

Адрес отправки

Номер телефона

Почтовый индекс

Ваши пожелания для подарка и что дарить не стоит

Рис. 10 – Страница анкеты

Команда_1	test_user1
Команда_2	test_user1
Команда_3	test_user1

Минимальная цена подарка:
 От:
 До:

Максимальная цена подарка:
 От:
 До:

Срок заполнения анкет:
 От:
 До:

Срок отправки чека оплаты:
 От:
 До:

Срок отправки подарка:
 От:
 До:

Рис. 11 – Страница списка команд с диалоговым окном для фильтрации данных

Роль участника	Действие в системе	Тип подарка
<input checked="" type="radio"/> Игрок <input type="radio"/> Организатор	<input checked="" type="checkbox"/> Заполнил анкету <input checked="" type="checkbox"/> Купил подарок <input type="checkbox"/> Отправил подарок	<input checked="" type="radio"/> Билет <input type="radio"/> Сертификат <input type="radio"/> Подарок по почте <input type="radio"/> Подарок другими сервисами
Имя участника		
<input type="text" value="Введите имя игрока"/>		
Адрес получения		
<input type="text" value="Введите адрес получения"/>		
Пожелания участника для подарка		
<input type="text" value="sweets"/>		
Что дарить не следует		
<input type="text" value="Введите предметы из стоп-листа"/>		
<input type="button" value="Cancel"/> <input type="button" value="Применить фильтры"/>		

Рис. 12 – Диалоговое окно на страницы списка людей

2.2. Сценарии использования для импорта

2.2.1. Сценарий использования для импорта данных

Описание:

- Пользователь загружает данные в базу.

Действующие лица:

- Пользователь (авторизованный или неавторизованный).

Основной сценарий:

- Пользователь выбирает данные для импорта и нажимает кнопку “Импортировать”.

Результат основного сценария:

- Данные будут импортированы в базу данных.

Альтернативный сценарий:

- Выбранный файл для импорта не поддерживается.

Результат альтернативного сценария:

- Появляется сообщение об ошибке "Выбран неподдерживаемый формат файла".

2.2.2. Сценарий использования для регистрации команды

Действующие лица:

- Пользователь сайта (неавторизованный пользователь).

Предусловие:

- Пользователь должен находиться на странице "Регистрация команды".

- Пользователь не должен быть авторизован.

Основной сценарий:

- Пользователь открывает страницу "Регистрация команды".
- Пользователь вводит в первой строке имя и адрес электронной почты организатора.
- Пользователь начиная со второй строки вводит в полях имя и адрес электронной почты игроков.
- Пользователь нажимает при необходимости на кнопку плюса, чтобы добавить ещё поля для заполнения информации об игроке.
- Пользователь нажимает на кнопку "Продолжить >>".
- Пользователь переходит на страницу "Продолжение страницы регистрация команды".
- Пользователь вводит даты дедлайнов для игры (дедлайн заполнения анкеты, дедлайн отправки чека, дедлайн отправки подарка).
- Пользователь вводит нижний и верхний предел суммы подарка.
- Пользователь нажимает на кнопку "Зарегистрировать".

Результат основного сценария:

- Пользователь (организатор) переходит на страницу "Данные об игроках (страница организатора)".

Альтернативный сценарий:

- Неправильный формат email.
- Неверное количество участников.
- Некорректное указание дедлайнов.

Результат альтернативного сценария:

- Система уведомляет пользователя об ошибке в введенном email, пользователь корректирует email и снова нажимает "Продолжить >>".
- Система уведомляет пользователя об неверном количестве игроков (добавлен всего один игрок), пользователь добавляет ещё игрока и снова нажимает "Продолжить >>".
- Система уведомляет пользователя о неверной хронологии дат дедлайнов, пользователь корректирует даты и снова нажимает "Зарегистрировать".

2.2.3. Сценарий использования «Заполнение анкеты пользователя»

Действующие лица:

- Авторизованный пользователь (игрок).

Предусловие:

- Пользователь должен являться игроком.
- Пользователь должен находиться на странице "Анкета пользователя".

Основной сценарий:

- Пользователь заполняет поля на странице:
 - Адрес отправки
 - Номер телефона
 - Почтовый индекс
 - Ваши пожелания для подарка и что дарить не стоит

- Пользователь может откорректировать поля Имя и Адрес электронной почты.

- Пользователь нажимает кнопку "Сохранить".

Результат основного сценария.

- Система уведомляет пользователя, что анкета успешно обновлена.

Альтернативный сценарий:

- Не все поля заполнены.

- Неверная длина номера телефона.

Результат альтернативного сценария:

- Система уведомляет пользователя, что не все поля анкеты заполнены, пользователь может заполнить все оставшиеся поля и повторно нажать на кнопку "Сохранить".

- Система уведомляет пользователя, что длина номера телефона должна быть 11 цифр, пользователь может корректно заполнить поле "Номер телефона" и повторно нажать кнопку "Сохранить".

2.3. Сценарии использования для предоставления данных

2.3.1. Сценарий использования «Просмотр действий в системе»

Действующие лица:

- Пользователь сайта (неавторизованный пользователь).

Основной сценарий:

- Пользователь просматривает действия, происходящие в системе.

- Пользователь нажимает на кнопку "Вернуться назад", его перенаправляет на страницу "Страница администратора".

Результат основного сценария.

- Пользователь узнает информацию о действиях игроков.

2.3.2. Сценарий использования «Просмотр команд в системе»

Действующие лица:

- Пользователь сайта (неавторизированный пользователь).

Основной сценарий:

- Пользователь нажимает на любую строчку списка.
- Открывается модульное окно "Профиль команды".
- Пользователь прочитывает информацию и закрывает окно, нажатием

на крестик.

- Пользователь нажимает на кнопку "Вернуться назад", его перенаправляет на страницу "Страница администратора".

Результат основного сценария.

- Пользователь узнает информацию о любой зарегистрированной команде системы:
 - Информация о команде
 - Дедлайны команды
 - Сумма подарка

2.3.3. Сценарий использования «Просмотр игроков в системе»

Действующие лица:

- Пользователь сайта (неавторизированный пользователь).

Основной сценарий:

- Пользователь нажимает на любую строчку списка.
- Открывается модульное окно "Профиль пользователя".
- Пользователь прочитывает информацию и закрывает окно, нажатием

на крестик.

- Пользователь нажимает на кнопку "Вернуться назад", его перенаправляет на страницу "Страница администратора".

Результат основного сценария.

- Пользователь узнает информацию о любом зарегистрированном пользователе системы:

- Анкета пользователя
- Информация о команде, в которой пользователь состоит
- Информация по операции Тайный Санта (если профиль имеет роль "организатор", то данный блок отсутствует)

2.3.4. Сценарий использования «Просмотр пожеланий получателя»

Действующие лица:

- Авторизованный пользователь (игрок).

Предусловие:

- Пользователь должен являться игроком.
- Пользователь должен находиться на странице "Текущее состояние".

Основной сценарий:

- Пользователь нажимает на кнопку "Пожелания получателя".
- Пользователю открывается модульное окно "Пожелания получателя".
- Пользователь после прочтения текста нажимает крестик.

Результат основного сценария.

- Пользователь узнает пожелания получателя для покупки подарка.

Альтернативный сценарий:

- Кнопка неактивна.

Результат альтернативного сценария:

- Система уведомляет пользователя, что дедлайн заполнения анкет не прошел.

2.3.5. Сценарий использования «Поиск по пользователям»

Действующие лица:

- пользователь сайта (неавторизованный пользователь)

Основной сценарий:

- Пользователь выбирает роль участника (Игрок/Организатор)
- Если пользователь выбрал роль «Игрок», то ему становится доступен выбор полей «Действия в системе», «Тип подарка», «Имя участника», «Адрес получения», «Пожелания участника для подарка», «Что дарить не следует»
- Если пользователь выбрал роль «Организатор», то остальные поля фильтрации остаются заблокированными
- Пользователь вводит в доступные текстовые поля значения, в доступных элементах выбора (checkbox, radio) выбирает желаемые опции. Если пользователь оставил поле пустым, то фильтрация по этому полю не производится
- Пользователь нажимает «Применить фильтры» и узнает информацию в соответствии с заданными значениями

Результат основного сценария:

- Пользователь узнает информацию в соответствии с заданными в полях фильтрации значениями

2.3.6. Сценарий использования «Поиск по командам»

Действующие лица:

- Пользователь сайта (неавторизованный пользователь)

Основной сценарий:

- Пользователь вбивает целочисленные значения в поля «Минимальная цена подарка, от», «Минимальная цена подарка, до», «Максимальная цена подарка, от», «Максимальная цена подарка, до». Пользователь выбирает значения дат в полях «Срок заполнения анкет, от», «Срок заполнения анкет, до», «Срок отправки чека оплаты, от», «Срок отправки чека оплаты, до», «Срок отправки подарка, от», «Срок отправки подарка, до».

- Если пользователь оставляет какие-то поля незаполненными, то фильтрация по этим полям не производится

- Пользователь нажимает «Применить фильтры» и узнает информацию в соответствии с заданными значениями

Результат основного сценария:

- Пользователь узнает информацию в соответствии с заданными в полях фильтрации значениями

2.4. Сценарии использования для анализа данных

2.4.1. Сценарий использования «Просмотр статистики сайта»

Действующие лица:

- Пользователь сайта (неавторизованный пользователь).

Основной сценарий:

- Пользователь выбирает тип графика:
 - Или график с информацией о событиях
 - Или график с информацией о командах
 - Открывается окно с выбором различных графиков, доступных соответствующим типам, которые нужно отобразить на странице.
- Пользователь выбирает необходимую информацию.

Результат основного сценария.

- Пользователь узнаёт статистику о событиях или об игроках в системе.

2.4.2. Сценарий использования «Статистика команды»

Действующие лица:

- Авторизованный пользователь (организатор).

Предусловие:

- Пользователь должен являться организатором.

Основной сценарий:

- Пользователь выбирает один из четырех параметров:
 - Вся статистика
 - По анкетам
 - По чекам оплаты
 - По отправке подарков

Результат основного сценария:

- Строится статистика по заданному параметру.

2.5. Сценарий использования для экспорта данных

Действующие лица:

- Пользователь авторизованный.

Предусловие:

- Если пользователь является организатором, он должен находиться на любой из страниц: "Данные об игроках (страница организатора)", "Статистика команды".
- Если пользователь является игроком, он должен находиться на любой из страниц: "Страница пользователя", "Анкета", "Текущее состояние".

Основной сценарий:

- Пользователь нажимает на кнопку "Экспортировать данные".

Результат основного сценария:

- Данные будут скачаны на устройство.

2.6 Вывод

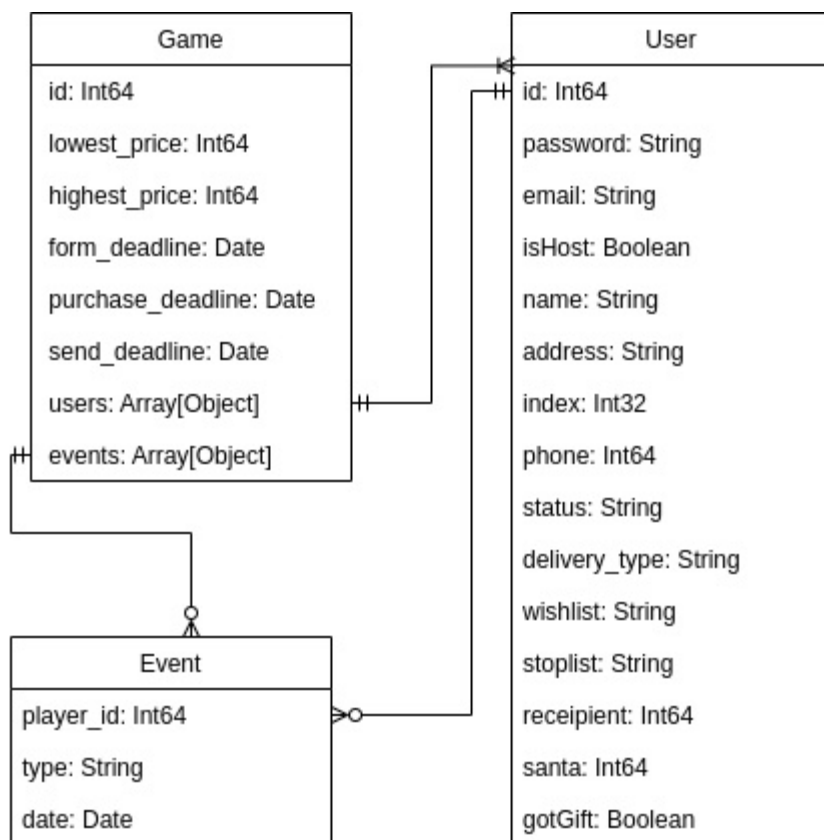
Для нашего решения преобладают операции записи и чтения данных в системе. Пользователь (организатор) регистрирует команды, выполняет активные действия подтверждения данных и просмотра данных своей команды. Пользователя (игрок) добавляет данные в анкету. В панели администратора в

равной степени осуществляется анализ и просмотр данных (поиск данных и просмотр статистики).

3. МОДЕЛЬ ДАННЫХ

3.1. Нереляционная модель данных

3.1.1. Графическое представление



3.1.2. Описание назначения коллекций, типов данных и сущностей

БД содержит 3 коллекции - "Game", "User", "Event":

- Коллекция "Game"
 - id - идентификатор каждой команды/игры в Тайного Санту. 8 байт
 - lowest_price - нижняя граница стоимости подарка. 8 байт
 - highest_price - верхняя граница стоимости подарка. 8 байт
 - form_deadline - срок, к которому нужно заполнить анкету. 8 байт
 - purchase_deadline - срок, к которому нужно купить подарок. 8 байт

- send_deadline - срок, к которому нужно отправить подарок. 8 байт
- created_at - дата создания. 8 байт
- updated_at - дата обновления. 8 байт
- users - игроки данной команды, в том числе и организатор.
- events - события, относящиеся к этой игре (заполнение анкеты, покупка подарка и т.д.).
- Коллекция "User"
 - id - пользователь. 8 байт
 - password - пароль. $L_password * 1 = 20$ байт
 - email - электронная почта. $L_email * 1 = 20$ байт
 - is_host - является ли организатором. 1 байт
 - name - имя. $L_name * 1 = 20$ байт
 - address - адрес. $L_address * 1 = 60$ байт
 - index - индекс. 4 байта
 - phone - телефон. 8 байт
 - status - статус игрока(ничего не сделал/ заполнил анкету/ купил подарок/ отправил его). 4 байта (none, form, paid, sent)
 - delivery_type - способ доставки подарка. 6 байт (mail, other, ticket, cert, none)
 - wishlist - желания игрока. $L_wishlist * 1 = 200$ байт
 - stoplist - "стоплист". $L_stoplist * 1 = 100$ байт
 - recipient - идентификатор получателя. 8 байт
 - santa - идентификатор того, кто будет "Тайным Сантой" данного игрока. 8 байт
 - got_gift - получил ли пользователь подарок. 1 байт
- Коллекция "Event"

- `player_id` - идентификатор игрока, совершившего определенное действие 8 байт
- `type` - тип события 4 байта
- `date` - дата 8 байт

3.1.3. Оценка объема информации, хранимой в модели

Далее используемые обозначения для расчета среднего объёма

1. $N_{players} = 15$ - количество игроков в одной команде без организатора игры
2. $L_{name} = 20$
3. $L_{address} = 60$
4. $L_{pass} = 20$
5. $L_{email} = 40$
6. $L_{wishlist} = 200$
7. $L_{stoplist} = 100$
8. N_{games} - количество команд, переменная

Значения для расчетов объема прописаны в графическом представлении модели.

Для коллекции Game: $V_{game} = 8844$ байт

Для поля users: $N_{players} * V_{user} + V_{host} + N_{players} * 8 = 7480 +$

Верхняя граница(произошли все события для каждого игрока): $5 N_{players} * V_{event} = 1300$ байт

Для коллекции User: $V_{user} = 468$ байта

Частный случай пользователя организатора – $V_{host} = V_{id} + V_{pass} + V_{email} + V_{name} + V_{ishost} + V_{createdat} + V_{updatedat} = 85$ байт

Для коллекции Event: $V_{event} = 20$ байт

$V(N_{games}) = 8879 * N_{games}$ - занимаемый объём.

Занимаемый объём зависит от количества игр, т.е. от количества команд.

3.1.4. Примеры запросов для совершения сценариев использования

1. Запрос на отображение списка желаний пользователя по id

```
users.find_one({'id': '1'}).get('wishlist')
```

2. Запрос на обновление анкеты

```
users.update_one({'id': 1}, {
    '$set':
    {
        'wishlist': "flowers",
        'stoplist': "candles",
        'phone': 8999999999999999,
        'index': 111111,
        'address': "nevsky str.1"
    }
})
```

3. Запрос на отображение участников конкретной игры

```
for player_id in games.find_one({'id': 1}).get('users'):
    print(users.find_one({'id': player_id}))
```

4. Запрос на отображение участников, чью покупку админ посчитал пересекающейся со стоплистом

```
game_column = db["games"]
player_column = db["users"]
```

```

wrong = []
for game in game_column.find():
    for player in player_column.find({ "id": { "$in":
game["players"]} }):
        if player["wrong_gift"]:
            wrong.append(player)

```

5. Запрос на отображение всех игр, где есть участники, просрочившие дедлайн покупки подарка

```

game_column = db["games"]
player_column = db["users"]
expired = []
current_date = dateutil.utils.today()
enum = ["none", "form", "paid", "sent"]
for game in game_column.find():
    if game["form_deadline"] < current_date:
        for player in player_column.find({ "id": { "$in":
game["players"]} }):
            if not player["is_host"] and enum.index(player["status"]) < 2:
                expired.append(game)
                break

```

6. Запрос на отображение участников конкретной игры, ещё не купивших подарок

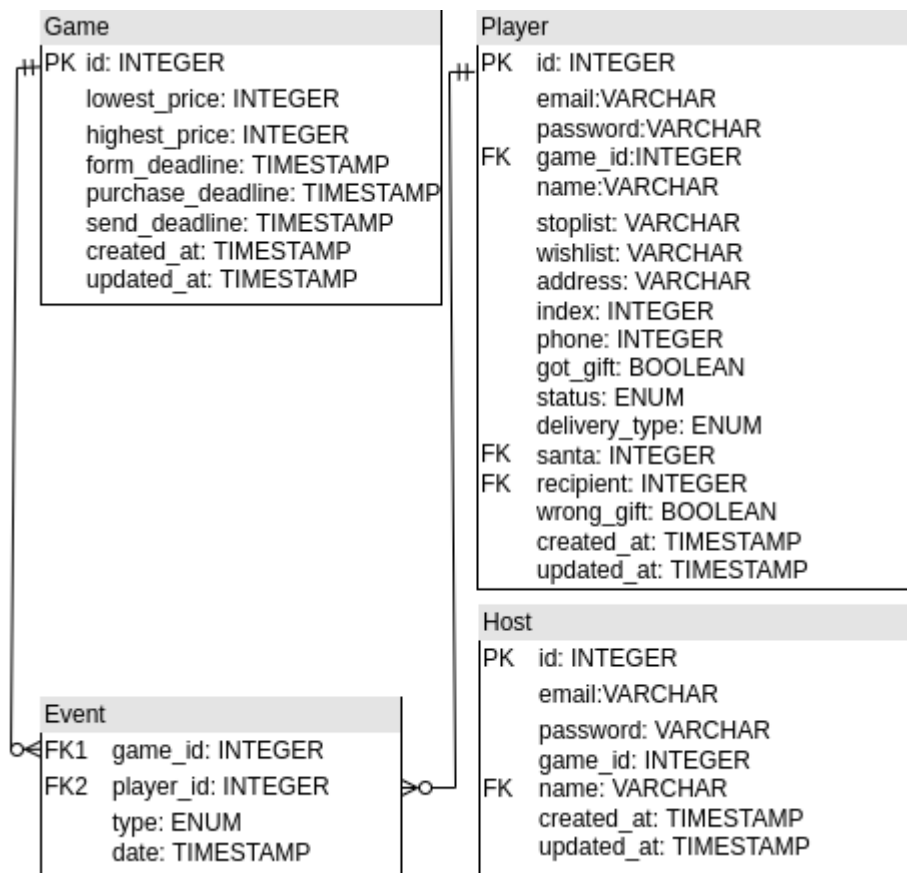
```

game = db["games"].find_one({ "id": { "$in": [game_id]}})
player_column = db["users"]
late = []
enum = ["none", "form", "paid", "sent"]
for player in player_column.find({ "id": { "$in":
game["players"]} }):
    if not player["is_host"] and enum.index(player["status"]) < 2:
        late.append(player)

```

3.2. Реляционная модель данных

3.2.1. Графическое представление



3.2.2. Описание назначения коллекций, типов данных и сущностей

БД содержит 4 отношения - "Game", "Host", "Player", "Event":

- Отношение "Game"
 - id - идентификатор каждой команды/игры в Тайного Санту. **Ключ 8 байт**
 - lowest_price - нижняя граница стоимости подарка 8 байт
 - highest_price - верхняя граница стоимости подарка 8 байт
 - form_deadline - срок, к которому нужно заполнить анкету 8 байт
 - purchase_deadline - срок, к которому нужно купить подарок 8 байт
 - send_deadline - срок, к которому нужно отправить подарок 8 байт
 - created_at - дата создания. 8 байт
 - updated_at - дата обновления. 8 байт

- Отношение "Host"
 - id - идентификатор организатора. **Ключ** 8 байт
 - password - пароль $L_{p a s s} * 1 = 20$ байт
 - email - электронная почта $L_{e m a i l} * 1 = 40$ байт
 - game_id - идентификатор игры, в которой данный пользователь является организатором 8 байт
 - name - имя $L_{n a m e} * 1 = 20$ байт
 - created_at - дата создания. 8 байт
 - updated_at - дата обновления. 8 байт
- Отношение "Player"
 - id - идентификатор игрока. **Ключ** 8 байт
 - password - пароль $\$L_{\{pass\}} * 1 = 20 \$$ байт
 - email - электронная почта $L_{e m a i l} * 1 = 40$ байт
 - game_id - идентификатор команды, в которой находится пользователь 8 байт
 - name - имя $L_{n a m e} * 1 = 20$ байт
 - wishlist - желания игрока $L_{w i s h l i s t} * 1 = 200$ байт
 - stoplist - "стоплист" $L_{s t o p l i s t} * 1 = 100$ байт
 - address - адрес $L_{a d d r e s s} * 1 = 60$ байт
 - index - индекс 4 байта
 - phone - телефон 8 байт
 - got_gift - получил ли пользователь подарок 1 байт
 - status - статус игрока(ничего не сделал/ заполнил анкету/ купил подарок/ отправил его) 1 байт
 - delivery_type - способ доставки подарка 1 байт
 - recipient - идентификатор получателя 8 байт
 - santa - идентификатор игрока, который будет "Тайным Сантой"данного игрока 8 байт

- wrong_gift - купил ли игрок подарок, пересекающийся со стоплистом 1 байт
- created_at - дата создания. 8 байт
- updated_at - дата обновления. 8 байт
- Отношение "Event"
 - game_id - идентификатор игры, к которой относится данное событие. 8 байт
 - player_id - идентификатор игрока, совершившего определенное действие. 8 байт
 - type - тип события. **game_id + player_id + type - ключ данной таблицы** 4 байт
 - date - дата 8 байт

3.2.3. Оценка объема информации, хранимой в модели

Далее используемые обозначения для расчета среднего объёма из п. 3.1.3.

Значения для расчетов объема прописаны в графическом представлении модели.

Для отношения Game: $V_{game} = 64$ байта

Для отношения Host: $V_{host} = 112$ байт

Для отношения Player: $V_{player} = 504$ байт

Для отношения Event: $V_{event} = 28$ байт

$V(N_{games}) = 48 * N_{games} + 479 * N_{players} * N_{games} + 96 * N_{games} + N_{players} * N_{games} * 5 = N_{games} * (64 + 504 * 15 + 112 + 15 * 5 * 28) = N_{games} * 9556$ байт - занимаемый объём.

Занимаемый объем зависит от количества игр, т.е. от количества команд.

3.2.4. Примеры запросов для совершения сценариев использования

1. Запрос на отображение списка желаний пользователя по id

```
SELECT wishlist
FROM players
WHERE id=1
```

2. Запрос на обновление анкеты

```
UPDATE users
SET wishlist = "flowers", stoplist = "sweets", phone=8999999999999999,
index=111111, address="nevsky str.1"
WHERE id = 1
```

3. Запрос на отображение участников конкретной игры

```
SELECT *
FROM players
WHERE game_id=1
```

4. Запрос на отображение участников, чью покупку админ посчитал пересекающейся со стоплистом

```
SELECT player_id FROM player
WHERE wrong_gift = TRUE;
```

5. Запрос на отображение всех игр, где есть участники, просрочившие дедлайн покупки подарка

```
SELECT game.game_id
FROM game
INNER JOIN player
ON game.game_id = player.game_id
WHERE status < 'bought' AND purchase_deadline < NOW()
GROUP BY game.game_id;
```

6. Запрос на отображение участников конкретной игры, ещё не купивших подарок

```
FROM player
INNER JOIN game
ON game.game_id = player.game_id
WHERE game.game_id = 2 AND status < 'bought' AND purchase_deadline <
NOW()
GROUP BY player.player_id;
```

3.3. Сравнение моделей

3.3.1. Объем информации

Объём, зависящий от количества зарегистрированных команд в нереляционной БД немного меньше, чем в реляционной - $8844 * N_{games}$ и $9556 * N_{games}$

3.3.2. Количество запросов по юзкейсам

Для всех сценариев использования по двум моделями используется 1 запрос и 1 коллекция. Количество запросов различно только для импорта/экспорта, входа в систему и регистрации команды. Для нереляционной БД используется 3 запроса и три коллекции, а для реляционной 4 запроса и 4 коллекции для сценария использования импорт/экспорт. Для регистрации команды 2 запроса и 2 коллекции для нереляционной БД, 3 запроса и 3 коллекции для реляционной. Для входа в систему 1 запрос и 1 коллекция для нереляционной БД, 2 запроса и 2 коллекции для реляционной БД.

3.3.3 Вывод

Благодаря меньшему объёму, количеству коллекций, чуть меньшему количеству запросов, производительности, количеству запросов, в данном проекте немного предпочтительнее использовать нереляционную БД. Но, тем не менее, писать запросы со сложной фильтрацией намного легче именно в реляционной БД.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

4.1. Краткое описание

Весь код приложения разделен на две части: back-end и front-end.

На части back-end развернут сервер с использованием фреймворка Flask, а также база данных приложения, основанная на MongoDB, сервер обеспечивает маршрутизацию по сайту, ловит запросы со стороны клиента, высылает ответы на запросы, обращается к базе данных.

На части front-end реализована пользовательская логика при помощи JavaScript.

4.2. Используемые технологии

БД: MongoDB.

Back-end: Python, Flask

Front-end: HTML, CSS, JavaScript.

4.3. Схема экранов приложения

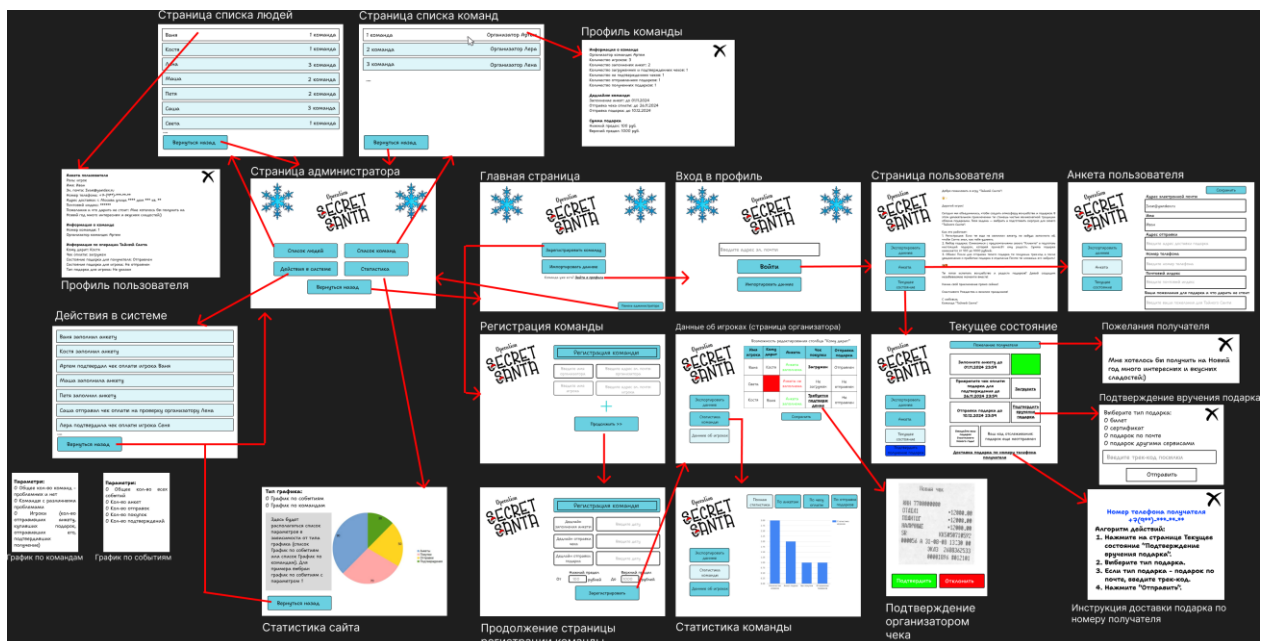


Рис. 11 – Схема экранов приложения

5. ВЫВОДЫ

5.1. Достигнутые результаты

В ходе работы было разработано веб-приложение Тайный Санта, представляющее собой сервис для создания игры обмена подарками для независимых команд. Приложение включает в себя такие возможности: регистрация команды, контролирование всего процесса игры организатором и удобное хранение всей информации для игры (организатор имеет всю информацию по совершению игры в сервисе).

Для пользователя (игрока) имеется возможность заполнения анкеты и контролирования всех сроков процесса игры (заполнение анкеты, отправка чека, отправка подарка).

Также в приложении есть возможность отслеживать действия всех игроков администратором. Возможен просмотр статистики сайта по командам и событиям.

5.2. Недостатки и пути для улучшения

Отсутствует возможность играть индивидуально, есть возможность играть только заранее сформированной командой, которую регистрирует организатор команды.

Нет возможности добавить или удалить людей из команды после того, как она уже зарегистрирована.

Нет возможности отфильтровать организаторов по имени, фильтрация по имени доступна только у роли «Игрок».

5.3. Будущее развитие решения

В качестве будущего развития решения рассматривается добавление возможности индивидуальной игры, добавление возможности изменять состав

команды после регистрации, добавление новых возможностей фильтрации, добавление контроля трек-номера посылки подарка через API сервисов доставки.

6. ПРИЛОЖЕНИЕ

6.1. Документация по сборке и разворачиванию приложения

Нужно скачать докер, клонировать репозиторий, зайти в папку с ним:

```
docker-compose build --no-cache  
docker-compose up
```

Скачивание образов:

```
docker pull mongo:7.0.15  
docker pull ubuntu:22.04
```

6.2. Инструкция для пользователя

Чтобы зайти в аккаунт и управлять состоянием профиля, необходимо пройти регистрацию. Регистрацию команды осуществляет организатор команды.

Если вы хотите зарегистрировать команду, необходимо выбрать «Зарегистрировать команду», далее необходимо ввести информацию об участниках команды.

Если ваша команда уже зарегистрирована и вы хотите авторизоваться, следует выбрать «Зайти в профиль», после чего необходимо ввести email, с которым вас зарегистрировал организатор вашей команды, после чего нажать «Войти».

Чтобы заполнить данные в анкете, в том числе информацию о пожеланиях по подарку, необходимо после авторизации выбрать «Анкета» и обновить информацию в анкете.

Чтобы просмотреть текущее состояние, необходимо после авторизации выбрать «Текущее состояние».

Чтобы экспортировать данные, необходимо после авторизации выбрать «Экспортировать данные».

Чтобы просмотреть статистику сайта, необходимо с главной страницы сайта перейти на «Панель администратора», а далее выбрать одну из опций

«Список людей», «Список команд», «Статистика», «Действия в системе». На страницах «Список людей» и «Список команд» есть возможность для фильтрации по различным параметрам. На странице «Статистика» есть возможность выбрать одну из опций «График по событиям», «График по командам», в каждой из которых также есть возможность вывести различные графики на ваш выбор.

7. ЛИТЕРАТУРА

1. Ссылка на GitHub. – [Электронный ресурс]. – URL: <https://github.com/moevm/nosql2h24-secret-santa>
2. Ссылка на ресурс по Mongo db – [Электронный ресурс]. – URL: <https://itgap.ru/post/vvedenie-v-mongodb?ysclid=m5264jbvaj433331572>
3. Ссылка на документацию по Flask – [Электронный ресурс] – URL: <https://docs.yandex.ru/docs/view?tm=1735027211&tld=ru&lang=en&name=Flask.pdf&text=flask%20python%20%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F&url=https%3A%2F%2Ftheswissbay.ch%2Fpdf%2FBooks%2FComputer%2520science%2Fflask%2FFlask.pdf&lr=165868&mime=pdf&l10n=ru&sign=87b6461b1e51feb0691d39d66f2abda1&keyno=0&nosw=1&serpParams=tm%3D1735027211%26tld%3Dru%26lang%3Den%26name%3DFlask.pdf%26text%3Dflask%2Bpython%2B%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F%26url%3Dhttps%253A%2F%2Ftheswissbay.ch%2Fpdf%2FBooks%2FComputer%2520science%2Fflask%2FFlask.pdf%26lr%3D165868%26mime%3Dpdf%26l10n%3Dru%26sign%3D87b6461b1e51feb0691d39d66f2abda1%26keyno%3D0%26nosw%3D1>