МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

**Анализ функциональных и эксплуатационных требований к  
программному продукту**

Выполнил студент группы ИВТ-31 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М.Л./

Киров 2021

1. Функциональные диаграммы

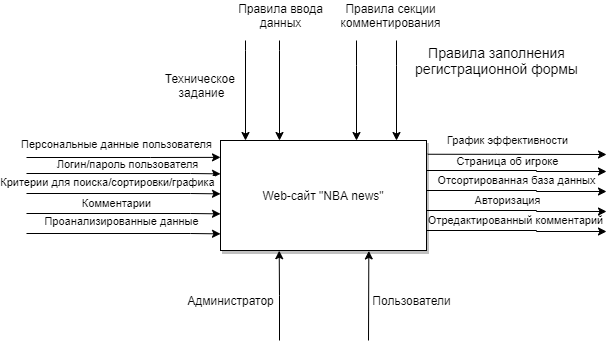


Рисунок 1. Функциональная диаграмма верхнего уровня

* **Функциональная диаграмма верхнего уровня** позволяет сформировать наиболее точное и описание разрабатываемого программного обеспечения. В данном случае в качестве отображения взаимосвязей была выбрана нотация IDEF0. В качестве входных данных выступают параметры, которые пользователь может задать для формирования графика или базы данных, персональные данные, логин/пароль от аккаунта и комментарии, оставленные в разделе «Новости». В качестве субъекта может выступать пользователь и администратор, имеющий более широкий спектр прав. Управление задаётся в соответствии с техническим заданием и некоторыми особенностями НБА, отличающими его от других баскетбольных лиг, таких как длительность четверти и подсчёт рейтинга. В выходных данных будет происходить отображение, выбранных пользователем данных: график, страница об игроке или таблица со статистикой игроков. После ввода логина/пароля происходит авторизация пользователя в системе. Администратор имеет доступ к редактированию комментариев, имеющих нарушения.

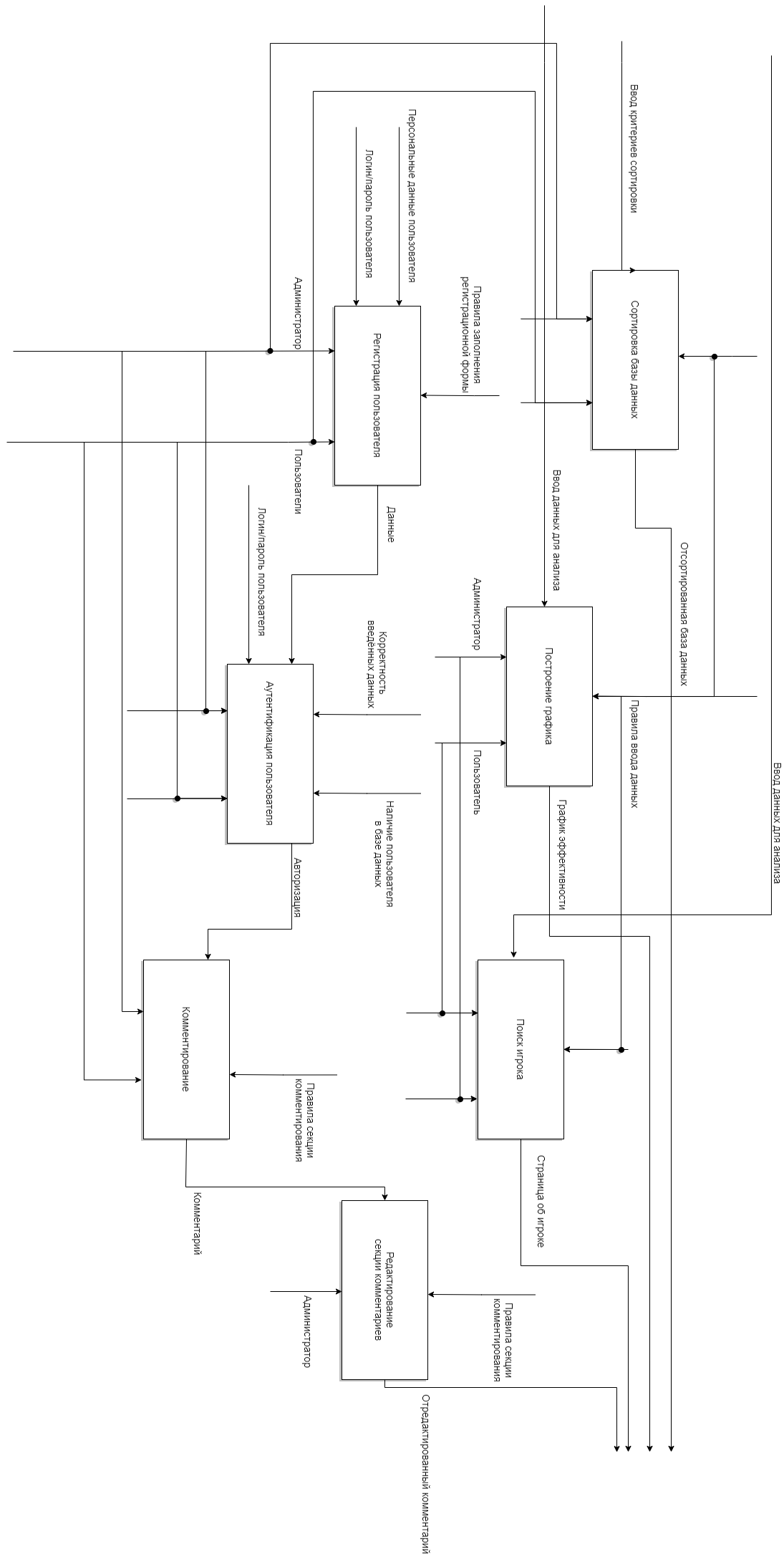


Рисунок 2. Функциональная диаграмма детализированного уровня

* **Детализирующая функциональная диаграмма** более подробно раскрывает функциональную диаграмму верхнего уровня: описывает взаимодействия и связи процессов, происходящих внутри системы. На ней можно увидеть, какие процессы взаимосвязаны и что между ними общего. Так практически во всех процедурах в качестве субъекта выступает пользователь и администратор, помимо части редактирования комментариев. Процедура регистрации нового пользователя протекает в 2 этапа: ввод данных и вход в систему. Результатом таких операций, как построение графика, сортировка базы данных и вывод списка игроков на экран являются данные, которые впоследствии анализирует пользователь. После анализа все склоняется к выбору следующего действия.

1. Диаграммы переходов состояний

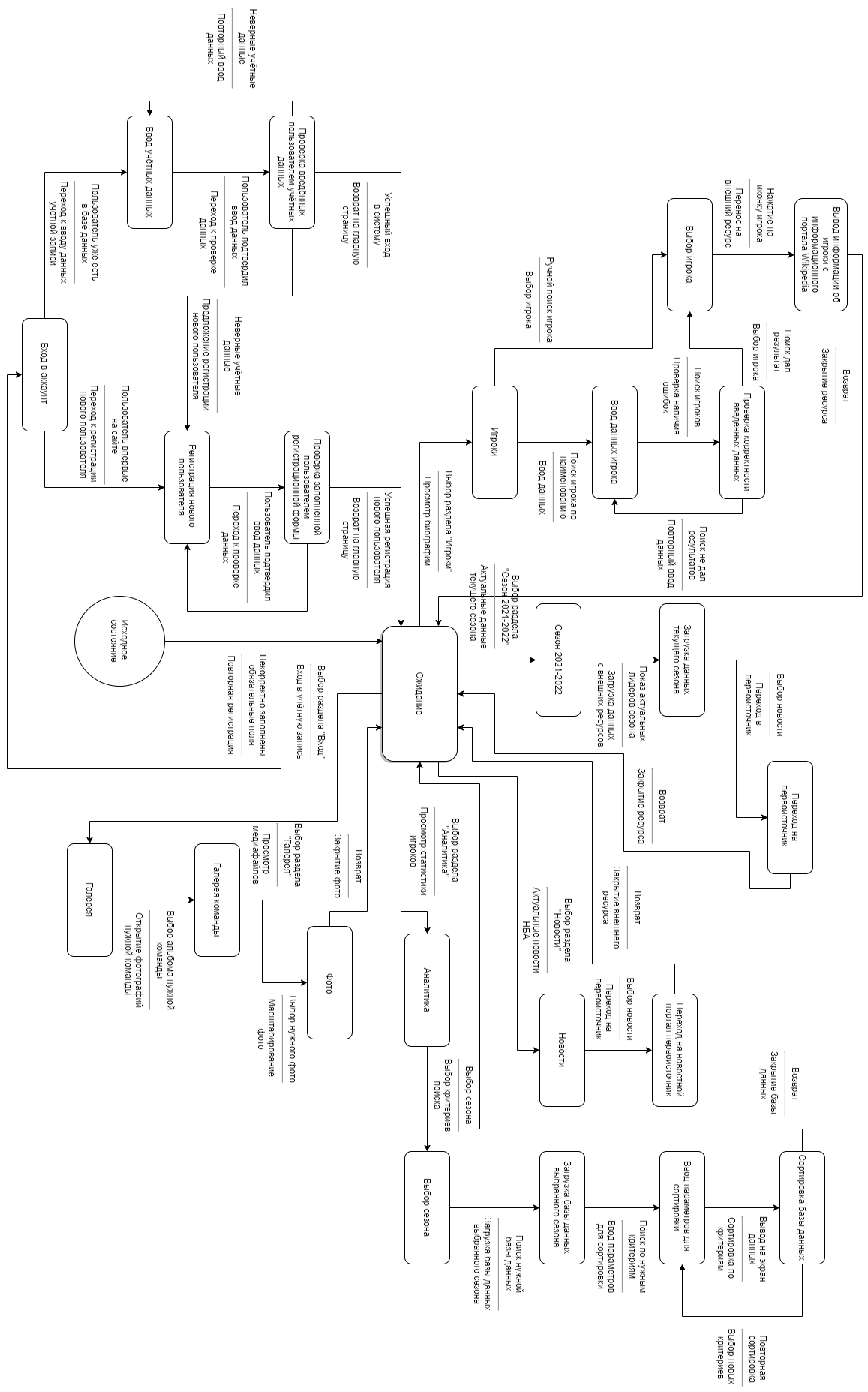


Рисунок 3. Диаграмма переходов состояний

* **Диаграмма переходов состояний** используется для описания динамического поведения объекта в ходе его существования. Она наиболее точно выражает как последовательность состояний, испытываемых объектом, событий, вызывающих переходы между состояниями, и действий, сопровождающих переход состояний, происходят в системе. В любом разрабатываемом ПО есть исходное состояние, с которого начинается работа программы, в данном случае – это начальная страница web-сайта «NBA news». Наиболее подробно описаны действия и события, к которым они приводят. После завершения какого-либо события оно переходит в блок «Ожидание», в котором происходит ожидание дальнейших действий пользователя.

1. Диаграммы потоков данных

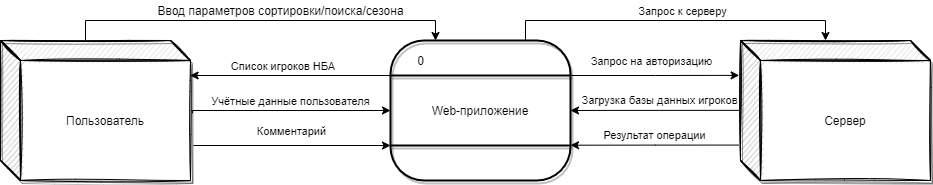


Рисунок 4. Контекстная диаграмма потоков данных

* **Контекстная диаграмма потоков данных** представляет собой иерархию функциональных процессов, связанных потоками данных. Контекстная диаграмма является диаграммой начального уровня и изображает все «заинтересованные» внешние сущности, общающиеся с системой и основные потоки данных между системой и внешним миром. В данном случае в роли сущностей будут выступать пользователь и сервер, которые взаимодействуют друг с другом путём передачи и получения информации с web-приложения. Диаграмма отображает какие запросы приложение передает серверу, а какие получает от него. Помимо этого, можно увидеть, что передает пользователь в качестве входных данных, и что он получает в ответ. Более подробное описание процессов приведена на следующей диаграмме.

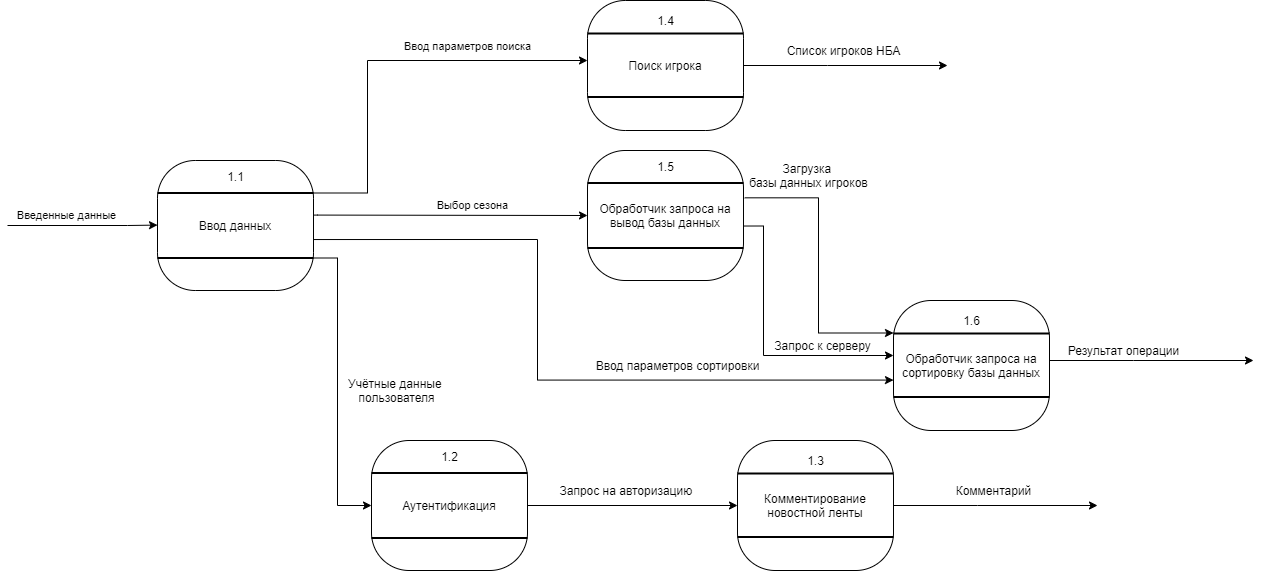


Рисунок 5. Детализированная диаграмма потоков данных

* **Детализированная диаграмма потоков данных** демонстрирует, как каждый процесс преобразует свои входные данные в выходные, а также выявляет отношения между этими процессами. В данной системе все процессы связаны вводом данных: выбор сезона, ввод параметров поиска, ввод параметров сортировки, ввод учётных данных пользователя, - что позволяет установить между процессами связь и задать начальную точку иерархии системы. Использование секции комментирования невозможно без аутентификации пользователя, которая позволяет быть точно уверенным в том, кто оставил комментарий. При сортировке базы данных сначала происходит выбор команды для сортировки и загрузка базы данных игроков, и уже только потом становится возможен выбор других критериев, позволяющих наиболее точно отобразить статистику игрока в том или ином аспекте игры. Результатом любой операции в конечном счёте является какое-либо визуальное представление конечных данных.

1. Укрупнённая схема алгоритма входа на web-сайта

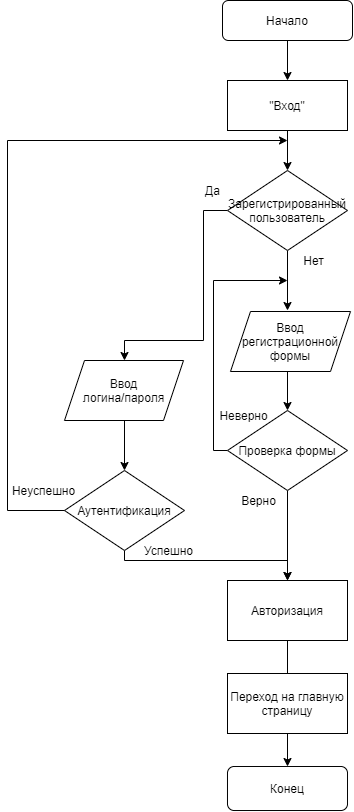


Рисунок 6. Укрупнённая схема алгоритма входа на web-сайта

* **Укрупнённая схема алгоритма** нужна для демонстрирования работы алгоритма того или иного процесса. В данном случае изображён алгоритм авторизации в системе. После нажатия вкладки «Вход» пользователь имеет выбор: зарегистрироваться на сайте или войти в уже существующую учётную запись. В обоих случаях присутствует проверка на корректность введённых данных. После успешно введённых данных происходит авторизация пользователя и вход в учётную запись с переходом на главную страницу проекта.