

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Вятский государственный университет»  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Допущено к защите  
Руководитель проекта  
\_\_\_\_\_ (Клюкин В. Л.)  
«\_\_» \_\_\_\_\_ 2022г.

Проектирование архитектуры платформы для создания телеграм ботов для  
спортзала. Telegram бот

Пояснительная записка курсового проекта по дисциплине  
«Комплекс знаний бакалавра в области программного и аппаратного  
обеспечения вычислительной техники»  
ТПЖА.090301.331 ПЗ

Разработал студент группы ИВТ-31 \_\_\_\_\_/Крючков И. С./

Руководитель \_\_\_\_\_/Клюкин В. Л./

Консультант \_\_\_\_\_/Кошкин О. В./

Проект защищен с оценкой «\_\_\_\_\_» \_\_\_\_\_  
(оценка) (дата)

Члены комиссии \_\_\_\_\_/\_\_\_\_\_  
(подпись) (Ф.И.О.)

\_\_\_\_\_/  
(подпись) (Ф.И.О.)

\_\_\_\_\_/  
(подпись) (Ф.И.О.)

## Реферат

Крючков И. С. Проектирование архитектуры платформы для создания телеграм ботов для спортзала. Telegram бот. ТПЖА.090301.331 ПЗ: Курс. проект / ВятГУ, каф. ЭВМ; рук. Клюкин В. Л. - Киров, 2022. – ПЗ 54 с, 23 рис., 3 табл., 4 источника, 6 прил.

АРХИТЕКТУРА ПЛАТФОРМЫ, АДМИНИСТРАТИВНАЯ ПАНЕЛЬ, КОНСТРУКТОР, TELEGRAM БОТ, БИЗНЕС-ПРОЦЕССЫ СПОРТЗАЛОВ, UML, ERD, JSON.

Объект курсового проекта – информационная платформа, позволяющая создавать и запускать Telegram ботов для организаций, оказывающих услуги спортклубов.

Цель курсового проекта – сокращение трудозатрат на разработку Telegram ботов для спортклубов.

Результат работы – спроектированная архитектура информационной системы, выполняющей функции по построению и запуску Telegram ботов для оказания доступа к услугам спортклубов.

## Содержание

Введение.....	5
1 Анализ предметной области .....	6
1.1 Актуальность темы .....	6
1.2 Обзор аналогов .....	7
1.2.1 Бот-платформа «ManyBot».....	8
1.2.2 Конструктор Telegram ботов «Puzzlebot».....	8
1.2.3 Конструктор чат-ботов «Botmother».....	9
2 Расширенное техническое задание.....	12
2.1 Основание для разработки .....	12
2.2 Цель и задачи проектирования .....	12
2.2.1 Структура информационной системы. ....	12
2.2.2 Требования к функциональным характеристикам .....	13
2.2.2.1 Административная панель .....	13
2.2.2.2 Telegram бот.....	14
2.2.2.3 Конструктор.....	15
2.2.3 Ролевая модель управления доступом .....	15
2.2.4 Требования к пользовательскому интерфейсу административной панели .....	17
2.2.5 Требования к программному обеспечению серверной части.....	17
2.2.6 Требования к клиентскому программному обеспечению.....	18
2.3 Стадии разработки .....	18
2.4 Общие положения .....	18
3 Разработка структуры системы.....	19
3.1 Обобщенная структура системы.....	19
3.2 Административная панель .....	22
3.2.1 Структура административной панели.....	22

					<b>ТПЖА.090301.331 ПЗ</b>		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.	Крючков				<b>Проектирование архитектуры платформы для создания телеграм ботов для спортзала. Telegram бот</b>	Литера	Лист
Пров.	Клюкин						3
						<b>Кафедра ЭВМ Группа ИВТ-31</b>	
Реценз.							
						Листов	54

3.2.2 Обработка запросов .....	24
3.2.3 Ролевая модель доступа.....	26
3.2.4 Структура передаваемых данных.....	27
3.2.5 Пользовательский интерфейс административной панели .....	30
3.3 Конструктор .....	31
3.3.1 Структура конструктора.....	31
3.3.2 Пользовательский интерфейс конструктора .....	34
3.4 Telegram бот.....	35
3.4.1 Структура бота .....	37
3.4.2 Пример взаимодействия пользователя с ботом .....	39
4 Структура базы данных .....	43
4.1 Общие сведения.....	43
4.2 Концептуальная структура.....	44
4.3 Логическая структура .....	45
Приложение А. Диаграмма состояний для административной панели .....	49
Приложение Б. Диаграмма состояний для модуля «бот» административной панели .....	50
Приложение В. Диаграмма состояний для модуля «организация» административной панели.....	51
Приложение Г. Диаграмма состояний для конструктора .....	52
Приложение Д. Перечень сокращений .....	53
Приложение Е. Библиографический список .....	54

					ТПЖА.090301.331 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

## Введение

В современном мире стали популярными такие приложения для быстрого общения как мессенджеры. Таких приложений достаточно много, но большинство пользователей сети интернет все чаще отдают предпочтение мессенджеру Telegram как наиболее удобному и надежному.

У Telegram имеется удобное API для создания ботов. Бот способен выполнять определенные команды, заданные пользователем через интерфейс Telegram. Данный функционал вполне может удовлетворять потребности компании в предоставлении некоторых услуг в разных сферах. Спортивные залы являются одной из таких сфер.

Создание ботов — это трудоемкий процесс, требующий квалифицированных программистов, что довольно затратно для бизнеса.

Платформа для создания ботов для спортзала будет решением, предоставляющим компаниям средство для быстрого создания и запуска Telegram ботов с требуемым функционалом для закрытия базовых потребностей этой области.

					ТПЖА.090301.331 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

## 1 Анализ предметной области

В данном разделе проводится анализ предметной области, выполняется обзор существующих программных решений для выполнения поставленной задачи.

### 1.1 Актуальность темы

Боты в мессенджере Telegram становятся все более популярными и число их пользователей постоянно растет. Они помогают пользователям выполнять типичные рутинные действия в автоматизированном режиме, значительно упрощая им жизнь. Для владельцев же самих ботов они стали незаменимыми помощниками в работе.

Telegram-боты имеют множество плюсов, таких как:

- Круглосуточный доступ;
- Моментальный ответ на запрос пользователя;
- Удобство использования, интуитивно понятный интерфейс;
- Не требуют установки дополнительных программ, общение с ботом ведется через мессенджер;
- Широкий набор реализуемых функций

Telegram-бот используют в коммерческой деятельности для следующих сфер и задач:

- Развлечения;
- Поиск и обмен файлами;
- Предоставление новостей;
- Утилиты и инструменты;
- Интеграция с другими сервисами;
- Осуществление онлайн-платежей.

					ТПЖА.090301.331 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

С популярностью ботов стали появляться все больше различных конструкторов, которые позволяют без наличия специальных знаний и навыков создать своего бота всего в несколько кликов.

Конструктором называется NoCode инструмент, предназначенный для быстрого создания ботов без знания каких-либо языков программирования. Иными словами, весь процесс создания – это нажатие тех или иных кнопок и ввода текста (например, название кнопки, текст сообщения и т.д.).

Первое предназначение – упрощение работы. Ведь не все обладают знаниями и навыками программирования. Когда боты только появились, их могли разрабатывать только программисты, обладающие соответствующим опытом и навыками

Помимо того, что конструкторы позволяют расширить аудиторию, способную создавать Telegram-ботов, они экономят время разработчикам. При наличии конструктора нет необходимости разрабатывать каждый раз отдельное приложение для выполнения типовых задач, так как конструктор предоставляет необходимый набор инструментов для быстрого создания бота без необходимости писать код.

Но у конструкторов есть некоторые ограничения, например, при их использовании нельзя выйти за рамки возможностей самого конструктора, а также при выходе нового функционала Telegram API, его реализация в конструкторе происходит с некоторой задержкой. Кроме того, боты, реализованные с помощью NoCode платформы могут быть менее производительные, чем их аналоги, написанные языке программирования.

## 1.2 Обзор аналогов

В подпунктах данного раздела рассматриваются существующие аналоги. В качестве рассматриваемых аналогов были выбраны приложения,

					ТПЖА.090301.331 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

реализующие функционал создания Telegram ботов с помощью конструктора.

### 1.2.1 Бот-платформа «ManyBot»

Один из наиболее популярных конструкторов. Работает внутри мессенджера Телеграм. Он бесплатный и прост в использовании.

Бот предоставляет возможность создания бота со следующими функциональными возможностями:

- Отправка сообщений;
- Создание меню;
- Автопостинг из VK, Twitter, YouTube;
- Поддержка нескольких языков.

Минусы конструктора состоят в том, что:

- нет администрирования бота за пределами мессенджера;
- наличие рекламного сообщения в созданном боте;
- малое количество компонентов и их модификаций;
- отсутствие статистических данных по созданному боту.

### 1.2.2 Конструктор Telegram ботов «Puzzlebot»

Данный конструктор имеет намного больше возможностей, чем предыдущий сервис:

- удобный личный кабинет;
- интуитивный интерфейс;
- имеет намного больше компонентов, позволяющих реализовывать сложных ботов.

Минусы данного решения:

					ТПЖА.090301.331 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		



- на бесплатном тарифе количество участников бота ограничено 150 пользователями;
- присутствует реклама.

### 1.2.3 Конструктор чат-ботов «Botmother»

Довольно мощный сервис по созданию ботов, который имеет удобный интерфейс для администрирования и создания ботов, предоставляет много различных компонентов, возможность просмотра статистики созданного бота. Но бесплатный пакет ограничен в функционале.

В таблице 1 представлено сравнение вышеперечисленных аналогов.

Таблица 1 – Сравнение аналогов

Аналоги Требования	Manybot	Puzzlebot	Botmother
Удобный доступ для администрирования бота	-	+	+
Изменение порядка вызовов компонентов	-	+	+
Изменение свойств компонента	-	+	+
Нет ограничений на использования компонентов	+	+	-
Отсутствие рекламы	-	-	+

Как видно из таблицы 1, существующие решения имеют ряд недостатков. Также конструкторы больше ориентированы на получение прибыли и ограничивают функционал для бесплатного пользования.

Рассмотрение аналогов демонстрирует актуальность задачи разработки платформы для создания чат-ботов, так как существующие решения

обладают недостатками. Перечень функциональных требований в расширенном техническом задании на разработку платформы должен включать требования, которые качественно будут отличать ее от существующих аналогов, исключать рассмотренные недостатки.

					ТПЖА.090301.331 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

## Вывод

В данном разделе был проведен анализ предметной области и осуществлен обзор аналогов. Telegram боты являются функциональными инструментами для многих пользователей, однако для их разработки зачастую требуются навыки программирования, что усложняет их внедрение в бизнес-процессы компаний. Из рассмотренных аналогов были выявлены требуемые функциональные возможности разрабатываемого продукта. Таким образом, проблема является актуальной, и разработка платформы для создания ботов для спортивных залов является целесообразной.

					ТПЖА.090301.331 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

## 2 Расширенное техническое задание

В данном разделе представлено расширенное техническое задание на разработку конструктора Telegram-ботов для спортазала.

### 2.1 Основание для разработки

Программа разрабатывается на основе задания на курсовое проектирование, полученного от заказчика в лице директора ООО «Синаптик», г. Киров.

### 2.2 Цель и задачи проектирования

Целью проектирования является сокращение трудозатрат на разработку Telegram ботов для спортклубов.

Задачи:

- обеспечить возможность быстрого создания Telegram ботов для спортивных залов в среде визуального проектирования;
- предусмотреть возможность задания собственного поведения бота при взаимодействии с клиентом;
- предоставить клиентам спортивного клуба удаленный доступ к сервисам компании, таким как покупка абонемента, запись на тренировки через мессенджер Telegram.

#### 2.2.1 Структура информационной системы.

Структура приложения состоит из трех основных блоков:

					ТПЖА.090301.331 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

- Административная панель для компании, предоставляющая услуги создания и управления ботом;
- Конструктор для редактирования функционала бота, представляющая из себя визуальную среду разработки;
- Телеграм бот.

## 2.2.2 Требования к функциональным характеристикам

Ниже представлены требования к функциональным характеристикам системы.

### 2.2.2.1 Административная панель

Административная панель должна предоставлять такие функции, как:

- создание аккаунта пользователя;
- создание организации пользователем;
- создание ролей для организации;
- задание полномочий для роли;
- привязка к организации других пользователей;
- создание ботов;
- привязка пользователей с определенной ролью к ботам;
- создание залов и привязка их к определенным ботам;
- создание абонементов;
- привязка абонементов к определенным залам;
- создание расписание для зала;
- создание тренировок и привязка к ним тренеров;
- создание новостей.

Диаграмма вариантов использования представлена на рисунке 1.

					ТГЖА.090301.331 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

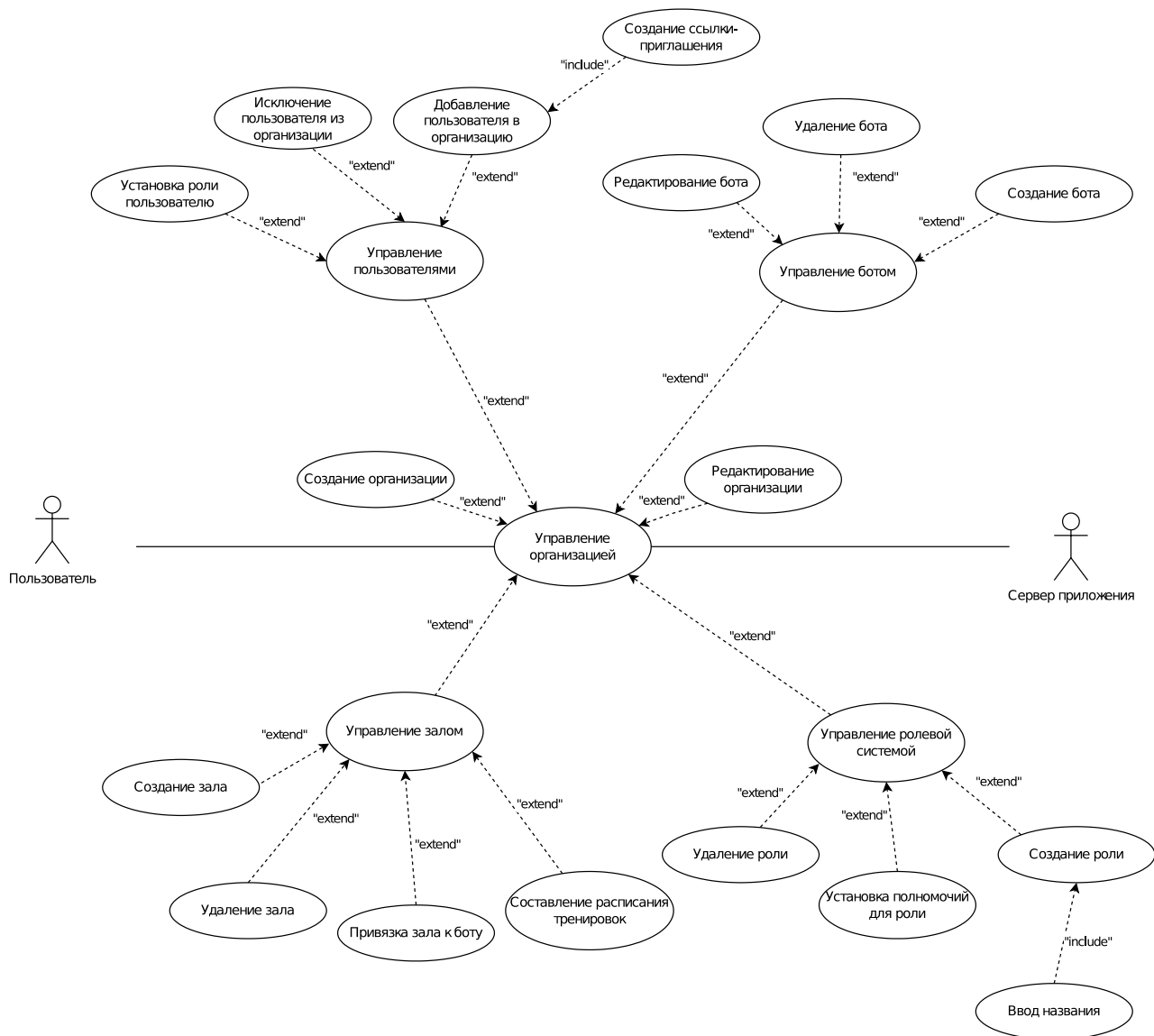


Рисунок 1 – Диаграмма вариантов использования

Административная панель имеет ролевую модель управления доступом, которая рассматривается в п. 2.3.3.

#### 2.2.2.2 Telegram бот

Бот должен предоставлять такие функции, как:

- покупка абонемента;
- уведомления клиентов спортзала о событиях;
- запись на тренировки;

- новости спортивного клуба.

### 2.2.2.3 Конструктор

Конструктор состоит из компонентов, которые можно описать определенным шаблоном.

Структура шаблона:

- Заголовок, кратко описывающий функционал;
- Параметры.

Функции конструктора:

- включение в структуру бота определенных компонентов;
- изменение порядка вызова компонентов;
- изменение значений параметров компонента.

### 2.2.3 Ролевая модель управления доступом

При регистрации пользователя в системе для него автоматически создается организация, в которой ему присваивается роль Администратор.

Пользователь с ролью «Администратор» может добавлять новые роли, действующие в его организации и сформированные из доступных полномочий.

Доступные привилегий для ролей

- изменение информации о компании;
- добавление пользователя в организацию;
- удаление пользователя из организации;
- создание ролей;
- редактирование полномочий пользователя;
- редактирование полномочий для ролей;

					ТПЖА.090301.331 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

- редактирование настроек бота;
- доступ к конструктору ботов;
- добавление/редактирование/удаление новостей в боте;
- изменение статуса новости;
- редактирование расписания;
- редактирование информации о зале;
- создание/редактирование/удаление залов;
- создание/редактирование/удаление ботов;
- просмотр статистики бота.

Список ролей представлен в таблице 2.

Таблица 2 – Роли

Наименование роли	Привилегии роли
Администратор	<ul style="list-style-type: none"> <li>– Добавление новых ролей;</li> <li>– Назначение роли пользователю;</li> <li>– Добавление и изменение ботов;</li> </ul>
Редактор расписания тренировок	<ul style="list-style-type: none"> <li>– Изменение информации о тренировках;</li> <li>– Добавление тренировок.</li> </ul>
Редактор бота	<ul style="list-style-type: none"> <li>– Изменять информацию о боте;</li> <li>– Добавлять пользователей к боту;</li> </ul>

Система управления доступом должна позволять редактировать существующие роли, создать новые роли на основе списка привилегий и удалять неиспользуемые роли.



## 2.2.4 Требования к пользовательскому интерфейсу административной панели

Интерфейс системы должен поддерживать русский язык.

Интерфейс системы должен быть спроектирован с учетом ролевой системы и уровней доступа пользователей.

Интерфейс системы должен обеспечивать наглядное, интуитивно понятное представление структуры размещенной информации, быстрый и логичный переход к соответствующим разделам системы.

Навигационные элементы интерфейса системы должны обеспечивать однозначное понимание пользователем их смысла и обеспечивать навигацию по всем доступным пользователю разделам системы и отображать соответствующую информацию.

Интерфейс должен быть адаптивным, т.е. обеспечивать высокую степень удобства использования не только на настольных экранах с шириной до 2560 пикселей, но и на мобильных устройствах с шириной экрана от 375 пикселей.

## 2.2.5 Требования к системному программному обеспечению серверной части

Для функционирования приложения необходимо следующее программное обеспечение:

- Операционная система – Ubuntu версии не ниже 18 или Debian версии не ниже 9;
- СУБД PostgreSQL версии не ниже 12.0.

					ТПЖА.090301.331 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

## 2.2.6 Требования к клиентскому программному обеспечению

Сайт должен быть доступен для полнофункционального просмотра с помощью следующих браузеров:

- Edge 88.0 и выше;
- Opera 43.0 и выше;
- Mozilla Firefox 55.0;
- Google Chrome 64.0 и выше.

## 2.3 Стадии разработки

Разработка должна быть проведена в четыре стадии:

- разработка технического задания;
- проектирование архитектуры приложения;
- разработка документации.

## 2.4 Требования к приемо-сдаточным процедурам

Приемка осуществляется комиссией, в состав которой должны входить представители Заказчика. Вниманию комиссии должны быть представлены эксплуатационные документы, разработанная архитектура и доклад.

## Вывод

В данном разделе было составлено расширенное техническое задания. Определены основные требования к разрабатываемому программному продукту.

					ТПЖА.090301.331 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

### 3 Разработка структуры системы

На данном этапе работы необходимо в соответствии с требованиями, поставленными в техническом задании, разработать алгоритмы функционирования, модульную структуру и механизмы взаимодействия компонентов информационной системы.

#### 3.1 Обобщенная структура системы

Система состоит из трех основных блоков:

- Административная панель для компании, предоставляющая услуги создания и управления ботом;
- Конструктор для редактирования функционала бота;
- Телеграм бот.

Приложения взаимодействуют с общей базой данных через единую систему управления базами данных, тем самым управляя состоянием системы.

Архитектура системы представлена на рисунке 2.

					ТПЖА.090301.331 ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

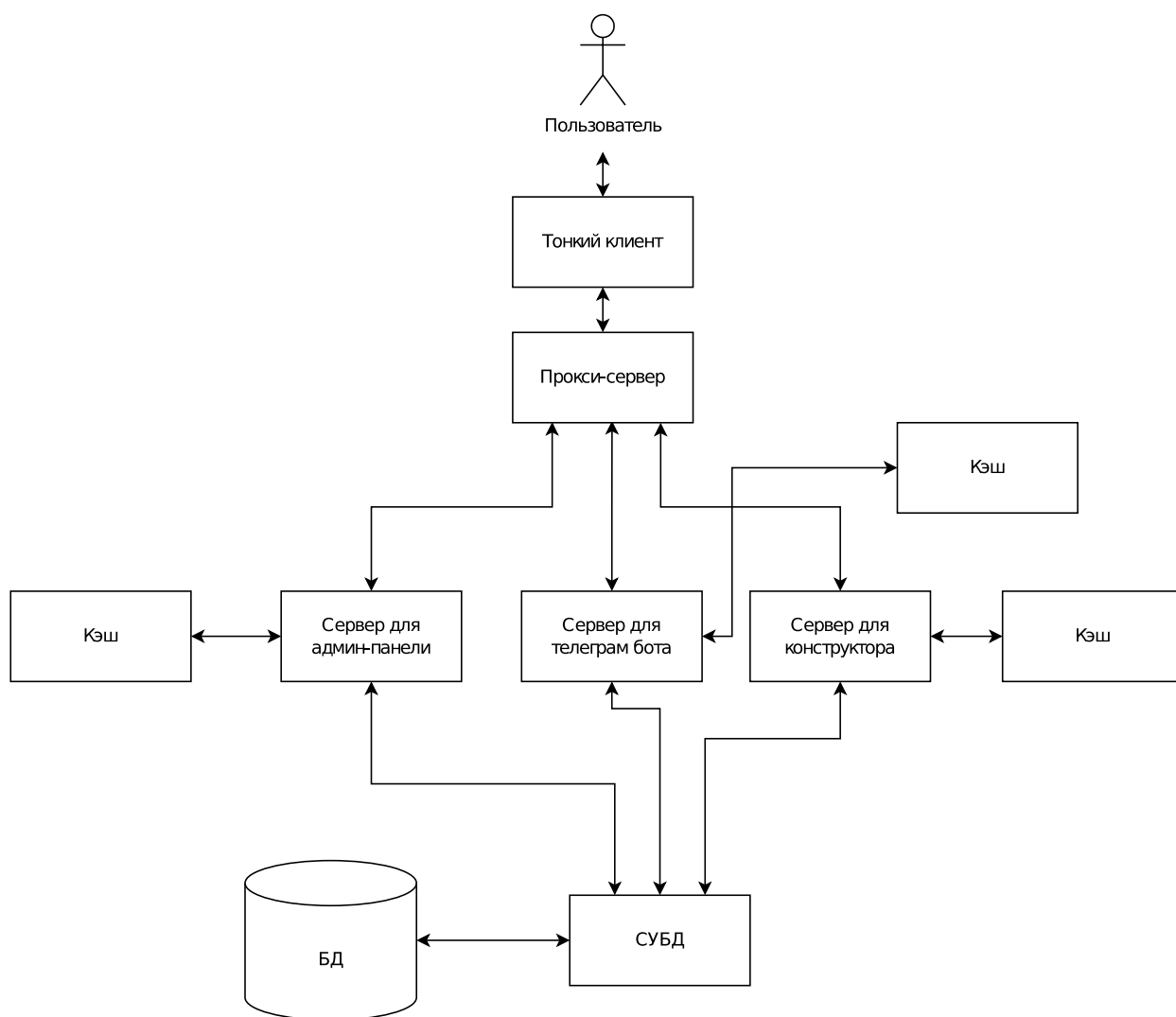


Рисунок 2 – Архитектура системы

Каждое приложение реализует клиент-серверную архитектуру, используя протокол HTTP(S).

В структуре системы используется общая реляционная база данных, доступная для всех указанных приложений. К достоинствам использования разделяемой базы данных можно отнести:

- 1) Обеспечение согласованности хранящейся в ней информации

Неоспоримым преимуществом использования реляционной базы данных является гарантия согласованности данных. Реляционные СУБД предоставляют механизм транзакций, гарантирующий согласованность данных в любой момент времени. Транзакция – это конечный набор

операций, который переводит одно согласованное состояние в другое согласованное состояние. В случае ошибки на любом шаге СУБД отменяет все выполненные ранее операции и возвращает данные к первоначальному согласованному состоянию.

## 2) Единый формат данных для всех приложений системы

Все приложения выполняют приведение данных к единым типам, тем самым исключая конфликты, связанные с обработкой данных.

Использование общей базы данных имеет ряд недостатков:

### 1) Высокая связанность приложений

Изменения в некотором приложении системы могут потребовать изменений в структуре базы данных, что в свою очередь скажется на работе всех приложений. Это затрудняет обновление существующих и внедрение новых функциональных возможностей в систему.

### 2) Общая база данных является точкой отказа всей системы

При выходе из строя БД, функционирование всей системы в целом становится невозможным.

### 3) Узкое место в обмене данными системы

При интенсивном обмене данными БД может стать узким местом системы. Появляется конкурентность доступа к данным, возможны возникновения блокировок на изменения данных.

При построении структуры проекта был выбран вариант с единой базой данных в связи с простотой дальнейшей реализации.

Структура базы данных приведена в разделе 4.

					ТПЖА.090301.331 ПЗ	Лист
						21
Изм.	Лист	№ докум.	Подпись	Дата		

## 3.2 Административная панель

Административная панель – это веб-приложение для управления ботами. Она позволяет контролировать текущее состояние ботов: их конфигурацию, содержимое и пользователей.

### 3.2.1 Структура административной панели

Модульная структура административной панели представлена на рисунке 3.

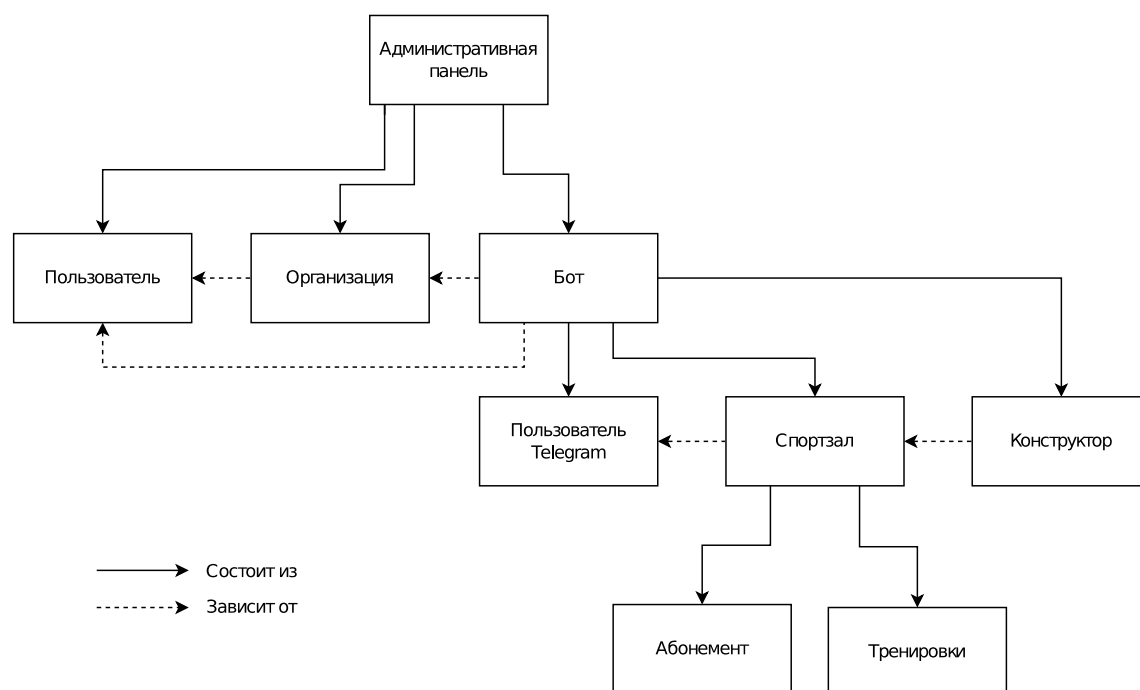


Рисунок 3 – Модульная структура административной панели

Структура административной панели представляет из себя иерархию модулей, каждый из которых выполняет определенную задачу. Так, административная панель зависит от трех модулей:

- Пользователь;
- Организация;

– Бот.

Модуль пользователя отвечает за регистрацию, авторизацию, редактирование, получение информации о пользователе и его удаление.

Модуль организации позволяет создавать организации, присоединять пользователей к организации, а также исключать их оттуда. Этот модуль зависим от модуля пользователя, т. к. обращается к нему за информацией. Подробное описание организаций приведено в п. 3.2.3.

Модуль бота содержит в себе ряд блоков(модулей):

- Модуль пользователя Telegram;
- Модуль конструктора;
- Модуль спортивного зала.

Модуль пользователя Telegram позволяет изменять и получать информацию о пользователе Telegram бота.

Модуль конструктора подробно в п. 3.3.

Модуль спортивного зала содержит подмодули, такие как абонемент, тренировки и т.д. Этот модуль и подмодули содержат функционал, который позволяет пользователям административной панели добавлять новые залы, редактировать и удалять их, составлять расписание тренировок для них и добавлять абонементы.

Исходя из описания модулей административной панели и их функций были составлены диаграммы:

- диаграмма состояний, описывающая поведение административной панели, представленная в приложении А;
- диаграмма состояний, описывающая поведение модуля «бот» административной панели, представленная в приложении Б;
- диаграмма состояний, описывающая поведение модуля «организация» административной панели, представленная в приложении В;

					ТПЖА.090301.331 ПЗ	Лист
						23
Изм.	Лист	№ докум.	Подпись	Дата		

– диаграмма состояний, описывающая поведение конструктора, представленная в приложении Г.

### 3.2.2 Обработка запросов

Стартовая строка запроса содержит путь к запрашиваемому ресурсу. По этому пути на сервере происходит вызов определенного обработчика.

Обработчики функций в административной панели имеют похожий алгоритм, представленный на рисунке 4.

					ТПЖА.090301.331 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		



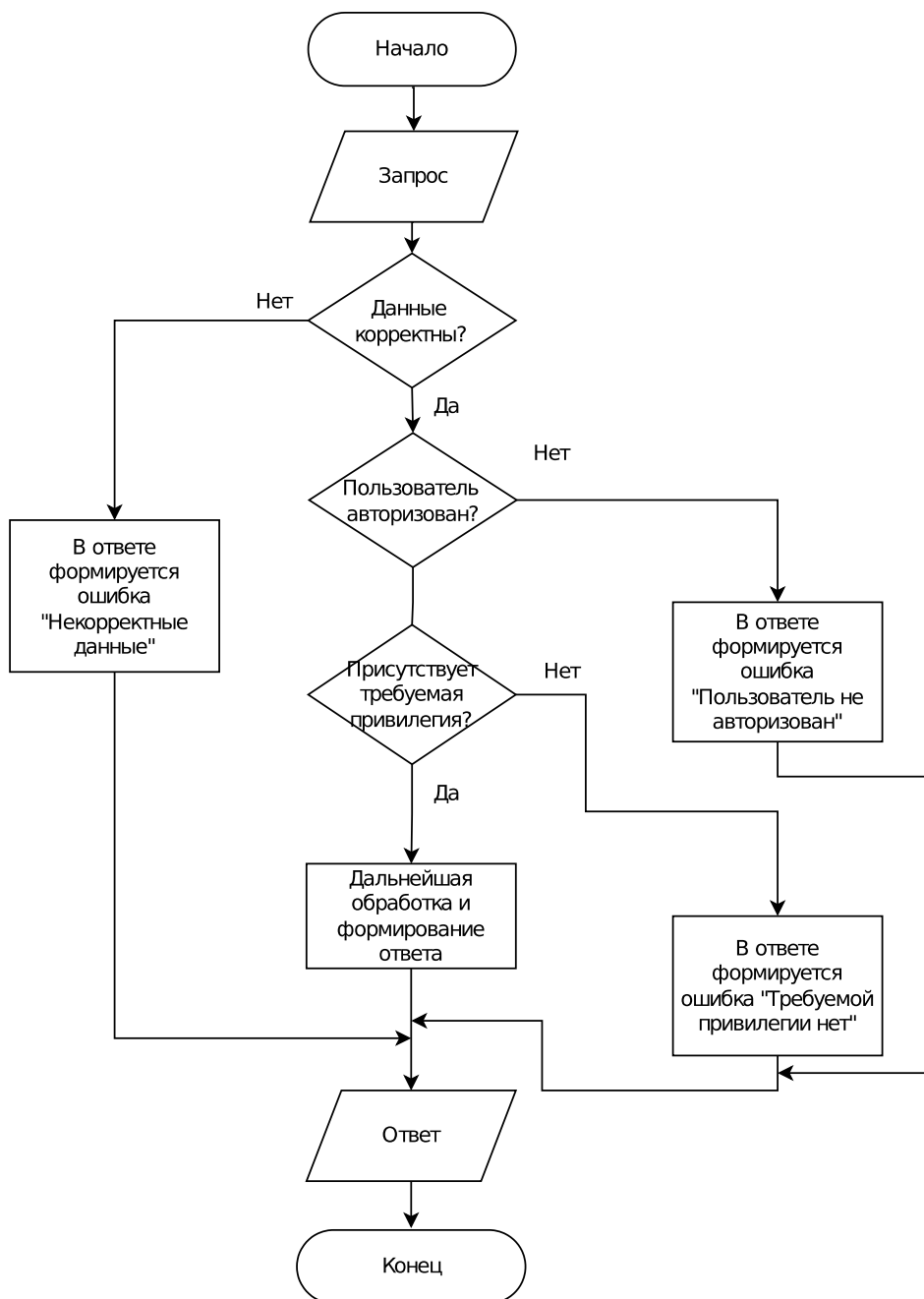


Рисунок 4 – Алгоритм обработчика

При вызове обработчика сначала проверяется корректность принятых данных, и в случае некорректности - выдается соответствующая ошибка. Далее идет проверка авторизации и наличия требуемой привилегии у пользователя. При их отсутствии обработка прекращается и выдается ошибка доступа. Если ошибок в запросе не нашлось и доступ предоставлен, то происходит дальнейшая обработка запроса.

Доступ к просмотру или изменению той или иной информации предоставляется системой ролей, представленной в п. 3.2.3.

Структура пересылаемых данных подробно рассматривается в п. 3.2.4.

### 3.2.3 Ролевая модель доступа

При регистрации пользователя в системе для него автоматически создается организация, в которой ему присваивается роль Администратор. Администратор может добавить других пользователей в свою организацию и назначить им определенные роли.

Управление доступом к инструментам платформы выполнено на базе ролей. Роли определяют круг доступных пользователю операций для контролирования той или иной информации в организации. Администраторы используют роли для объединения привилегий в наборы, которые можно назначать учетным записям пользователей. Каждый связанный с ролью пользователь обладает всеми включенными в нее привилегиями.

Привилегии пользователей хранятся в промежуточных таблицах между пользователем - организацией и пользователем - ботом, позволяя тем самым разграничить доступ как к функционалу организаций, так и к функционалу бота.

Набор привилегий в базе данных хранится в 64-битном формате целого числа, позволяя работать с ними с помощью битовых масок.

Для расширения наборов привилегий, в случае выхода за заданный диапазон представления числа, осуществляется их группировка.

Группировка представляет собой целое число, содержащее набор привилегии. Программист дополняет таблицу базы данных новыми группами в случае расширения функционала, требующего сопоставлений с новыми привилегиями.

					ТПЖА.090301.331 ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

Структура хранения привилегий представлена на рисунке 5.

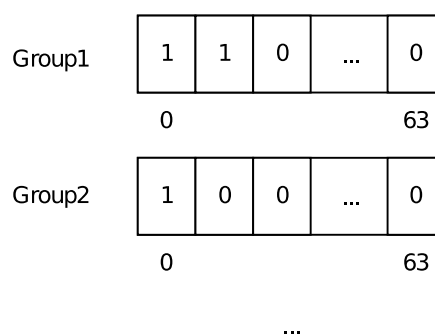


Рисунок 5 – Структура хранения привилегий

Как было сказано выше, привилегии объединяются в роли, которые хранятся в отдельной таблице базы данных. Администратор может назначать роль пользователям, но также есть возможность изменения некоторых привилегий для отдельного пользователя.

### 3.2.4 Структура передаваемых данных

Рассмотрим два формата передачи данных: XML и JSON

JSON – текстовый формат обмена данными, основанный на JavaScript. Применяется в веб-приложениях как для обмена данными между браузером и сервером, так и между серверами (программные HTTP-сопряжения).

Преимущество данного формата:

- удобен для чтения человеком;
- поскольку JSON является подмножеством синтаксиса языка JavaScript, то он может быть быстро десериализован стандартной библиотекой этого языка на стороне браузера.

XML - формат документа, в котором использованы теги для определения объектов и их атрибутов. Используется для формирования структуры документа и как формат обмена данными.

Преимущества данного формата:

- удобен для создания структуры документа;
- расширяем.

Минусы:

- трудно читаем по сравнению с вышеописанным форматом;
- из-за тегов избыточен при обмене данными.

Из-за удобочитаемости и нативной поддержки в JavaScript для взаимодействия клиента и сервера был выбран формат JSON.

Запросы имеют общий шаблон, состоящий из передаваемых данных.

Ответы состоят из возвращаемых полей и информирующих признаков, таких как кода ошибки и сообщения, описывающего её.

Шаблон ответа представлен на рисунке 6.

```
{  
  // Поля  
  "errorCode": ...,  
  "message": "..."  
}
```

Рисунок 6 – Шаблон ответа сервера

Примеры запросов и ответов в формате JSON представлены в таблице 3.

					ТПЖА.090301.331 ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

Таблица 3 – Примеры запросов и ответов при взаимодействии клиента с сервером.

URL	Метод	Запрос	Ответ	Описание
/registration	POST	{ "login": "Ivan" "password": "klsdja112" "email": "Ivan@gmail.com" }	{ "errorCode":0, "message": "Success!" }	Регистрация
/auth	POST	{ "login": "Ivan" "password": "klsd112" }	{ "errorCode":2, "message": "Incorrect password!" }	Авторизация
/users/<id>	GET	id = 2	{ "name": "Ivan", "age":25, "email": Ivan@gmail.com, "errorCode":2, "message": "Success" }	Получить информацию о пользователе
/users/<id>?field =name	GET	id = 2	{ "name": "Ivan", "errorCode":2, "message": "Success" }	Получить имя пользователя
/user	GET	-	{ "name": "Ivan", "age":25, "email": Ivan@gmail.com, "errorCode":2, "message": "Success" }	Получить информацию об авторизованно м пользователе
/users/<id>?field =organizations	GET	id = 2	{ organizations: [ "A", "B" ], "errorCode":2, "message": "Success" }	Получить организации, в которых находится пользователь

### 3.2.5 Пользовательский интерфейс административной панели

Интерфейс административной панели содержит ряд компонентов:

- Верхнее меню, где находятся ссылки на основные разделы веб-приложения: пользователь, организация, бот.
- Меню слева, которое отражает возможные функции данного раздела;
- Область, где можно просматривать, манипулировать данными.

Обычно представлено в виде формы или списка.

Примеры шаблонов страниц административной панели приведены на рисунках 7-8.

Пункт раздела	Пункт раздела	Пользователь
Функция 1	<div>Форма</div> <div><div>Ввод</div><div><input type="text"/></div><div><input type="checkbox"/> Свойство 1</div><div><input type="checkbox"/> Свойство 2</div><div>Отправить</div></div>	

Рисунок 7 – Шаблон страницы с формой ввода


	Пункт раздела	Пункт раздела	 Пользователь												
Функция 1	<div>Список</div> <table border="1"> <thead> <tr> <th>id</th> <th>Поле1</th> <th>Поле2</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Поле1 объекта1</td> <td>Поле2 объекта1</td> </tr> <tr> <td>2</td> <td>Поле1 объекта2</td> <td>Поле2 объекта2</td> </tr> <tr> <td>3</td> <td>Поле1 объекта3</td> <td>Поле2 объекта3</td> </tr> </tbody> </table>			id	Поле1	Поле2	1	Поле1 объекта1	Поле2 объекта1	2	Поле1 объекта2	Поле2 объекта2	3	Поле1 объекта3	Поле2 объекта3
id				Поле1	Поле2										
1				Поле1 объекта1	Поле2 объекта1										
2				Поле1 объекта2	Поле2 объекта2										
3				Поле1 объекта3	Поле2 объекта3										
Функция 2															
Функция 3															
Функция 4															
Функция 5															

Рисунок 8 – Шаблон страницы с некоторым списком

### 3.3 Конструктор

#### 3.3.1 Структура конструктора

Конструктор - приложение, предоставляющее набор визуальных компонентов, которое позволяет быстро спроектировать и запустить Telegram бота. Для создания бота пользователю не нужно обладать глубокими знаниями о языках программирования. Конструктор предлагает набор компонентов, из которых можно собрать бота в несколько кликов.

Компоненты можно описать определенным шаблоном.

Структура шаблона компонента:

- Идентификатор
- Имя;
- Параметры
- Связи.

Идентификатор – уникальный номер компонента.

Имя - кратко описывает функционал компонента.

Параметры - значения, позволяющие настраивать изменяемые данные компонента. Например, у компонента «Отправка сообщения» одним из параметров будет «Текст сообщения».

Связи позволяют соединять компоненты между собой, реализуя тем самым переход от одного компонента к другому по определенному событию.

Каждый компонент обладает кроме специфичных только для него параметров, опциональным параметром “keyboard”, который определяет клавиатуру в боте. Клавиатура состоит из набора кнопок с определенным текстом. При нажатии на кнопку данной клавиатуры в интерфейсе Telegram бота происходит отправка сообщения, что значительно упрощает взаимодействие пользователя с ботом.

Для описания структуры и связи между компонентами используется формат данных JSON.

Шаблон структуры компонента представлен на рисунке 9.

```
{
  "id": ...,
  "name": ...,
  // параметры компонента
  "next_block": ...
}
```

Рисунок 9 – Шаблон структуры компонента

Примеры компонентов представлены на рисунках 10 и 12, их структуры в формате JSON представлены соответственно на рисунках 11 и 13.

					ТПЖА.090301.331 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		



Отправка сообщения

Текст сообщения

Добавить клавиатуру

Рисунок 10 – Компонент «Отправка сообщения»

```
{
  "id": 1,
  "name": "Отправка сообщения",
  "text": "Текст сообщения",
  "next_block": 3
}
```

Рисунок 11 – JSON структура компонента «Отправка сообщения»

Отправка изображения

Ссылка на изображение

Подпись

Выберите действие

Клавиатура

Пункт1

Пункт2

Рисунок 12 – Компонент «Меню»

```

{
  "id": 1,
  "name": "Отправка изображения",
  "imgUrl": "Ссылка на изображение",
  "caption": "Выберите действие",
  "keyboard": {
    "inline": false,
    "buttons": [
      {
        "text": "Пункт1",
        "next_block": 2,
      },
      {
        "text": "Пункт2",
        "next_block": 3,
      }
    ]
  }
}

```

Рисунок 13 – JSON структура компонента «Меню»

### 3.3.2 Пользовательский интерфейс конструктора

Интерфейс конструктора предоставляет наглядное, интуитивно понятное представление размещенной информации. Элементы управления обеспечивают однозначное понимание пользователем их смысла.

В структуре интерфейса конструктора можно выделить две основополагающие части:

- панель со списком доступных компонентов;
- визуальный редактор для размещения и связывания компонентов.

Шаблон интерфейса конструктора представлен на рисунке 14.

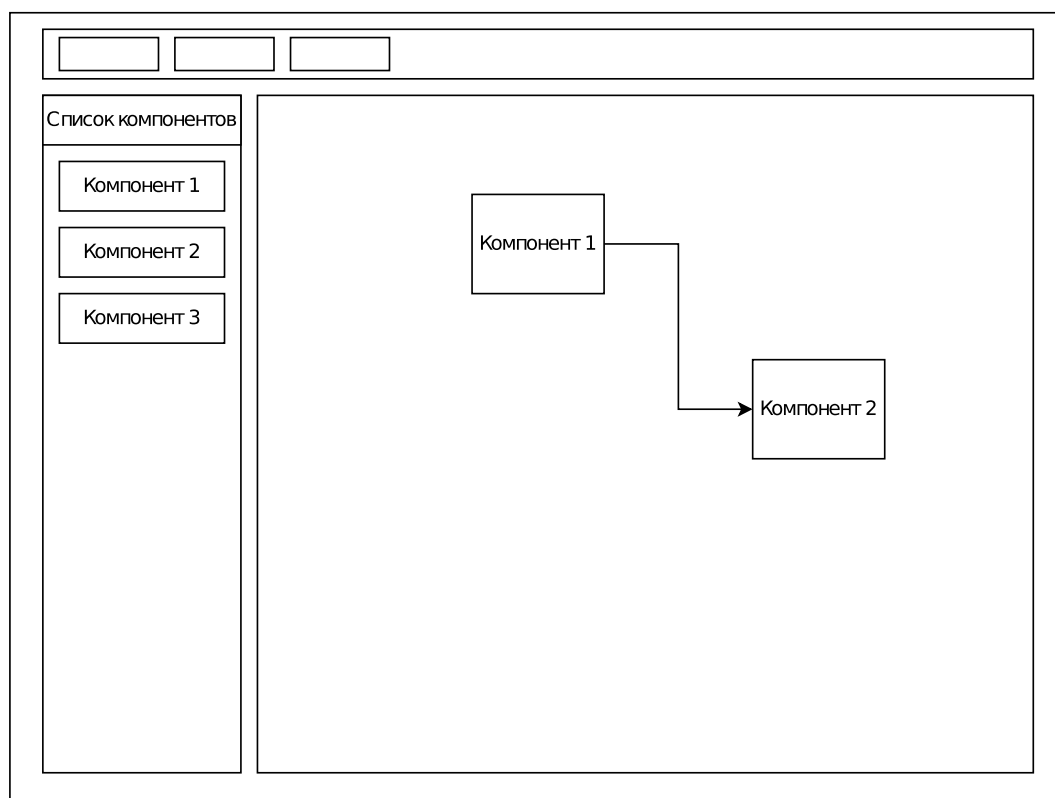


Рисунок 14 – Шаблон интерфейса конструктора

### 3.4 Telegram бот

Для эффективного взаимодействия пользователя с ботом он должен обладать набором таких функций как:

- регистрация клиента;
- покупка абонемента;
- запись на тренировки;
- уведомления клиентов спортзала о событиях;
- новости спортивного клуба.

Взаимодействие между пользователем и Telegram ботов происходит следующим образом:

- 1) Пользователь отправляет команду боту;
- 2) Telegram передает полученную команду на сервер с ботом;

- 3) Программа на сервере обрабатывает полученный от бота запрос и формирует определенный ответ;
- 4) Сервер отдает ответ пользователю.

Этот цикл повторяется каждый раз взаимодействия пользователя с Telegram ботом.

Процесс взаимодействия пользователя с ботом показан на диаграмме последовательности – рисунок 15.

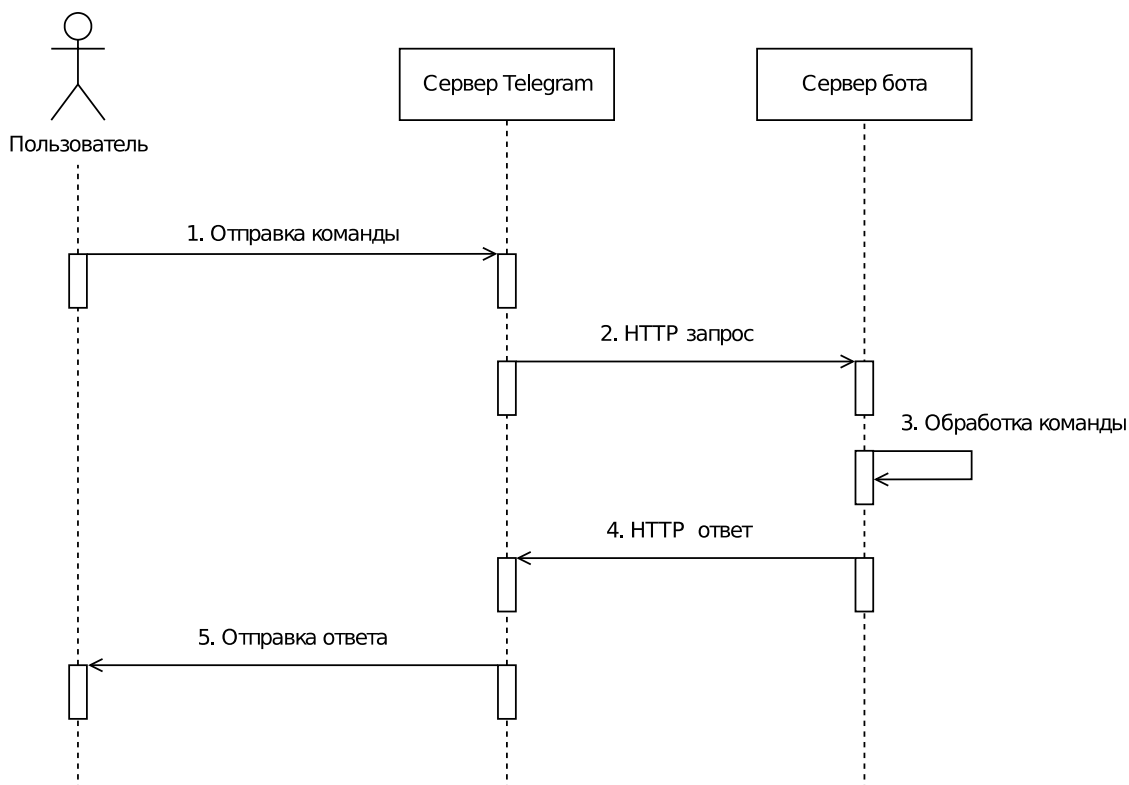


Рисунок 15 – Схема взаимодействия пользователя с ботом

### 3.4.1 Структура бота

Так как все создаваемые в конструкторе боты состоят из набора определенных компонентов и имеют схожие функции, для обработки входящих запросов на сервере запускается один экземпляр приложения, который обрабатывает входящие запросы всех для всех ботов. Telegram отправляет запросы о событиях в боте на url адреса, генерируемые для каждого бота при его первом запуске. При поступлении HTTP запроса от Telegram происходит получение информации о состоянии пользователя бота. После чего выполняется проверка соответствия полученных данных с данными, допустимыми в текущем состоянии пользователя. Затем происходит обработка полученных данных в соответствии с текущим состоянием пользователя и переход в новое состояние, для которого формируется и отправляется ответ. Схема обработки запросов пользователя показана на рисунке 16.

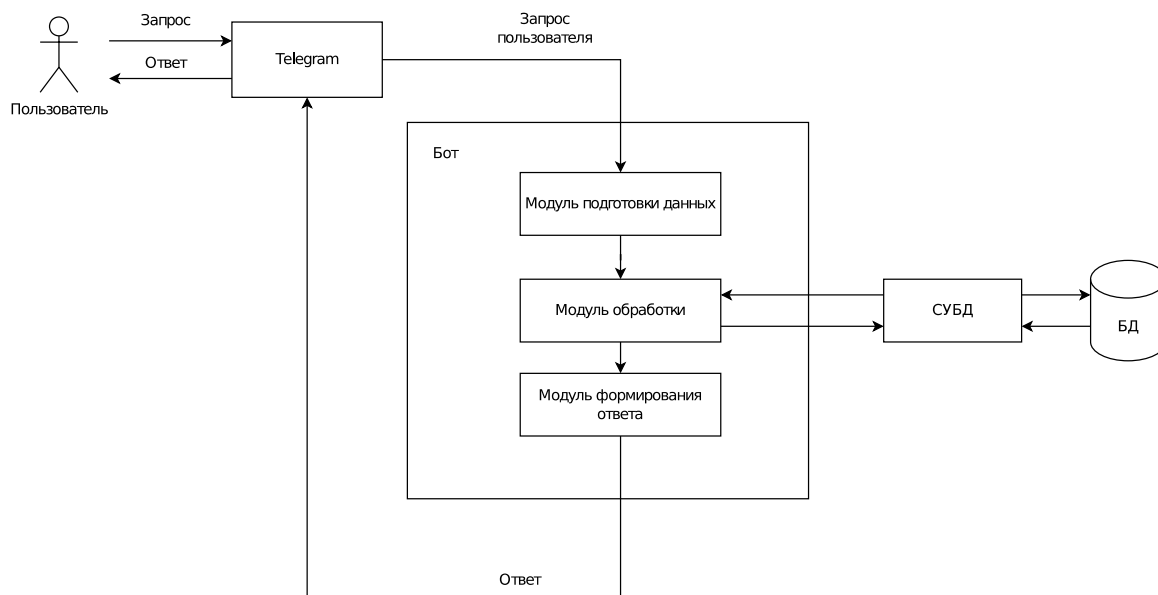


Рисунок 16 – Схема обработки запросов пользователя

Общий алгоритм обработки пользовательских запросов к боту в виде граф-схемы алгоритма приведен на рисунке 17.

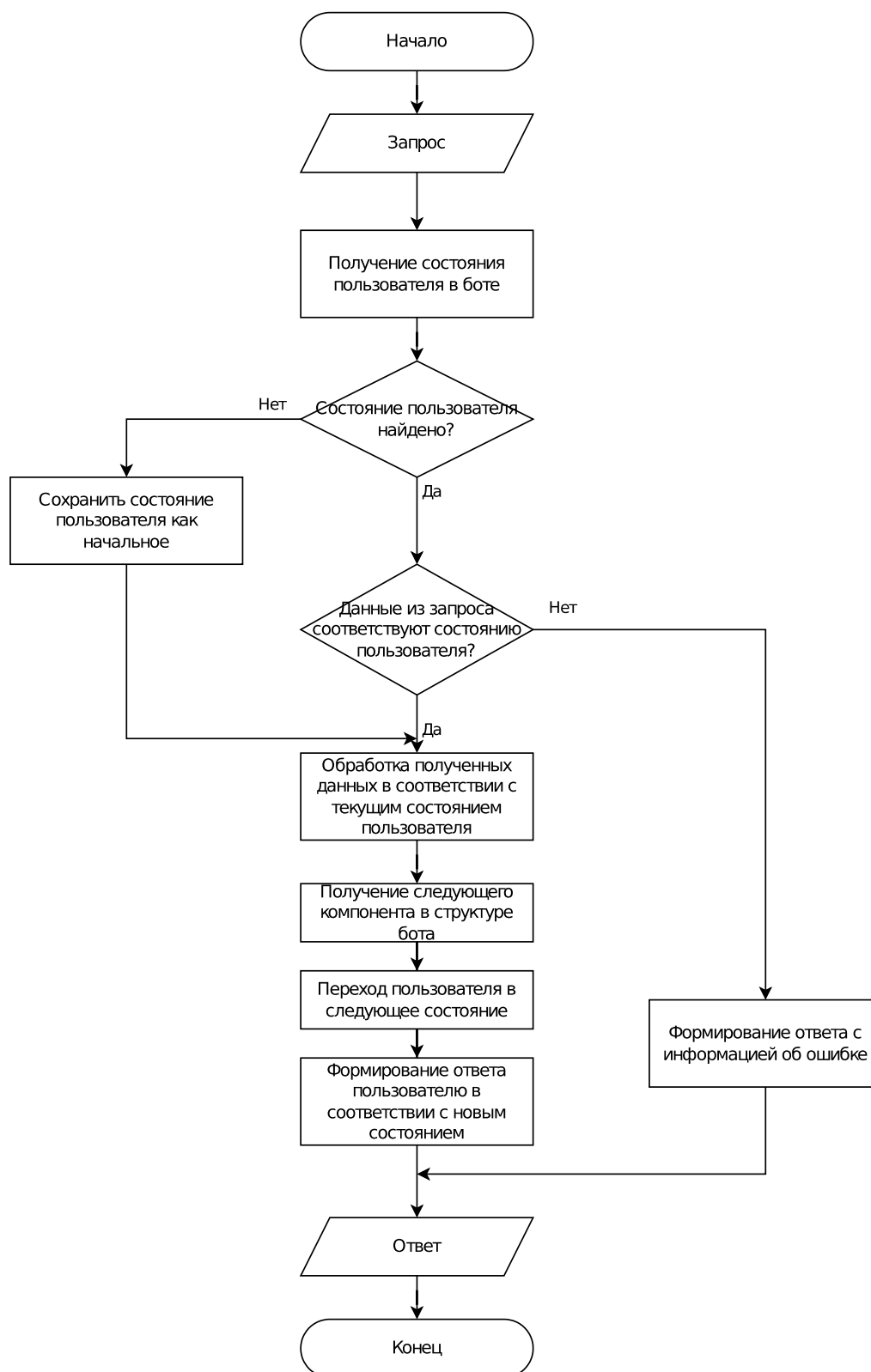


Рисунок 17 – ГСА обработки запроса пользователя

### 3.4.2 Пример взаимодействия пользователя с ботом

В качестве примера взаимодействия пользователя с Telegram ботом приводится бот, реализующий функционал настроек языка пользовательского интерфейса бота и просмотра правил клуба. Навигация по боту выполнена в виде меню, реализованного с помощью кнопок клавиатуры бота. Компонентная структура данного бота приведена на рисунке 18.

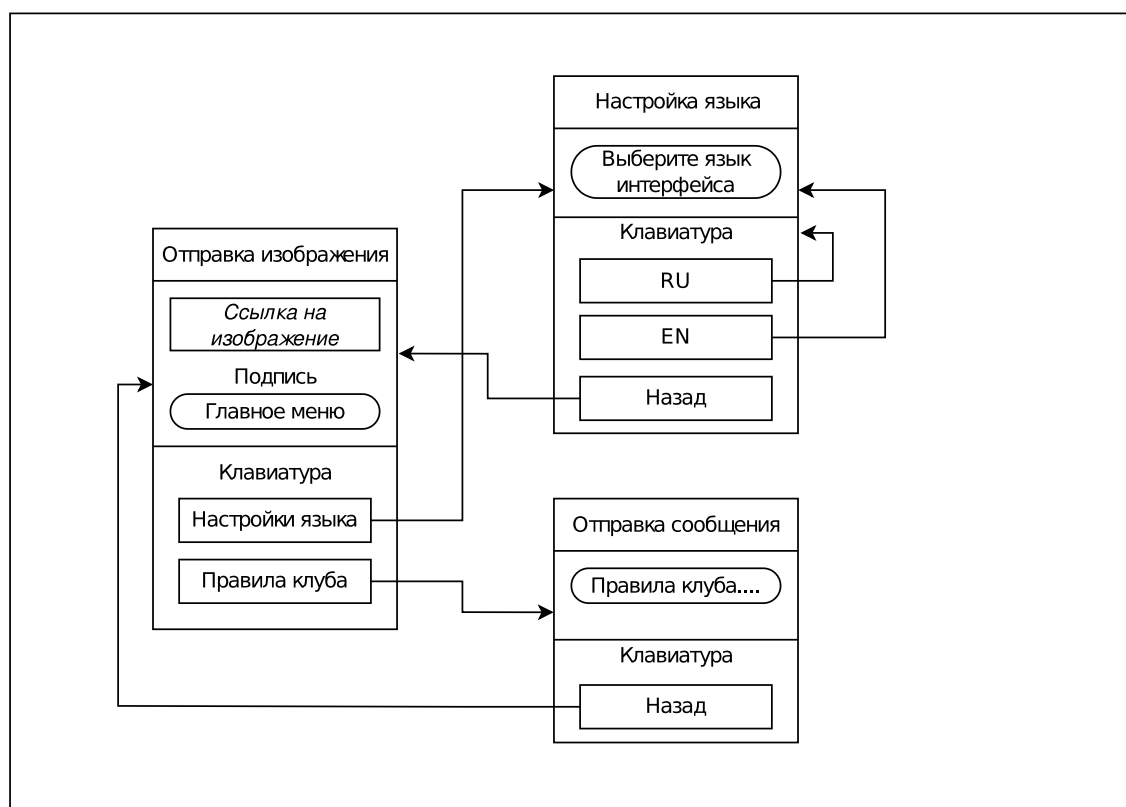


Рисунок 18 – Компонентная структура бота

Бот состоит из трех компонентов: “отправка изображения” для отправки изображения с приветствием и клавиатурой меню, компонента “Настройка языка” для выбора языка пользовательского интерфейса бота и компонента “Отправка сообщения”, который используется для отправки пользователю правил клуба. Связь компонентов можно описать в виде графа состояний.

Структуры компонентов в формате JSON представлены на рисунках 19-21.

```
{
  "id": 1,
  "name": "Отправка изображения",
  "imgUrl": "Ссылка на изображение"
  "caption": "Главное меню",
  "keyboard": {
    "inline": false,
    "buttons": [
      {
        "text": "Настройка языка",
        "next_block": 2,
      },
      {
        "text": "Правила клуба",
        "next_block": 3,
      }
    ]
  }
}
```

Рисунок 19 – JSON структура компонента «Отправка изображения»



```

{
  "id": 2,
  "name": "Настройка языка",
  "text": "Выберите язык интерфейса",
  "keyboard": {
    "inline": false,
    "buttons": [
      {
        "text": "RU",
        "next_block": 2,
      },
      {
        "text": "EN",
        "next_block": 2,
      },
      {
        "text": "Назад",
        "next_block": 1
      }
    ]
  }
}

```

Рисунок 20 – JSON структура компонента «Настройка языка»

```

{
  "id": 3,
  "name": "Отправка сообщения",
  "text": "Правила клуба....",
  "keyboard": {
    "inline": false,
    "buttons": [
      {
        "text": "Назад",
        "next_block": 1
      }
    ]
  }
}

```

Рисунок 21 – JSON структура компонента «Отправка сообщения»

## Вывод

В данном разделе выполнено проектирование структуры информационной системы. Разработаны алгоритмы функционирования административной панели, конструктора и Telegram бота. Построены шаблоны пользовательского интерфейса.

					ТПЖА.090301.331 ПЗ	Лист
						42
Изм.	Лист	№ докум.	Подпись	Дата		

## 4 Структура базы данных

В данном разделе необходимо спроектировать структуру базы данных, соответствующую предметной области.

### 4.1 Общие сведения

База данных — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, которая поддерживает одну или более областей применения. База данных обладает следующими свойствами:

- Быстродействие
- Простота получения и обновления данных
- Стандартизация
- Безопасность данных
- Целостность данных

Рассмотрим два вида баз данных: реляционную и нереляционную базы данных.

Реляционная база данных (SQL) – это набор данных с predetermined связями между ними. Эти данные организованы в виде набора таблиц (отношений), состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. Для взаимодействия с реляционными БД используется язык структурированных запросов - SQL.

Нереляционная база данных (NoSQL) - это база данных, в которой не используется табличная схема из строк и столбцов. В базах данных такого типа используется модель хранения, ориентированная на конкретные

					ТПЖА.090301.331 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

требования хранимых данных. Выделяют несколько видов NoSQL баз данных:

- БД «Ключ-значение»;
- Документно-ориентированные БД;
- Графовые;
- Колоночные БД.

#### 4.2 Концептуальная структура

Концептуальная модель является моделью наиболее высокого уровня абстракции, она создается без ориентации на какую-либо конкретную систему управления базами данных и модель данных. Она строится для описания и структуры базы данных. Концептуальная структура базы данных представлена на рисунке 22.

					ТПЖА.090301.331 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

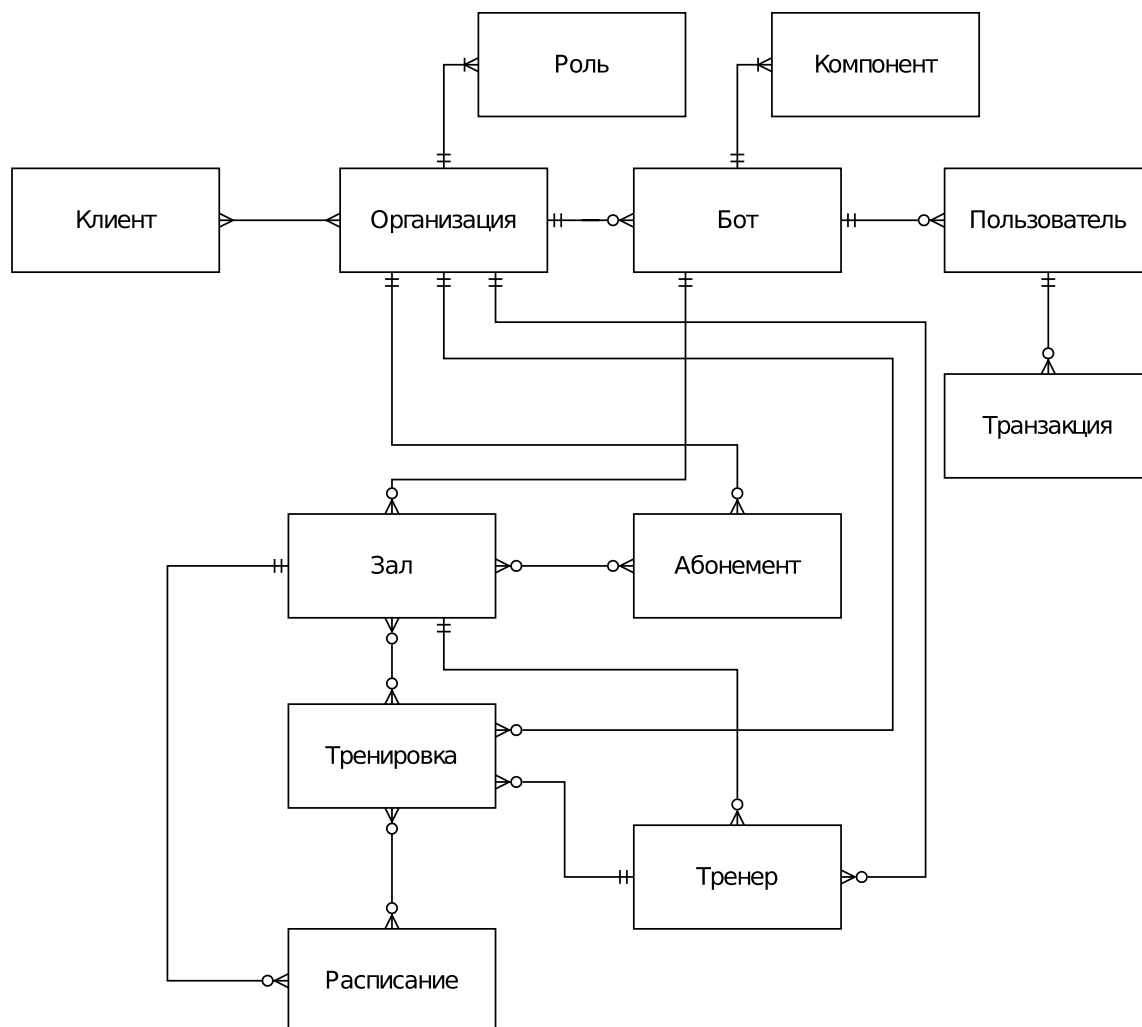


Рисунок 22 – Концептуальная схема базы данных

### 4.3 Логическая структура

Для проектирования логической схема базы данных используется реляционная модель, так как хранящиеся в ней данные представляют собой набор двумерных таблиц (отношений), в которых каждая строка уникальна и имеет один и тот же формат. Логическая схема базы данных представлена в виде ER диаграммы на рисунке 23.



организации. Также в промежуточной таблице содержится набор привилегий в виде 64-битного числа.

Сущность Бот содержит такие поля, как идентификатор, название бота, Telegram токен и идентификатор организации, к которой принадлежит бот. Также существует промежуточная таблица, которая служит для сопоставления пользователя и бота, и дает первому определенные привилегии для управления последним.

Сущность Роль содержит в себе идентификатор, имя, идентификатор организации и привилегии в виде 64-битного числа, и служит для хранения привилегий, привязанных к типовым ролям.

Сущность Зал содержит идентификатор, имя, адрес расположения спортзала и идентификатор бота, который способствует принадлежности спортивного зала определенному боту.

Сущность Абонемент содержит идентификатор, имя, цену, количество тренировок, время начало и конца, а также идентификатор организации. Абонемент принадлежит определенной организации. Также Абонементы могут быть привязаны ко многим залам при помощи промежуточной таблицы, которая их сопоставляет по идентификаторам.

Сущность Telegram пользователь содержит идентификатор Telegram, возраст, имя, идентификатор зала, телефон. Он привязывается к определенному боту через идентификатор бота.

## Вывод

В данном разделе в соответствии с предметной областью были спроектированы концептуальная и логическая схемы базы данных.

					ТПЖА.090301.331 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

## Заключение

В ходе выполнения курсового проекта была построена структура платформы для создания Telegram ботов для спортивных залов, представляющая собой набор блоков, которые взаимодействуют с общей базой данных.

Для блока административной панели был спроектирован механизм обработки запросов клиента, были созданы шаблоны пользовательского интерфейса для формы и списка и организация системы управления доступом на базе ролевой модели с помощью битовых масок.

Для блока конструктора были определены шаблоны компонента в формате JSON и пользовательского интерфейса.

Для блока бота был разработан алгоритм обработки запросов.

На основании предметной области и функциональных требований была спроектирована принципиальная и логическая структуры базы данных.

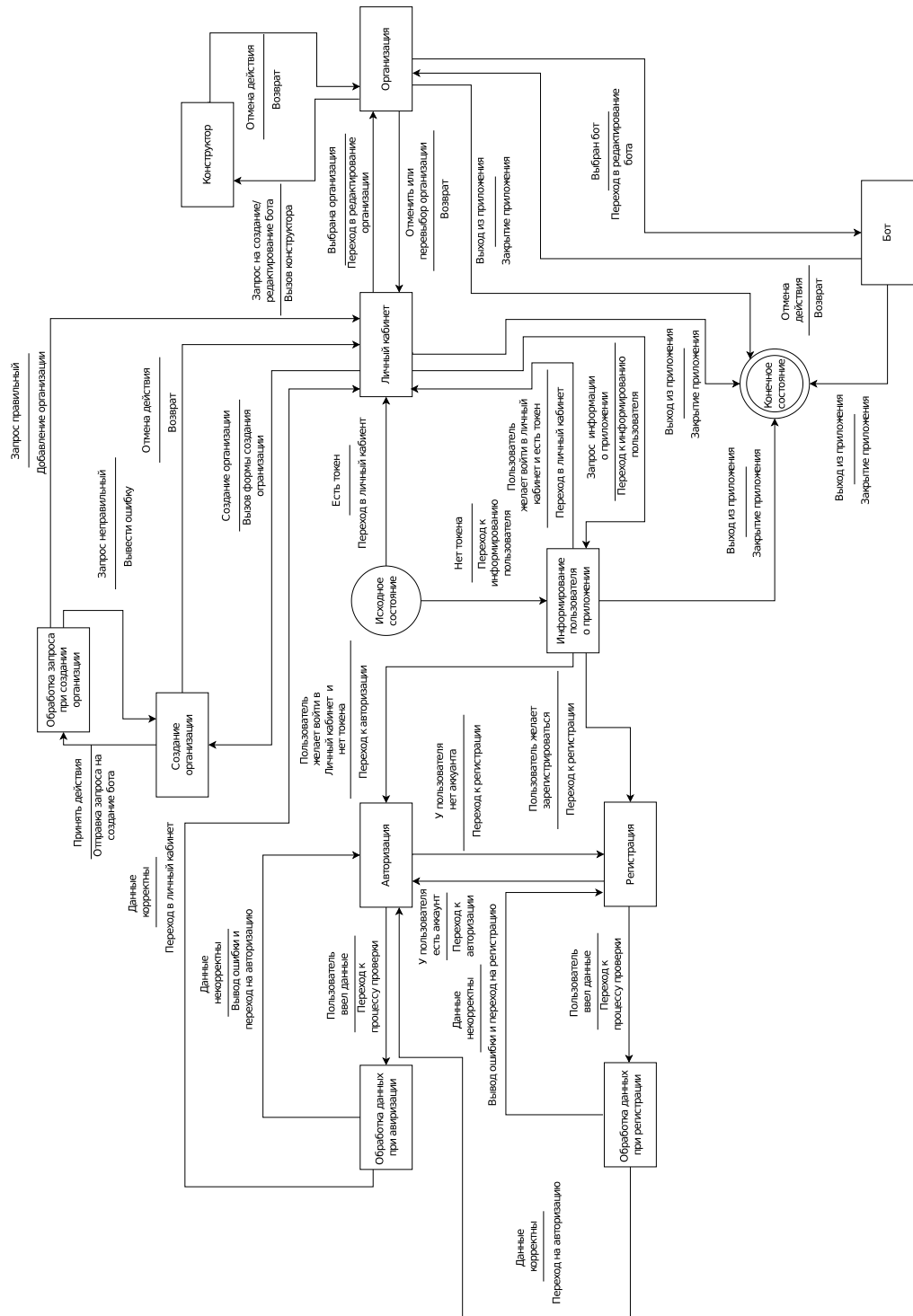
В качестве направления дальнейшего развития можно рассмотреть переход на использование микросервисной архитектуры.

					ТПЖА.090301.331 ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

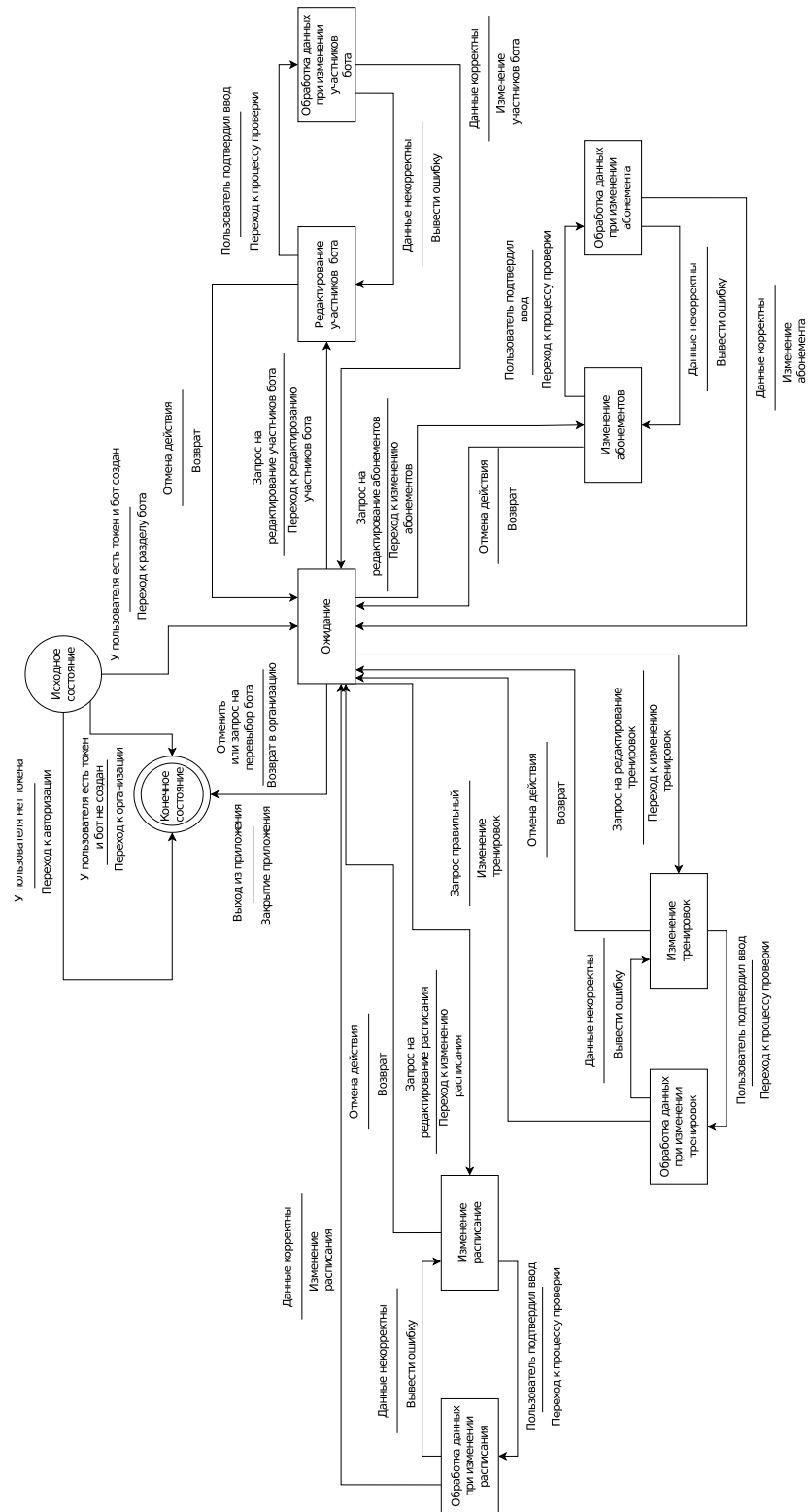


# Приложение А (обязательное)

## Диаграмма состояний для административной панели

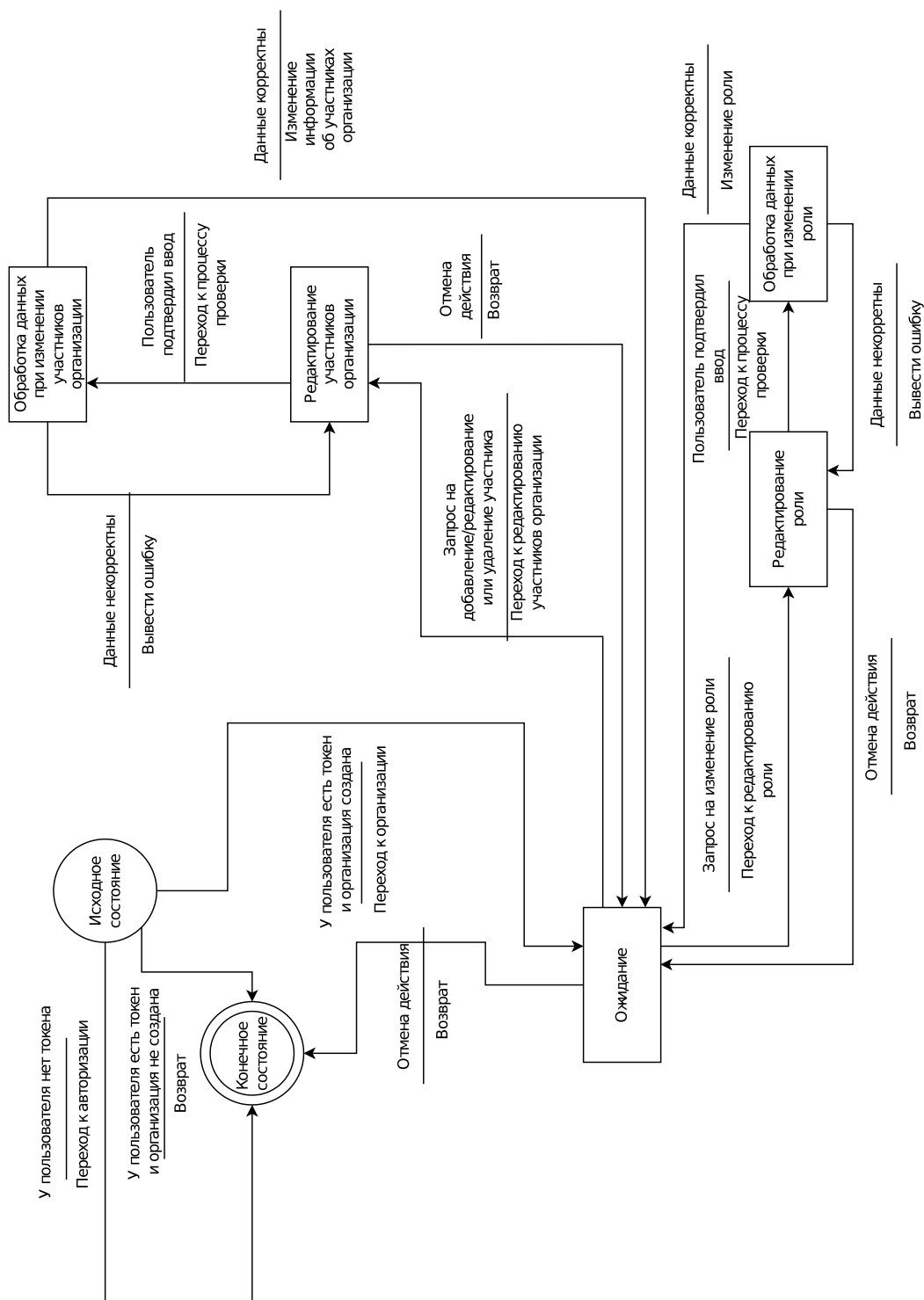


### Диаграмма состояний для модуля «бот» административной панели



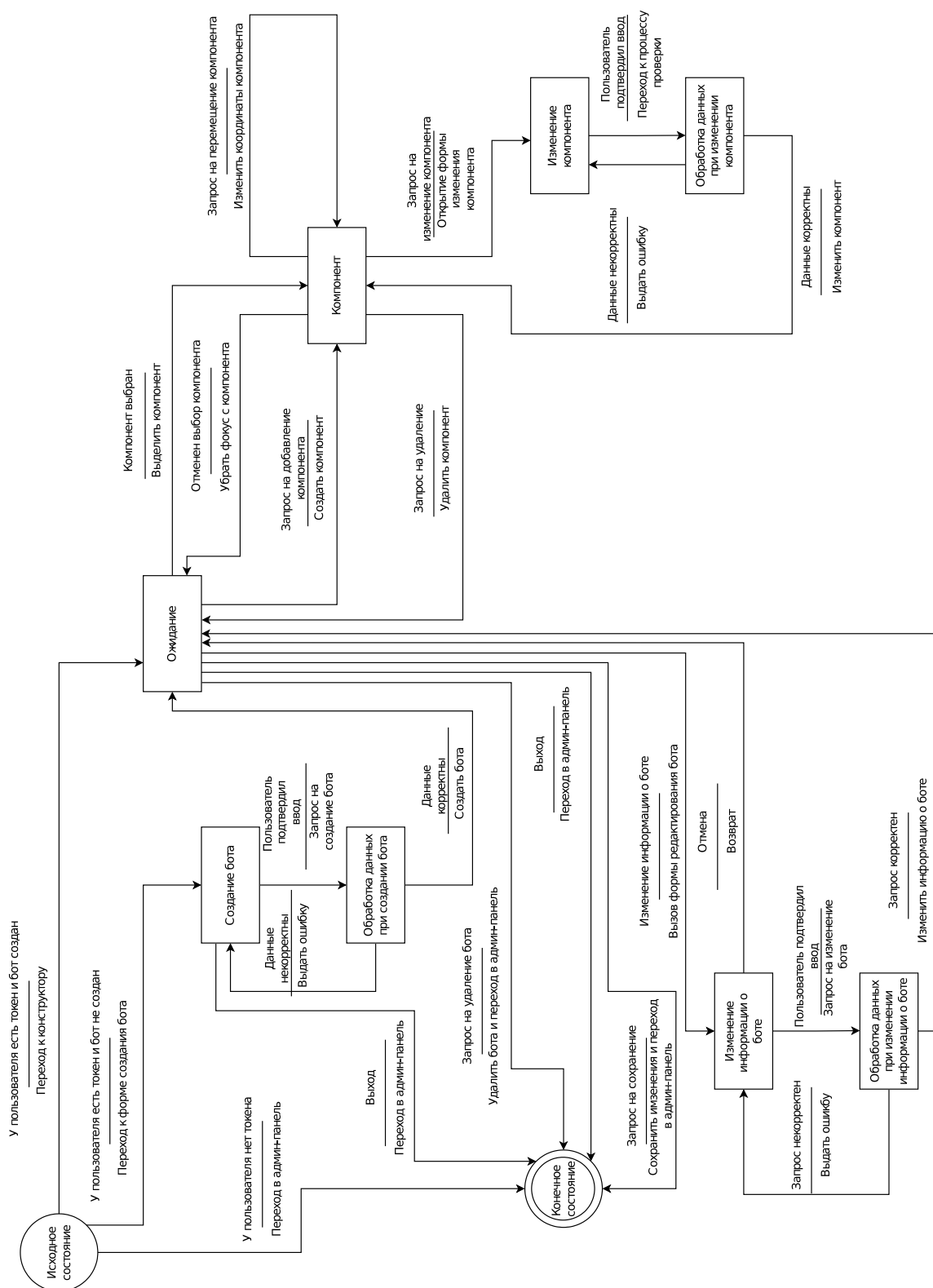
(обязательное)

Диаграмма состояний для модуля «организация» административной панели



# Приложение Г (обязательное)

## Диаграмма состояний для конструктора



Приложение Д

(Обязательное)

Перечень сокращений

БД – база данных

СУБД – система управления базами данных

JSON – JavaScript Object Notation

ГСА - граф-схема алгоритма

ER- entity relationship

SQL - Structured Query Language

					ТПЖА.090301.331 ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

## Приложение Е

(справочное)

### Библиографический список

1. HTTP – Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/HTTP>.
2. JSON – Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/JSON>.
3. Telegram Bot API – Википедия [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots/api>.
4. 15 сервисов для простого создания ботов в Telegram [Электронный ресурс]. – Режим доступа: <https://vc.ru/services/89824-15-servisov-dlya-prostogo-sozdaniya-botov-v-telegram>.

					ТПЖА.090301.331 ПЗ	Лист
						54
Изм.	Лист	№ докум.	Подпись	Дата		