



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

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Компьютерные системы и сети»

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ БАКАЛАВРА

НА ТЕМУ:

Система поддержки организации футбольного турнира

Студент ИУ6-85Б

(Подпись, дата)

Э.А. Степанов

(И.О. Фамилия)

Руководитель ВКР

(Подпись, дата)

А.А. Малахов

(И.О. Фамилия)

Нормоконтроллер

(Подпись, дата)

А.М. Суровов

(И.О. Фамилия)

2020

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

УТВЕРЖДАЮ

Заведующий кафедрой ИУ6

А.В. Пролетарский
«2» марта 2020 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы
бакалавра

Студент группы ИУ6-85Б

Степанов Эрнест Артурович
(Фамилия, имя, отчество)

Тема квалификационной работы Система поддержки организации
футбольного турнира

Источник тематики (НИР кафедры, заказ организаций и т.п.)

Инициативная НИР кафедры

Тема квалификационной работы утверждена распоряжением по факультету ИУ № 03.02.01-04.03/25 от «13» ноября 2019 г.

Часть I. Исследовательская

Исследовать существующие сервисы для организации массовых мероприятий и
проанализировать системы по обработке данных.

Часть 2. Конструкторская

Спроектировать структурную схему системы. Спроектировать бизнес-процессы и разработать функциональные диаграммы. Спроектировать диаграммы вариантов использования. Выбрать средство реализации разработки базы данных и спроектировать серверную часть системы. Выбрать язык программирования для реализации системы. Спроектировать основные алгоритмы работы системы. Спроектировать формы интерфейса.

Часть 3. Технологическая

Разработать технологию тестирования программной системы и провести тестирование системы.

Оформление квалификационной работы:

Расчетно-пояснительная записка на 55–65 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

1. Результаты анализа существующих систем;
2. Структурная схема информационной системы;
3. Функциональная модель IDEF0;
4. Схема структуры базы данных;
5. Диаграмма вариантов использования системы;
6. Результаты тестирования системы;
7. Формы интерфейса.

Дата выдачи задания «4» _сентября_ 2019 г.

В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до «1» июня 2020 г.

**Руководитель квалификационной
работы**

(Подпись, дата)

А.А. Малахов

(И.О. Фамилия)

Студент



Э.А. Степанов

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

ФАКУЛЬТЕТ Информатика и системы управления

УТВЕРЖДАЮ

КАФЕДРА Компьютерные системы и сети

Заведующий кафедрой ИУ6

ГРУППА ИУ6-85Б

А.В. Пролетарский

« 2 » марта 2020 г.

КАЛЕНДАРНЫЙ ПЛАН

выполнения выпускной квалификационной работы бакалавра

студента: Степанова Эрнеста Артуровича

(фамилия, имя, отчество)

Тема квалификационной работы Система поддержки организации футбольного турнира.

№ п/п	Наименование этапов выпускной квалификационной работы	Сроки выполнения этапов		Отметка о выполнении	
		план	факт	Должность	ФИО, подпись
1.	Задание на выполнение работы. Формулирование проблемы, цели и задач работы	<u>09.2019</u> <small>Планируемая дата</small>	<u>09.2019</u>	Руководитель ВКР	А.А. Малахов
2.	1 часть <u>Исследовательская</u>	<u>12.2019</u> <small>Планируемая дата</small>	<u>12.2019</u>	Руководитель ВКР	А.А. Малахов
3.	Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач	<u>02.2020</u> <small>Планируемая дата</small>	<u>02.2020</u>	Заведующий кафедрой	А.В. Пролетарский
4.	2 часть <u>Конструкторская</u>	<u>05.2020</u> <small>Планируемая дата</small>	<u>05.2020</u>	Руководитель ВКР	А.А. Малахов
5.	3 часть <u>Технологическая</u>	<u>05.2020</u> <small>Планируемая дата</small>	<u>05.2020</u>	Руководитель ВКР	А.А. Малахов
6.	1-я редакция работы	<u>05.2020</u> <small>Планируемая дата</small>	<u>05.2020</u>	Руководитель ВКР	А.А. Малахов
7.	Подготовка доклада и презентации	<u>06.2020</u> <small>Планируемая дата</small>	<u>06.2020</u>	Руководитель ВКР	А.А. Малахов
8.	Заключение руководителя	<u>06.2020</u> <small>Планируемая дата</small>	<u>06.2020</u>	Руководитель ВКР	А.А. Малахов
9.	Нормоконтроль	<u>06.2020</u> <small>Планируемая дата</small>	<u>06.2020</u>	Нормоконтролер	А.М. Суровов
10.	Внешняя рецензия	<u>06.2020</u> <small>Планируемая дата</small>	<u>06.2020</u>	Руководитель ВКР	А.А. Малахов
11.	Защита работы на ГЭК	<u>06.2020</u> <small>Планируемая дата</small>	<u>06.2020</u>	Руководитель ВКР	А.А. Малахов

Студент Степанов 2.03.2020
(подпись, дата)

Руководитель работы А.А. Малахов 2.03.2020
(подпись, дата)

АННОТАЦИЯ

В ходе выполнения выпускной квалификационной работы был описан процесс создания системы поддержки организации футбольного турнира при помощи OLAP системы.

Выполнен анализ существующих систем по организации массовых мероприятий. Также выполнен анализ существующих систем по обработке данных и был выбран тип HOLAP наиболее подходящий для организации турнира.

Для того, чтобы улучшить процесс организации турнира была разработана система поддержки. Система будет обрабатывать входные данные. Исходя из полученных данных и регламента турнира будут сформированы матчи футбольного турнира наиболее оптимальные, с учетом местоположения как участников, так и стадионов. Также система формирует и предоставляет данные о проведенных матчах и статистики игроков зрителю.

ABSTRACT

During the final qualifying work, the process of creating a support system for organizing a football tournament using the OLAP system was described.

The analysis of existing systems for the organization of public events. An analysis of existing data processing systems was also performed, and the HOLAP type that was most suitable for organizing the tournament was selected.

In order to improve the organization of the tournament, a support system was developed. The system will process the input. Based on the obtained data and the tournament regulations, the most optimal football tournament matches will be formed, taking into account the location of both participants and stadiums. The system also generates and provides data on matches and player statistics to the viewer.

РЕФЕРАТ

Записка 70 с., 45 рис., 14 табл., 14 ист., 4 прил.

ИССЛЕДОВАНИЕ, OLAP СИСТЕМА, HOLAP, МАССОВОЕ МЕРОПРИЯТИЕ, ФУТБОЛЬНЫЙ ТУРНИР, ОРГАНИЗАТОР, УЧАСТНИК, ЗРИТЕЛЬ.

Целью выпускной квалификационной работы является разработка системы поддержки организации футбольного турнира. Данная система предназначена для помощи организатору, а именно: для генерирования матчей футбольного турнира.

В процессе выполнения данной работы был произведен сравнительный анализ существующих систем по организации массовых мероприятий. Также были рассмотрены и проанализированы системы по обработке данных. На основе анализа сделан вывод о применимости OLAP к разрабатываемой системе. Была спроектирована и реализована система поддержки организации футбольного турнира, а также проведено функциональное и оценочное тестирование готового продукта.

Материалы работы представлены запиской, включающей введение, три главы основной части, заключение, список использованных источников, а также приложения: техническое задание на выпускную квалификационную работу бакалавра, руководство пользователя, фрагменты исходного кода, копии листов графической части.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1. Исследование существующих сервисов для организации футбольного турнира	8
1.1 Сравнительный анализ существующих решений по организации массовых мероприятий	8
1.2 Анализ существующих систем обработки данных	11
1.3 Принцип построения OLAP системы	15
1.3.1 Классификация OLAP систем	15
1.3.2 Витрина данных	20
ЗАКЛЮЧЕНИЕ ПО ПЕРВОЙ ГЛАВЕ	22
2. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ	23
2.1 Проектирование структурной схемы системы	23
2.2 Проектирование бизнес-процессов системы и разработка функциональных диаграмм	24
2.3 Проектирование диаграммы вариантов использования и уточнение функционала	30
2.4 Выбор средств реализации разработки бд и проектирование серверной части системы.....	40
2.4.1 SQL	40
2.4.2 Выбор языка программирования для реализации	44
2.4.3 JavaFX и JFoenix	45
2.4.4 Hibernate	47
2.4.5 Объектно-реляционное отображение	48
2.5 Проектирование основных алгоритмов работы системы	49
2.6 Анализ функционального состава информационной системы.....	52
2.7 Примеры форм интерфейса пользователя	54
ЗАКЛЮЧЕНИЕ ПО ВТОРОЙ ГЛАВЕ	59
3 РАЗРАБОТКА ТЕХНОЛОГИИ ТЕСТИРОВАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ	60
3.1 Тестирование «Черный Ящик»	60
3.2 Оценочное тестирование	63
ЗАКЛЮЧЕНИЕ ПО ТРЕТЬЕЙ ГЛАВЕ	66
ЗАКЛЮЧЕНИЕ	67

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	68
ПРИЛОЖЕНИЕ А	70
ПРИЛОЖЕНИЕ Б.....	71
ПРИЛОЖЕНИЕ В	72
ПРИЛОЖЕНИЕ Г.....	73

Введение

Массовое мероприятие — это организованная, активная форма реализации прав, свобод и законных интересов больших групп (масс) людей в общественных местах, а также способ удовлетворения экономических, политических, культурных, религиозных и других потребностей граждан.

На сегодняшний день проводится большое количество массовых мероприятий: концерты, спортивные соревнования, городские праздники, выставки и т.д.

Человеку, который руководит организацией массового мероприятия или группе людей, принимающей участие, сложно ориентироваться в потоке информации без использования каких-либо приемов по систематизации и представлению полученных данных. Также еще одной особенностью является частота сменяемости информации, что следует учитывать при составлении плана или регламента проведения мероприятия. Именно поэтому система, которая позволяет накопить, обработать и в любой момент времени представить данные организатору является актуальной с точки зрения получения конкретного и максимально быстрого эффекта, слаженности, отсутствия временных накладок, эффективного действия всех участников. Эти условия требуют быстрой обработки данных и моментального решения полученных задач, для этого будет использована OLAP система.

Для успешной разработки системы необходимо произвести анализ существующих систем с похожим функционалом. В связи с тем, что затрагиваемая в данной работе проблема является узконаправленной, рассмотрим системы по организации массовых мероприятий: Eventbrite, Smart Sheet, Eventleaf. Анализ перечисленных систем показал, что эти системы имеют недостатки такие как: медленная обработка входящих данных, отсутствие генерирования матчей, а также высокая стоимость использования системы.

Таким образом необходимо создать систему, которая будет обладать всеми необходимыми функциями для организации футбольного турнира. Программа разработана на языке Java в IntelliJ IDEA.

Решаемые задачи при разработке выпускной квалификационной работы:

1. Анализ существующих систем;
2. Анализ систем обработки данных;
3. Проектирование структурной схемы и разработка функциональных диаграмм;
4. Выбор средств реализации разработки и проектирование серверной части системы;
5. Выбор языка программирования;
6. Проектирование основных алгоритмов системы;
7. Разработка технологий тестирования и тестирование системы.

1. Исследование существующих сервисов для организации футбольного турнира

1.1 Сравнительный анализ существующих решений по организации массовых мероприятий

Так как решаемая в данной работе проблема – организация футбольного турнира, является довольно узкоспециализированной, то для сравнения и анализа были выбраны системы, имеющие в своем функционале возможность организовывать массовые мероприятия, что будет являться аналогом разрабатываемой системы [1].

В процессе анализа был произведён поиск разрабатываемых систем в сети Интернет. По результатам поиска были выбраны три системы, имеющие наиболее близкий набор функций:

1. Eventbrite — онлайн-инструмент для организации мероприятий. Для бесплатных мероприятий использование Eventbrite ничего не стоит. Небольшая комиссия взимается за продажу билетов с организатора. Сервис позволяет спланировать мероприятие, время и место проведения, а также провести регистрацию участников. Логотип сервиса расположен на рисунке 1.



Рисунок 1 - Логотип сервиса Eventbrite

2. Smart Sheet – ведущая в мире платформа по управлению и автоматизации совместной работы. Пользователям предлагается максимальная гибкость в управлении логикой проекта с помощью знакомого и простого в использовании интерфейса электронной таблицы. Логотип сервиса расположен на рисунке 2.



Рисунок 2 - Логотип сервиса Smart Sheet

3. Eventleaf — этот сервис служит бесплатным инструментом для отправки индивидуальных приглашений, онлайн-регистрации участников. За небольшую комиссию платформа предлагает интеграцию с платежными системами для оплаты посещения мероприятия. Сервис Eventleaf универсален. Логотип сервиса расположен на рисунке 3.



Рисунок 3 - Логотип сервиса Eventleaf

Проведем анализ систем по следующим критериям:

- доступность информации о системе;
- простота использования;
- стоимость использования;
- возможность проведения масштабного мероприятия;
- быстрая обработка данных;
- удобная система регистрации.

Выбор критериев обусловлен предметной областью и техническими возможностями для проведения анализа.

Таблица 1 - Анализ существующих систем

№	Критерий	Сервис		
		Eventbrite	Smart Sheet	Eventleaf
1	Доступность информации о системе	-	+	+
2	Простота использования	+	-	-
3	Стоимость использования	-	-	-
4	Возможность проведения масштабного мероприятия	+	+	+
5	Быстрая обработка данных	+	+	-
6	Удобная система регистрации	+	+	+

После проведенного анализа существующих были выявлены следующие недостатки. Это необходимо учесть при создании новой системы, так как уже существующие системы не отражают нужную информацию, которая будет

полезна как организатору, при составлении матчей, так и участнику, при удобном процессе регистрации и связи с организатором, так и зрителю, который хочет просмотреть информацию о турнире или купить билет на матч.

На примере систем по организации массовых мероприятий можно сделать вывод, что нет системы, для организации футбольного турнира. Есть компании, которые могут предоставить услуги по организации мероприятий, но они будут платные.

Автоматизация процесса организации футбольного мероприятия заключается в том, что система должна быстро и моментально обрабатывать полученные данные с минимальными рисками, заключающимися в том, что участнику может быть неудобно место и время проведения матча в связи с их расположением, и предоставлять сгенерированный список времени и места проведения матчей, наиболее подходящий участникам с точки зрения их логистики. Также отображение результата матча и статистики игроков сразу же после его завершения.

Для того, чтобы улучшить существующую структуру организации футбольного мероприятия, была придумана система, которая должна реализовывать такие функции, такие как:

- Регистрацию для участников, стадионов, трансферных компаний и зрителей;
- Обработку и анализ полученных данных;
- Составление таблицы предстоящих матчей;
- Предоставление доступа к данным о командах, стадионах, трансферных компаний и о предстоящих матчах организатору;
- Просмотр данных о футбольном турнире пользователю;
- Отчет о проведенных матчах;
- Итоги футбольного турнира.

1.2 Анализ существующих систем обработки данных

Для разработки системы поддержки организации футбольного нужно будет использовать одну из уже существующих систем.

OLTP (Online Transaction Processing)— обработка данных в реальном времени. Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы минимальное время отклика [2].

OLAP (Online Analytical Processing) – аналитическая обработка данных в реальном времени) — это ключевой компонент организации традиционных хранилищ данных [3].

Приведем сравнение решаемости задач между двумя системами OLTP и OLAP, наиболее подходящих для создания новой системы.

Таблица 2 - Решаемые задачи двух систем

Характеристика	OLTP	OLAP
Частота обновления данных	Высокая частота	Малая частота
Источники данных	В основном, внутренние	По отношению к аналитической системе, в основном, внешние
Возраст данных	Текущие	Любой
Уровень агрегации данных	Детализированные данные	В основном агрегированные данные
Возможности аналитических операций	Регламентированные отчеты	Последовательность интерактивных отчётов, динамическое изменение уровней агрегаций и срезов данных
Назначение системы	Фиксация, оперативный поиск и обработка данных, регламентированная аналитическая обработка	Работа с любым видом данных, аналитическая обработка, прогнозирование, моделирование
Объем данных	Малый	Большой

Из таблицы 2 можно сделать вывод, что для организации и проведения футбольного турнира следует использовать OLAP систему, так она обладает следующими преимуществами и функциями:

- Анализ данных;
- Сложные транзакции;
- Имеет большой срок хранения данных;
- Имеет возможность использования разных типов данных;
- Обрабатывает большое количество данных;
- Имеет возможность прогнозирования и моделирования.

Основная технология, с помощью которой будет реализована система –

Это OLAP система. Применение такой технологии позволит быстро анализировать полученные данные и предоставлять их организатору.

Для эффективного хранения информации используются операционные базы данных (OLAP) и соответствующие программные средства СУБД используются как можно шире. Вы можете извлечь данные из базы данных, организовав запрос на языке SQL.

Однако во многих случаях сложно принимать организационные решения на основе информации из OLTP-системы. Попытка качественно и всесторонне проанализировать большой объем оперативной информации особенно трудна. Поэтому ретроспективные данные теперь хранятся в отдельных базах данных, которые доступны только для чтения и дальнейшей аналитической обработки.

Хранилище данных — это оптимально организованная база данных, содержащая данные, сгруппированные по многим измерениям и обеспечивающая быстрый доступ к информации, необходимой для принятия управленческих решений. Агрегаты (или суммарные показатели) хранятся в явном виде, чтобы ускорить выполнение аналитических запросов. База данных регулярно обновляется из различных внешних источников, включая статистические отчеты о футбольных матчах [4].

Системы поддержки принятия решений обычно имеют средства для предоставления пользователю агрегированных данных для различных выборок из исходного файла в форме, удобной для восприятия и анализа (таблицы, диаграммы и т. д.). Традиционный подход к сегментации исходных данных заключается в извлечении одного или нескольких многомерных наборов данных из исходных данных.

Концепция OLAP была описана в 1993 году Эдгаром Коддом, известным исследователем баз данных и автором реляционной модели данных. В 1995 году на основе требований, изложенных Коддом, был сформулирован так называемый тест FASMI (Fast Analysis of Shared Multidimensional Information — быстрый анализ разделяемой многомерной информации), включающий следующие требования к приложениям для многомерного анализа:

- предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;

- возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;

- многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;

- многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);

- возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

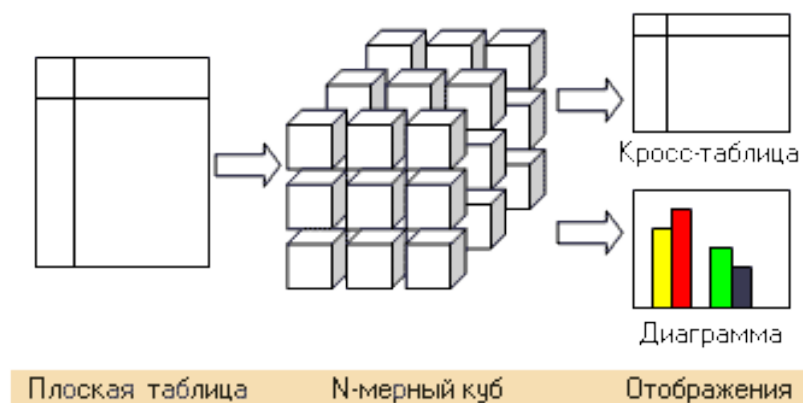


Рисунок 4 - Общая схема OLAP

Построение на технологии OLAP, предоставляет практически безграничные возможности по составлению отчетов, выполнению сложных аналитических расчетов, построению прогнозов и сценариев, разработке множества вариантов планов. Также применение данной системы позволяет автоматизировать стратегический уровень управления организацией.

В основе работы OLAP системы лежит обработка многомерных массивов данных. Многомерные массивы устроены так, что каждый элемент массива имеет множество связей с другими элементами. Чтобы сформировать массив данных система должна получить данные из других систем или через внешний ввод. Пользователь OLAP системы получает необходимые данные в структурированном виде в соответствии со своим запросом [4].



Рисунок 5 - Принцип OLAP системы

Для организации массовых мероприятий важны свойства обработки данных и работы с данными в реальном времени, поэтому OLAP система

позволит организаторам массовых мероприятий эффективнее проводить обработку большого количества информации.

1.3 Принцип построения OLAP системы

Механизм OLAP является на сегодня одним из популярных методов анализа данных. Каждый из этих подходов имеет свои плюсы и минусы.

Такая задача возникла после применения ROLAP системы, построенной на основе компонентов Decision Cube, входящих в состав Borland Delphi. К сожалению, использование этого набора компонентов показало низкую производительность на больших объемах данных. Остроту этой проблемы можно снизить, стараясь отсечь как можно больше данных перед подачей их для построения кубов, но этого не всегда бывает достаточно.

В Интернете и прессе можно найти много информации об OLAP системах, но практически нигде не сказано о том, как система устроена внутри. Поэтому решение большинства проблем нам давалось методом проб и ошибок [5].

1.3.1 Классификация OLAP систем

На сегодняшний день в мире разработано множество продуктов, реализующих OLAP-продукцию: по способу хранения данных для анализа и по месту нахождения OLAP-машины.

Таблица 3 - Виды OLAP систем

Тип OLAP	Значение
Реляционная OLAP (ROLAP)	ROLAP — это расширенная СУБД вместе с многомерным отображением данных для выполнения стандартной реляционной операции.
Многомерный OLAP (MOLAP)	MOLAP - реализует работу в многомерных данных.
Гибридная онлайн-аналитическая обработка (HOLAP)	В подходе HOLAP агрегированные итоговые значения хранятся в многомерной базе данных, а подробная информация хранится в реляционной базе. Это обеспечивает как эффективность модели ROLAP, так и производительность модели MOLAP

Продолжение таблицы 3

Веб-OLAP (WOLAP)	Web OLAP является системой OLAP, доступной через веб-браузер. Он состоит из трех компонентов: клиент, промежуточное программное обеспечение и сервер базы данных
Мобильный OLAP	Мобильный OLAP помогает пользователям получать и анализировать данные OLAP с помощью своих мобильных устройств

Исследование предметной области приводит нас к тому, что многомерные кубы строятся на основе исходных и агрегатных данных. И исходные, и агрегатные данные для кубов могут храниться как в реляционных, так и многомерных базах данных. Поэтому в настоящее время применяются три способа хранения данных: MOLAP (Multidimensional OLAP), ROLAP (Relational OLAP) и HOLAP (Hybrid OLAP). Соответственно, OLAP-продукты по способу хранения данных делятся на три аналогичные категории:

В случае **MOLAP**, исходные и агрегатные данные хранятся в многомерной БД или в многомерном локальном кубе.

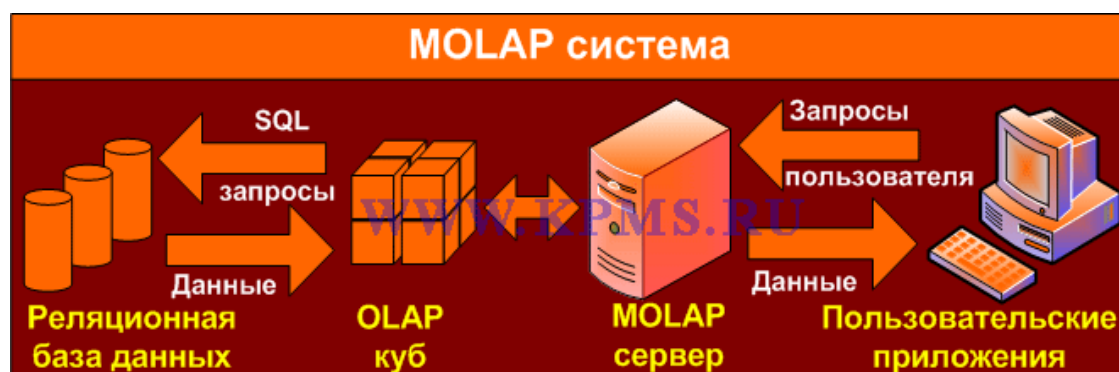


Рисунок 6 - MOLAP

В **ROLAP**-продуктах исходные данные хранятся в реляционных БД или в плоских локальных таблицах на файл-сервере. Агрегатные данные могут помещаться в служебные таблицы в той же БД. Преобразование данных из реляционной БД в многомерные кубы происходит по запросу OLAP-средства.

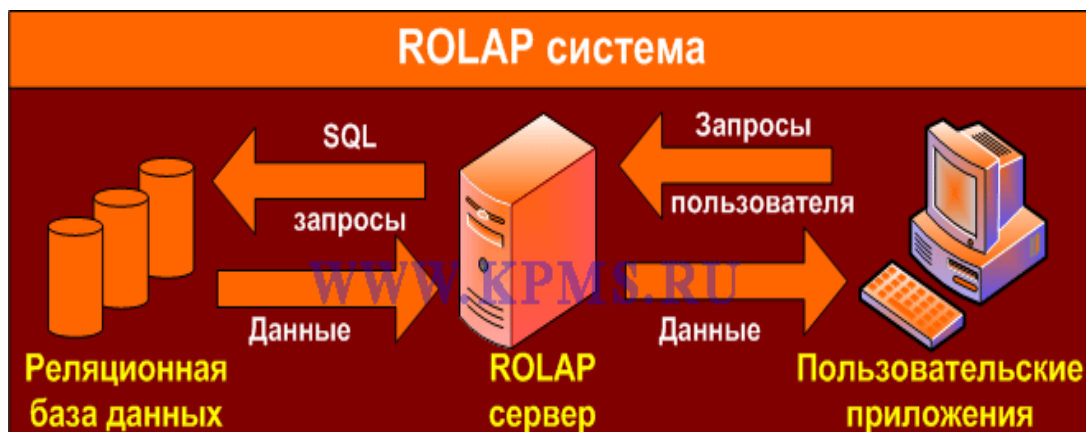


Рисунок 7 - ROLAP

В случае использования **HOLAP** архитектуры исходные данные остаются в реляционной базе, а агрегаты размещаются в многомерной. Построение OLAP-куба выполняется по запросу OLAP-средства на основе реляционных и многомерных данных.

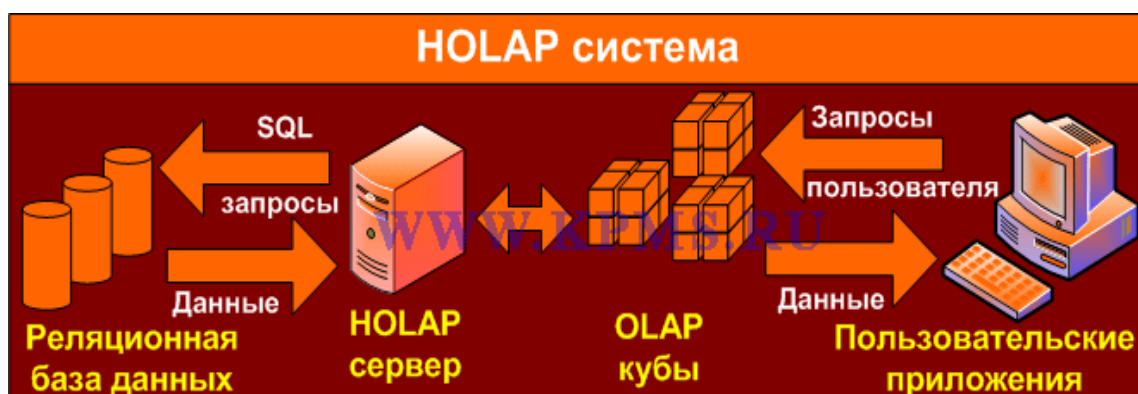


Рисунок 8 - HOLAP

Следующая классификация - по месту размещения OLAP-машины. По этому признаку OLAP-продукты делятся на OLAP-серверы и OLAP-клиенты:

В серверных OLAP-средствах вычисления и хранение агрегатных данных выполняются отдельным процессом - сервером. Клиентское приложение получает только результаты запросов к многомерным кубам, которые хранятся на сервере. Некоторые OLAP-серверы поддерживают хранение данных только в реляционных базах, некоторые - только в многомерных. Многие современные OLAP-серверы поддерживают все три способа хранения данных: MOLAP, ROLAP и HOLAP.

OLAP-клиент устроен по-другому. Построение многомерного куба и OLAP-вычисления выполняются в памяти клиентского компьютера. OLAP-

клиенты также делятся на ROLAP и MOLAP. А некоторые могут поддерживать оба варианта доступа к данным.

У каждого из этих подходов, есть свои "плюсы" и "минусы". Вопреки распространенному мнению о преимуществах серверных средств перед клиентскими, в целом ряде случаев применение OLAP-клиента для пользователей может оказаться эффективнее и выгоднее использования OLAP-сервера. Далее мы подробнее остановимся на сильных сторонах клиентских OLAP-средств. В качестве примера будем использовать OLAP-клиент со встроенной OLAP-машиной "Контур Стандарт" разработки российской компании Intersoft Lab.

OLAP-клиент - OLAP-сервер: "за" и "против"

При построении информационной системы OLAP-функциональность может быть реализована как серверными, так и клиентскими OLAP-средствами. На практике выбор является результатом компромисса эксплуатационных показателей и стоимости программного обеспечения. Посмотрим, из чего складывается покупательская привлекательность инструментов класса OLAP-клиент, и на что следует обратить внимание, отдавая предпочтение этому классу OLAP-продуктов [6].

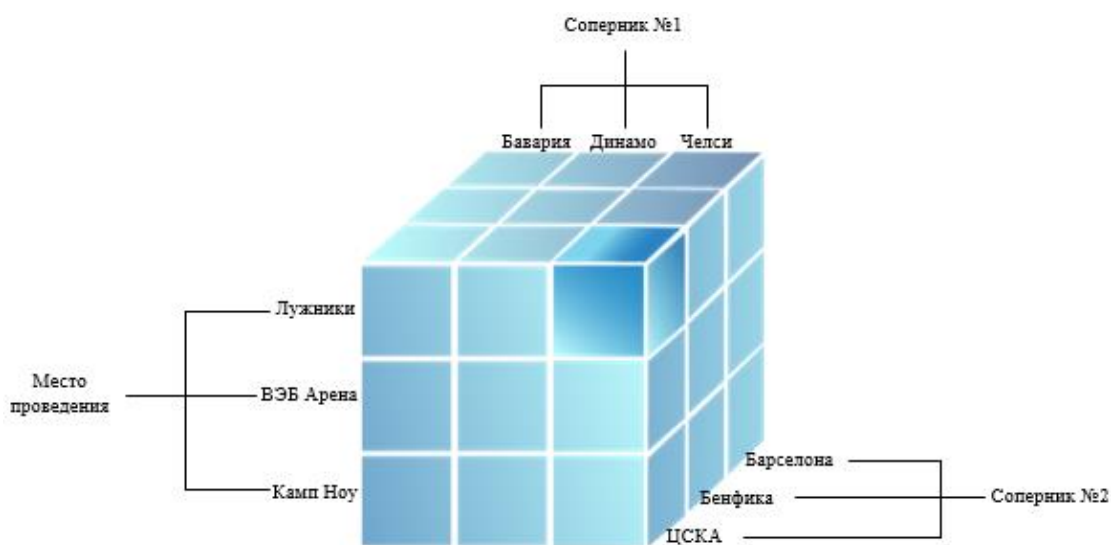


Рисунок 9 - Пример реализации данных при помощи OLAP куба

Из рисунка 9 можно сделать вывод, что наиболее удобный для двух участников матч будет проходить на Стадионе «Лужники», а соперниками будут команды Челси и ЦСКА.

Проведем анализ систем (ROLAP, MOLAP, HOLAP) для выявления системы, с помощью которой будет в дальнейшем произведена разработка системы.

Таблица 4 - Виды OLAP систем.

№	Критерий	Вид OLAP системы		
		MOLAP	ROLAP	HOLAP
1.	Безопасность и администрирование обеспечивается реляционными СУБД	+	+	+
2.	Возможность манипулирования большими объемами данных	-	+	+
3.	Эффективность	+	-	+
4.	Производительность	-	+	+
5.	Возможность использования системы с хранилищами данных и различными OLTP-системами	+	+	+
6.	Быстрая скорость обработки запросов	-	+	+
7.	Сложно изменять данные без повторной агрегации.	+	+	+
8.	Нет необходимости в специальных инструментах для формирования кубов	+	-	+

По результатам анализа всех систем, можно сделать вывод, что наиболее удобной системой для организации футбольного турнира является тип HOLAP. В подходе HOLAP агрегированные итоговые значения хранятся в многомерной базе данных, а подробная информация хранится в реляционной базе. Это обеспечивает как эффективность модели ROLAP, так и производительность модели MOLAP.

1.3.2 Витрина данных

Идея создания витрин данных (ВД) была предложена в 1991 году Forrester Research. Авторы представляли данное хранилище информации как определенное множество специфических баз данных, которые содержат в себе сведения, относящиеся к конкретным векторам деятельности [7].

Forrester Research выделяли следующие сильные стороны своего проекта – витрин данных:

- Представление аналитикам только той информации, которая действительно нужна для определенного рабочего задания, профиля служебной деятельности.

- Максимальная приближенность целевой части хранилища данных к конкретному пользователю.

- Содержание тематических подмножеств, заранее агрегированных специалистами данных, которые в дальнейшем проще настраивать и проектировать.

- Для реализации витрины данных (хранилища данных специализированного типа) не требуется вычислительная техника большой мощности.

Процесс получения витрины данных изображен на рисунке 10.

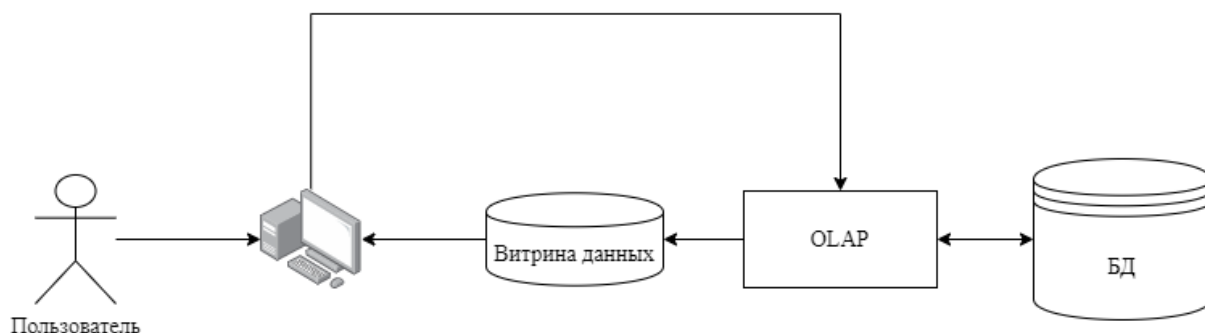


Рисунок 10 - Процесс получения Витрины Данных

Процесс получения витрины данных проходит следующим образом:

1. Пользователь отправляет запрос на получение информации через ПК в OLAP, которая ему необходима;
2. OLAP система обрабатывает полученный запрос и обращается к базе данных за нужной информацией;
3. База данных выдает информацию, которая была запрошена OLAP системой;
4. OLAP система предоставляет полученные данные пользователю через витрину данных;
5. Пользователь просматривает запрошенные данные.

Пример такого запроса может быть:

- Список команд;
- Список всех игроков, которые в одной команде или во всей системе;
- Список матчей;
- Результаты матчей или результативность игроков.

Выделим ключевые преимущества витрин данных:

- Максимально ориентированы на пользователя, предоставляют ему только ту информацию, которая необходима при выполнении служебного задания.
- Значительно меньше “весят”, нежели базы данных.
- Содержат в себе агрегированную информацию по определенным тематикам.
- Достаточно быстрое внедрение витрин данных.
- Создание для ответов на конкретный ряд вопросов.
- Данные оптимизированы для использования определенным кругом пользователей. Это облегчает процедуру наполнения ВД, способствует увеличению производительности системы.

Заключение по первой главе

В результате анализа было выявлено что существующие системы не отражают той специфики, которая необходима при организации футбольного турнира. Разрабатываемая система, будет удобна для организатора, потому что она будет бесплатной, удобной в использование, к тому же при применении OLAP системы будут быстро обрабатываться данные, генерироваться матчи с учетом местоположения стадионов и предоставлять отчет о проведенных матчах организатору и зрителям. Был выбран тип HOLAP, так как он объединяет в себя как MOLAP, так и ROLAP тем самым лучше всего подходит, для организации футбольного турнира.

2. Разработка информационной системы

2.1 Проектирование структурной схемы системы

Организация процесса проведения футбольного турнира без применения системы показана на рисунке 11.

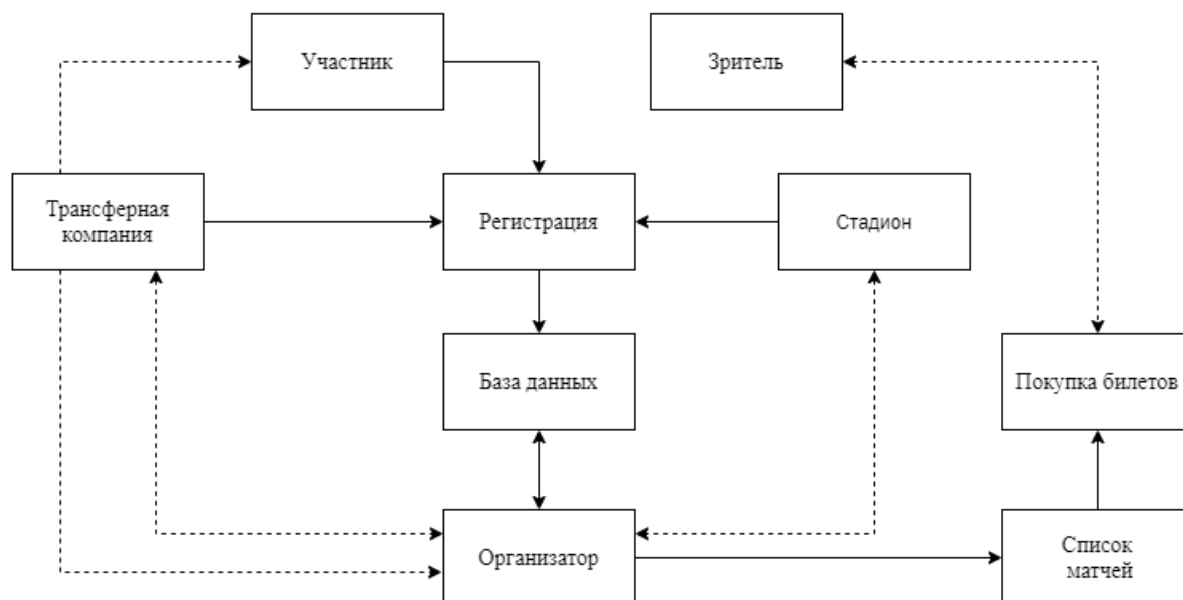


Рисунок 11 - Структурная схема проведения турнира без OLAP

Из структурной схемы, изображенной на рисунке 11 можно увидеть, что такими процессами являются:

- Регистрация участников;
- Регистрация стадионов;
- Регистрация транспортных компаний;
- Составление списка матчей.

Этими процессами занимается непосредственно организатор и затрачивает много времени на обработку и анализ полученных данных для наиболее удачного составления плана проведения турнира.

С применением технологии, которую предоставляет OLAP система процесс организации футбольного турнира становится менее время затратным и более автоматизированным процессом так как вместо организатора обработкой и анализом данных будет заниматься сама система.

На рисунке 12, представленном ниже, показана структурная схема футбольного турнира с применением OLAP системы.

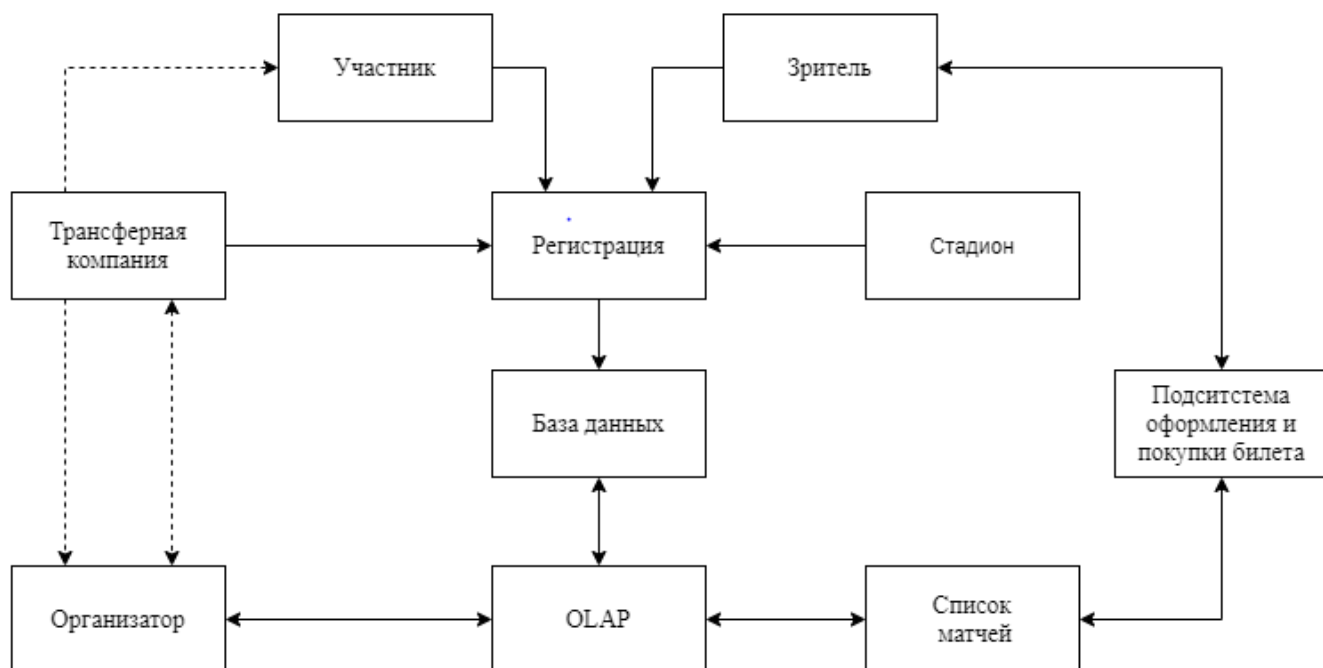


Рисунок 12 - Структурная схема футбольного турнира

На схеме показан структура проведения футбольного турнира и связи между объектами ПО. Как видно из представленной схеме на рисунке 12 процесс организации футбольного турнира будет происходить в большей степени при помощи OLAP системы. Основные процессы регистрации и составления списков матчей будет происходить автоматически, при этом будут учитываться логистические особенности как участников, так и стадионов. Данные, которые были переданы в OLAP систему могут быть просмотрены организатором этого турнира

2.2 Проектирование бизнес-процессов системы и разработка функциональных диаграмм

Для того, чтобы разобраться в функциональных характеристиках проектируемой системы и рассмотреть все ее бизнес-процессы, нужно построить функциональную диаграмму.

В разрабатываемой системе присутствует 3 типа пользователей – организатор (тот, кто организует турнир и взаимодействует с OLAP системой), участник (команда, которая участвует в турнире) и зритель (тот, кто

приобретает билеты, чтобы посетить турнир на стадионе). Рассмотрим систему с этих точек зрения.

Существующая модель демонстрирует организацию футбольного турнира. Этот процесс регламентируется правилами проведения организации футбольного турнира. Проведение турнира ведется под контролем организатора. Также в процессе организации турнира задействованы программные средства. В течении всего турнира проводится несколько этапов, после каждого этапа определяются победившие и проигравшие команды, первые продолжают участие, проходя в следующий этап, проигравшие команды выбывают. По результатам проведения турнира определяется победитель.

В качестве входных данных в соответствии с предъявляемыми требованиями выделим:

Список команд – перед началом турнира, участником необходимо пройти процесс регистрации;

В качестве управляющих данных правила проведения турнира.

В качестве механизмов выделим информационную (разрабатываемую) систему и СУБД (системы управления базами данных).

В качестве выходных данных выделим:

1. Список участников;
2. Местоположение участников;
3. Список предстоящих матчей;
4. Результаты сыгранных матчей и отдельно результативность самих игроков.

На основе этих данных можно сделать анализ и провести футбольный турнир.

На основе вышеперечисленных данных была разработана начальная контекстная диаграмма IDEF0 с точки зрения организатора.



Рисунок 13 - IDEF0 диаграмма с точки зрения организатора

При детализации диаграммы А0 выделим основные функциональные блоки:

- Регистрация (составление списка команд);
- Футбольный турнир (проведение футбольных матчей);
- Итоги турнира (подведение итогов).

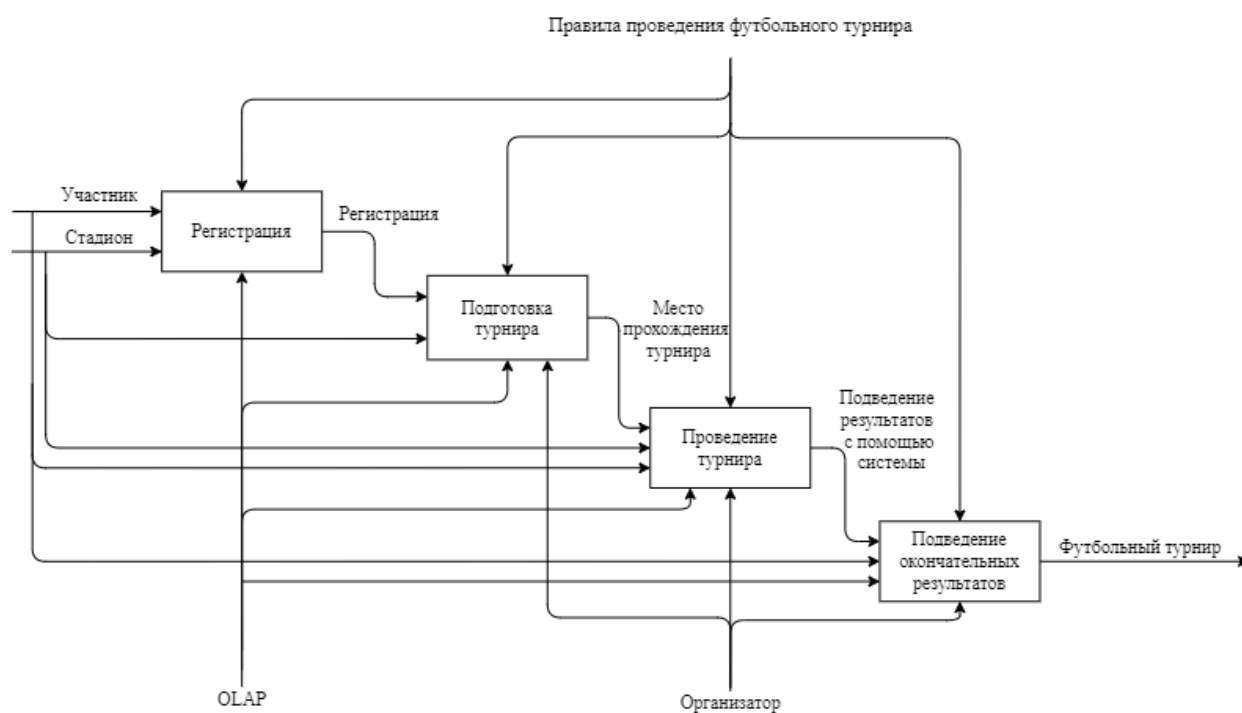


Рисунок 14 -Модель ТО-ВЕ, декомпозиция “А0. Футбольный турнир”

С помощью диаграммы, приведённой на рисунке 14 изображен процесс проведения футбольного турнира при помощи OLAP системы. Перед началом проведения турнира участникам и стадионам необходимо пройти процесс регистрации, после чего при помощи системы будут сформированы команды противников и места проведения матчей. Вторым этапом организатор при помощи OLAP системы подготавливает место и время проведения футбольного матча. После прохождения предыдущих этапов проводится турнир. Затем после завершения последнего матча в турнире, определяется победитель и турнир завершается.

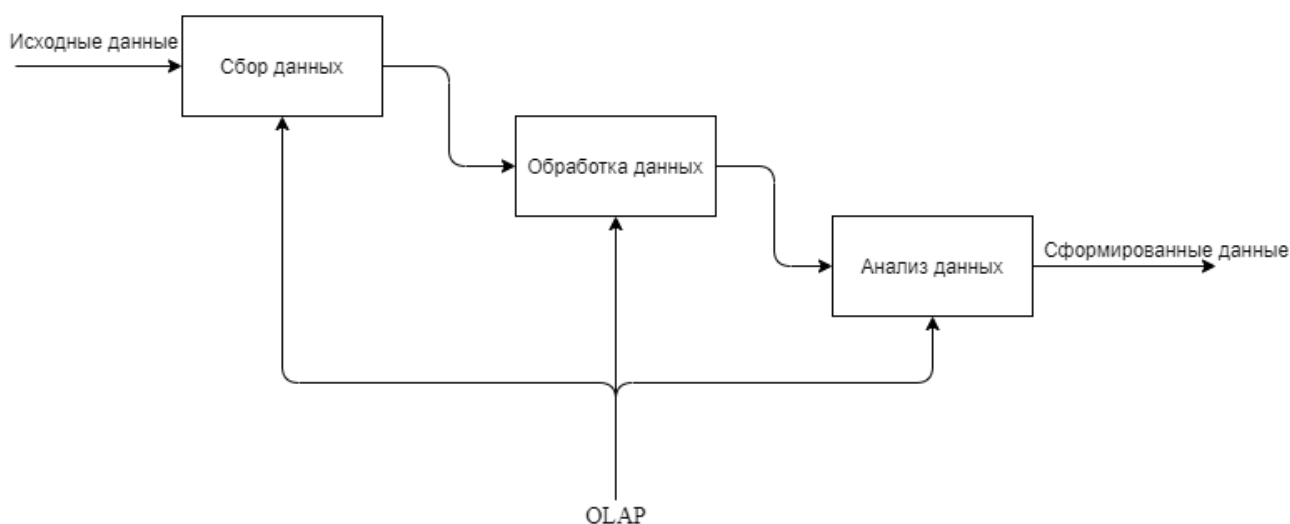


Рисунок 15 - Модель TO-BE, декомпозиция “A1. Регистрация”

На рисунке 15 представлена диаграмма формирования данных. Целью этого этапа является загрузка данных об участниках и о стадионах в систему.

После прохождения регистрации OLAP система обрабатывает и анализирует данные, а затем сформировывает полученные ранее данные, для проведения футбольного турнира.



Рисунок 16 - Модель TO-BE, декомпозиция “А2 Подготовка турнира”

На данной диаграмме представлен этап «подготовка к проведению футбольного турнира». Процесс начинается с получения заявок участниками турнира и затем принятии решения о дальнейшем участии, каждой команды в турнире. Затем, при помощи OLAP системы происходит распределение игр между участниками. После распределения игр между участниками команд, организатор при помощи OLAP системы определяет наилучшее время и место для проведения футбольных матчей и создается список предстоящих матчей.

Далее происходит процесс проведения футбольного турнира. Ниже на рисунке детально расписан процесс «Проведения турнира».



Рисунок 17 - Модель ТО-БЕ, декомпозиция “А3 Проведение турнира”

На рисунке 17 представлена диаграмма «Проведение турнира». Система после обработки данных выдает список матчей и стадион, на котором будут проводиться игры между командами. Затем по окончании одного из этапов футбольного турнира, на котором определяются победители, которые проходят в следующий этап, система получает данные о прошедших играх, для того, чтобы их проанализировать для дальнейшего определения соперников и составления новых списков предстоящих матчей. По итогу всех проведенных игр на футбольном турнире система определяет и подводит результаты турнира и на выходе выдает итоги матчей на прошедшем турнире.

На рисунке, представленном ниже показан заключительный этап проведения футбольного турнира «подведение окончательных результатов».



Рисунок 18 - Модель ТО-ВЕ, декомпозиция “А4 Подведение окончательных результатов”

После проведения финального матча турнира организатор, при помощи OLAP системы определяет победителя турнира. Затем идёт подготовка организации церемонии награждения участников финального матча. Когда все готово, происходит награждение команды и футбольный турнир считается завершённым.

Работа со зрителем заключается только в регистрировании их для возможности покупки билета на матчи, во избежание предотвращения перепродажи билетов иными лицами.

На этапе рассмотрения моделей проведения турнира, можно сделать вывод, что критичными этапами с точки зрения быстрой обработки и предоставления информации пользователю (организатору) являются этапы: обработка входящих данных (Рисунок 15), генерирование списка игр между участниками (Рисунок 16), а также анализ матча (Рисунок 17).

2.3 Проектирование диаграммы вариантов использования и уточнение функционала

Каждый объект обладает своим набором данных и функций. Данные могут как повторяться, так и быть совершенно уникальными для каждого объекта, в предметной области.

Организатор (или группа лиц) обладает следующим набором данных:

1.ФИО

2.Должность на предстоящем турнире:

- 1) главный организатор
- 2) куратор команд
- 3) менеджер по логистики

3.Обязанности относительно выбранной должности:

1) Главный организатор следит за всем проходящем турнире и выборе места, даты и времени проведения матча.

2) Куратор команды общается с командами по разным вопросам, которые их интересуют.

3) Менеджер по логистики занимается вопросами транспортировки команд до места проведения матча и общается с трансферной компанией.

Пример объекта «Организатор» представлен на рисунке 19.

Организатор
ФИО
Должность
Обязанности

Рисунок 19 - Объект организатор

Для участия в футбольном турнире участнику необходимо пройти процесс регистрации.

Участник, обладает таким набором данных:

1.Название команды.

2.Штаб. В штаб команды входят: Главный тренер, помощники тренера, медицинский штаб.

3.Состав команды

4.Местоположение (из какого города команда).

5.Трансфер, при его необходимости.

Пример объекта «Участник» представлен на рисунке 20.

Участник
Название
Штаб
Состав
Местоположение
Трансфер

Рисунок 20 - Объект участник

Опишем более подробно взаимодействие Участника с Организатором.



Рисунок 21 - Взаимодействие Организатора и участника

На рисунке 21 изображено взаимодействие организатора с участниками. Организатор во время всего турнира связывается со всеми участниками, поэтому связь между организатором и участниками имеет значение равное n .

После регистрации, участник передает название своей команды организатору. По названию команды организатор может узнать какое количество человек входит в команду и требуется ли команде трансфер до места проведения матча. Далее организатор отправляет участнику дату и время проведения футбольного матча, а также соперника, с которым предстоит сыграть после того, как получит эти данные от OLAP системы. Если участнику необходим трансфер до места проведения матча, то организатор сообщает время, к которому будет их ждать автобус. Алгоритм взаимодействия участника с организатором представлен на рисунке 22.



Рисунок 22 - Алгоритм взаимодействия организатора с участниками

Трансферная компания проходит процесс регистрации, для того чтобы у организатора, который отвечает за логистику был список водителей и машин.

Трансферная компания обладает данными:

- Название компании;
- Администратор компании;
- Список водителей;
- Список машин.

Трансферная компания
Название компании
Администратор
Список водителей
Список машин

Рисунок 23 - Трансферная компания

Опишем более подробно взаимодействие организатора с трансферной компанией.



Рисунок 24 - Взаимодействие организатора и трансферной компании

Организатор, который занимается перевозкой участников, связывается с администратором трансферной компании в случае, если есть необходимость по перевозке команды до стадиона, на котором будет проводиться матч. Так же, организатор сообщает администратору время, к которому необходимо предоставить машину. Далее, администратор отправляет организатору данные о машине и водителе, который будет заниматься перевозкой участника.

Стадион обладает следующим набором данных:

- Название;
- Местоположение стадиона;
- Вместимость стадиона;
- Судьи.

Стадион
Название
Местоположение
Вместимость
Судьи

Рисунок 25 - Стадион

Стадиону, чтобы принимать на своем поле матчи футбольного турнира, также необходима регистрация. При регистрации, необходимо указать название

арены, вместимость, местоположение и бригаду судей. Местоположение указывается для того, чтобы эти данные отправить в трансферную компанию. Вместимость указывается для того, чтобы система могла регулировать процесс продажи билетов. На каждом стадионе будет своя бригада арбитров, которые судят матчи, но только те, которые проходят именно на том стадионе, на котором они зарегистрированы.

Зритель обладает следующим набором данных:

- ФИО;
- Почта;
- Номер телефона.

Зритель
ФИО
Почта
Телефон

Рисунок 26 - Зритель

Зрителю, который хочет посмотреть матч вживую, также нужно пройти процесс регистрации, чтобы у него была возможность приобрести билет.

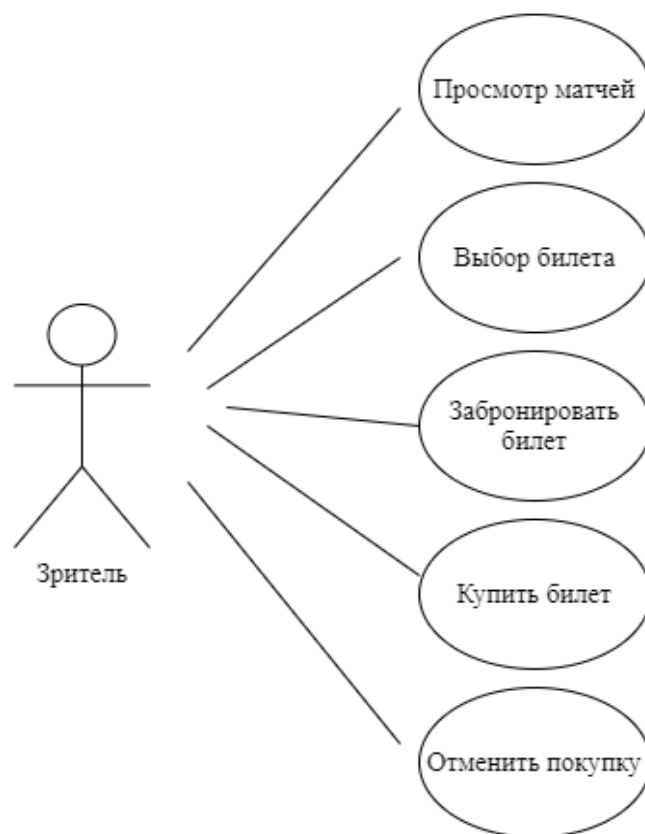


Рисунок 27 - Диаграмма вариантов для зрителя

В диаграмме вариантов использования представлены различные действия, которые может выполнять зритель во время работы с программой.

У зрителя есть возможности:

- Просмотреть предстоящие матчи;
- Выбрать нужный билет;
- Забронировать билет;
- Купить билет на матч;
- Отменить уже ранее купленный билет.

Таблица 5 - Сценарий использования

Сценарий	Действие
Просмотр матча	Зритель просматривает матч, на который хочет пойти.
Выбор билета	Зритель просматривает какие билет(-ы) есть в наличии и выбирает нужный(-ые) билет(-ы).
Забронировать билет	Зритель бронирует нужное количество билетов.
Купить билет	Зритель покупает билет.
Отмена покупки	Зритель может сдать купленные ранее билеты.

Далее опишем действия участника с системой.

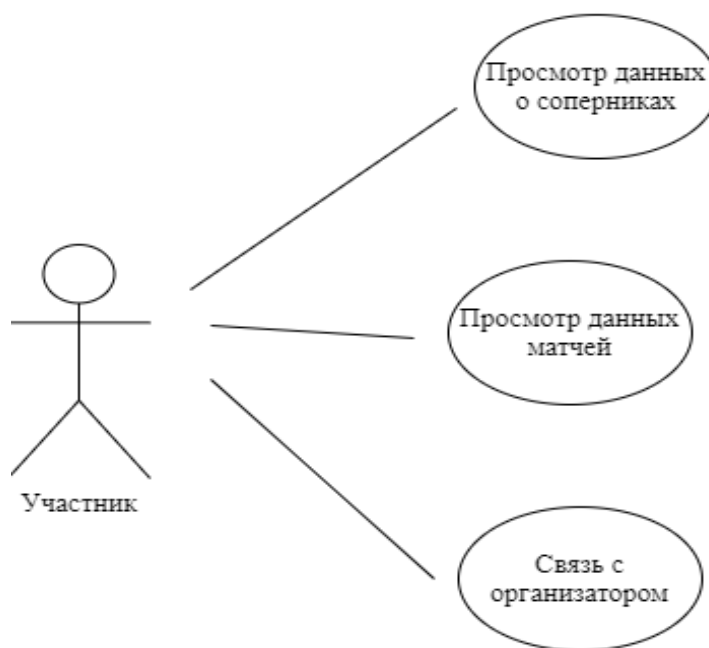


Рисунок 28 - Диаграмма вариантов для участника

В диаграмме вариантов использования представлены различные действия, которые может выполнять участник во время работы с программой.

У участника есть возможности:

–Просмотр данных о соперниках, которые будут также участвовать в турнире;

–Просмотр данных о предстоящем матче: команду, против которой будут играть, дата и время проведения матча, место проведения;

–Связь с организатором.

Таблица 6 - Сценарий использования.

Сценарий	Действие
Просмотр данных о соперниках	Участник может просмотреть информацию о будущих соперниках.
Просмотр данных матчей	Участник может просмотреть информацию о предстоящих матчах, такую как: <ul style="list-style-type: none">– Соперника, с которым предстоит сыграть;

Продолжение таблицы 6

	<ul style="list-style-type: none"> – Стадион, на котором будет проводиться матч; – Дата и время проведения матча.
Связь с организатором	Участник может связываться с организатором по разным вопросам.

Опишем более подробно взаимодействие между организатором и OLAP системой.

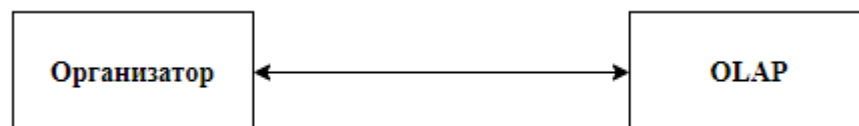


Рисунок 29 - Взаимодействие организатора и OLAP системы

При обращении организатора к OLAP системе, организатор получает данные о командах и предстоящих матчах с указанием даты и времени. Так как одна из главных его задач — это контролирование даты и времени проведения матча. В свою очередь OLAP система в дальнейшем будет сразу отправлять организаторам результаты матчей и дальнейшее расписание игр.

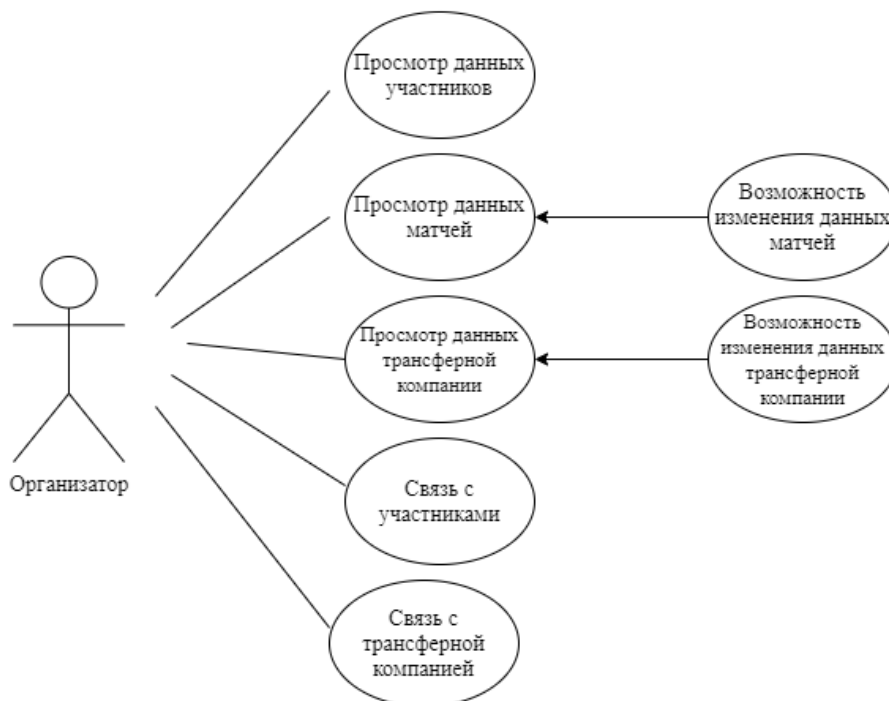


Рисунок 30 - Диаграмма вариантов для организатора

В диаграмме вариантов использования представлены различные действия, которые может выполнять организатор во время работы с программой.

У организатора есть возможности:

–Просматривать данные участников, которые собираются участвовать в турнире;

–Просмотр данных о предстоящих матчах: команды, которые будут играть, дата и время проведения матча, место проведения;

–Изменять данные о предстоящих матчах;

–Просматривать данные трансферной компании: какой автомобиль прибудет, время прибытия автомобиля;

–Изменять данные трансферной компании;

–Связь с участниками;

–Связь с трансферной компанией: непосредственно с водителем, если будут какие-то накладки.

Таблица 7 - Сценарий использования

Действие	Сценарий
Просмотр данных участников	Организатор может просматривать всех участников футбольного турнира.
Просмотр данных матчей	Организатор может как просматривать данные о предстоящих матчах, так и изменять их, по каким-либо причинам. Организатор может изменять такие данные о матчах: <ul style="list-style-type: none">– Соперников;– Дату и время проведения;– Место проведения.
Просмотр данных трансферной компании.	Организатор просматривает данные трансферной компании <ul style="list-style-type: none">– Водителей– Машины
Связь с участником	Организатор может связаться с участником.
Связь с трансферной компанией	Организатор может связаться непосредственно с водителем, который будет осуществлять трансфер участника, до места проведения матча.

2.4 Выбор средств реализации разработки бд и проектирование серверной части системы

2.4.1 SQL

SQL (structured query language — «язык структурированных запросов») — формальный непроцедурный декларативный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей СУБД.

SQL отличается от стандартных языков программирования, поскольку он не может быть использован для создания обособленных приложений. Он не может существовать вне ядра СУБД, которое необходимо для преобразования инструкций в выполняемые машинные команды [8].

После анализа задания на выпускную квалификационную работу бакалавра был сделан вывод, что данные, используемые для организации футбольного турнира и поддержания ее работоспособности, необходимо хранить в базе данных и обновлять данные о матчах. Следовательно, при проектировании базы данных нужно учитывать следующие требования:

- Обеспечивать хранение в БД всей необходимой информации.
- Обеспечивать возможность получения данных по всем необходимым запросам.
- Сокращать избыточность и дублирование данных.
- Обеспечивать целостность базы данных.

Для начала выполним концептуальное проектирование – построим семантическую модель предметной области, то есть информационную модель наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных.

Чаще всего концептуальная модель базы данных включает в себя:

- Описание информационных объектов или понятий предметной области и связей между ними.
- Описание ограничений целостности, то есть требований к допустимым

значениям данных и к связям между ними.

Концептуальная схема представлена на рисунке 31.

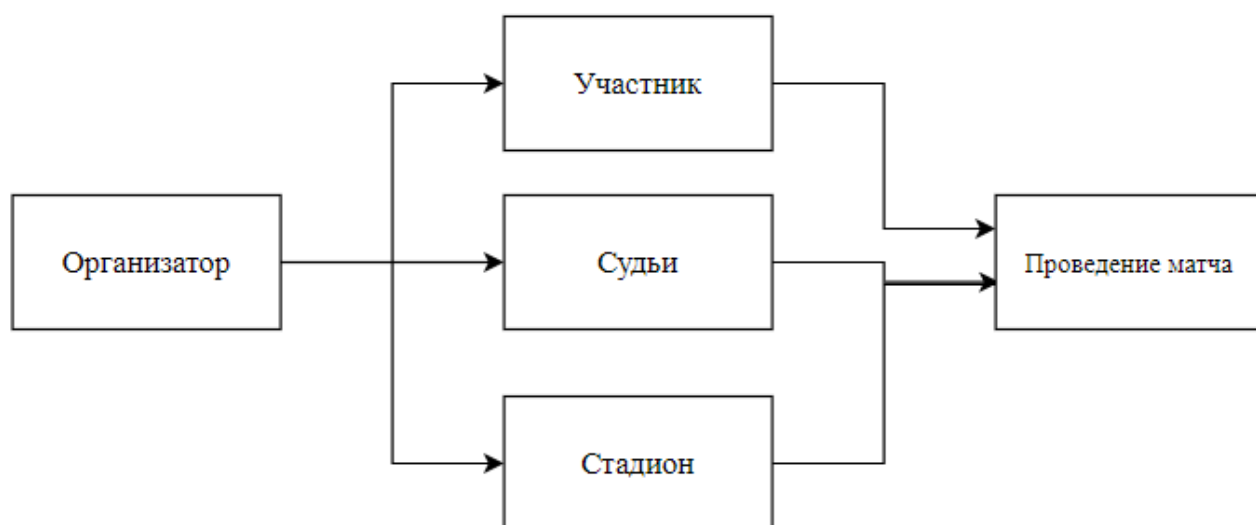


Рисунок 31 - Концептуальная схема

Как видно из схемы основными объектами в проведение матча являются: организатор, участник, судьи и стадион. Организатор связан с объектами: участник, судьи и стадион. Судьи и стадион так же связаны друг с другом, потому что за стадионом закреплена определенная бригада судей.

Вторым шагом является логическое проектирование – создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Рассмотрим возможности OLAP системы, реализуемой для системы:

1. Быстрый доступ к данным

Быстрый доступ к данным, независимо от размеров массива, является основой OLAP систем. Хранилище данных обычно строится по принципам, отличным от принципов реляционных баз данных. Данные в разрабатываемой системе организованы в виде куба. При этом время на выборку данных измеряется в долях секунды.

2.Преагрегация

Кроме быстрой выборки существующих данных, в системе также предоставляется возможность преагрегировать «наиболее вероятно-используемые» значения. В системе преагрегация происходит не всегда – потому что теоретически возможных комбинаций матчей/времени/ и т.д. может быть огромное количество, поэтому установлены четкие правила для каких элементов агрегация будет построена, а для каких нет.

3.Иерархии

OLAP кубы изначально рассматривают данные с точки зрения иерархий и взаимоотношений с другими параметрами одной и той же сущности, так что построение и использования иерархией в кубах – легко выполнимое OLAP системой.

В системе данные реализованы с помощью иерархии «Звезда». Модель данных состоит из двух типов таблиц: одной таблицы фактов (fact table) — центр «звезды» — и нескольких таблиц измерений (dimension table) по числу измерений в модели данных — лучи «звезды». Центр «Звезды» — это участник, а лучами звезды являются: местоположение участника, местоположение стадиона, информация о клубе и т.д.

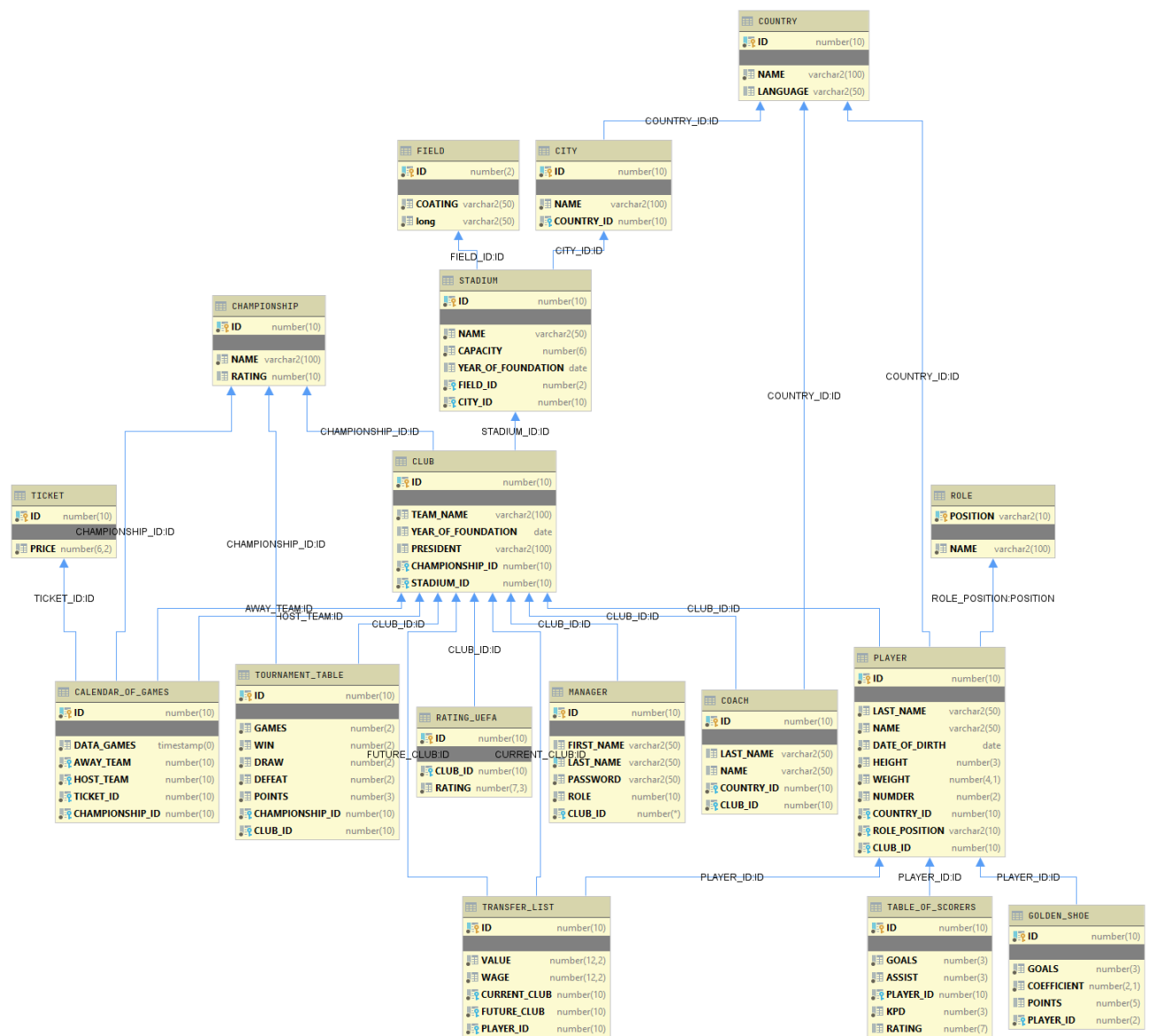


Рисунок 32 - Схема «Звезда»

4. Работа с временем

Так как в основном анализ данных происходит на временных участках, именно время в OLAP системах имеет особое значение, а значит просто определив для системы, где именно обозначается время, в дальнейшем можно с легкостью пользоваться функциями типа Year To Date, Month To Date (период от начала года/месяца и до текущей даты), Parallel Period (в этот же день или месяц, но в прошлом году) и т.п.

5. Key Performance Indicators (KPI)

Ключевые показатели эффективности — это финансовая и нефинансовая система оценки, которая помогает организатору определить достижение

стратегических целей: оптимальное время и место проведения матча, с точки зрения местоположения участников, заинтересованность зрителями футбольным турниром, регламент проведения турнира и т.д. Ключевые показатели эффективности могут быть достаточно просто определены в OLAP системах и использоваться в отчетах.

6. Многоуровневое кэширование

Для обеспечения наиболее высокой скорости доступа к данным в разрабатываемой системе, кроме хитрых структур данных и преагрегаций, OLAP системы поддерживают многоуровневое кэширование. Кроме кэширования простых запросов, также кэшируются части вычитанных из хранилища данных, агрегированные значения, вычисленные значения. Таким образом, чем дольше работаешь с OLAP кубом, тем быстрее он, по сути, начинает работать.

2.4.2 Выбор языка программирования для реализации

При разработке системы был выбран язык программирования Java. Выбор именно этого языка вызван тем, что он является платформонезависимым, в отличии от других языков программирования, простой в использование, а также динамическим.

Для стандартизации работы с SQL-серверами взаимодействие с ней можно выполнять через единую точку — **JDBC** (Java DataBase Connectivity). Она представляет собой реализацию пакета `java.sql` для работы с СУБД. Производители всех популярных SQL-серверов выпускают для них драйверы JDBC. Рассмотрим схему ниже. Приложение использует экземпляры классов из `java.sql`. Затем мы передаем необходимые команды для получения/модификации данных. Далее `java.sql` через jdbc-драйвер взаимодействует с СУБД и возвращает нам готовый результат [9].

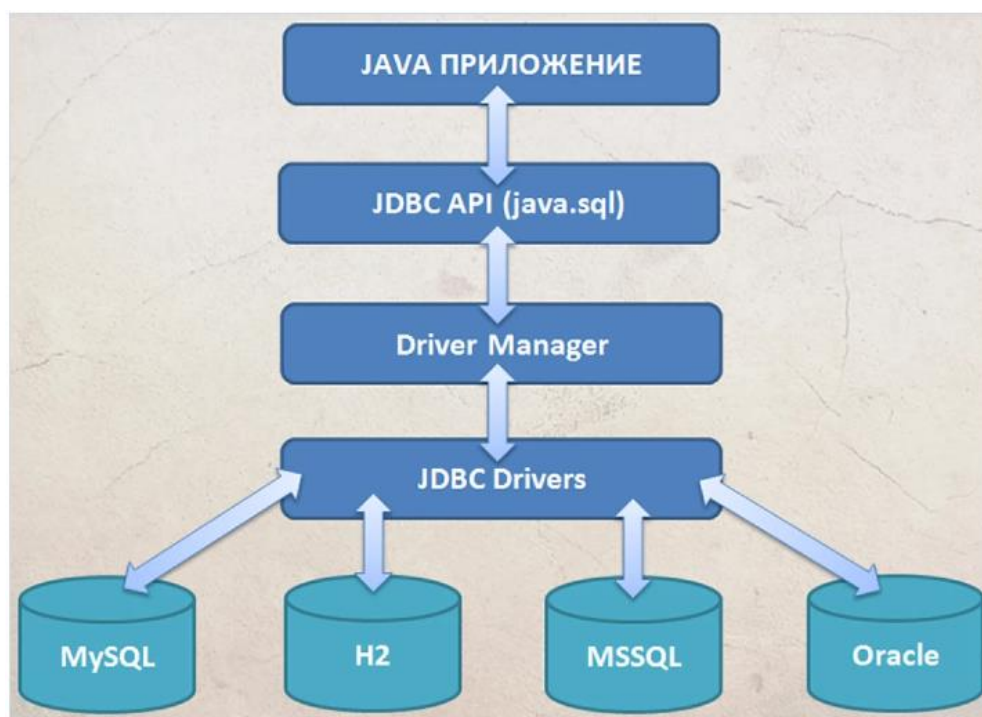


Рисунок 33 - Архитектура JDBC

Как следует из рисунка, приложение работает с абстракцией JDBC в виде набора интерфейсов. А вот реализация для каждого типа СУБД используется своя. Эта реализация называется “JDBC-драйвер”. Для каждого типа СУБД используется свой JDBC-драйвер — для Oracle свой, для MySQL — свой.

Система JDBC позволяет загрузить JDBC-драйвер для конкретной СУБД и единообразно использовать компоненты этого драйвера за счет того, что мы к этим компонентам обращаемся не напрямую, а через интерфейсы [10].

2.4.3 JavaFX и JFoenix

JavaFX представляет из себя библиотеку с открытым исходным кодом для создания кроссплатформенных приложений на языке Java. Благодаря широкому спектру возможностей может использоваться как для создания настольных приложений, так и для создания интернет приложений или приложений на мобильных устройствах.

На данный момент использование JavaFX является предпочтительным вариантом для создания графических приложений на языке Java, который пришел на замену устаревшим Swing и AWT. Так же стоит отметить, что для работы с JavaFX можно использовать не только Java, но и любой другой язык,

который поддерживает JVM. С 2018 года было принято решение отделить JavaFX от основной функциональности JavaSE и поставлять как отдельный модуль.

Использование данной библиотеки вызвано тем, что встроенные графические библиотеки имели ряд минусов:

- отсутствие структурированной иерархии классов
- при разрастании приложения необходимо вводить несколько уровней вложенных макетов и множество компонентов, из-за этого код становится очень сложным для понимания
- присутствие в одном классе кода, который отвечает за структуру, визуальные эффекты и поведение одновременно

Особенностью библиотеки является структурирование кода при помощи использования многочисленных шаблонов дизайна, такие как Model-View-Whatever, Model-View-Controller, Model-View-View-Model и другие. Это возможно потому, что в дополнение к процедурной конструкции нашего пользовательского интерфейса мы можем использовать декларативную разметку XML. Иерархичная структура XML — это отличный способ описать иерархию компонентов в пользовательском интерфейсе.

JavaFX использует специфичный формат XML который называется FXML. В нём возможно определять все компоненты и их свойства, указывать связь с контроллером, который отвечает за управление, определять стили и визуальные эффекты при помощи языка CSS.

JFoenix представляет из себя java-библиотеку с открытым исходным кодом, которая реализуется принципы Google Material Design с использованием компонентов Java.

2.4.4 Hibernate

Hibernate – это ORM Фреймворк для Java с открытым исходным кодом. Эта технология является крайне мощной и имеет высокие показатели производительности.

Hibernate создаёт связь между таблицами в базе данных и Java-классами и наоборот. Это избавляет разработчиков от огромного количества лишней, рутинной работы, в которой крайне легко допустить ошибку и крайне трудно потом её найти.

Hibernate поддерживает все основные СУБД: MySQL, Oracle, PostgreSQL, Microsoft SQL Server Database, HSQL, DB2 [10].

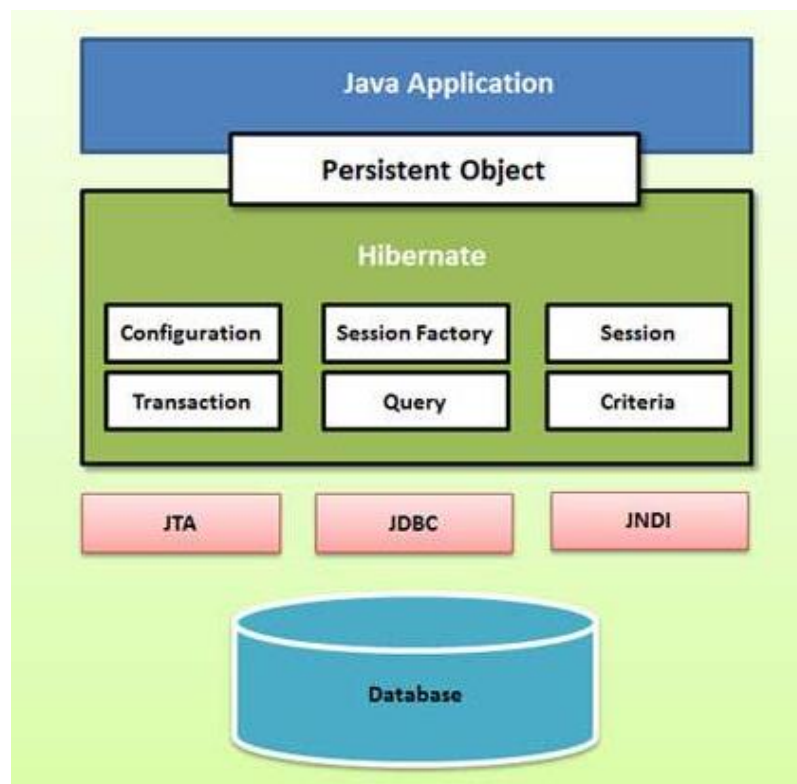


Рисунок 34 - Архитектура Hibernate

Hibernate использует различные существующие API Java, такие как JDBC, API транзакций Java (JTA) и интерфейс именования и каталогов Java (JNDI). JDBC обеспечивает элементарный уровень абстракции функциональности, общей для реляционных баз данных, что позволяет Hibernate поддерживать практически любую базу данных с драйвером JDBC [11].

Объект конфигурации содержит два ключевых компонента:

–Соединение с базой данных – это обрабатывается через один или несколько файлов конфигурации, поддерживаемых Hibernate.

–Настройка отображения классов – этот компонент создает связь между классами Java и таблицами базы данных.

2.4.5 Объектно-реляционное отображение

ORM или Object-relational mapping (рус. *Объектно-реляционное отображение*) — это технология программирования, которая позволяет преобразовывать несовместимые типы моделей в ООП, в частности, между хранилищем данных и объектами программирования. ORM используется для упрощения процесса сохранения объектов в реляционную базу данных и их извлечения, при этом ORM сама заботится о преобразовании данных между двумя несовместимыми состояниями. Большинство ORM-инструментов в значительной мере полагаются на метаданные базы данных и объектов, так что объектам ничего не нужно знать о структуре базы данных, а базе данных — ничего о том, как данные организованы в приложении. ORM обеспечивает полное разделение задач в хорошо спроектированных приложениях, при котором и база данных, и приложение могут работать с данными каждый в своей исходной форме [12].



Рисунок 35 - Структурное представление ORM

Ключевой особенностью ORM является отображение, которое используется для привязки объекта к его данным в БД. ORM как бы создает «виртуальную» схему базы данных в памяти и позволяет манипулировать данными уже на уровне объектов. Отображение показывает, как объект и его

свойства связаны с одной или несколькими таблицами и их полями в базе данных. ORM использует информацию этого отображения для управления процессом преобразования данных между базой и формами объектов, а также для создания SQL-запросов для вставки, обновления и удаления данных в ответ на изменения, которые приложение вносит в эти объекты [13].

2.5 Проектирование основных алгоритмов работы системы

Схемы основных алгоритмов системы приведены на рисунках ниже.

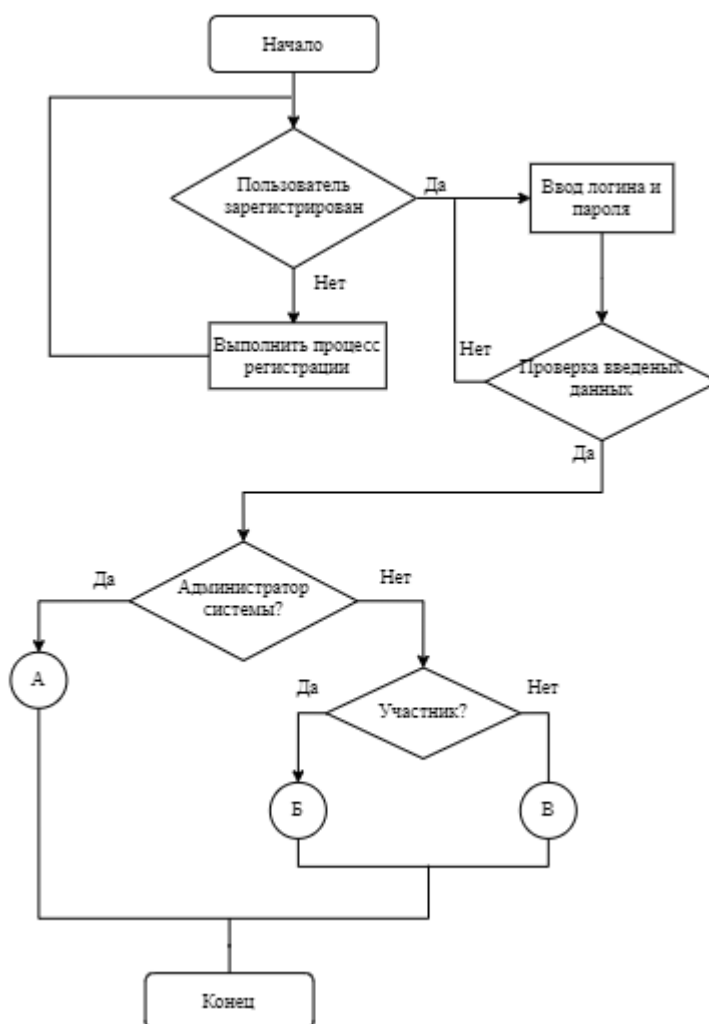


Рисунок 36 - Общий алгоритм работы системы

В связи с тем, что от роли пользователя, работа в системе будет немного отличаться, нужно показать работу системы для разных пользователей, таких как:

- А (Организатора турнира);
- Б (Участника турнира);

–В (Зрителя, который хочет попасть на матч).

Соответствующие схемы приведены ниже.

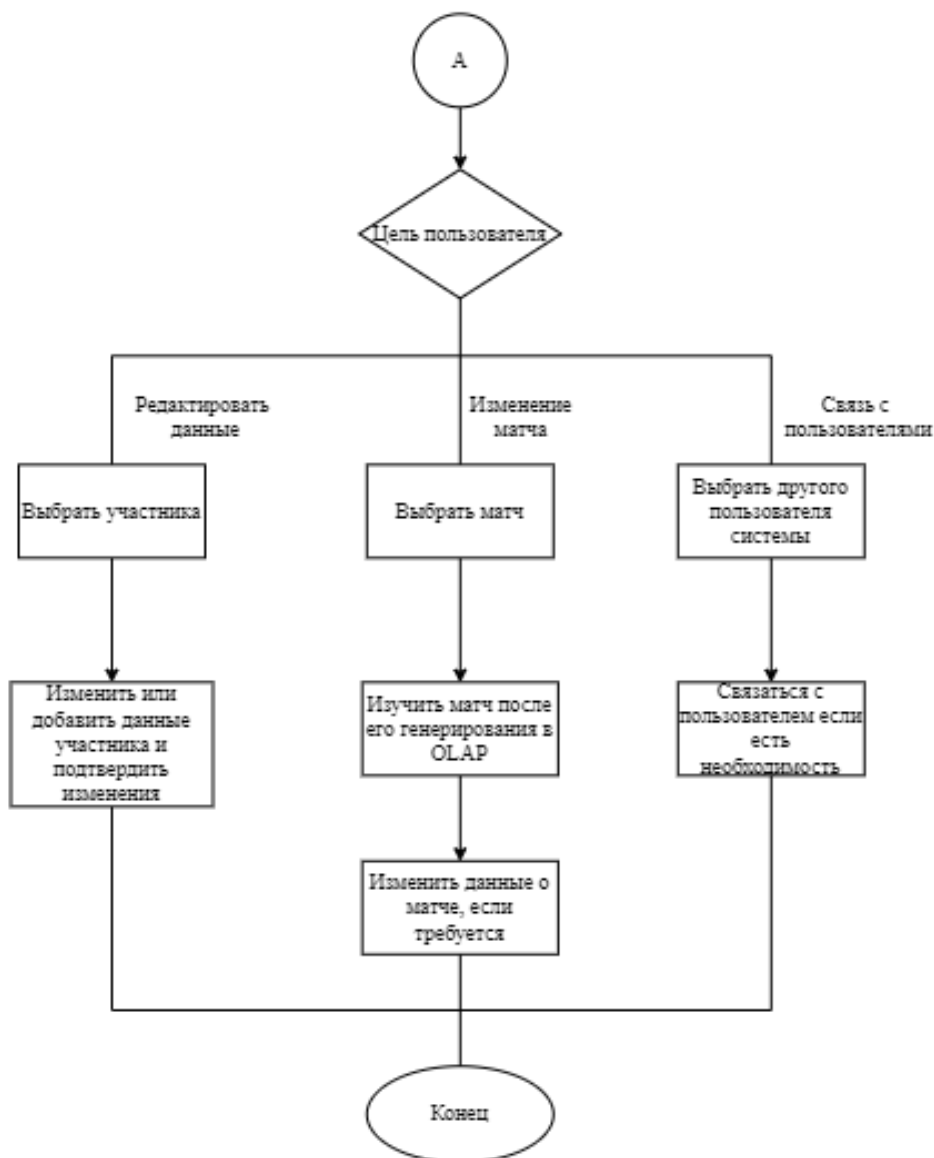


Рисунок 37 - Алгоритм работы системы для использования под ролью «Организатор»

На алгоритме видно, что у организатора есть 3 отличающиеся цели использования системы от других пользователей. Редактирование данных уже полученных при регистрирование других пользователей в системе, которые будут приниматься участия. Изменение матча, если есть такая необходимость. Связь с другим пользователем в системе, для оповещения его о предстоящем матче.

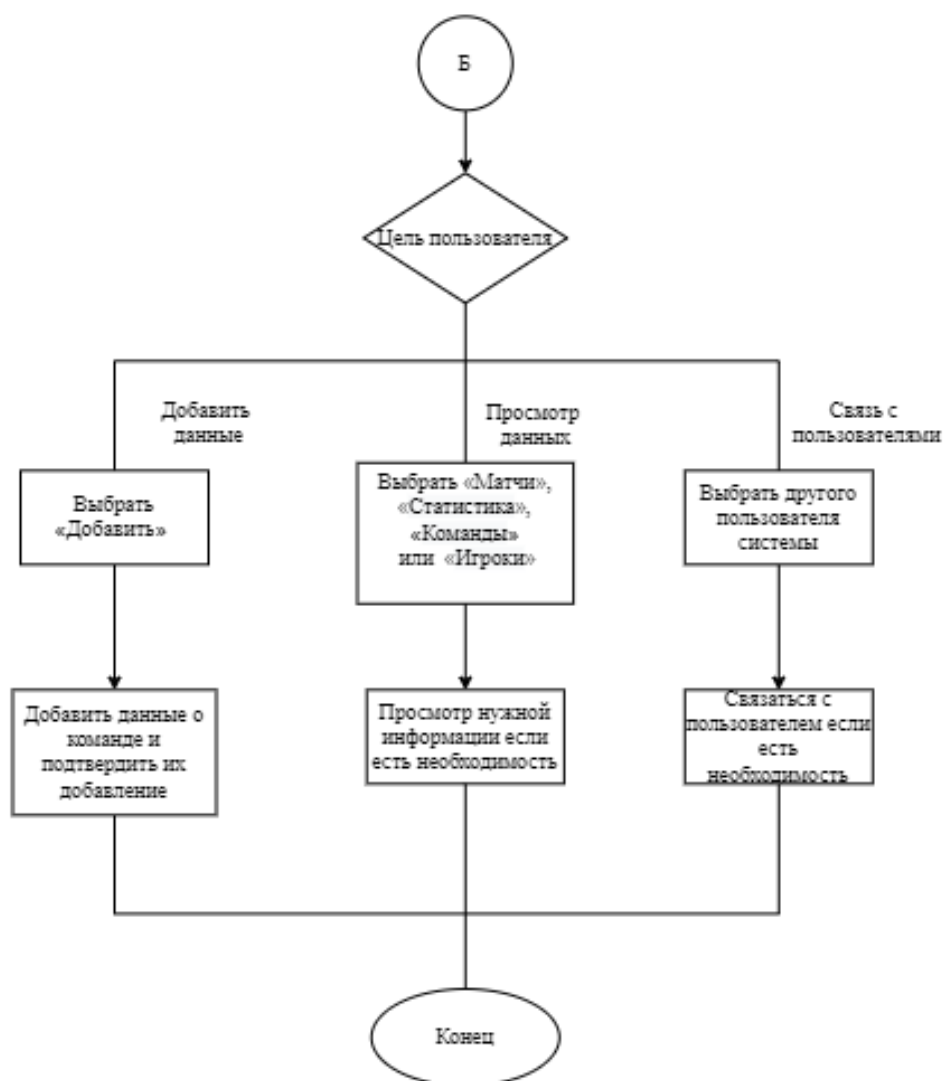


Рисунок 38 - Алгоритм работы системы для использования под ролью «Участник»

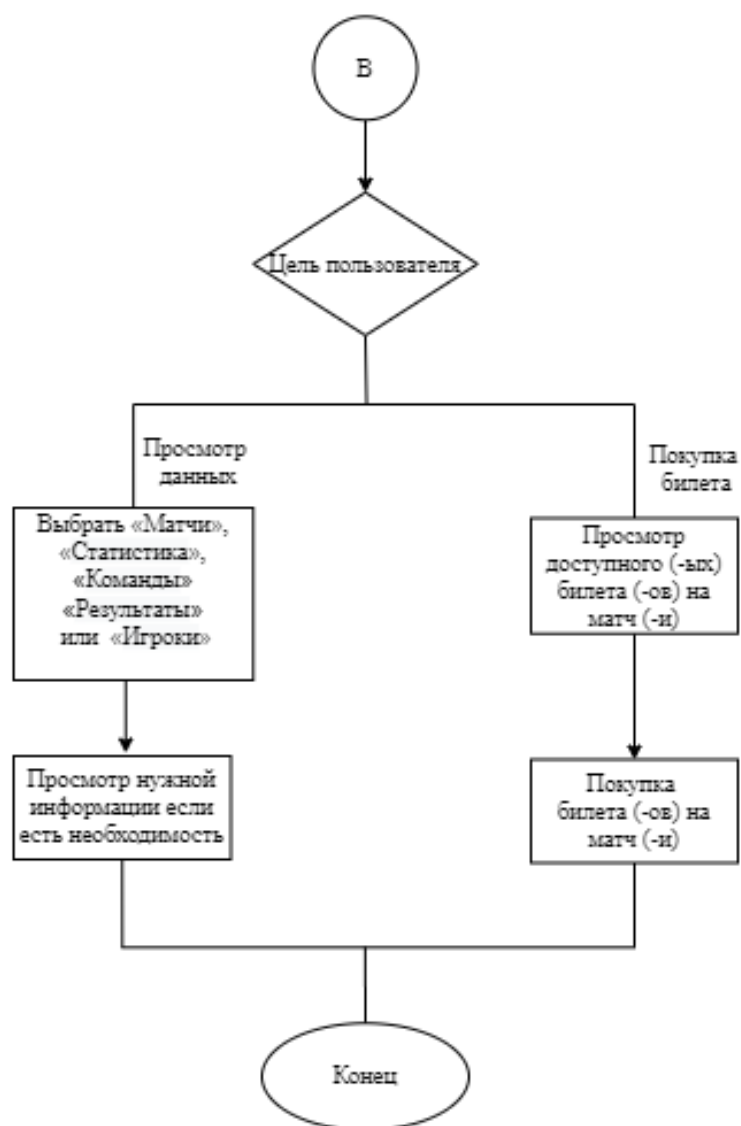


Рисунок 39 - Алгоритм работы системы для использования под ролью «Зритель»

2.6 Анализ функционального состава информационной системы

Информационная система представляет организатору основную функцию— генерирование матчей футбольного турнира в зависимости от входных данных. Начальные данные вводятся пользователями при процессе регистрации на футбольный турнир в качестве участника. Далее происходит процесс генерации матчей, места проведения и времени в зависимости от входных данных.

Особенность системы заключается в том, что данные во время турнира меняются и система будет их быстро обновлять и генерировать следующие матчи.

Процесс генерирования матча показан на рисунке ниже.



Рисунок 40 - Этапы генерирования матчей футбольного турнира

Генерирование данных футбольного турнира требует:

1. Ввод данных. При помощи стандартных возможностей платформы JavaFX происходит захват вводимых значений в поля ввода данных и сохранение их в базу данных.
2. После того как данные были введены и занесены в БД они будут переданы в OLAP систему для генерирования матчей футбольного турнира.
3. Генерирование матчей. Генерирование осуществляется при помощи OLAP системы.
4. Запись сгенерированных матчей в БД.
5. Отправка матчей организатору.

Таблица 8 - Генерирование матчей

Название варианта	Генерирование матчей
Цель	Просмотреть на экране результата сгенерированных матчей футбольного турнира
Действующие лицо	Пользователь (Организатор)
Краткое описание	Генерирование футбольных матчей получается при анализе и обработке полученных данных.

При разработке системы был выбран объектно-ориентированный подход к проектированию, поскольку основными объектами программного продукта являются компоненты пользовательского интерфейса, которые можно легко представить в виде иерархии классов.

Был использован нисходящий подход к разработке и реализации компонентов. Этот подход был выбран для разработки, поскольку он обеспечивает полное определение спецификаций предлагаемого компонента и согласованности компонентов друг с другом, позволяет определить пользовательский интерфейс, который демонстрирует требования к созданному программному продукту и позволяет проводить последующее тестирование и комплексную отладку.

Наиболее выгодно использовать спиральную модель жизненного цикла, поскольку выбранный тип модели предполагает разработку в форме последовательности версий, в которой требования определяются в результате разработки версии. Такой подход позволит вам получить наименьший возможный результат работы за счет реализации базового интерфейса и функций приложения, а затем добавить новые функции в программное обеспечение и дополнить существующие новыми версиями.

2.7 Примеры форм интерфейса пользователя

В данном разделе приведены примеры форм интерфейса информационной системы поддержки организации футбольного турнира.

На рисунке 41 представленном ниже представлена форма входа в систему.

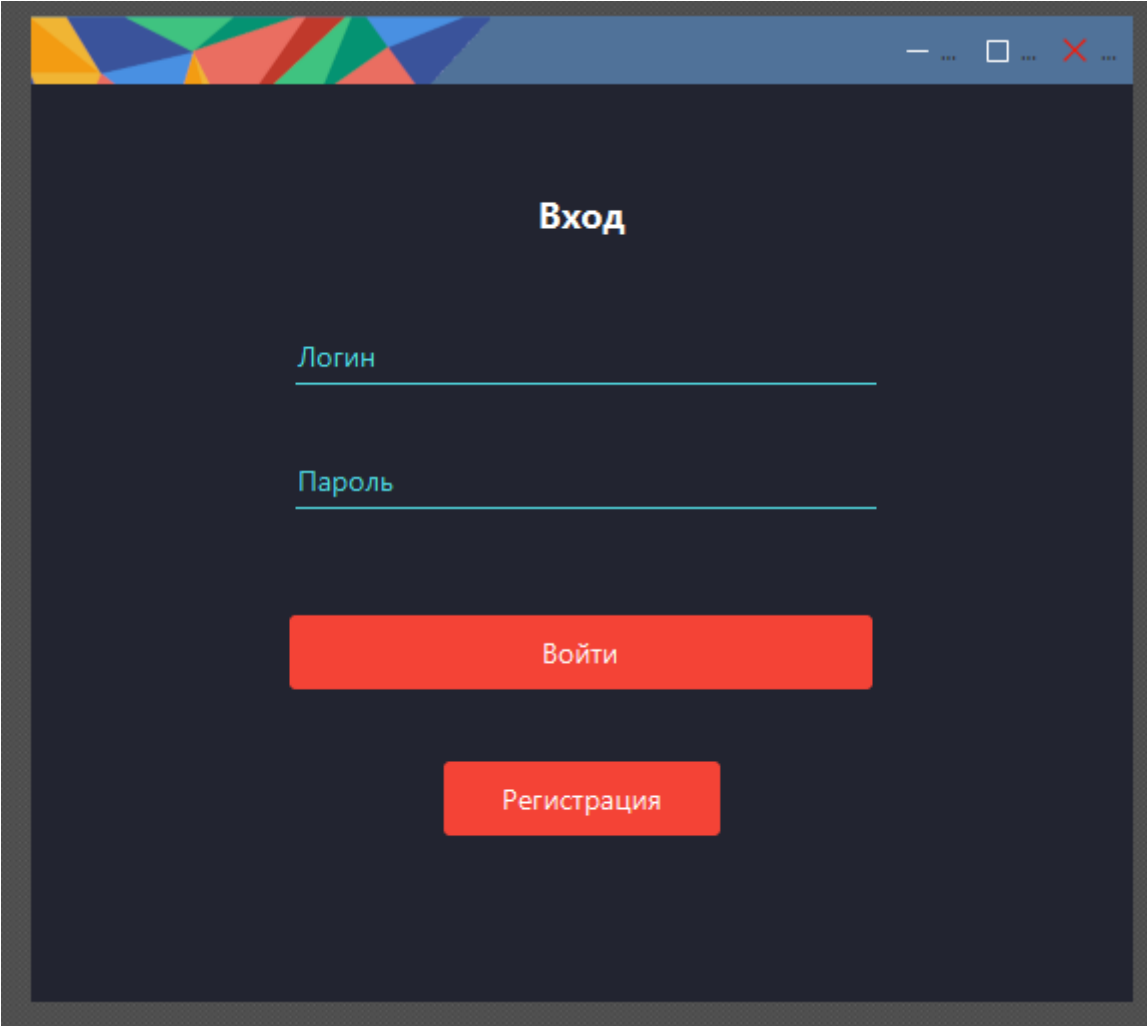
The image shows a web browser window with a login form. The browser's address bar is at the top with a blue background and some icons. The main content area has a dark blue background. At the top center, the word 'Вход' (Login) is written in white. Below it, there are two input fields: 'Логин' (Login) and 'Пароль' (Password), both with light blue text and underlined. Below the password field, there is a large red button with the text 'Войти' (Login) in white. Below that, there is a smaller red button with the text 'Регистрация' (Registration) in white.

Рисунок 41 - Форма входа в систему

Данные заносятся в систему после процесса регистрации. Форма содержит поля для ввода логина и пароля если пользователь уже зарегистрирован, а также кнопку по нажатию которой можно пройти процесс регистрации в системе для новых пользователей.

Чтобы войти в систему пользователь должен быть зарегистрирован, если нет, то он проходит процесс регистрации. Если пользователь зарегистрирован, то он вводит логин и пароль и система проверяет введенные данные на наличие

ошибок. После проверки данных, пользователь может войти в систему, иначе ему выведется сообщение об ошибки.

На рисунке 42 представленном ниже показана форма интерфейса регистрации в систему.

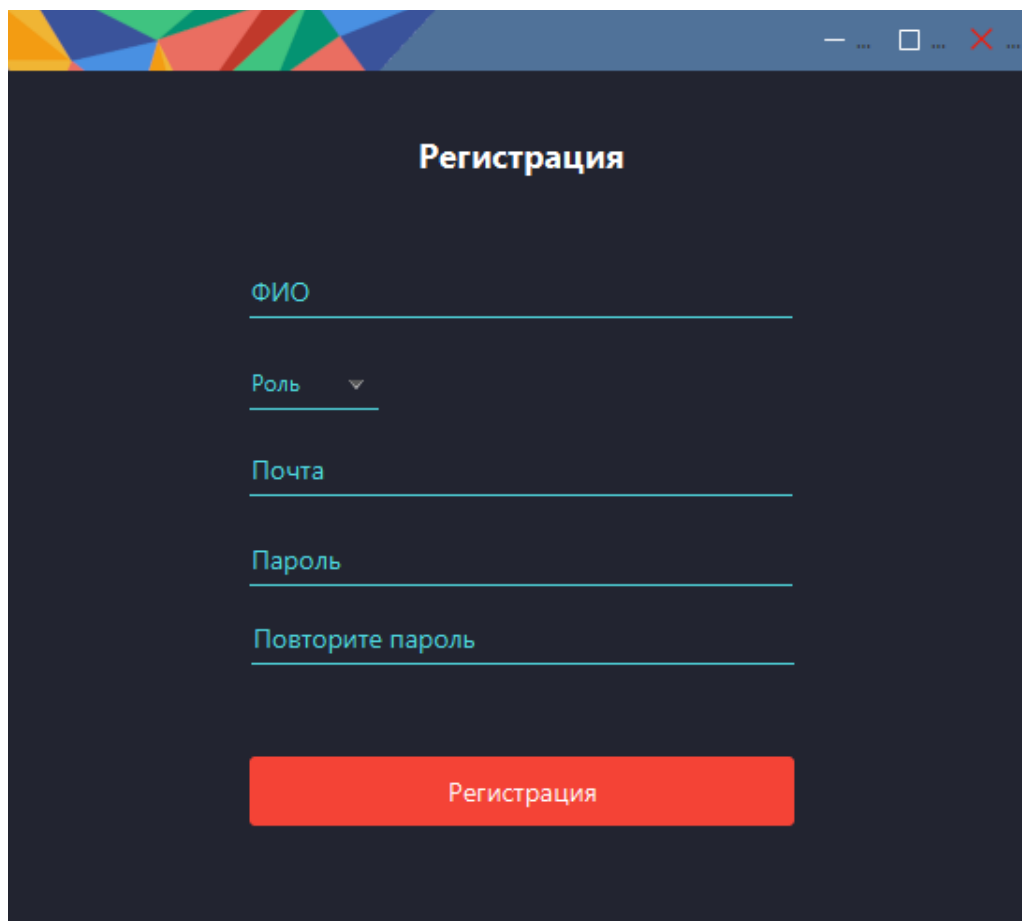
The image shows a web browser window with a registration form. The window has a title bar with standard minimize, maximize, and close buttons. The form itself has a dark blue background. At the top, there's a header with a colorful geometric pattern on the left and the title 'Регистрация' in white text. Below the title, there are five input fields, each with a light blue label and a corresponding input line: 'ФИО', 'Роль' (which includes a small downward arrow indicating a dropdown menu), 'Почта', 'Пароль', and 'Повторите пароль'. At the bottom of the form is a prominent red button with the text 'Регистрация' in white.

Рисунок 42 - Форма регистрации

Пользователю чтобы зарегистрироваться в системе необходимо заполнить поля, такие как:

- ФИО;
- Роль пользователя в турнире;
- Почта;
- Пароль;
- Повтор пароля.

Добавление игроков

ФИО

Возраст

Рост

Номер

Позиция ▼

Команда

Сохранить

Отмена

Рисунок 43 - Форма добавления игроков в команду

Менеджер команды заполняет форму добавления игроков, так как он является представителем команды. Форма содержит такие поля как:

- ФИО;
- Возраст;
- Рост;
- Номер, под которым будет выступать игрок на матчах;
- Позиция, на которой он будет играть во время матчей;
- Команда, поле заполняется автоматически.

Команды	Стадион	Дата	Время
ЦСКА - Спартак	ВЕБ Арена	21.04.2020	18:00
Реал Мадрид - Челси	Лужники	07.04.2020	14:15
Бавария - Боруссия	Альянс Арена	19.03.2020	17:00
Ювентус - Ливерпуль	Энфилд	13.03.2020	15:00
Зенит - Локомотив	РЖД Арена	09.04.2020	20:30
Рубин - Севилья	Казань Арена	10.04.2020	19:45
Шахтер - Арсенал	Металлист	15.03.2020	19:00
Бенфика - Милан	Эштадиу да Луш	04.04.2020	18:30
Байер - Базель	Бай Арена	24.03.2020	12:30
Вердер - Зальцбург	Дженари	09.03.2020	16:00
Порту - Бенфика	Драган	18.03.2020	21:00

Рисунок 44 - Форма списка матчей

На рисунке 44 показана форма списка предстоящих матчей футбольного турнира. Это поле могут просмотреть все пользователи системы.

Матч	Результат	Проход в следующий этап
ЦСКА - Спартак	2 - 0	ЦСКА
Реал Мадрид - Челси	3 - 1	Реал Мадрид
Бавария - Боруссия	1 - 0	Бавария
Ювентус - Ливерпуль	0 - 2	Ливерпуль
Зенит - Локомотив	0 - 1	Локомотив
Рубин - Севилья	1 - 2	Севилья
Шахтер - Арсенал	2 - 3	Арсенал
Бенфика - Милан	1 - 2	Милан
Байер - Базель	0 - 1	Базель
Вердер - Зальцбург	2 - 1	Вердер
Порту - Бенфика	1 - 3	Бенфика

Рисунок 45 - Форма результатов матчей

На рисунке 45 показана форма результатов матчей футбольного турнира. Это поле могут просмотреть все пользователи системы.

Заключение по второй главе

На данном этапе было спроектировано и реализовано десктопное приложение для организации и проведения футбольных турниров. При проектировании были выделены основные задачи:

- Определения наилучшего способа хранения данных
- Проектирование иерархии классов приложения
- проектирование базы данных

В ходе разработки был выбран набор наиболее оптимальных технологий для решения поставленных задач.

3 Разработка технологии тестирования информационной системы

Тестирование программного продукта — приоритетная часть разработки и дальнейшей поддержки информационной системы. Выявление программных ошибок на ранних этапах разработки сокращает время, необходимое для поиска и устранения неполадок в сравнении с тестированием уже готового продукта. Помимо этого, даже после окончания основного цикла разработки программного обеспечения, любые изменения, внесённые в исходный код и связанные с добавлением, улучшением или исправлением функциональности, могут привести к новым ошибкам. Потому разработанная технология испытаний должна учитывать необходимость частого повторного тестирования [14].

Систему необходимо тестировать при каждом добавление новых функций, для проверки того, что система работает верно и без каких-либо изменений.

3.1 Тестирование «Черный Ящик»

Проведём тестирование для системы по стратегии «черный ящик». Тестирование по принципу «черного ящика» – это тестирование с управлением по данным без знания кода программы. Такое тестирование имеет целью выяснение обстоятельств, в которых поведение программы не соответствует спецификации.

При тестировании с использованием стратегии «черного ящика» структура программы считается неизвестной, и тесты подбирают так, чтобы проверить выполнение всех функций программы, а затем отследить реакцию на ввод некорректных данных. То есть будем перебирать всевозможные варианты входных данных, для того чтобы выявить ошибки и уязвимые места системы, если таковы имеются [15].

Данные проведенных тестов представлены в таблицах

Проведем тестирование входа в систему. Рассмотренные тесты приведены в таблице.

Таблица 9 - Тестирование экрана авторизации

№	Логин	Пароль	Ожидаемый результат	Полученный результат
1	Ничего не введено	Ничего не введено.	Сообщение об ошибке	Сообщение о необходимости ввести данные
2	Введён логин существующего пользователя.	Ничего не введено.	Сообщение об ошибке	Сообщение о необходимости ввести данные
3	Введён логин не существующего пользователя.	Ничего не введено	Сообщение об ошибке	Сообщение о необходимости ввести данные
4	Введён логин не существующего пользователя.	Введён какой-либо пароль.	Сообщение об ошибке	Сообщение о неверности введенных данных
5	Введён логин существующего пользователя.	Введён некорректный пароль	Сообщение об ошибке	Сообщение о неверности введенных данных
6	Введён логин существующего пользователя.	Введён корректный пароль	Успешная авторизация	Успешная авторизация и переход на главный экран

Проведем тестирование экрана регистрации. Рассмотренные тесты приведены в таблице

Таблица 10 - Тестирование формы регистрации

№	ФИО	Роль	Е-mail	Пароль	Подтверждение пароля	Ожидаемый результат	Полученный результат
1	Ничего не введено	Ничего не введено	Ничего не введено	Ничего не введено	Ничего не введено	Сообщение об ошибке	Сообщение: «Заполните пустые поля»
2	Ничего не введено	Ничего не введено	Введена корректная почта	Ничего не введено	Ничего не введено	Сообщение об ошибке	Сообщение: «Заполните пустые поля»
3	Ничего не введено	Ничего не введено	Введена корректная почта	Введен пароль	Ничего не введено	Сообщение об ошибке	Сообщение: «Заполните пустые поля»

Продолжение таблицы 10

4	Ничего не введено	Ничего не введено	Введена корректная почта	Введен пароль	Введен не совпадающий пароль	Сообщение об ошибке	Сообщение : «Заполните пустые поля»
5	Ничего не введено	Ничего не введено	Введена корректная почта	Введен пароль	Введен совпадающий пароль	Сообщение об ошибке	Сообщение : «Заполните пустые поля»
6	Ничего не введено	Выбрана роль пользователя	Введена корректная почта	Введен пароль	Введен совпадающий пароль	Сообщение об ошибке	Сообщение : «Пароли не совпадают»
7	Введены корректные ФИО	Выбрана роль пользователя	Введена корректная почта	Введен пароль	Введен совпадающий пароль	Успешная регистрация и вход в приложение.	Успешная регистрация и вход в систему.

Далее проведем тестирование формы добавления игроков. Процесс и результаты тестирования приведены на таблице.

Таблица 11 - Тестирование формы добавления игрока

№	ФИО	Возраст	Рост	Номер	Позиция	Ожидаемый результат	Полученный результат
1	Ничего не введено	Ничего не введено	Ничего не введено	Ничего не введено	Ничего не выбрано	Сообщение об ошибке	Ошибка, заполните необходимые поля для добавления игрока
2	Введено ФИО	Ничего не введено	Ничего не введено	Ничего не введено	Ничего не выбрано	Сообщение об ошибке	Ошибка, заполните необходимые поля для добавления игрока
3	Введено ФИО	Введен возраст	Ничего не введено	Ничего не введено	Ничего не выбрано	Сообщение об ошибке	Ошибка, заполните необходимые поля для добавления игрока

Продолжение таблицы 11

4	Введено ФИО	Введен возраст	Введен рост	Ничего не введено	Ничего не выбрано	Сообщение об ошибке	Ошибка, заполните необходимые поля для добавления игрока
5	Введено ФИО	Введен возраст	Введен рост	Введен номер	Ничего не выбрано	Сообщение об ошибке	Ошибка, заполните необходимые поля для добавления игрока
6	Введено ФИО	Введен возраст	Введен рост	Введен номер	Выбрана позиция	Успешное добавление игрока	Успешное добавление игрока

При тестировании основных форм информационной системы стратегией черного ящика не было выявлено никаких критических недочетов. Однако, несмотря на это система может дорабатываться и улучшаться.

3.2 Оценочное тестирование

С помощью оценочного тестирования можно проверить варианты использования системы, которые могут быть протестированы только последовательностью ручных действий пользователями.

Проведем оценочное тестирование двух видов:

–тестирование удобства эксплуатации - последовательная проверка соответствия программного продукта и документации на него основным положениям технического задания;

–тестирование конфигураций - проверка работоспособности программного обеспечения на разном оборудовании.

Таблица 12 - Тестирование удобства эксплуатации всей системы

Критерий	Варианты ответа	1 пользователь	2 пользователь	3 пользователь	4 пользователь
Интерфейс	Приятный/ Неприятный	Приятный	Приятный	Приятный	Приятный

Продолжение таблицы 12

Читаемость текста	Хорошо/ Плохо	Хорошо	Хорошо	Хорошо	Хорошо
Размер шрифта	Хорошо/ Плохо	Хорошо	Плохо	Хорошо	Хорошо
Расположение полей ввода и кнопок	Удобное/Не удобное	Удобное	Удобное	Удобное	Не удобное
Цветовая гамма	Приятная/ Неприятная	Приятная	Не приятная	Приятная	Приятная
Понятность назначения кнопок	Понятно/Не понятно	Понятно	Понятно	Понятно	Понятно

По результатам тестирования на удобство эксплуатации можно сделать следующий вывод.

Таблица 13 - Выводы тестирования

Критерий	Итого
Интерфейс	100%
Читаемость текста	100%
Размер шрифта	75%
Расположение полей ввода и кнопок	75%
Цветовая гамма	50%
Понятность назначения кнопок	100%

Исходя из результатов тестирования можно сказать, что система удобна большинству пользователей, которые проходили тестирование.

Таблица 14 -Тестирование конфигураций оборудования

Критерий	Результат тестирования
Тестирование удобства использования	Сервис полностью соответствует техническому заданию, выполняет все заявленные функции.
Тестирование конфигурации оборудования	Проверка была осуществлена на 2 устройствах и в 3 браузерах. При наличии подключения к Интернету, сервис хорошо работает на таких браузерах, как: <ul style="list-style-type: none"> –Google Chrome; –Internet Explorer11; –Mozilla Firefox.

Таким образом, с помощью оценочного тестирования была проверена система. Тестирование конфигурации также показало соответствие разработанного технического задания.

Заключение по третьей главе

Разработана технология тестирования информационной системы с учетом особенностей предметной области и используемых технологий. Выполнено тестирование «Черным Ящиком», а также выполнено оценочное тестирование удобства эксплуатации и конфигурации системы. По итогам тестирования сделан вывод о соответствии разработанного программного продукта всем функциям и конфигурационным требованиям, указанным в техническом задании.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы поставленная цель была достигнута. Разработана система поддержки организации футбольного турнира. Для разработки был использован язык программирования Java, с использованием библиотеки JavaFX, а также языка SQL для создания хранилища данных.

Также выполнены следующие задачи:

1. Был проведен анализ существующих систем;
2. Выполнен обзор и сравнительный анализ систем обработки данных. На основе исследования была OLAP система, тип HOLAP;
3. Спроектирована структурная схема и разработаны функциональные диаграммы;
4. Спроектирована серверная часть, а также основные алгоритмы системы;
5. Разработан интерфейс;
6. Разработана технология тестирования и проведено тестирование системы.

В дальнейшем планируется доработка той части системы, которая позволит: менять язык системы, для иностранных пользователей, а также возможность поменять вид спорта организации турнира, для большей вариативности использования системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сервисы для организации мероприятий [Электронный ресурс]
<https://www.polylog.ru/ru/pr-blog/top-10-besplatnyh-servisov-dlya-organizatsii-meropriyatiy> (дата обращения 12.12.2019)
2. Национальная библиотека им Н. Э. Баумана [Электронный ресурс]
[https://ru.bmstu.wiki/OLTP_\(Online_Transaction_Processing\)](https://ru.bmstu.wiki/OLTP_(Online_Transaction_Processing)) (дата обращения 12.12.2019)
3. Введение в OLAP и многомерные базы данных [Электронный ресурс]
<http://www.olap.ru/basic/alpero2i.asp> (дата обращения 12.12.2019)
4. Введение в многомерный анализ [Электронный ресурс]
<https://habr.com/ru/post/126810/> (дата обращения 22.12.2019)
5. Принципы построения систем, ориентированных на анализ данных [Электронный ресурс] <https://ami.nstu.ru/~vms/lecture/lecture9/LECTURE9.HTM> (дата обращения 22.12.2019)
6. Многомерные кубы [Электронный ресурс] <https://habr.com/ru/post/66356/> (дата обращения 30.12.2019)
7. Что такое витрина данных? [Электронный ресурс] <http://asu-analitika.ru/chto-takoe-vitrina-dannyh-opredelenie-raznovidnosti-i-primery> (дата обращения 15.03.2020)
8. SQL [Электронный ресурс] <http://progopedia.ru/language/sql/> (дата обращения 30.03.2020)
9. База данных на Java - введение [Электронный ресурс] <https://java-course.ru/begin/database01/> (дата обращения 12.02.2020)
10. Основы Hibernate [Электронный ресурс] <https://habr.com/ru/post/29694/> (дата обращения 16.03.2020)
11. Встраиваемые базы данных в Java [Электронный ресурс]
<https://easyjava.ru/data/vstraivaemye-bazy-dannyh-v-java/> (дата обращения 15.02.2020)

12. Введение в ORM [Электронный ресурс] <http://internetka.in.ua/orm-intro/>
(дата обращения 15.03.2020)

13. ORM или как проектировать БД [Электронный ресурс]
<https://habr.com/ru/post/237889/> (дата обращения 15.02.2020)

14. Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. Объектно-ориентированное программирование. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 320 с. (дата обращения 25.05.2020)

15. Технология программирования: Учебник для вузов. – 3-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 336 с. (дата обращения 25.05.2020)

ПРИЛОЖЕНИЕ А

(Обязательное)

Техническое задание

Листов 9

ПРИЛОЖЕНИЕ Б

(Обязательное)

Руководство пользователя

Листов 8

ПРИЛОЖЕНИЕ В

(Обязательное)

Фрагменты исходных кодов программы

Листов 15

ПРИЛОЖЕНИЕ Г

(Обязательное)

Копии листов графической части

Листов 8