|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий |
|  |
| Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ) |

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ**  **ПО ПРАКТИЧЕСКИМ РАБОТАМ 5-8** | | |
| **по дисциплине** «Системная и программная инженерия» | | |
| **по теме** «Веб-приложение по организации спортивных мероприятий» | | |
|  | | |
| Выполнили студенты группы ИКБО-20-21 | Мухаметшин А. Р.  Квашнин Ю. В.  Хитров Н. С.  Емельянов И. В. | |
| Принял преподаватель | Дворникова Е.М. | |
|  |  | |
|  | | |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_24 г. | *(подпись студента)* |
|  | | |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_24 г. | *(подпись руководителя)* |

# СОДЕРЖАНИЕ

[5 СОЗДАНИЕ СТРУКТУРНОЙ ДИАГРАММЫ СИСТЕМЫ 3](#_Toc162023804)

[5.1. Задание 3](#_Toc162023805)

[5.2 Структурная диаграмма 3](#_Toc162023806)

[5.2.1. Диаграмма классов 3](#_Toc162023807)

[5.2.2. Диаграмма объектов 4](#_Toc162023808)

[5.3. Построение процессов проекта в нотациях IDEF0, BPMN 2.0, EPC 5](#_Toc162023809)

[5.3.1. Нотация IDEF0 6](#_Toc162023810)

[6 СОЗДАНИЕ ИНФОРМАЦИОННОЙ ДИАГРАММЫ СИСТЕМЫ И ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ 10](#_Toc162023811)

[6.1 Задание 10](#_Toc162023812)

[6.2 Диаграмма в нотации DFD 10](#_Toc162023813)

[6.3 Модель базы данных 13](#_Toc162023814)

[6.3.1 Различия между логической и физической моделью БД 13](#_Toc162023815)

[6.3.2 Построение нормализованной логической БД 15](#_Toc162023816)

[7 АРХИТЕКТУРА СИСТЕМЫ 16](#_Toc162023817)

[7.1 Задание 16](#_Toc162023818)

[7.2 Архитектура системы 16](#_Toc162023819)

[7.3 Дополнение матрицы требований 18](#_Toc162023820)

[8 ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ 25](#_Toc162023821)

[8.1. Задание 25](#_Toc162023822)

[8.2 Выбор ГОСТа 25](#_Toc162023823)

[8.3 Техническое задание 26](#_Toc162023824)

[ЗАКЛЮЧЕНИЕ 46](#_Toc162023825)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 47](#_Toc162023826)

# 5 СОЗДАНИЕ СТРУКТУРНОЙ ДИАГРАММЫ СИСТЕМЫ

## 5.1. Задание

1.Построить структурные диаграммы своего проекта.

1.1.Диаграмма классов.

1.2.Диаграмма объектов.

2.Разобрать процесс работы внутри проекта в нотации по выбору, IDEF0 или BPMN 2.0. Построить собственный процесс в выбранной нотации.

## 5.2 Структурная диаграмма

Основное назначение структурных диаграмм заключается в графическом представлении состава статистических совокупностей, характеризующихся как соотношение различных частей каждой из совокупностей.

Структурная диаграмма – это инструмент модульного дизайна сверху вниз, построенный из квадратов, представляющих различные модули в системе и соединяющие их линии. Линии представляют связь и / или право собственности между видами деятельности и вспомогательными видами деятельности, как они используются в организационных диаграммах.

Структурная диаграмма отображает:

1. размер и сложность системы;
2. количество легко идентифицируемых функций и модулей в каждой функции;
3. является ли каждая идентифицируемая функция управляемым объектом или должна быть разбита на более мелкие компоненты.

Существует три основных типа структурных диаграмм:

1. Диаграмма классов.
2. Диаграмма развертывания.
3. Диаграмма объектов.

### 5.2.1. Диаграмма классов

Диаграмма классов (рис. 5.1) – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

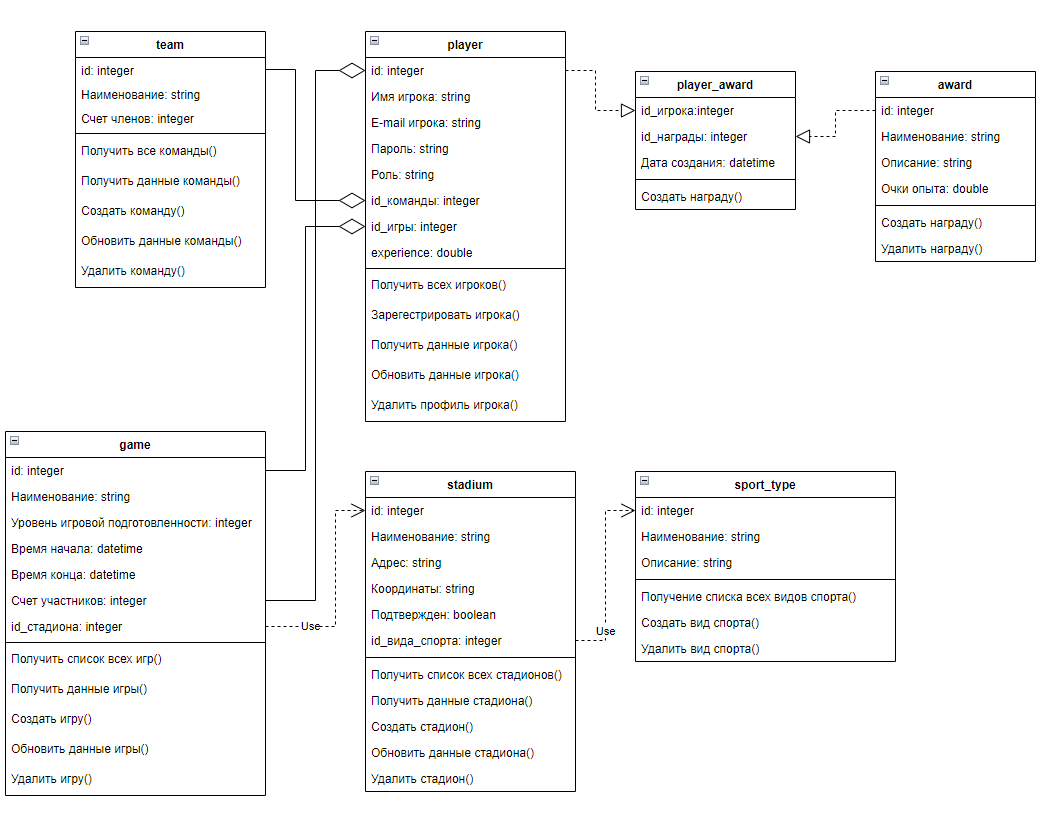


Рисунок 5.1 Диаграмма классов

### 5.2.2. Диаграмма объектов

Диаграмма объектов в языке моделирования UML предназначена для демонстрации совокупности моделируемых объектов и связей между ними в фиксированный момент времени.

Диаграмма объектов (рис. 5.2) описывает конкретные экземпляры объектов и напрямую соотносится с диаграммой классов, которая дает общее представление о конфигурации системы. Она используется для документирования структур данных и создания статических снимков состояний объектов принимая во внимание реальные экземпляры или прототипы. Динамику поведения объектов обычно изображают в виде последовательности таких диаграмм.

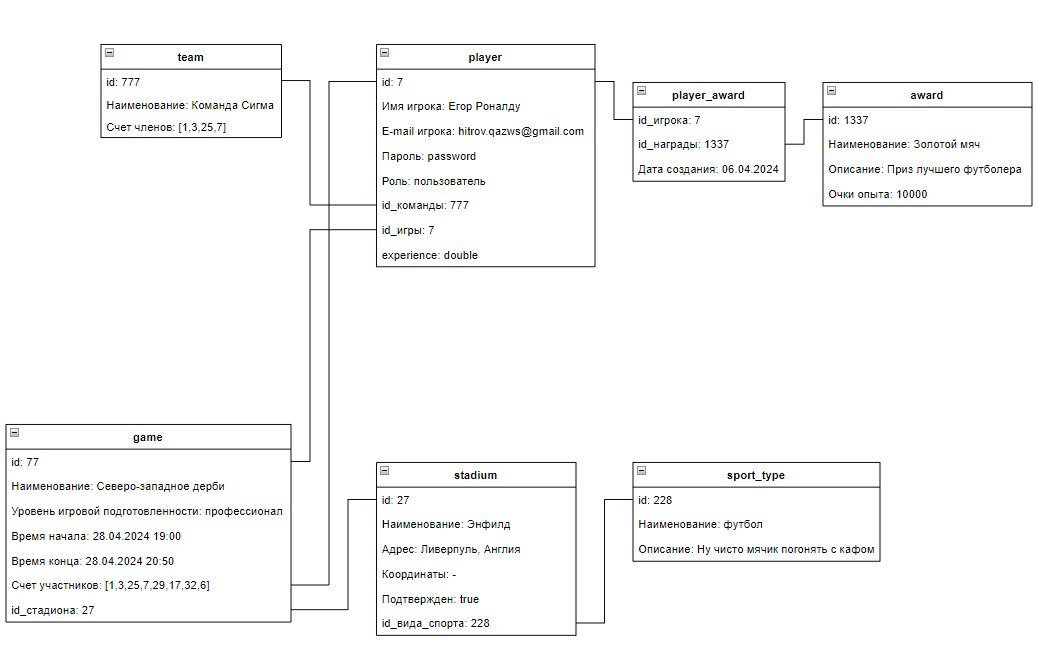


Рисунок 5.2 Диаграмма объектов

## 5.3. Построение процессов проекта в нотациях IDEF0, BPMN 2.0, EPC

Построение процессов выполняются посредством построения диаграмм в различных нотациях. Нотации – графические модели, которые используются, чтобы фиксировать бизнес-процессы, анализировать их и оптимизировать. По сравнению с текстовыми описаниям, графические модели занимают меньше места, помогают увидеть алгоритм наглядно, представить, как он проходит от начала до конца. Однако, в отличие от текстового описания, графическая модель хуже передает детали.

Нотации применяются, чтобы сотрудники могли понять и запомнить схему, по которой они должны, к примеру, обрабатывать заявку на поставку партии товара. А руководителю схема будет полезна, чтобы он мог найти проблемные или избыточные элементы (этапы, сотрудников), внести нужные корректировки. Часто это помогает ускорить или удешевить работу компании.

По аналогии с языками программирования, нотации называют языками моделирования бизнес-процессов.

Сегодня в мире наиболее популярны 3 нотации:

1. IDEF0.
2. EPC.
3. BPMN.

### 5.3.1. Нотация IDEF0

Она позволяет создать модель, которая будет отражать:

* структуру системы;
* функции;
* потоки ресурсов, информации.

Модель IDEF0 разворачивается одновременно слева направо и сверху вниз, по диагонали. Объекты, расположенные левее/выше, доминируют над теми, которые находятся правее/ниже. Доминирующие объекты могут включать в себя зависимые: например, доставка заказа – это элемент, входящий в состав более масштабного процесса управления заказами. Также доминирующие объекты могут являться предшествующими этапами для зависимых: получение заявки – согласование заявки.

Графические элементы:

* прямоугольники – действия или этапы;
* стрелки – ресурсы, исполнители, необходимые для совершения действия или прохождения этапа.

Главное достоинство IDEF0 – крайне высокая степень детализации, можно создать модель, которая будет учитывать на каждом этапе практически все ресурсы, сотрудников, которые потребуются даже для самых сложных алгоритмов. Недостатком является то, что графическая модель занимает очень много места, её тяжело читать, не имея специальных навыков.

Процесс «Оформление участия в игре» от лица Пользователя был рассмотрен в рамках построения диаграммы в нотации IDEF0 (рис.5.3-5.7).

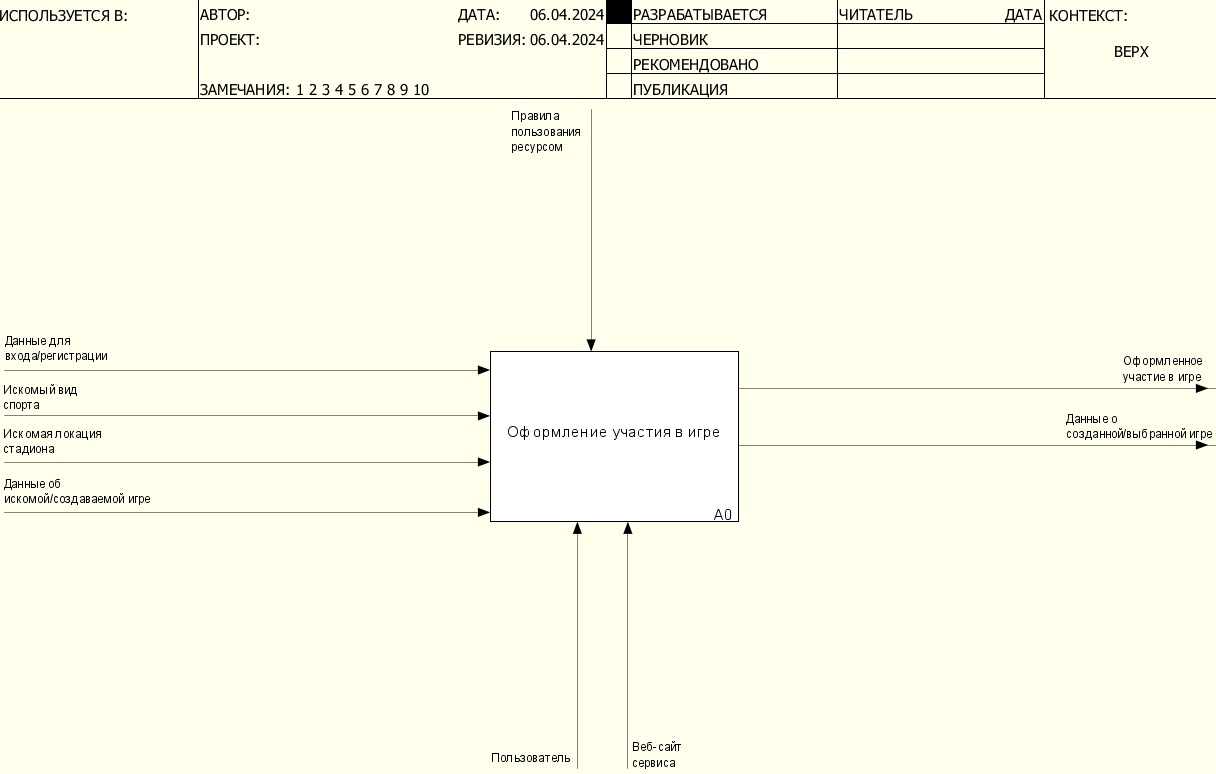


Рисунок 5.3 Процесс «Оформление участия в игре»

Процесс «Оформление участия в игре» был декомпозирован на три подпроцесса «Вход/Регистрация”, “Выбор места участия», «Добавление/Выбор игры».

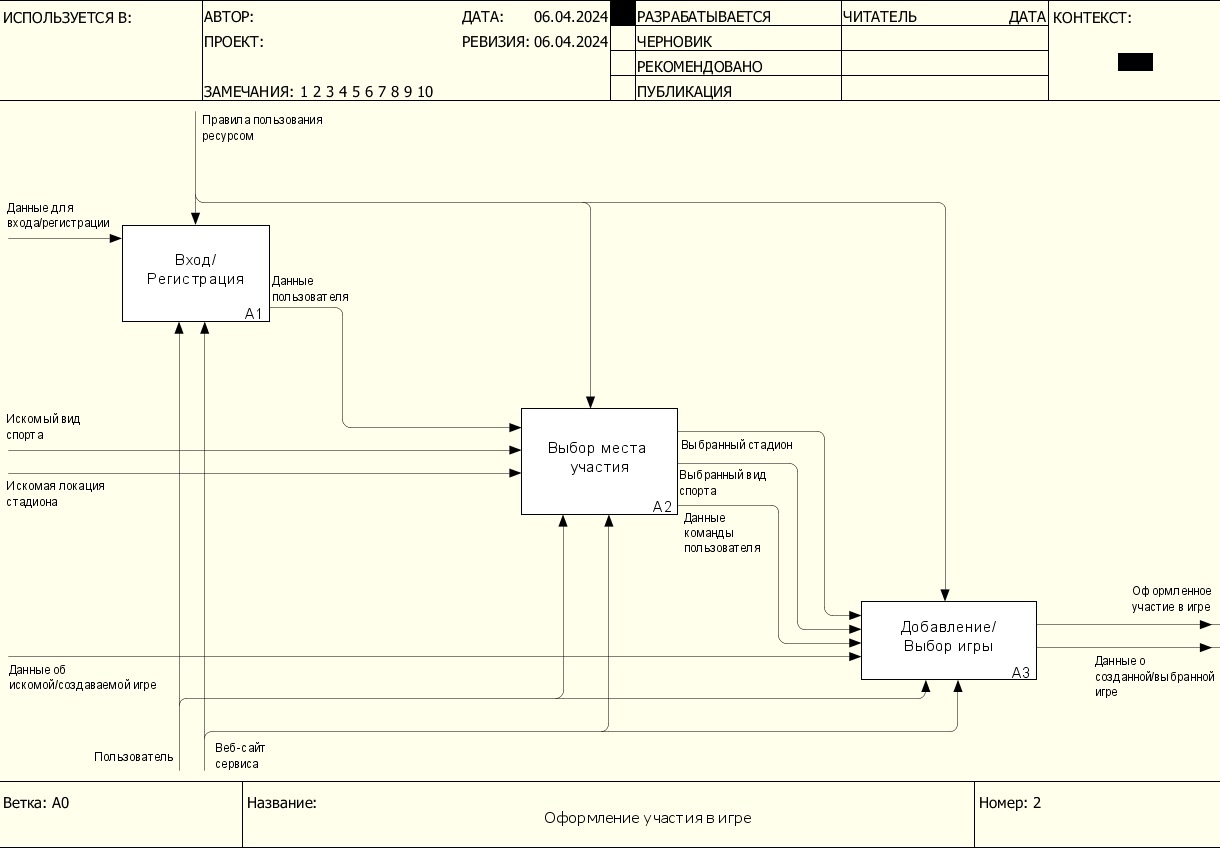


Рисунок 5.4 Декомпозиция процесса «Оформление участия в игре»

Каждый подпроцесс был также декомпозирован на три подпроцесса (рис.5.5 – 5.7)

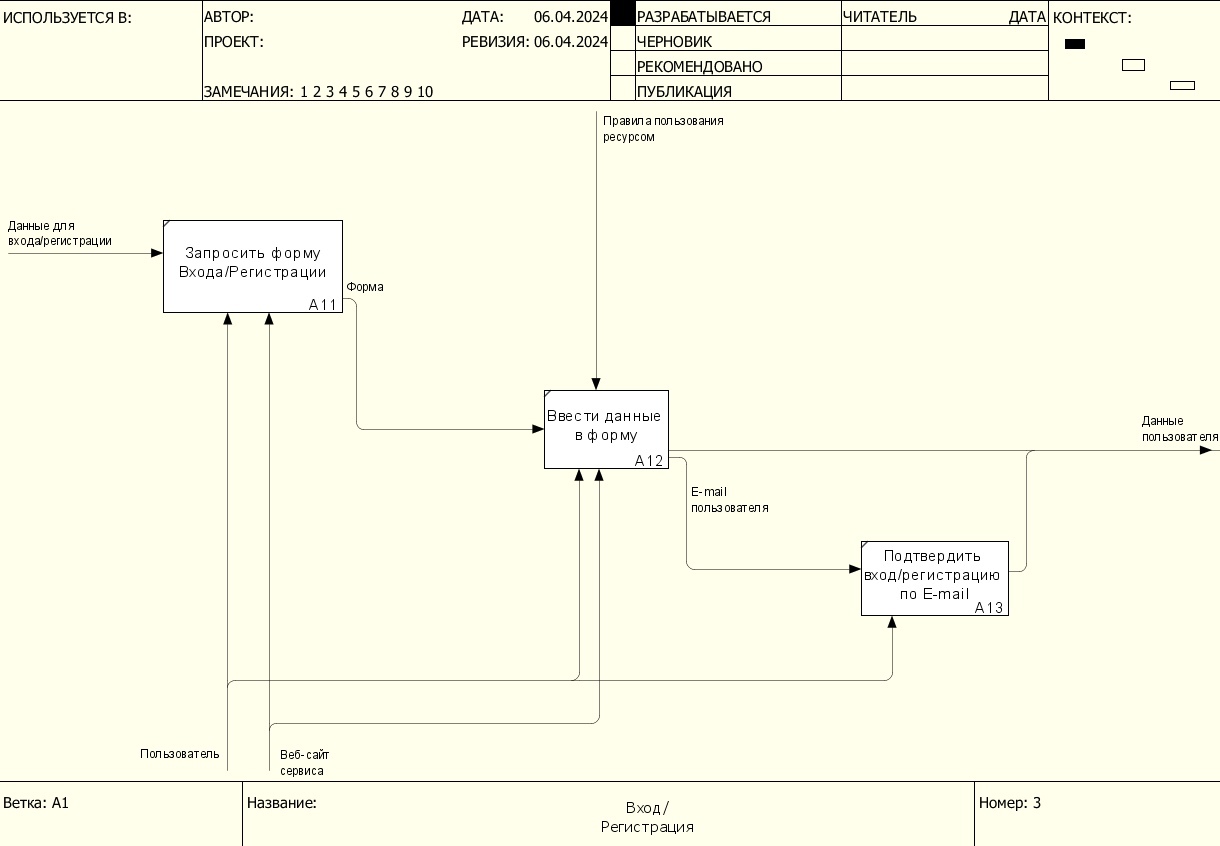


Рисунок 5.5 Декомпозиция процесса «Вход/Регистрация»

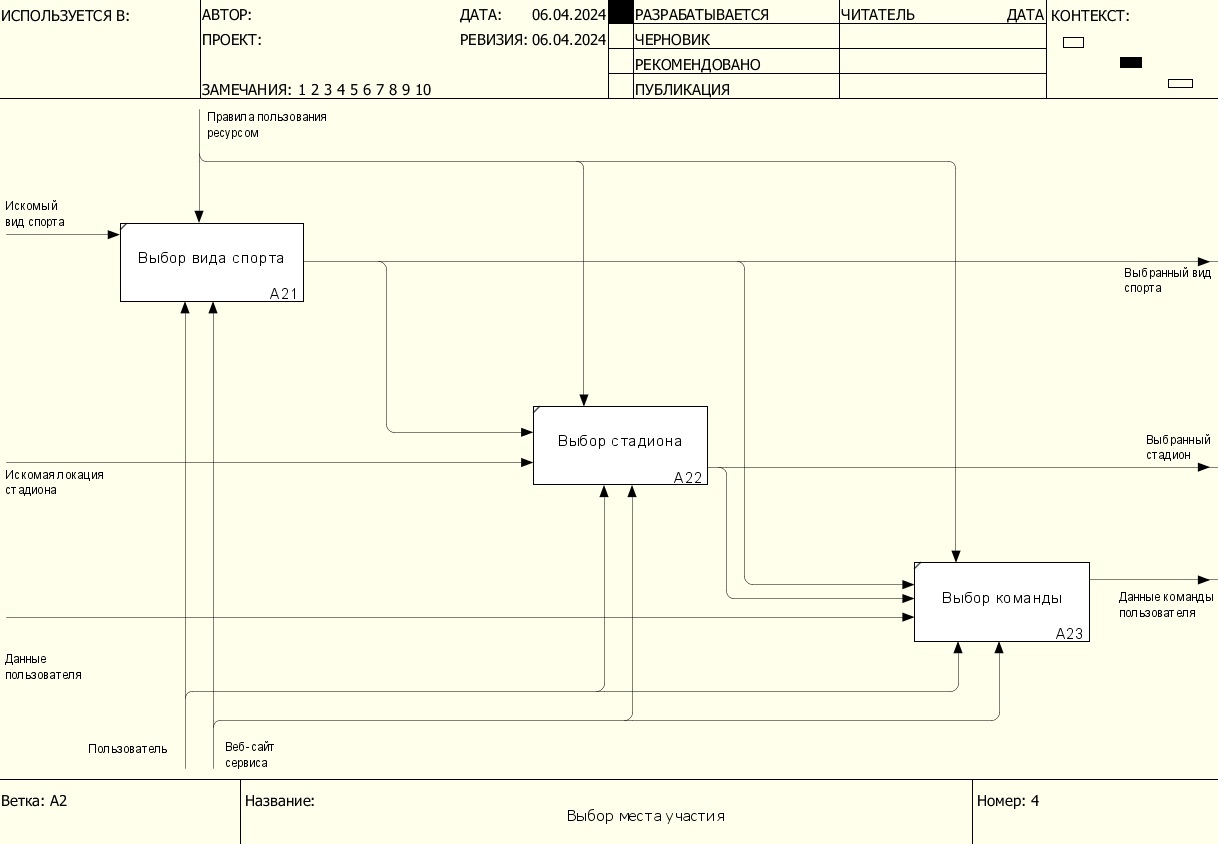


Рисунок 5.6 Декомпозиция процесса «Выбор места участия»

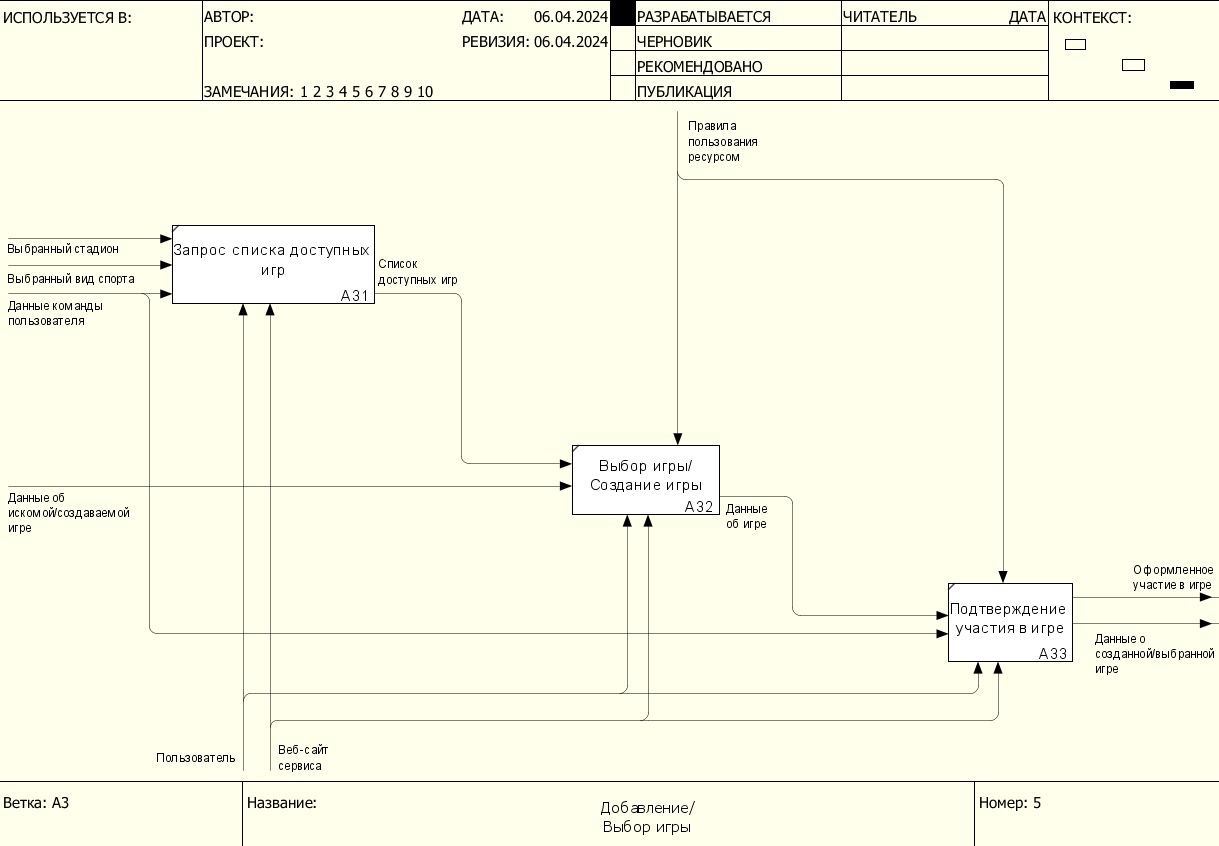


Рисунок 5.7 Декомпозиция процесса «Выбор/Создание игры»

# 6 СОЗДАНИЕ ИНФОРМАЦИОННОЙ ДИАГРАММЫ СИСТЕМЫ И ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ

## **6.1 Задание**

1. Посредством анализа прошлых диаграмм и изучив материал практики, выполнить построение диаграммы в нотации DFD для своего проекта.
2. Словесно описать информационное взаимодействие компонентов системы.
3. Построить нормализованную логическую модель базы данных собственного проекта.

## **6.2 Диаграмма в нотации DFD**

DFD — общепринятое сокращение от англ. data flow diagrams — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе, источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ. Диаграмма потоков данных (data flow diagram, DFD) — один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения UML.

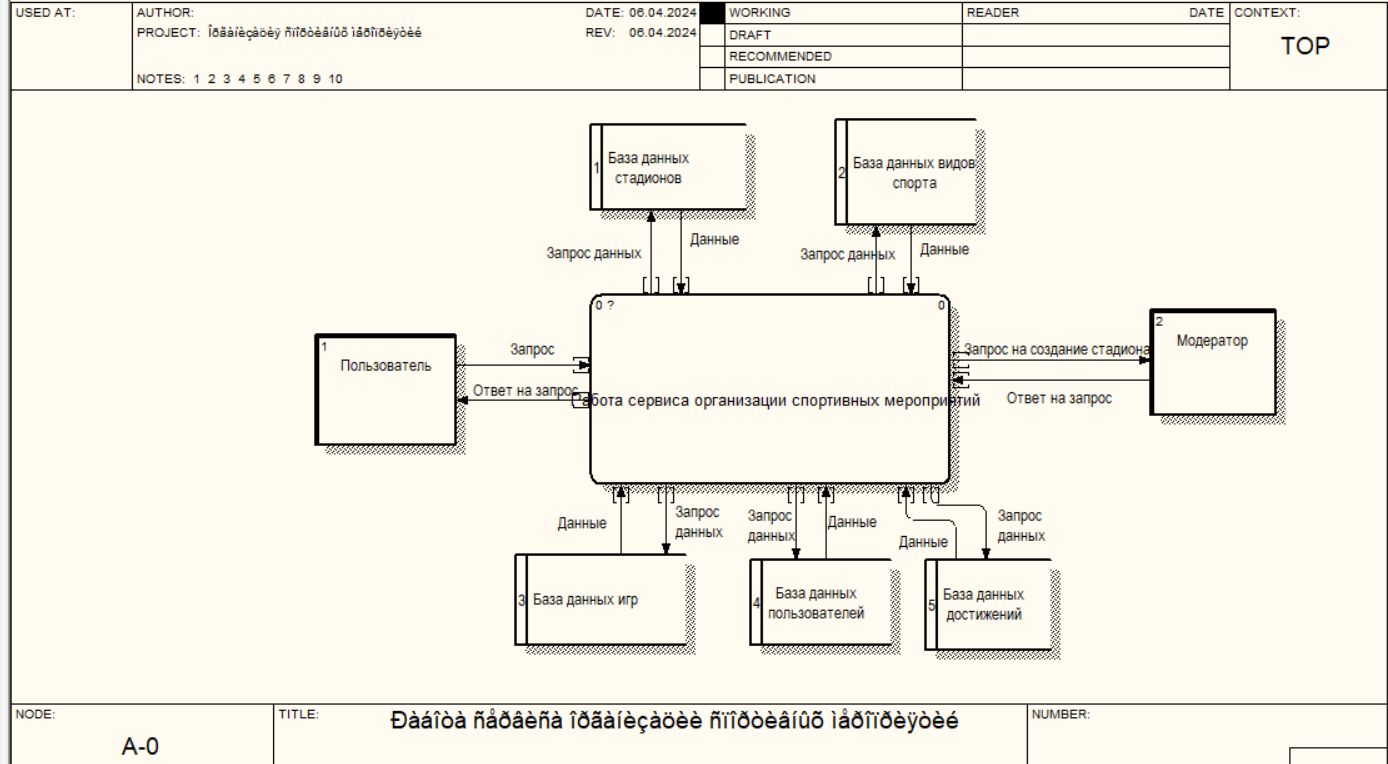


Рисунок 6.1 Диаграмма потоков данных «Работа сервиса организации спортивных мероприятий»

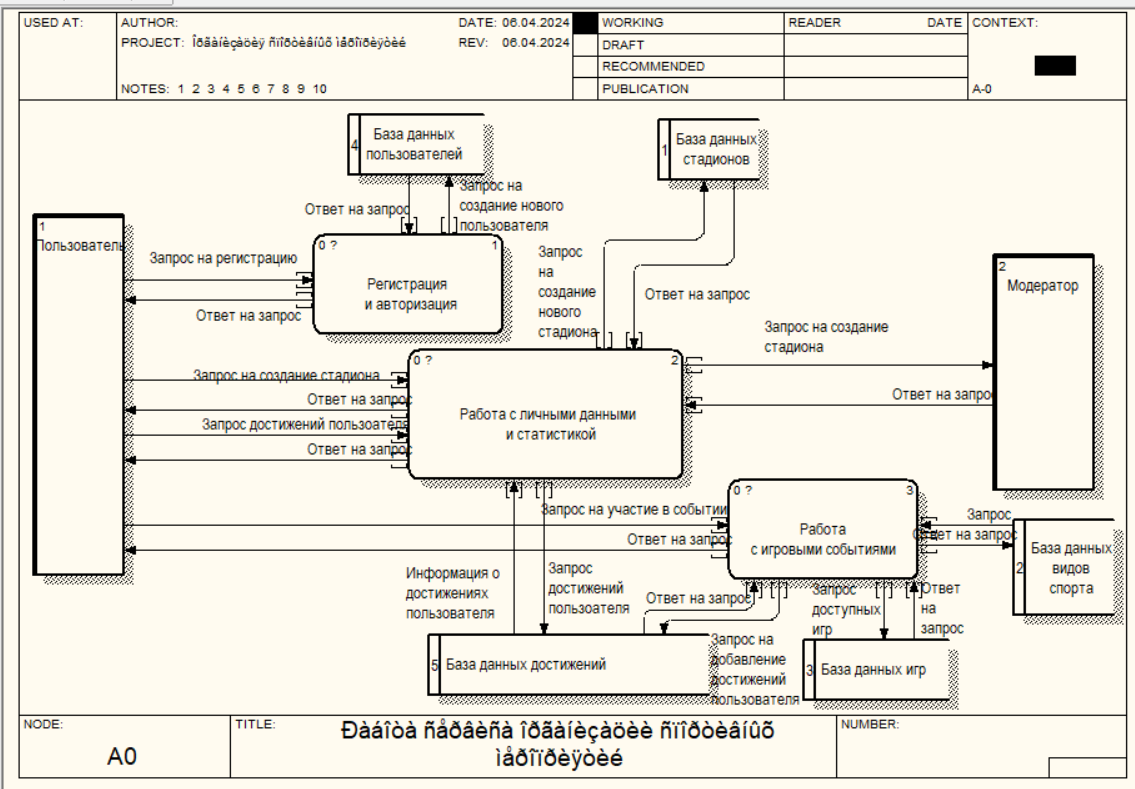


Рисунок 6.2 Диаграмма потоков данных «Работа сервиса организации спортивных мероприятий»

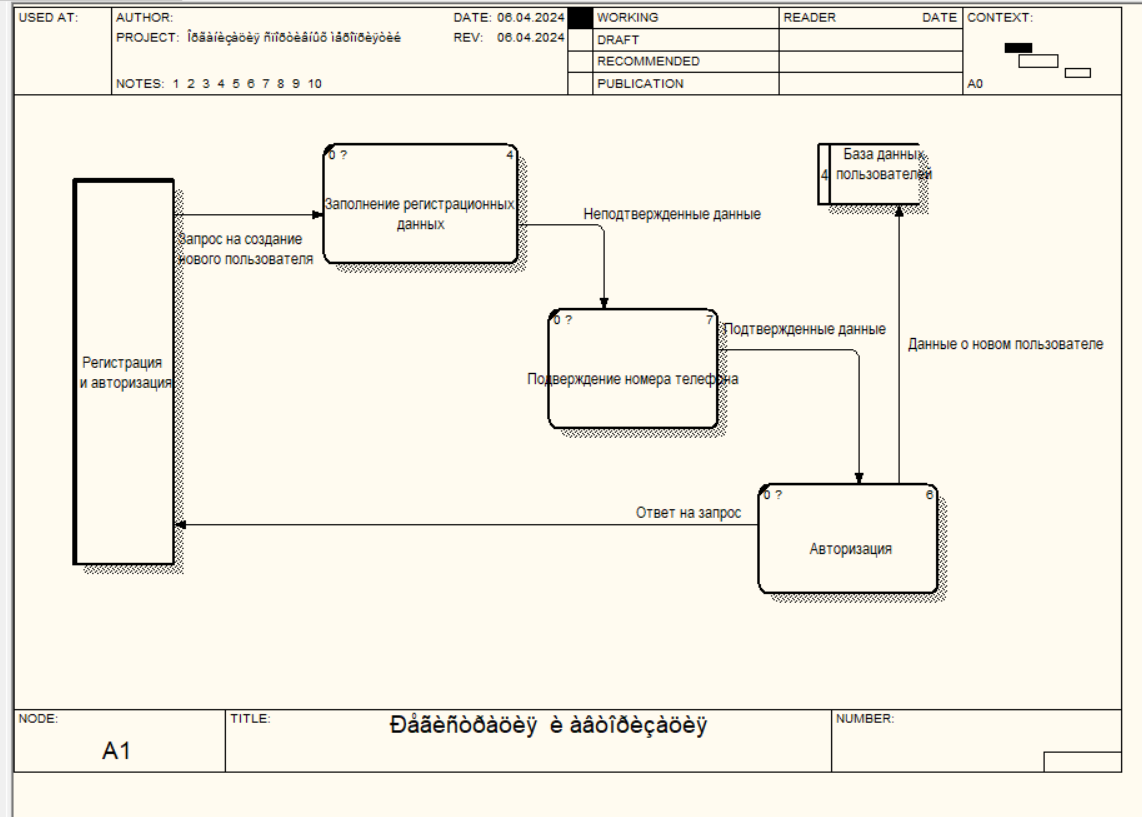


Рисунок 6.3 Диаграмма потоков данных «Регистрация и авторизация»

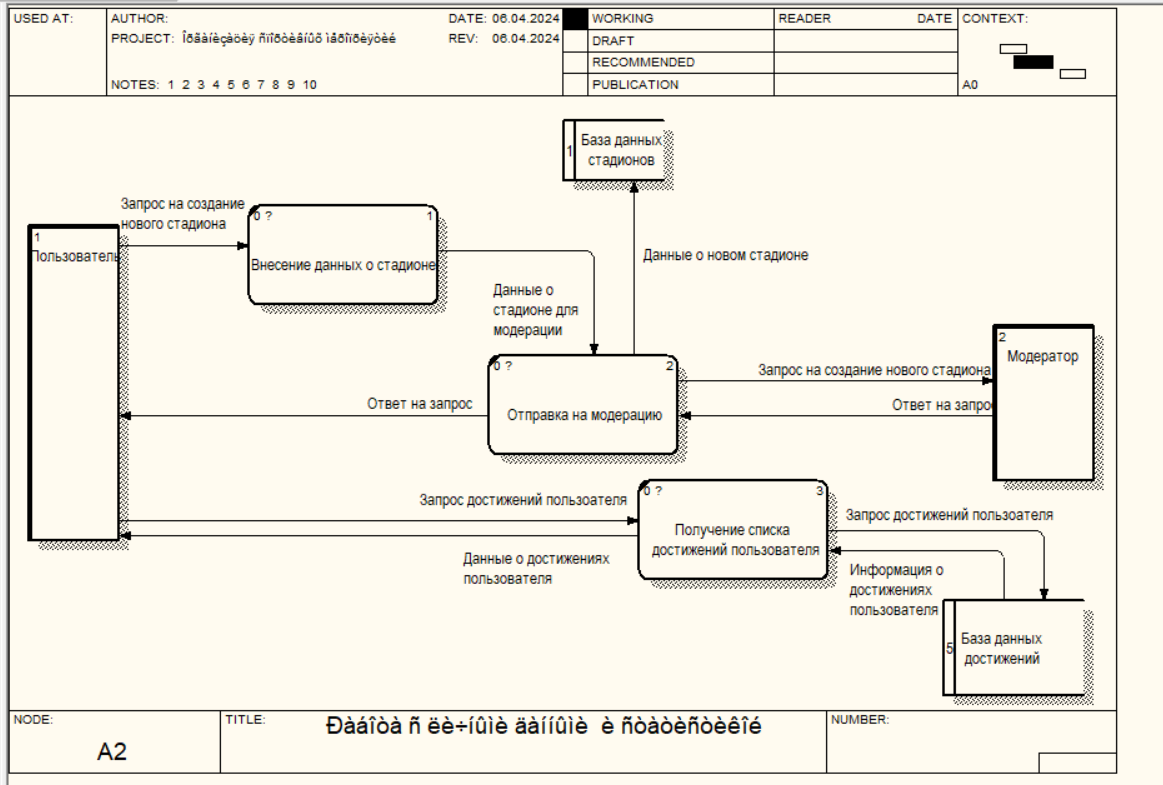


Рисунок 6.4 Диаграмма потоков данных «Работа с личными данными

и статистикой»

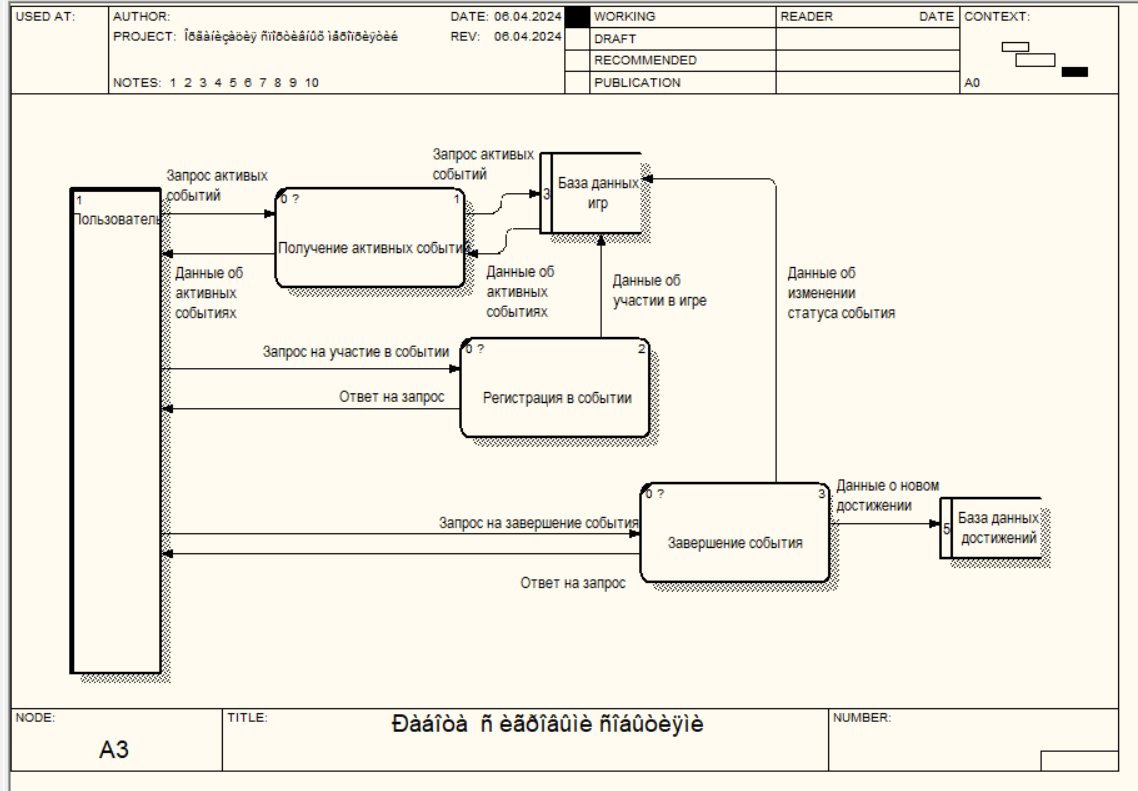


Рисунок 6.5 Диаграмма потоков данных «Работа

с игровыми событиями»

## **6.3 Модель базы данных**

Модель данных – это совокупность структур данных и операций их обработки.

### **6.3.1 Различия между логической и физической моделью БД**

Основные различия между логической и физической моделью БД:

* физическая модель данных описывает физическую структуру базы данных. Логическая модель данных – это модель высокого уровня, которая не описывает физическую структуру базы данных.
* физическая модель данных зависит от используемой системы управления базой данных. Однако логическая модель данных не зависит от используемой системы управления базами данных.
* логическая модель данных включает сущности, атрибуты, отношения и ключи. Физическая модель данных включает таблицы, столбцы, типы данных, ограничения первичного и внешнего ключей, триггеры и хранимые процедуры.
* в логической модели данных для сущностей и атрибутов используются длинные неформальные имена. Однако в физических данных для имен таблиц и столбцов используются сокращенные формальные имена.
* логическая модель данных сначала выводится из описания. После этого выводится только физическая модель данных.
* логическая модель данных нормализована до четвертой нормальной формы. При необходимости физическая модель базы данных будет деформализована для соответствия требованиям.

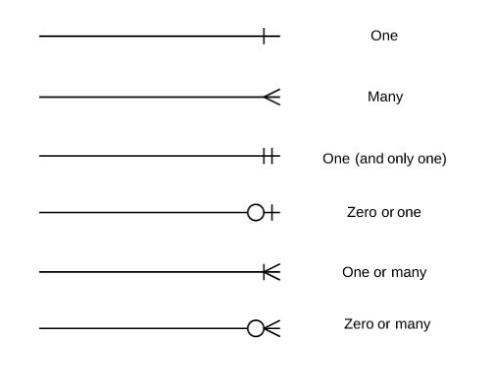


Рисунок 6.6 Связи между сущностями

### **6.3.2 Построение нормализованной логической БД**

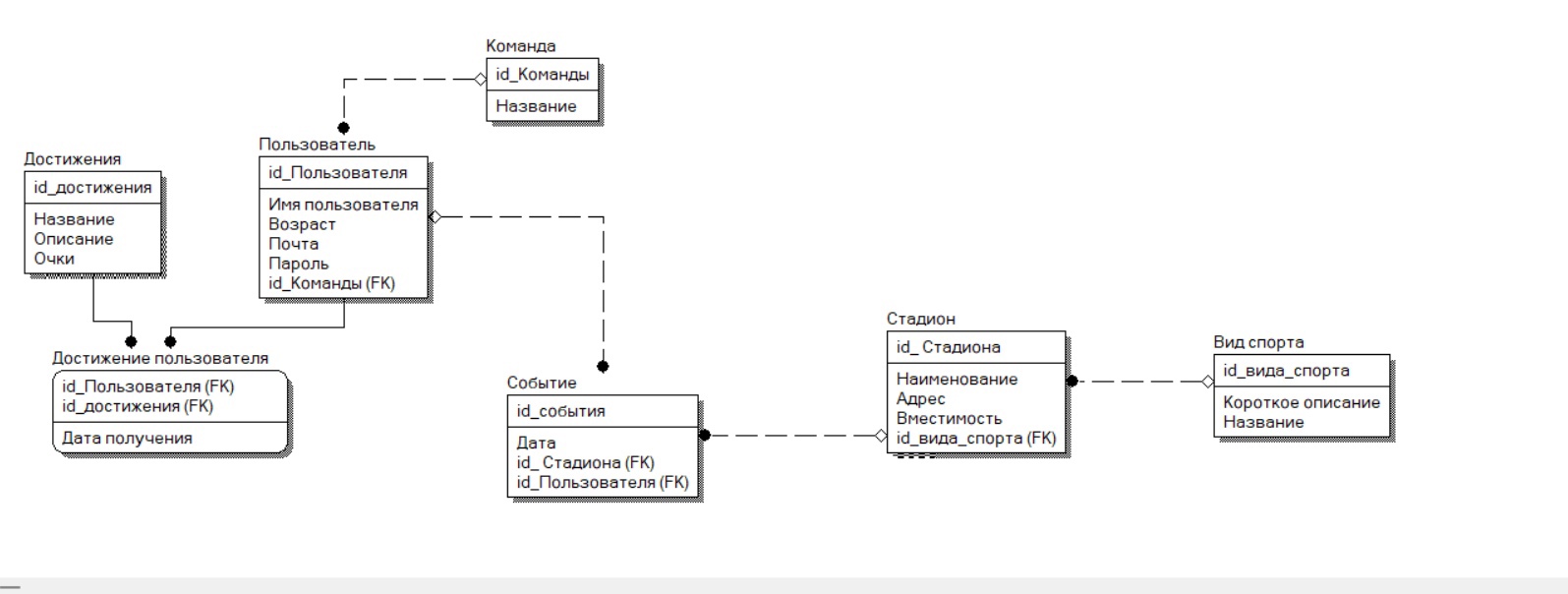


Рисунок 6.7 Нормализованная логическая модель БД

# 7 АРХИТЕКТУРА СИСТЕМЫ

## 7.1 Задание

1. Необходимо описать предлагаемую архитектуру системы и обосновать выбор определенных программных решений для реализации ее компонентов. Программные решения должны включать в себя языки реализации будущей системы, планируемые к применению фреймворки, движки, базы данных и т.д.
2. Построить архитектурную диаграмму своей разработки.
3. Отразить в матрице требований соответствие каждого требования компоненту архитектуры, в котором оно реализуется.

## 7.2 Архитектура системы

В качестве архитектуры системы был выбран монолит. Монолитная архитектура представляет собой подход к разработке программного обеспечения, при котором вся функциональность приложения интегрирована в один модуль, обычно известный как "монолит". Этот модуль развертывается и запускается как единое целое, без явного разделения на отдельные микросервисы.

В контексте нашей системы – агрегатора спортивных мероприятий, который представляет собой инструмент для добавления стадионов, организации игр и управления участниками через веб-портал, монолитная архитектура позволяет сосредоточиться на разработке и поддержке одного приложения, что упрощает управление и развертывание.

Для разработки приложения с монолитной архитектурой очень хорошо подходит архитектурный паттерн MVC (Model, View, Controller). Это схема (Рисунок 7.1) разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер. Разделение приложения на три компонента позволяет независимо изменять каждый из них без влияния на другие компоненты. Это упрощает разработку, тестирование и сопровождение приложения.

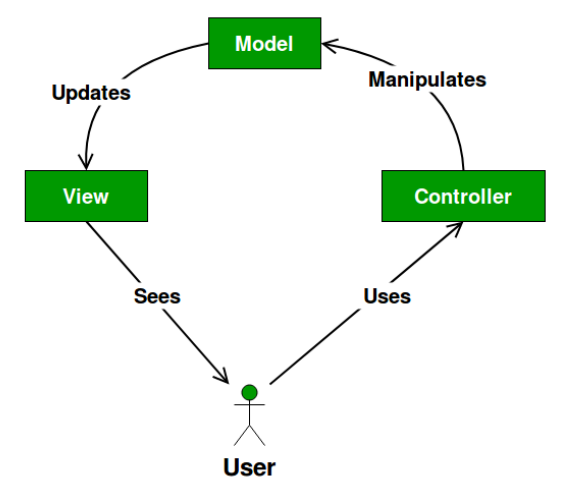


Рисунок 7.1 – Архитектурная диаграмма

Для разработки веб-приложения использовались только свободные (open-source) программные решения.

При разработке приложения была выбрана архитектура клиент-серверного взаимодействия, где сервер выполняет основную обработку данных, а фронтенд является тонким клиентом. Для обеспечения этого подхода были использованы PostgreSQL, FastAPI (Python) и React.

FastAPI был выбран для серверной части приложения. Он предоставляет эффективные инструменты для обработки запросов, интеграции с другими технологиями и обеспечения безопасности, обеспечивает структурирование кода, его переносимость и масштабируемость.

PostgreSQL была выбрана в качестве системы управления базами данных благодаря своей надежности и открытости.

Для клиентской части приложения был выбран React. Его компонентная архитектура облегчает поддержку и переиспользование кода. Использование виртуального DOM улучшает производительность интерфейса.

Таким образом, приложение реализовано на основе клиент-серверной архитектуры с использованием FastAPI, PostgreSQL и React. FastAPI обеспечивает эффективное взаимодействие с базой данных, а React — легкость в разработке интерфейса и поддержке.

## 7.3 Дополнение матрицы требований

После того, как будет спроектирована архитектурная диаграмма, необходимо дополнить матрицу требований. В матрице (табл. 7.1) добавляется новый столбец с названием “Компоненты архитектуры”.

Таблица 7.1 – Матрица требований

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Требование** | **Суть** | **Автор** | **Источники** | **Критерий проверки** | **Компоненты архитектуры** |
| 1. | Веб-сервис для пользователя | | | | | |
| 1.1. | Регистрация участника/организатора | «Система должна предоставлять новому участнику или организатору функцию регистрации» | Квашнин Ю.В. | https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\_Nodejs/forms | Регистрация нового участника или организатора | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, хранение в единой базе данных. |
| 1.2 | Авторизация участника/организатора | «Система должна предоставлять участнику или организатору функцию авторизации» | Квашнин Ю.В. | https://auth0.com/blog/complete-guide-to-nodejs-express-user-authentication/ | Авторизация зарегистрированного участника или организатора | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, хранение в единой базе данных. |
| 1.3 | Ограничение на валидный адрес электронной почты | «Система должна ограничивать ввод пользователя так, чтобы он соответствовал валидному адресу электронной почты» | Квашнин Ю.В. | https://www.twilio.com/blog/2017/03/phone-verification-with-php-laravel-and-twilio-verify.html | Отсутствие невалидных номеров телефона у зарегистрированных участников или организаторов | Ввод данных реализован на устройстве пользователя, проверка происходит на сервере |
| 1.4 | Ограничение на уникальность логина | «Система должна ограничивать ввод логина при регистрации таким образом, чтобы среди участников или организаторов не имелось такого же логина» | Квашнин Ю.В. | https://stackoverflow.com/questions/49547/how-to-control-web-page-caching-across-all-browsers | Отсутствие невалидных адресов электронной почты у зарегистрированных участников или организаторов | Ввод данных реализован на устройстве пользователя, проверка происходит на сервере, данные берутся из единой базы данных |
| 1.5 | Ограничение на сложность пароля | «Система должна ограничивать допустимый пароль наличием хотя бы одной заглавной буквы, прописной буквы и числа, при общей длине пароля не менее 8-ми символов» | Квашнин Ю.В. | https://owasp.org/www-community/password-special-characters | Отсутствие слабых паролей у зарегистрированных участников или организаторов | Ввод данных реализован на устройстве пользователя, проверка происходит на сервере |
| 1.6 | Соответствие новых данных фильтрам при регистрации | «Система должна поддерживать ограничения, применяемые при регистрации, при изменении данных участника или организатора» | Квашнин Ю.В. | https://www.postgresql.org/docs/current/ddl-constraints.html | Отсутствие недопустимых данных участников или организаторов после изменения профиля | Ввод данных реализован на устройстве пользователя, проверка происходит на сервере, хранение в единой базе данных |
| 1.7 | Возможность поиска по названию мероприятия | «Система должна предоставлять участнику или организатору поле поиска, в котором можно вводить название спортивного мероприятия» | Квашнин Ю.В. | https://www.eventbrite.com | Вывод мероприятия в соответствии с введенным названием/вывод, что мероприятие с данным названием отсутствует | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, данные берутся из базы данных |
| 1.8 | Возможность фильтрации по различным параметрам | «Система должна предоставлять возможность фильтрации списка мероприятий по видам спорта, уровню сложности, дате и другим критериям» | Квашнин Ю.В. | https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/filter | Отфильтрованный список мероприятий | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, данные берутся из базы данных |
| 1.9 | Возможность сортировки по различным параметрам | «Система должна предоставлять возможность сортировки мероприятий по дате, местоположению, популярности и другим параметрам» | Квашнин Ю.В. | https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/sort | Отсортированный список мероприятий | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, данные берутся из базы данных |
| 1.10 | Возможность просмотра подробной информации о мероприятии | «Система должна предоставлять подробную информацию о мероприятии, включая описание, местоположение, отзывы участников» | Квашнин Ю.В. | https://www.eventbrite.com | Наличие подробной информации у каждого мероприятия | При выборе данной опции из базы данных берется подробная информация о мероприятии и выводится на устройство |
| 1.11 | Возможность регистрации на мероприятие | «Система должна давать возможность регистрации на мероприятие каждому зарегистрированному участнику» | Квашнин Ю.В. | https://www.eventbrite.com | Наличие зарегистрированных участников на мероприятиях | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, хранение в единой базе данных |
| 1.12 | Возможность просмотра прошедших мероприятий | «Система должна давать возможность зарегистрированному пользователю просматривать историю его участий в мероприятиях» | Квашнин Ю.В. | https://habr.com/ru/companies/Centrobit/articles/164163/ | Список мероприятий, в которых участвовал пользователь | При выборе данной опции из базы данных берется информация о мероприятиях и выводится на устройство. |
| 1.13 | Возможность просмотра списка мероприятий | «Система должна давать возможность пользователю изучать актуальный список мероприятий» | Квашнин Ю.В. | https://habr.com/ru/articles/456604/ | Наличие полного актуального списка мероприятий на странице пользователя | При выборе данной опции из базы данных берется список мероприятий и выводится на устройство. |
| 1.14 | Управление записью на мероприятия | «Система должна предоставлять пользователю возможность записи на мероприятия и управления своими записями» | Квашнин Ю.В. | https://habr.com/ru/articles/46265/ | Наличие записей на мероприятия и управление ими | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, хранение в единой базе данных. |
| 1.15 | Личный кабинет участника | «Система должна предоставлять пользователю личный кабинет, в котором он может просмотреть историю своих участий в мероприятиях» | Квашнин Ю.В. | https://habr.com/ru/articles/725942/ | Наличие личной информации в кабинете пользователя | При выборе данной опции из базы данных берется подробная информация о текущем пользователе и выводится на устройство. |
| 1.16 | Возможность добавления мероприятий в закладки | «Система должна предоставлять пользователю возможность добавления понравившихся мероприятий в закладки» | Квашнин Ю.В. | https://www.chitai-gorod.ru/ | Наличие мероприятия в закладках | Ввод данных реализован на устройстве пользователя, обработка происходит на сервере, хранение в единой базе данных. |
| 1.17 | Возможность просмотра информации о мероприятии | «Система должна показывать информацию о мероприятии на странице мероприятия» | Квашнин Ю.В. | https://www.chitai-gorod.ru/ | Вывод статуса мероприятия | При выборе данной опции из базы данных берется информация о мероприятии и выводится на устройство. |
| 2. | Веб-сервис для модератора | | | | | |
| 2.1 | Авторизация для модератора | «Система должна предоставлять модератору функцию авторизации для доступа к функционалу модератора» | Квашнин Ю.В. | https://www.sciencedirect.com/science/article/pii/S1877050920305041 | Доступ к функционалу после авторизации | Ввод данных реализован на устройстве модератора, обработка происходит на сервере, хранение в единой базе данных. |
| 2.2 | Управление страницей стадионов | «Модератор должен иметь возможность добавлять, редактировать и удалять информацию о стадионах» | Квашнин Ю.В. | https://link.springer.com/article/10.1007/s12525-017-0270-8 | Актуализированный список стадионов, доступный для всех пользователей | Ввод данных реализован на устройстве модератора, обработка происходит на сервере, хранение в единой базе данных. |
| 2.3 | Модерация видов спорта | «Система должна предоставлять инструменты для добавления, редактирования и удаления видов спорта в каталоге» | Квашнин Ю.В. | https://www.tandfonline.com/doi/abs/10.1080/16184742.2017.1318166 | Обновленный каталог видов спорта, доступный для организаторов мероприятий | Ввод данных реализован на устройстве модератора, обработка происходит на сервере, хранение в единой базе данных. |
| 2.4 | Модерация страницы наград | «Система должна обеспечивать модератору возможности по добавлению, редактированию и удалению наград» | Квашнин Ю.В. | https://journals.sagepub.com/doi/abs/10.1177/2158244019835914 | Обновленный список наград, отображаемый у всех пользователей | Ввод данных реализован на устройстве модератора, обработка происходит на сервере, хранение в единой базе данных. |
| 2.5 | Управление страницей пользователей | «Модератор должен иметь возможность управлять пользователями: блокировать и разблокировать аккаунты, редактировать профили» | Квашнин Ю.В. | https://ieeexplore.ieee.org/abstract/document/9155567 | Перечень пользователей и статус их активности, доступный для модерации | Ввод данных реализован на устройстве модератора, обработка происходит на сервере, хранение в единой базе данных. |
| 3. | Веб-сервис | | | | |  |
| 3.1 | Шифрование паролей | «Для хэширования пароля необходимо использовать алгоритм BCrypt» | Квашнин Ю.В. | https://ru.wikipedia.org/wiki/Bcrypt | Пароли захэшированы алгоритмом BCrypt | Для хэширования пароля на сервере используется алгоритм BCrypt. |
| 3.2 | Способность обрабатывать большое количество запросов | «На быстродействии приложения не должно сказываться несколько (т. е. не более 25) одновременных запросов к серверу.» | Квашнин Ю.В. | https://simpleone.ru/blog/bystrodejstvie-sistemy-s-tochki-zreniya-polzovatelya/ | Проведение тестирования быстродействия сервера в условиях высокой нагрузки | Увеличение допустимых пиковых нагрузок на сервер достигается с помощью улучшения характеристик серверного оборудования, а также улучшения методов работы с данными. |
| 3.3 | Кроссплатформенность | «Веб-приложение не должно иметь привязки к одной ОС и должно иметь возможность развертывания на Windows и Linux» | Квашнин Ю.В. | https://ru.wikipedia.org/wiki/Кроссплатформенность | Тестирование нормальной работы приложения, развёртывание на разных ОС | Пользователи могут взаимодействовать с сервером используя любую ОС. |
| 3.4 | Сохранность данных | «Приложение не должно давать пользователю без прав администратора доступ к данным других пользователей» | Квашнин Ю.В. | https://habr.com/ru/post/51327/ | Тестирование недоступности данных других пользователей текущему пользователю без прав администратора | На устройствах пользователей доступ должен быть только к тем функциям, которые предусмотрены их ролью, распределение функций для разных ролей происходит на сервере. |
| 3.5 | Безопасность исходного кода | «Все пароли для доступа к базе данных и внешним сервисам должны храниться в отдельном конфигурируемом файле в папке проекта» | Квашнин Ю.В. | https://www.youtube.com/watch?v=Y9MRCxq4DIc | Отсутствие паролей в исходном коде проекта | Все пользователи должны получать доступ к базе данных и другим внешним сервисам только через приложение, все возможности определяются сервером. |
| 4. | Правовые нормы регулирования | | | | | |
| 4.1 | Закон "О защите персональных данных" | «Приложение, при получении, хранении и обработки персональных данных клиентов и сотрудников компании, должна руководствоваться нормами закона Федерального закона "О персональных данных" от 27.07.2006 N 152-ФЗ (последняя редакция)» | Квашнин Ю.В. | http://www.consultant.ru/document/cons\_doc\_LAW\_61801/ | Прохождение теста на невозможность получения закрытых персональных данных третьими лицами | Выполнение норм должны обеспечивать все компоненты программной системы. |

# 8 ФОРМИРОВАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Техническое задание (далее – ТЗ) – основной документ проекта, которым Заявитель устанавливает цели и задачи проекта, номенклатуру и назначение продуктов проекта, технические и иные значимые характеристики проектируемого производства и/или продукта проекта, порядок и последовательность необходимых стадий реализации проекта, создания продукта проекта (в том числе описание технологии) и контроля его качественных параметров.

Техническое задание должно давать исчерпывающее описание соответствия заявленного проекта требованиям, описанным в прошедших практиках собственного проекта.

## 8.1. Задание

1.Необходимо обосновать выбор ГОСТа для разработки технического задания, почему был выбран тот или иной стандарт, по каким разделам данный ГОСТ подходит больше и т.д.

2.Составить в соответствии с выбором техническое задание по ГОСТу.

## 8.2 Выбор ГОСТа

Наиболее подходящим стандартом для написания ТЗ для нашего проекта может быть ГОСТ 34.602-2020 "Автоматизированные системы. Техническое задание. Структура и правила оформления". Этот стандарт предоставляет структуру для описания требований к автоматизированным системам. Он включает разделы, такие как введение, область применения, функциональное назначение, требования к функциональным характеристикам. Вот некоторые из наиболее важных разделов:

1. **Функциональное назначение**: этот раздел позволяет подробно описать функциональные требования к системе, такие как управление каталогом книг, оформление заказов и т.д.
2. **Требования к интерфейсам**: здесь можно определить требования к пользовательскому интерфейсу веб-сайта магазина, включая дизайн, навигацию, формы ввода данных и т.д.
3. **Требования к программным средствам**: В этом разделе можно указать требования к программным средствам, необходимым для работы онлайн магазина, такие как язык программирования, базы данных и т.д.
4. **Требования к техническому обеспечению**: здесь следует определить требования к серверному оборудованию, хостингу, безопасности данных и т.д.
5. **Требования к документации**: важно указать требования к документации, которая должна быть предоставлена в рамках проекта, например, техническая документация, руководства пользователя и т.д.

### 8.3 Техническое задание

**СОДЕРЖАНИЕ**

[8.3.1 Общие сведения 28](#_Toc162023753)

[8.3.1.1 Полное наименование системы и ее условное обозначение 28](#_Toc162023754)

[8.3.2.1 Шифр темы или шифр (номер) договора 28](#_Toc162023755)

[8.3.3.1 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты 28](#_Toc162023756)

[8.3.4.1 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы 28](#_Toc162023757)

[8.3.5.1 Плановые сроки начала и окончания работы по созданию системы 28](#_Toc162023758)

[8.3.6.1 Сведения об источниках и порядке финансирования работ 29](#_Toc162023759)

[8.3.7.1 Порядок оформления и предъявления заказчику результатов работ по созданию системы 29](#_Toc162023760)

[8.3.2 Назначение и цели создания (развития) системы 30](#_Toc162023761)

[8.3.2.1 Назначение системы 30](#_Toc162023762)

[8.3.2.1 Цели создания системы 30](#_Toc162023763)

[8.3.3 Характеристика объекта автоматизации 31](#_Toc162023764)

[8.3.3.1 Краткие сведения об объекте автоматизации 31](#_Toc162023765)

[8.3.3.2 Сведения об условиях эксплуатации объекта автоматизации 31](#_Toc162023766)

[8.3.4 Требования к системе 32](#_Toc162023767)

[8.3.4.1 Требования к системе в целом 32](#_Toc162023768)

[8.3.4.1.1 Требования к структуре и функционированию системы 32](#_Toc162023769)

[8.3.4.1.2 Требования к численности и квалификации персонала системы и режиму его работы 33](#_Toc162023770)

[8.3.4.1.3 Показатели назначения 33](#_Toc162023771)

[8.3.4.1.4 Требования к надежности 34](#_Toc162023772)

[8.3.4.1.5 Требования к безопасности 35](#_Toc162023773)

[8.3.4.1.6 Требования к эргономике и технической эстетике 35](#_Toc162023774)

[8.3.4.1.7 Требования к транспортабельности для подвижных АС 36](#_Toc162023775)

[8.3.4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы 36](#_Toc162023776)

[8.3.4.1.9 Требования к защите информации от несанкционированного доступа 36](#_Toc162023777)

[8.3.4.1.10 Требования по сохранности информации при авариях 36](#_Toc162023778)

[8.3.4.1.11 Требования к защите от влияния внешних воздействий 36](#_Toc162023779)

[8.3.4.1.12 Требования к патентной чистоте 36](#_Toc162023780)

[8.3.4.1.13 Требования по стандартизации и унификации 36](#_Toc162023781)

[8.3.4.1.14 Дополнительные требования 37](#_Toc162023782)

[8.3.4.2 Требования к функциям (задачам), выполняемым системой 37](#_Toc162023783)

[8.3.4.3 Требования к видам обеспечения 37](#_Toc162023784)

[8.3.4.3.1 Требования к математическому обеспечению системы 37](#_Toc162023785)

[8.3.4.3.2 Требования к информационному обеспечению системы 38](#_Toc162023786)

[8.3.4.3.3 Требования к лингвистическому обеспечению системы 38](#_Toc162023787)

[8.3.4.3.4 Требования к программному обеспечению системы 38](#_Toc162023788)

[8.3.4.3.5 Требования к техническому обеспечению системы 38](#_Toc162023789)

[8.3.4.3.6 Требования к метрологическому обеспечению системы 39](#_Toc162023790)

[8.3.4.3.7 Требования к организационному обеспечению системы 39](#_Toc162023791)

[8.3.4.3.8 Требования к методическому обеспечению системы 39](#_Toc162023792)

[8.3.5 Состав и содержание работ по созданию системы 40](#_Toc162023793)

[8.3.6 Порядок контроля и приемки системы 41](#_Toc162023794)

[8.3.7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие 42](#_Toc162023795)

[8.3.7.1 Приведение поступающей в систему информации к виду, пригодному для обработки с помощью ЭВМ 42](#_Toc162023796)

[8.3.7.2 Изменения, которые необходимо осуществить в объекте автоматизации 42](#_Toc162023797)

[8.3.7.3 Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ 42](#_Toc162023798)

[8.3.7.4 Создание необходимых для функционирования системы подразделений и служб 42](#_Toc162023799)

[8.3.7.5 Сроки и порядок комплектования штатов и обучения персонала 43](#_Toc162023800)

[8.3.8 Требования к документированию 44](#_Toc162023801)

[8.3.9 Источники разработки 45](#_Toc162023802)

#### 8.3.1 Общие сведения

##### Полное наименование системы и ее условное обозначение

Название системы: Спортивный портал PlayHub.

Условное обозначение: PH.

##### Шифр темы или шифр (номер) договора

Номер договора: №1/11-11-11-001 от 01.03.2024.

##### Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты

Заказчиком системы является юридическое лицо «RTU MIREA».

Адрес заказчика: г. Москва.

Разработчиком системы является ООО «Students».

##### Перечень документов, на основании которых создается система, кем и когда утверждены эти документы

«Договор на разработку автоматизированной системы интернет-ресурса по организации спортивных мероприятий от 01.03.2024»

##### Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ по созданию системы интернет-ресурса – 1 марта 2024 года.

Плановый срок окончания работ по созданию системы интернет-ресурса – 30 мая 2024 года.

##### Сведения об источниках и порядке финансирования работ

Источником финансирования работ является ООО «RTU MIREA».

##### Порядок оформления и предъявления заказчику результатов работ по созданию системы

Результат работ передается Заказчику в порядке, определенном контрактом в соответствии с Календарным планом работ контракта на основании Актов сдачи-приемки выполненных работ (этапа работ).

Документация PH передается на машинном носители (флеш-накопитель). Текстовые документы, передаваемые на машинных носителях, должны быть представлены в форматах PDF.

Все материалы передаются с сопроводительным документами Исполнителя.

#### 8.3.2 Назначение и цели создания (развития) системы

##### 8.3.2.1 Назначение системы

Информационная система организации спортивных мероприятий предназначена для автоматизации процесса организации спортивных мероприятий и участия в них.

##### 8.3.2.1 Цели создания системы

Основными целями создания ИС являются:

1. увеличение уровня спортивного интереса пользователей;
2. упрощение организации и участия в различных спортивных мероприятиях.

#### 8.3.3 Характеристика объекта автоматизации

##### 8.3.3.1 Краткие сведения об объекте автоматизации

Объектом автоматизации является совокупность процессов, составляющих деятельность сервиса организации спортивных мероприятий заказчика.

##### 8.3.3.2 Сведения об условиях эксплуатации объекта автоматизации

Условия эксплуатации комплекса технических средств Системы должны соответствовать условиям эксплуатации группы 2 ГОСТ 21552-84 «Средства вычислительной техники. Общие технические требования, приемка, методы испытаний, маркировка, упаковка, транспортировка, хранение».

Условия эксплуатации персональных компьютеров Системы соответствуют Гигиеническим требованиям к видео-дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы (Санитарные правила и нормы. СанПиН 2.2.2.542-96).

Исполнитель должен проверить соблюдение условий эксплуатации комплекса технических средств на этапе технического проектирования

#### 8.3.4 Требования к системе

##### 8.3.4.1 Требования к системе в целом

###### 8.3.4.1.1 Требования к структуре и функционированию системы

Система должна иметь модульную структуру, включающую в себя следующие модули:

* модуль раздела «Список мероприятий»;
* модуль раздела «Список наград»;
* модуль раздела «Профиль»;
* модуль раздела «Регистрация и авторизация»
* модуль раздела «Модерация».

Система должна выполнять следующие функции:

* просмотр доступных команд и мероприятий;
* просмотр информации о командах и мероприятиях;
* Создание новых команд и мероприятий;
* Присоединение пользователя к команде и мероприятию;
* Просмотр доступных видов спорта и стадионов;
* Добавление нового стадиона пользователем;
* Управление ролями пользователя (модератор, обычный пользователь);
* назначение и присвоение наград за достижения и участие в играх;
* просмотр своего профиля и редактирование своих данных;
* редактирование информации о командах, играх и стадионах модераторами.

Взаимодействие между модулями происходит внутри сервера, за счет монолитной структуры веб-приложения.

Взаимодействие со смежными системами происходит в формате JSON.

Система должна функционировать круглосуточно, а её диагностика проводиться ежедневно.

Система может быть модернизирована, путем разработки возможности добавления стадионов на онлайн карты.

###### 8.3.4.1.2 Требования к численности и квалификации персонала системы и режиму его работы

Для поддержания работоспособности системы и эксплуатации веб-интерфейса системы управления сайтом интернет-магазина от модератора не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб-браузером.

Конкретные требования к роли администратора и модератора представлены ниже, а режим работы других пользователей не ограничен.

Администратор:

* Численность: 1-2 человека.
* Квалификационные требования: продвинутое пользование персональным компьютером, опыт работы в области информационных технологий не менее 2 лет, опыт работы с базами данных и операционными системами, умение работать с Linux и Windows, управление конфигурацией веб-сервера, решение технических проблем и взаимодействие с авторами изображений при их обращении.

Модератор:

* Численность: 2-5 человек.
* Квалификационные требования: умение пользоваться панелью модерации отзывов, объективно и внимательно оценивать отзыв на соответствие законодательству РФ и правилам сайта для принятия решения об его публикации или удаления, внимательность к деталям и умение работать с большим объемом информации.

Режим работы: свободный график, но не менее 4 часов в день.

###### 8.3.4.1.3 Показатели назначения

Объектом автоматизации является совокупность процессов, составляющих деятельность сервиса организации спортивных мероприятий заказчика.

Подсистемы, разработанные и доработанные в рамках данного раздела, обязательно должны отвечать следующим требованиям:

1. Время на полный запуск (или перезапуск) системы и компонентов системы должно составлять не более 5 минут.

1. Коэффициент юзабилити не менее 85%.

2. Коэффициент интерактивности не менее 88%

3. Среднее время загрузки страницы интернет-ресурса не должно превышать 3 секунды.

4. Время реагирования администратора на возникшую внештатную ситуацию не более 5 минут.

5. Время реагирования администратора на запрос правообладателя товара не более суток.

6. Время ожидания прохождения модерации стадиона не более суток.

7. Сервер системы: 100 запросов в минуту при времени отклика не более трёх секунд.

8. Интервал создания резервных копий данных раз в 3 месяца.

Требования к аппаратной части и масштабированию для обеспечения перечисленных показателей должны быть определены на этапе технического проектирования.

###### 8.3.4.1.4 Требования к надежности

Объектом автоматизации является совокупность процессов, составляющих деятельность сервиса организации спортивных мероприятий заказчика.

Программное обеспечение не должно выходить из строя более чем на 10 минут. Наиболее вероятные нештатные ситуации и реакция на них описана в руководстве пользования системой для администратора.

Для устойчивости к потере данных необходимо регулярно производить выгрузку хранимой информации.

Надежность требуемого уровня достигается путем комплексного применения организационных и организационно-технических мероприятий. При этом необходимо использовать соответствующие требованиям программно-аппаратные средств. В частности, можно использовать следующие базовые подходы:

* системное и базовое ПО и технические средства, соответствующие классу решаемой задачи;
* четкое соблюдение правил эксплуатации, а также регламентных сроков обслуживания используемых программно-аппаратных средств;
* допуск к системе управления только пользователей, прошедших предварительное обучение.

###### 8.3.4.1.5 Требования к безопасности

Безопасность данных пользователей должна обеспечиваться шифрованием, а также обеспечением устойчивости программно-технических средств к возможным кибератакам.

Безопасность аппаратной части системы обеспечивается организацией, которая предоставляет свои услуги для хостинга сервера веб-приложения заказчика.

###### 8.3.4.1.6 Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением осуществляется через визуальный графический интерфейс (GUI).

Интерфейс системы должен быть понятным и удобным для пользователей всех категорий, он не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм.

Дизайн и компоновка элементов пользовательского интерфейса должны быть дружелюбными и современными, создавая приятное визуальное впечатление у пользователей.

Важно обеспечить согласованность и целостность дизайна интерфейса по всему приложению, чтобы пользователи могли легко ориентироваться и использовать функционал системы без лишних усилий.

###### 8.3.4.1.7 Требования к транспортабельности для подвижных АС

Для системы требования к транспортабельности не применимы, так как она представляет собой веб-приложение, которое размещается на сервере и доступно через интернет.

###### 8.3.4.1.8 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Эксплуатацией, техническим обслуживанием, ремонтом и хранением сервера АС занимается организация, которая предоставляет свои услуги для хостинга сервера веб-приложения заказчика.

###### 8.3.4.1.9 Требования к защите информации от несанкционированного доступа

Необходимо использовать криптографические методы защиты данных и SSL-сертификаты, чтобы информация, размещенная на сайте, и данные пользователей были защищены от доступа или модификации несанкционированными лицами.

###### 8.3.4.1.10 Требования по сохранности информации при авариях

Серверное программное обеспечение системы должно восстанавливать свое функционирование при перезапуске аппаратных средств. Для обеспечения сохранности данных требуется предусмотреть резервное копирование.

###### 8.3.4.1.11 Требования к защите от влияния внешних воздействий

Требования к защите от влияния внешних воздействий не предъявляются, так как этим руководит организация, которая предоставляет свои услуги для хостинга сервера веб-приложения заказчика.

###### 8.3.4.1.12 Требования к патентной чистоте

Требования к патентной чистоте не предъявляются, если о таковых не заявит правообладатель товара, сообщив об этом администрации сайта.

###### 8.3.4.1.13 Требования по стандартизации и унификации

Для обеспечения совместимости и эффективной работы, система должна использовать языки HTML и CSS в соответствии со стандартами W3C, для создания интерактивных элементов клиентской части – язык JavaScript и его библиотека React. Для серверной части приложения должен быть использован язык Python и его фреймворк FastAPI. Система должна соответствовать стандартам безопасности информации. Интерфейс системы должен соответствовать современным требованиям юзабилити.

###### 8.3.4.1.14 Дополнительные требования

Дополнительные требования не предъявляются.

##### 8.3.4.2 Требования к функциям (задачам), выполняемым системой

Таблица 7.1 – Требования к функциям, выполняемым системой

|  |  |
| --- | --- |
| **Функция** | **Задача** |
| Работа с мероприятиями | Просмотр списка видов спорта |
| Просмотр списка стадионов |
| Просмотр списка доступных игр |
| Добавление нового стадиона |
| Добавление новой игры |
| Запись на игру |
| Работа с пользователями | Регистрация пользователей |
| Авторизация пользователей |
| Аутентификация пользователей |
| Использование профиля |
| Обработка, хранение и поддержка БД | Создание резервных копий в соответствии с графиком |
| Функция поиска и просмотр | Поиск игр, стадионов и наград по названию |
| Фильтрация и сортировка поиска |
| Функция модерации | Просмотр и обработка предложенных стадионов |
| Удаление неактуальных/некорректных данных |
| Редактирование некорректных данных |
| Функция администрирования | Просмотр и изменение профиля пользователя |
| Блокировка пользователя |
| Удаление пользователя |
| Изменение роли пользователя |

##### 8.3.4.3 Требования к видам обеспечения

###### 8.3.4.3.1 Требования к математическому обеспечению системы

Для эффективной работы системы необходимо подготовить соответствующее математическое обеспечение. Это включает в себя разработку алгоритмов для выполнения основных функций, а также выполнение операций конфигурирования, программирования, управления базами данных и документирования. Кроме того, важно разработать алгоритмы, которые могут корректно обрабатывать возможные ошибки во входных данных и предпринимать соответствующие действия в таких ситуациях. При сохранении паролей должны использоваться современные алгоритмы кодировки.

###### 8.3.4.3.2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования. Данные, используемые системой, должны храниться в реляционной СУБД и на сервере веб-приложения. Структура базы данных определяется с учетом особенностей внутренней модели системы принятия решений. Информационный обмен между серверной и клиентской частями системы должен осуществляться по протоколу HTTP.

###### 8.3.4.3.3 Требования к лингвистическому обеспечению системы

ИС организации спортивных мероприятий должна быть реализована на русском языке. Система ввода-вывода должна поддерживать и русский язык.

###### 8.3.4.3.4 Требования к программному обеспечению системы

Программное обеспечение пользователя должно удовлетворять следующим требованиям: веб-браузер: Internet Explorer 10.0 и выше, или Firefox 10.0 и выше, или Opera 12 и выше, или Safari 14 и выше, или Chrome 88 и выше; включенная поддержка JavaScript и cookies.

###### 8.3.4.3.5 Требования к техническому обеспечению системы

Платформа, на которой будет развернут сервер системы, должна удовлетворять следующим минимальным требованиям:

* операционная система: Linux (желательно Ubuntu 16.04 или более поздней версии);
* веб-сервер: Nginx (желательно версии 1.14 или более поздней);
* серверное ПО: Python 3.10 или более поздней версии, фреймворк FastAPI версии 3.2.3 или более поздней;
* база данных: PostgreSQL версии 10 или более поздней;
* оперативная память (RAM): 4 ГБ или более;
* процессор: 64-разрядный;
* свободное место на жестком диске: 500 ГБ или более;
* сетевое соединение: 100 Мбит/с или более быстрое.

###### 8.3.4.3.6 Требования к метрологическому обеспечению системы

Требования к метрологическому обеспечению не предъявляются.

###### 8.3.4.3.7 Требования к организационному обеспечению системы

Требования к организационному обеспечению не предъявляются.

###### 8.3.4.3.8 Требования к методическому обеспечению системы

Необходимо разработать несколько типов руководств:

* руководство пользователя для администраторов интернет-магазина;
* руководство пользователя для модераторов интернет-магазина.

#### 8.3.5 Состав и содержание работ по созданию системы

Организация-исполнитель работ – ООО «Students».

Документом, подтверждающим согласие организации-исполнителя на участие в создании системы, является договором на разработку.

По окончании разработки каждого соответствующего этапа работ организация-исполнитель должна в письменной отчетной форме в свободном оформлении отчитываться организации заказчику.

Разработка системы предполагается по укрупненному календарному плану, приведенному в таблице 5.1.

Таблица 5.1 – Календарный план работ по созданию ИС BS

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Содержание работ** | **Сроки** |
| 1. Исследование и обоснование создания ИС | 1.1. Обследование (сбор и анализ данных) автоматизированного объекта, включая сбор сведений о зарубежных и отечественных аналогах | 16.02.2024 – 01.03.2024 |
| 2. Составление технического задания | 2.1. Разработка функциональных и нефункциональных требований к системе | 1.03.2024 – 20.03.2024 |
| 3. Эскизное проектирование | 3.1. Разработка предварительных решений по выбранному варианту ИС и отдельным видам обеспечения | 21.03.2024 – 28.03.2024 |
| 4. Техническое проектирование | 4.1. Разработка диаграмм | 29.03.2024 – 03.04.2024 |
| 4.2. Разработка макетов интерфейса | 04.04.2024 – 07.04.2024 |
| 5. Разработка программной части | 5.1. Разработка всех модулей ИС | 08.04.2024 – 01.05.2024 |
| 6. Предварительные комплексные испытания | 6.1. Проверка работоспособности системы в условиях, приближенных к реальным | 02.05.2024 – 06.05.2024 |
| 7. Опытная эксплуатация | 7.1. Эксплуатация с привлечением небольшого количества участников | 07.05.2024 – 10.05.2024 |
| 7.2. Устранение замечаний, выявленных при эксплуатации ИС | 11.05.2024 – 15.05.2024 |
| 8. Ввод в промышленную эксплуатацию | 8.1. Приемка ИС в промышленную эксплуатацию (внедрение ИС) | 16.05.2024 – 25.05.2024 |

#### 8.3.6 Порядок контроля и приемки системы

Система должна иметь модульную структуру, включающую в себя следующие модули:

Разработка системы предполагается по укрупненному календарному плану, приведенному в таблице 5.1.

В соответствии с разделом 5 необходимо на каждой стадии создания системы установить контроль и приемку результатов работ.

На стадии 5 происходит прием готовой версии программного продукта (модели), а остальные результаты работ представляются в виде документов согласно таблице 5.1.

Приемка этапа включает в себя рассмотрение и оценку объема работ и предоставленной технической документации в соответствии с требованиями технического задания.

Организацию и проведение приемки системы должен осуществлять заказчик, а приемка системы должна производиться только после того, как будут выполнены все задачи системы.

Заказчик обязан предоставить материальную часть (технические средства), проектную документацию и специально выделенный персонал.

Последним этапом при приемке системы является составление акта приемки.

#### 8.3.7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Для обеспечения готовности объекта к вводу системы в действие провести комплекс мероприятий:

1. установить все необходимые компоненты программного обеспечения (открытое программное обеспечение);
2. завершить работы по установке технических средств;
3. завершить работы по развертыванию ИС;
4. провести диагностику сети, подключенной к ИС;
5. провести обучение сотрудников.

##### 8.3.7.1 Приведение поступающей в систему информации к виду, пригодному для обработки с помощью ЭВМ

Информация вводится пользователем в разработанные экранные формы компонентов системы.

##### 8.3.7.2 Изменения, которые необходимо осуществить в объекте автоматизации

Не требуется дополнительных манипуляций.

##### 8.3.7.3 Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ

Для функционирования создаваемой системы требуется платформа, технические характеристики которой соответствуют предъявленным.

##### 8.3.7.4 Создание необходимых для функционирования системы подразделений и служб

Для функционирования системы не требуется дополнительных подразделений и служб.

##### 8.3.7.5 Сроки и порядок комплектования штатов и обучения персонала

Комплектование штатов подразделений и служб, необходимых для функционирования системы, а также подготовка их сотрудников должны быть завершены до начала опытной эксплуатации системы.

#### 8.3.8 Требования к документированию

Проектная документация должна быть разработана в соответствии с ГОСТ 34.201-2020 и ГОСТ 7.32-2017.

Отчетные материалы должны включать в себя текстовые материалы (представленные в виде бумажной копии и на цифровом носителе в формате MS Word) и графические материалы.

Предоставить документы:

1) схема функциональной структуры автоматизируемой деятельности;

2) описание технологического процесса обработки данных;

3) описание информационного обеспечения;

4) описание программного обеспечения ИС;

5) схема логической структуры БД;

6) руководство пользователя, модератора и администратора;

7) описание контрольного примера (по ГОСТ 24.102);

8) протокол испытаний (по ГОСТ 24.102).

#### 8.3.9 Источники разработки

− ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

− ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.

− ГОСТ Р 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.

− ГОСТ 34.201-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.

− ГОСТ Р 59795-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.

− ГОСТ 2.105-78. Единая система программной документации. Общие требования к программным документам.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения практических работ по проектированию информационной системы для онлайн книжного магазина были проведены анализ, проектирование и формирование ключевых элементов системы.

На основе заданий, предоставленных в рамках проекта, были разработаны структурные диаграммы системы, включая диаграмму классов и диаграмму объектов. Эти инструменты позволили нам более детально проработать структуру системы и ее компонентов.

Также были построены процессы проекта с использованием нотации IDEF0, что способствовало ясному представлению последовательности действий и взаимосвязей между процессами.

Создание информационной диаграммы системы и логической модели базы данных позволило нам более детально определить структуру данных, необходимых для функционирования системы. Это включало как диаграмму в нотации DFD, так и разработку нормализованной логической модели базы данных.

В разделе, посвященном архитектуре системы, была представлена общая концепция ее организации, что включало в себя различные компоненты и их взаимодействие, а также дополнение матрицы требований.

Завершающий этап работы заключался в формировании технического задания, которое включало выбор соответствующего ГОСТа и разработку самого задания, определяющего требования к конечной системе.

В целом, выполнение данных практических работ позволило глубже понять процесс проектирования информационных систем и приобрести практические навыки по применению соответствующих методов и инструментов проектирования.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по выполнению практических работ по дисциплине «Системная и программная инженерия» [Электронный ресурс]: Режим доступа: <https://online-edu.mirea.ru/mod/resource/view.php?id=463631> [Дата обращения: 28.02.2024];
2. Статья на Хабр «Проверенный шаблон пользовательских историй» [Электронный ресурс]: Режим доступа: <https://online-edu.mirea.ru/mod/resource/view.php?id=463631> [Дата обращения: 28.02.2024];
3. Статья на Flexberry «Диаграмма последовательности» [Электронный ресурс]: Режим доступа: <https://flexberry.github.io/ru/fd_sequence-diagram.html> [Дата обращения: 28.02.2024];
4. Статья на Яндекс Практикум «Варианты на все случаи жизни: как написать полезный use case» [Электронный ресурс]: Режим доступа: <https://practicum.yandex.ru/blog/chto-takoe-use-case-kak-ih-napisat/> [Дата обращения: 28.02.2024];
5. Статья на Хабр «Матрица трассабилити» [Электронный ресурс]: Режим доступа: <https://habr.com/ru/companies/simbirsoft/articles/412677/> [Дата обращения: 06.03.2024];
6. Статья на Ensi «Классификация требований» [Электронный ресурс]: Режим доступа: <https://ensi-platform.gitlab.io/analyst-guides/tools/classification/> [Дата обращения: 06.03.2024];
7. КонсультантПлюс - Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ [Электронный ресурс]: Режим доступа: <https://www.consultant.ru/document/cons_doc_LAW_61801/> [Дата обращения: 06.03.2024];
8. IDEF0. Знакомство с нотацией и пример использования [Электронный ресурс]: Режим доступа: https://www.trinion.org/blog/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya [Дата обращения: 14.03.2024];
9. Использование диаграммы классов UML при проектировании и документировании программного обеспечения [Электронный ресурс]: Режим доступа: https://habr.com/ru/articles/572234/ [Дата обращения: 14.03.2024];
10. What is Object Diagram? [Электронный ресурс]: Режим доступа:  [https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/](%20https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/) [Дата обращения: 14.03.2024];
11. DFD (Data Flow Diagram) Диаграммы — зачем они нужны и какие бывают [Электронный ресурс]: Режим доступа: <https://www.trinion.org/blog/chto-takoe-dfd-diagrammy-potokov-dannykh> [Дата обращения: 15.03.2024];
12. Моделирование данных: обзор [Электронный ресурс]: Режим доступа: https://habr.com/ru/articles/556790/ [Дата обращения: 17.03.2024];
13. Архитектура приложений и интеграции [Электронный ресурс]: Режим доступа: https://habr.com/ru/companies/itq\_group/articles/705598/ [Дата обращения: 18.03.2024];
14. Как составить ТЗ? [Электронный ресурс]: Режим доступа: https://www.sravni.ru/kursy/info/kak-sostavit-tz/ [Дата обращения: 19.03.2024];