

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**  
**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

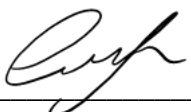
СОГЛАСОВАНО

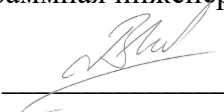
УТВЕРЖДАЮ

Приглашенный преподаватель  
департамента программной инженерии  
факультета компьютерных наук

Академический руководитель  
образовательной программы

«Программная инженерия», канд. техн. наук

 А. Н. Степанов

 В. В. Шилов

«3» мая 2022 г.

«03» мая 2022 г.

**СЕРВЕРНАЯ ЧАСТЬ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПОМОЩИ В  
ЗАПОМИНАНИИ ТЕОРИТИЧЕСКОЙ ЧАСТИ ДИСЦИПЛИН**


Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.10-01 ПЗ 01-1-ЛУ

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Исполнитель:

 / И. Н. Дедов /  
«2» мая 2022 г.

УТВЕРЖДЕН

RU.17701729.05.10-01 ПЗ 01-1-ЛУ

**СЕРВЕРНАЯ ЧАСТЬ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПОМОЩИ В  
ЗАПОМИНАНИИ ТЕОРИТИЧЕСКОЙ ЧАСТИ ДИСЦИПЛИН**

**Пояснительная записка**

**RU.17701729.05.10-01 ПЗ 01-1**

**Листов 52**

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**АННОТАЦИЯ**

Данный программный документ представляет собой пояснительную записку к программному проекту «Серверная часть мобильного приложения для помощи в запоминании теоретической части дисциплин», предназначенному для реализации внутренней логики мобильного приложения «Steady», которое будет использоваться студентами для помощи в запоминании теоретической части дисциплин. Серверная часть будет использоваться для управления безопасностью приложения, в том числе сохранности данных пользователей, обеспечения взаимодействия с базой данных, и, наконец, предоставления своего API для использования клиентской частью мобильного приложения.

Разделе «Введение» содержит наименование программы, условное обозначение темы разработки, документ, на основании которого ведётся разработка, а также организация, утвердившая данный документ.

В разделе «Назначение и область применения» указано функциональное и эксплуатационное назначение программы и краткая характеристика области её применения.

В разделе «Технические характеристики» содержатся следующие подразделы: постановка задачи на разработку программы, описание функционирования программы, описание алгоритма работы программы, описание и обоснование выбора метода организации входных и выходных данных, описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Программный документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102-77 Стадии разработки [2];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104-78 Основные надписи [4];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78[8], ГОСТ 19.604-78[9].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**СОДЕРЖАНИЕ**

ГЛОССАРИЙ .....	6
1. ВВЕДЕНИЕ .....	8
1.1. Наименование программы.....	8
1.2. Документы, на основании которых ведется разработка.....	8
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....	9
2.1. Функциональное назначение.....	9
2.2. Эксплуатационное назначение.....	9
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....	11
3.1. Постановка задачи на разработку .....	11
3.2. Общий принцип работы программы .....	11
3.3. Подробное описание обработки запроса .....	12
3.4. Разработка схемы базы данных.....	13
3.5. Описание и обоснование выбора метода организации входных данных .....	13
3.6. Описание и обоснование выбора метода организации выходных данных .....	14
3.7. Описание и обоснование выбора состава технических и программных средств... 14	
3.7.1. Состав технических и программных средств .....	14
3.7.2. Обоснование выбора технических и программных средств .....	15
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ .....	16
4.1. Предполагаемая потребность .....	16
4.2. Ориентировочная экономическая ценность .....	16
4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами .....	16
5. СПИСОК ИСТОЧНИКОВ.....	17
ПРИЛОЖЕНИЕ 1.....	18

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 2.....	19
ПРИЛОЖЕНИЕ 3.....	22
ПРИЛОЖЕНИЕ 4.....	25
ПРИЛОЖЕНИЕ 5.....	26
ПРИЛОЖЕНИЕ 6.....	30
ПРИЛОЖЕНИЕ 7.....	31
ПРИЛОЖЕНИЕ 8.....	40
ПРИЛОЖЕНИЕ 9.....	42
ПРИЛОЖЕНИЕ 10.....	44
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	52

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ГЛОССАРИЙ

- 1) *Пользователь* – человек, который использует приложение «Stoady». В контексте приложения «Stoady» существует два типа пользователей: «студенты» и «администраторы».
- 2) *Студент* (также, *участник, участник команды*) – пользователь, цель которого – лучше разобраться в теоретическом материале дисциплин.
- 3) *Администратор* – пользователь, имеющий доступ к редактированию материалов, содержащихся в приложении «Stoady», а именно:
  - добавление, удаление и редактирование прав участников команд;
  - добавление, удаление и редактирование предметов, тем и вопросов, содержащихся в команде.
- 4) *Команда* (также, *группа*) – группа людей, объединенных общим набором предметов.
- 5) *Предмет* (также, *дисциплина*) – верхний уровень иерархии теоретических материалов в приложении, который может содержать внутри себя одну или несколько тем.
- 6) *Тема* – раздел предмета, содержащий необходимую для изучения теорию и соответствующие ей вопросы.
- 7) *Вопрос* – элемент нижнего уровня иерархии теории в приложении. Вопрос содержит в себе сам текст вопроса, а также ответ на него.
- 8) *СУБД* – система управления базами данных.
- 9) *API* (Application Programming Interface) – описание способов (набор классов, методов и т. п.), которыми одна компьютерная программа (в данном случае, клиентская часть приложения) может взаимодействовать с другой (в данном случае, с сервером).
- 10) *REST API* (Representational State Transfer) – способ взаимодействия приложений с сервером с помощью протокола HTTP, соответствующий принципам REST, например, отделение клиента от сервера, отсутствие записи состояния клиента и многоуровневость системы.
- 11) *JSON* (JavaScript Object Notation) – текстовый формат обмена данными, основанный на языке программирования JavaScript. Но при этом формат независим от JavaScript и может использоваться в любом языке программирования.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 12) *Контроллер* – объект в API, предназначенный для обработки запросов. На вход он получает запрос, затем обрабатывает его одним из методов (хэндлеров) и выдает ответ пользователю.
- 13) *Хэндлер* – метод, обрабатывающий определенный запрос.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



**1. ВВЕДЕНИЕ****1.1. Наименование программы**

Наименование программы – «Серверная часть мобильного приложения для помощи в запоминании теоретической части дисциплин».

Наименование программы на английском языке – «Server Side of the Mobile Application for Assisting in the Memorization of Theoretical Parts of Disciplines».

**1.2. Документы, на основании которых ведется разработка**

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем программы тема курсового проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1. Функциональное назначение

Мобильное приложение «Stoady» представляет собой сервис для помощи учащимся вузов и других учебных заведений в подготовке к различным элементам контроля. Сервис реализует в себе функциональность, позволяющую пользователям выбирать нужный предмет, а также тему для изучения внутри этого предмета.

Каждый пользователь может создавать команды и приглашать в них других пользователей, чтобы предоставить им доступ к материалам, содержащимся в их команде. Это позволит учащимся адаптировать приложение под образовательную программу их учебного заведения, что позволит лучше ее усвоить: у учащихся не будет необходимости в поиске и фильтрации информации из сети Интернет – вся теория, нужная им, будет сосредоточена в одной команде.

Серверная часть мобильного приложения «Stoady» предназначена для реализации внутренней логики приложения, управления безопасностью приложения, в том числе сохранности данных пользователей приложения, обеспечения взаимодействия с базой данных, и, наконец, предоставления своего API для использования клиентской частью мобильного приложения.

### 2.2. Эксплуатационное назначение

Мобильное приложение «Stoady» будет применяться в сфере образования и обучения. Инициативные студенты (а также преподаватели учебных заведений) могут в доступном формате давать необходимую теорию своим сокурсникам (или же подопечным студентам), которая будет агрегирована в одном приложении, что избавит студентов от необходимости хранить все ресурсы и ссылки на них в каких-либо хранилищах (например, Google Drive или Яндекс.Диск). Это улучшит пользовательский опыт студентов при подготовке к элементам контроля, а наличие функциональности для отслеживания статистики при изучении теории будет способствовать более эффективному обучению: каждый студент будет понимать, какие темы ему стоит повторить, на каких вопросах следует заострить внимание.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Целевая аудитория приложения – студенты, которые хотят лучше изучить теоретическую часть дисциплин или помочь своим сокурсникам сделать это благодаря добавлению необходимой теории в свою команду.

Серверная часть приложения «Stoady» предоставляет API, который будет использоваться клиентской частью приложения. Клиент будет получать доступ к необходимым ему данным из базы данных посредством вызова конечных точек API.

### 2.3. Краткая характеристика области применения

Целевая аудитория мобильного приложения «Stoady» – студенты, которым требуется помощь в запоминании теоретической части учебных дисциплин (к примеру, формулировок, теорем, билетов к экзамену и т. п.). Студенты смогут выбирать необходимый для изучения предмет, проходить тесты по темам, изученным в курсе данной дисциплины, а также отслеживать свою статистику и выявлять слабые места в изученной по выбранному предмету теории.

Помимо этого, пользователям будет доступна возможность создавать команды, чтобы готовиться к экзаменам и другим элементам контроля совместно со своими друзьями или сокурсниками. Приложение также могут использовать преподаватели, публикуя теорию и сопутствующие вопросы по дисциплинам для студентов своих групп.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

### 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

#### 3.1. Постановка задачи на разработку

Разрабатываемая программа должна соответствовать требованиям, описанным в настоящем Техническом задании «Серверная часть мобильного приложения для помощи в запоминании теоретической части дисциплин».

Полученная программа предоставляет REST API для взаимодействия с клиентской частью приложения. API предоставляет конечные точки, вызывая которые пользователь имеет возможность получать обработанные данные из базы данных или добавлять в нее новые данные. Таким образом, API предоставляет пользователю (клиентской части приложения) выполнять следующие действия:

- 1) авторизацию и регистрацию пользователей;
- 2) редактирование профиля пользователей;
- 3) создание и управление командами и учебными материалами внутри команд, а именно:
  - a. создание, редактирование и удаление предметов внутри команд;
  - b. создание, редактирование и удаление тем внутри предметов;
  - c. создание, редактирование и удаление вопросов внутри тем;
  - d. сохранять вопросы и удалять вопросы из сохраненных;
- 4) прохождение тестирования по темам.

Для успешного предоставления пользователю данной функциональности API и база данных PostgreSQL были развернуты на удаленном хостинге Heroku [10].

#### 3.2. Общий принцип работы программы

Взаимодействие пользователя с базой данных происходит через API следующим образом (рисунок 1): пользователь обращается к конечным точкам API, которые обрабатывают запрос, обращаются к нужным таблицам в базе данных и возвращают пользователю требуемый ответ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

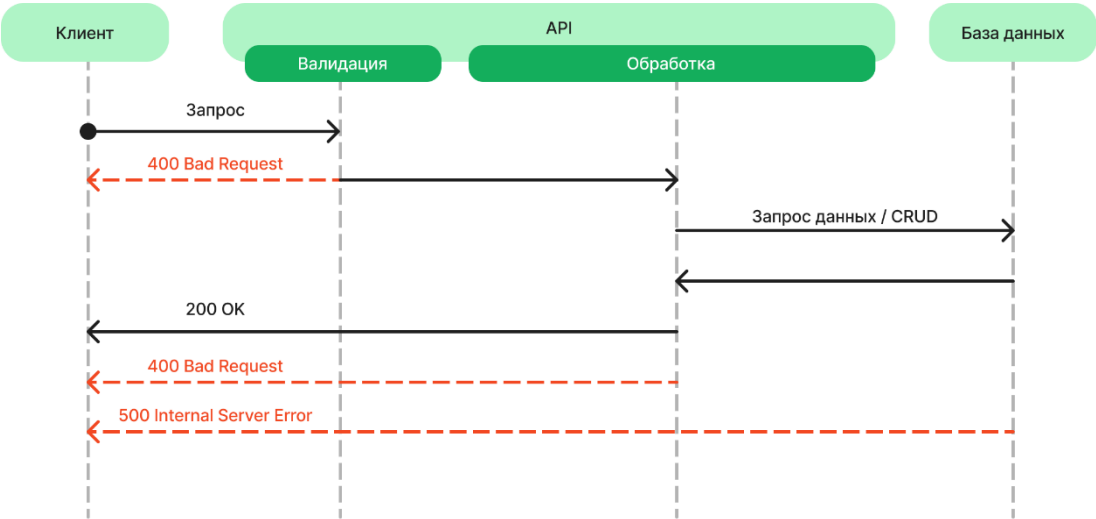


Рисунок 1. Общая схема взаимодействия клиента с API и базой данных.

Запросы, посылаемые клиентом, перенаправляются соответствующему контроллеру по первой части URL, а затем контроллер определяет соответствующий хэндлер, который будет этот запрос обрабатывать.

Далее хэндлер вызывает свои внутренние методы, которые ходят в нужные таблицы базы данных (выбираются в зависимости от запроса) и, либо создают новые записи в ней, либо достают из нее необходимую информацию, обрабатывают ее и возвращают пользователю ответ на его запрос.

Подробное описание контроллеров можно найти в Приложении 1, а подробное описание хэндлеров – в Приложении 2.

Чтобы программа соблюдала принципы SOLID, была применена Dependency Injection, и вместо классов с реализацией в методы передаются интерфейсы.

3.3. Подробное описание обработки запроса

Когда в серверную часть приложения приходит запрос, по его адресу, он передается в соответствующий контроллер, который определяет нужный хэндлер, которому перенаправить этот запрос.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

После этого, внутри каждого хэндлера запрос (класс, название которого оканчивается на Request) преобразовывается в команду (идентичный класс, в конце названия присутствует Command). Затем этот класс, прежде чем передаться в метод Handle и начать исполнение, проходит валидацию с помощью Nuget-библиотеки FluentValidation [13], и если она не была пройдена, возвращает ответ с сообщением об ошибке и кодом 400 Bad Request.

Если же валидация была успешно пройдена, запрос передается в метод Handle и исполняет необходимый код, посылая запросы в базу данных.

При возникновении исключений класс ExceptionHandlingMiddleware обрабатывает его и возвращает ответ с кодом 400 BadRequest (или 500 Internal Server Error в случае ошибки самой базы данных, например, несоответствия внешнего ключа), а также объект с информацией о произошедшем исключении: код ошибки, ресурс, в котором брошено исключение, сообщение исключения и трейс стека вызовов. Это необходимо для правильной обработки таких ситуаций клиентской частью приложения.

В случае, когда исключений в ходе работы метода не возникло, клиенту возвращается ответ с кодом 200 Success и, при необходимости, JSON-объектом, который получается сериализацией класса с именем, оканчивающимся на Response.

### 3.4. Разработка схемы базы данных

Для хранения данных о пользователях, командах, предметах внутри этих команд, а также тем и вопросов внутри предметов была выбрана реляционная СУБД PostgreSQL. Обращения к базе данных из кода API производятся с помощью фреймворка Dapper [11] и запросов, написанных на диалекте PostgreSQL.

Схема используемой базы данных находится в Приложении 3. Преобразование данных, полученных из таблиц базы, в классы, написанные на языке C# (см. Приложение 4 и 5), происходит с помощью ORM Dapper [11].

### 3.5. Описание и обоснование выбора метода организации входных данных

API принимает на вход HTTP-запросы типов GET, PUT, POST или DELETE, в теле которых могут присылаться JSON-объекты с различными полями, которые затем

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

преобразовываются в классы на языке C#. Описание этих классов и их свойств можно найти в Приложениях 6 и 7.

Для взаимодействия клиентской части с серверной частью приложения было решено использовать REST API, так как оно позволяет осуществлять такое взаимодействие, а наличие библиотеки Swagger [12] облегчает проверку работоспособности и корректности ответов конечных точек. Swagger также предоставляет удобный графический интерфейс для работы с конечными точками API.

Примеры запросов к API приведены в Приложении 8.

### **3.6. Описание и обоснование выбора метода организации выходных данных**

Общие требования к выходным данным включали себя возможность их обработки клиентской частью приложения.

Предоставляемое серверной частью REST API позволяет возвращать данные в универсальном формате JSON, которые могут быть обработаны клиентской частью.

### **3.7. Описание и обоснование выбора состава технических и программных средств**

#### **3.7.1. Состав технических и программных средств**

Для работоспособности всей серверной части приложения (API и база данных) необходим следующий состав технических средств:

- 1) не менее 10 Гб свободной памяти;
- 2) не менее 4 Гб оперативной памяти;
- 3) доступ к сети Интернет.

Для работоспособности серверной части приложения (API и база данных) необходим следующий состав программных средств:

- 1) операционная система Linux;
- 2) .NET версии 6.0;
- 3) PostgreSQL версии 13 или выше;
- 4) Docker.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**3.7.2. Обоснование выбора технических и программных средств**

Определенные выше требования позволяют предоставлять доступ к API сразу многим клиентам, а большое количество выделяемой памяти позволит хранить многое количество записей в базе данных, так как пользователи могут создавать множество вопросов и вступать в большое число команд.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



#### **4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

##### **4.1. Предполагаемая потребность**

Серверная часть мобильного приложения «Stoady» будет использоваться клиентской частью и являться посредником между конечным пользователем и базой данных.

##### **4.2. Ориентировочная экономическая ценность**

Расчет ориентировочной экономической ценности серверной части мобильного приложения «Stoady» не предусмотрен.

Расчет экономической ценности мобильного приложения «Stoady» в целом приведен в настоящем Техническом задании «Мобильное приложение для помощи в запоминании теоретической части дисциплин».

##### **4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**

Расчет экономических преимуществ серверной части приложения не предусмотрен.

Экономические преимущества приложения «Stoady» в целом можно найти в настоящем Техническом задании «Мобильное приложение для помощи в запоминании теоретической части дисциплин».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**5. СПИСОК ИСТОЧНИКОВ**

- 1) ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 10) Платформа Heroku [Электронный ресурс] / Режим доступа: [www.heroku.com](http://www.heroku.com), свободный.
- 11) Обучение фреймворку Dapper [Электронный ресурс] / Режим доступа: <https://www.learndapper.com>, свободный.
- 12) Библиотека Swagger [Электронный ресурс] / Режим доступа: <https://swagger.io>, свободный.
- 13) Документация FluentValidation [Электронный ресурс] / Режим доступа: <https://fluentvalidation.net>, свободный.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 1****ОПИСАНИЕ И НАЗНАЧЕНИЕ КОНТРОЛЛЕРОВ***Таблица 1.1 – Список контроллеров*

<b>Класс</b>	<b>Назначение</b>
AuthenticationController	Отвечает за авторизацию пользователей приложения
QuestionController	Отвечает за создание, редактирование, удаление и получение информации по вопросам
StatisticsController	Отвечает за получение статистики прохождения тем пользователями
SubjectController	Отвечает за создание, редактирование, удаление и получение информации по дисциплинам
TeamController	Отвечает за создание, редактирование, удаление и получение информации о командах пользователей
TestingController	Создает или редактирует информацию по пройденным пользователем тестам
TopicController	Отвечает за создание, редактирование, удаление и получение информации о темах
UserController	Отвечает за создание и удаление пользователей, а также за добавление пользователей в команды, их удаление и изменение ролей внутри команд

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 2****ОПИСАНИЕ И НАЗНАЧЕНИЕ МЕТОДОВ КОНТРОЛЛЕРОВ***Таблица 2.1 – Список методов контроллера AuthenticationController*

Метод	Назначение
Authorize	По введенному адресу электронной почты и паролю возвращает данные о пользователе

*Таблица 2.2 – Список методов контроллера QuestionController*

Метод	Назначение
GetQuestions	По введенному идентификатору темы получает вопросы этой темы
GetSavedQuestions	По введенному идентификатору пользователя получает его сохраненные вопросы
AddQuestion	Создает новый вопрос
EditQuestion	Редактирует существующий вопрос
SaveQuestion	По идентификатору пользователя и вопроса добавляет вопрос в сохраненные
RemoveQuestion	Удаляет вопрос с указанным идентификатором
RemoveSavedQuestion	Убирает вопрос из сохраненных у выбранного пользователя

*Таблица 2.3 – Список методов контроллера StatisticsController*

Метод	Назначение
GetUserStatistics	Возвращает статистику по пройденным темам у указанного пользователя

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 2.4 – Список методов контроллера *SubjectController*

Метод	Назначение
GetSubjectInfo	По введенному идентификатору предмета получает информацию о нем
AddSubject	Создает новую дисциплину
EditSubject	Редактирует информацию о дисциплине
RemoveSubject	Удаляет дисциплину

Таблица 2.5 – Список методов контроллера *TeamController*

Метод	Назначение
SelectTeam	По введенному идентификатору команды и пользователя возвращает роль этого пользователя в команде
GetTeamInfo	По введенному идентификатору команды получает информацию о ней
GetTeamMembers	По введенному идентификатору команды получает ее участников
ChangeMemberStatus	Изменяет роль выбранного участника в команде
CreateTeam	Создает новую команду
EditTeam	Редактирует название и аватар команды
AddMember	Добавляет пользователя в команду
RemoveMember	Удаляет участника из команды

Таблица 2.6 – Список методов контроллера *TestingController*

Метод	Назначение
SaveTestResults	Сохраняет результаты тестирования по теме

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 2.7 – Список методов контроллера *TopicController*

Метод	Назначение
GetTopicInfo	По введенному идентификатору темы получает информацию о ней
AddTopic	Создает новую тему
EditTopic	Редактирует информацию о теме
RemoveTopic	Удаляет тему

Таблица 2.8 – Список методов контроллера *UserController*

Метод	Назначение
GetUserTeams	По переданному идентификатору пользователя получает список команд, в которых он состоит
RegisterUser	Регистрирует нового пользователя приложения
UpdateUserAvatar	Обновляет аватар пользователя

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 3

## СХЕМА БАЗЫ ДАННЫХ

```
CREATE TABLE IF NOT EXISTS Questions (
    id          BIGSERIAL    PRIMARY KEY,
    questionText VARCHAR(100) NOT NULL,
    answerText   VARCHAR(100) NOT NULL,
    topicId      BIGINT NOT NULL REFERENCES Topics ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS Roles (
    id   BIGSERIAL    PRIMARY KEY,
    name VARCHAR(30) NOT NULL UNIQUE
);
```

```
CREATE TABLE IF NOT EXISTS Teams (
    id          BIGSERIAL    PRIMARY KEY,
    name        VARCHAR(30) NOT NULL,
    avatar      TEXT         NOT NULL,
    creatorId   BIGINT      NOT NULL REFERENCES Users ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS Subjects (
    id          BIGSERIAL    PRIMARY KEY,
    title        VARCHAR(50) NOT NULL,
    description   VARCHAR(250),
    teamId       BIGINT      NOT NULL REFERENCES Teams ON DELETE CASCADE
);
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```
CREATE TABLE IF NOT EXISTS Topics (
  id          BIGSERIAL PRIMARY KEY,
  title       VARCHAR(50),
  description VARCHAR(250),
  subjectId   BIGINT NOT NULL REFERENCES Subjects ON DELETE CASCADE
);
```

```
CREATE TABLE IF NOT EXISTS Users (
  id          BIGSERIAL PRIMARY KEY,
  username    VARCHAR(30) NOT NULL,
  email       VARCHAR(100) NOT NULL UNIQUE,
  password    VARCHAR(64) NOT NULL,
  avatarId    INTEGER NOT NULL,
  salt        VARCHAR(64) NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS Statistics (
  id          BIGSERIAL PRIMARY KEY,
  userId      BIGINT NOT NULL REFERENCES Users ON DELETE CASCADE,
  topicId     BIGINT NOT NULL REFERENCES Topics ON DELETE CASCADE,
  result      INTEGER DEFAULT 0 NOT NULL,
  UNIQUE (userId, topicId)
);
```

```
CREATE TABLE IF NOT EXISTS TeamUsers (
  id          BIGSERIAL PRIMARY KEY,
  teamId      BIGINT NOT NULL REFERENCES Teams ON DELETE CASCADE,
  userId      BIGINT NOT NULL REFERENCES Users ON DELETE CASCADE,
  roleId      BIGINT NOT NULL REFERENCES Roles ON DELETE CASCADE,
  UNIQUE (teamId, userId)
);
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



```
CREATE TABLE IF NOT EXISTS UserQuestions (  
    id          BIGSERIAL PRIMARY KEY,  
    userId      BIGINT NOT NULL REFERENCES Users ON DELETE CASCADE,  
    questionId BIGINT NOT NULL REFERENCES Questions ON DELETE CASCADE,  
    UNIQUE (userId, questionId)  
);
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 4

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ ДЛЯ  
ВЗАИМОДЕЙСТВИЯ С БАЗОЙ ДАННЫХ**

*Таблица 4.1 – Список классов, описывающих сущности, для взаимодействия с базой данных*

<b>Класс</b>	<b>Назначение</b>
QuestionDao	Содержит информацию о вопросе: идентификатор, текст вопроса, текст ответа и идентификатор темы, которой принадлежит этот вопрос
RoleDao	Содержит информацию о роли: идентификатор и название
StatisticsDao	Содержит информацию о статистике пользователя: идентификатор, идентификатор пользователя, идентификатор темы, ее название и результат выполнения (в процентах, округленное до целого числа)
SubjectDao	Содержит информацию о дисциплине: ее идентификатор, название, описание и идентификатор команды, которой принадлежит дисциплина
TeamDao	Содержит информацию о команде: ее идентификатор, название, идентификатор аватара и идентификатор пользователя, который является создателем команды
TopicDao	Содержит информацию о теме: ее идентификатор, название, описание и идентификатор дисциплины, которой принадлежит тема
UserDao	Содержит сокращенную информацию о пользователе: идентификатор, имя, адрес электронной почты и идентификатор аватара пользователя
UserWithPasswordDao	Содержит расширенную информацию о пользователе: помимо идентификатора, имени, адреса электронной почты и идентификатора аватара пользователя, в классе присутствует захешированный пароль и персональную «соль» к нему

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 5

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ КЛАССОВ ДЛЯ ВЗАИМОДЕЙСТВИЯ С БАЗОЙ ДАННЫХ

Таблица 5.1 – Список полей, методов и свойств класса *QuestionDao*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор вопроса в таблице Users
QuestionText	public	string	Текст вопроса
AnswerText	public	string	Текст ответа
TopicId	public	long	Идентификатор темы в таблице Topics, которой принадлежит вопрос

Таблица 5.2 – Список полей, методов и свойств класса *RoleDao*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор вопроса в таблице Roles
Name	public	string	Название роли

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.3 – Список полей, методов и свойств класса *StatisticsDao*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор статистики в таблице Statistics
UserId	public	long	Идентификатор пользователя в таблице Users
TopicId	public	long	Идентификатор темы в таблице Topics
TopicName	public	string	Название темы
Result	public	int	Результат прохождения, в процентах, округленных до целого числа

Таблица 5.4 – Список полей, методов и свойств класса *SubjectDao*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор предмета в таблице Subjects
Title	public	string	Название предмета
Description	public	string	Описание предмета
TeamId	public	long	Идентификатор команды в таблице Teams, которой принадлежит предмет

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.5 – Список полей, методов и свойств класса TeamDao

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор команды в таблице Teams
Name	public	string	Название команды
Avatar	public	int	Идентификатор аватара – число от 0 до 14 включительно
CreatorId	public	long	Идентификатор пользователя в таблице Users, который создал команду

Таблица 5.6 – Список полей, методов и свойств класса TopicDao

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор темы в таблице Topics
Title	public	string	Название темы
Description	public	string	Описание темы
SubjectId	public	long	Идентификатор предмета в таблице Subjects, которому принадлежит тема

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 5.7 – Список полей, методов и свойств класса UserDao

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор пользователя в таблице Users
Username	public	string	Имя пользователя
Email	public	string	Адрес электронной почты пользователя
AvatarId	public	int	Идентификатор аватара пользователя

Таблица 5.8 – Список полей, методов и свойств класса UserWithPasswordDao

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор пользователя в таблице Users
Username	public	string	Имя пользователя
Email	public	string	Адрес электронной почты пользователя
AvatarId	public	int	Идентификатор аватара пользователя
Password	public	string	Захэшированный пароль пользователя
Salt	public	string	Персональная «соль» к паролю

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

**ПРИЛОЖЕНИЕ 6****ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ-ПАРАМЕТРОВ  
HTTP-ЗАПРОСОВ***Таблица 6.1 – Список классов, отвечающих за принятие запросов и отправку ответов на HTTP-запросы*

<b>Класс</b>	<b>Назначение</b>
AutorizationRequest	Запрос на авторизацию пользователя
AuthorizationResponse	Информация об авторизованном пользователе
AddQuestionRequest	Запрос на создание нового вопроса
EditQuestionRequest	Запрос на редактирование вопроса
GetQuestionsResponse	Информация о вопросах в теме
GetSavedQuestionResponse	Информация о сохраненных вопросах пользователя
GetUserStatisticsResponse	Информация о статистике пользователя по пройденным темам
AddSubjectRequest	Запрос на создание нового предмета
EditSubjectRequest	Запрос на редактирование предмета
GetSubjectInfoResponse	Информация о предмете
EditTeamRequest	Запрос на редактирование команды
GetTeamInfoResponse	Информация о команде
GetTeamMembersResponse	Информация об участниках команды
SelectTeamResponse	Информация о роли пользователя в команде
SaveTestResultsRequest	Запрос на сохранение результатов тестирования по теме
AddTopicRequest	Запрос на создание новой темы
EditTopicRequest	Запрос на редактирование темы
GetTopicInfoResponse	Информация о теме
GetUserTeamsResponse	Информация о командах, в которых состоит пользователь
RegisterUserRequest	Запрос на регистрацию нового пользователя
RegisterUserResponse	Информация о новом зарегистрированном пользователе

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 7

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ КЛАССОВ-ПАРАМЕТРОВ HTTP-ЗАПРОСОВ

Таблица 7.1 – Список полей, методов и свойств класса *AuthorizationRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Email	public	string	Адрес электронной почты пользователя
Password	public	string	Пароль пользователя

Таблица 7.2 – Список полей, методов и свойств класса *AuthorizationResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор пользователя из таблицы Users
Name	public	string	Имя пользователя
AvatarId	public	int	Идентификатор аватара пользователя (число от 0 до 14 включительно)

Таблица 7.3 – Список полей, методов и свойств класса *AddQuestionRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
TopicId	public	long	Идентификатор темы из таблицы Topics
QuestionText	public	string	Текст вопроса
AnswerText	public	string	Текст ответа

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



Таблица 7.4 – Список полей, методов и свойств класса *EditQuestionRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
QuestionText	public	string	Новый текст вопроса
AnswerText	public	string	Новый текст ответа

Таблица 7.5 – Список полей, методов и свойств класса *GetQuestionsResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Questions	public	List<QuestionInTopic>	Список вопросов в теме

Таблица 7.5.1 – Список полей, методов и свойств класса *QuestionInTopic*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор вопроса в таблице Questions
QuestionText	public	string	Текст вопроса
AnswerText	public	string	Текст ответа

Таблица 7.6 – Список полей, методов и свойств класса *GetSavedQuestionsResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
SavedQuestions	public	List<SavedQuestion>	Список сохраненных вопросов пользователя

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 7.6.1 – Список полей, методов и свойств класса *SavedQuestion*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор вопроса в таблице Questions
QuestionText	public	string	Текст вопроса
AnswerText	public	string	Текст ответа

Таблица 7.7 – Список полей, методов и свойств класса *GetUserStatisticsResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Results	public	List<TopicStatistics>	Результаты прохождения тестирования по темам

Таблица 7.7.1 – Список полей, методов и свойств класса *TopicStatistics*

Свойства			
Название	Модификатор доступа	Тип	Назначение
TopicId	public	long	Идентификатор темы в таблице Topics
TopicName	public	string	Название темы
Result	public	int	Результат прохождения темы пользователем

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 7.8 – Список полей, методов и свойств класса *AddSubjectRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
TeamId	public	long	Идентификатор команды в таблице Teams, в которую нужно добавить предмет
SubjectName	public	string	Название предмета
SubjectDescription	public	string	Описание предмета

Таблица 7.9 – Список полей, методов и свойств класса *EditSubjectRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
SubjectName	public	string	Новое название предмета
SubjectDescription	public	string	Новое описание предмета

Таблица 7.10 – Список полей, методов и свойств класса *GetSubjectInfoResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Name	public	string	Название предмета
Description	public	string	Описание предмета
Topics	public	List<TopicInSubject>	Список тем в предмете

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 7.10.1 – Список полей, методов и свойств класса TopicInSubject

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор темы в таблице Topics
Name	public	string	Название темы

Таблица 7.11 – Список полей, методов и свойств класса EditTeamRequest

Свойства			
Название	Модификатор доступа	Тип	Назначение
TeamName	public	string	Новое название команды
TeamAvatar	public	string	Новая аватарка команды

Таблица 7.12 – Список полей, методов и свойств класса GetTeamInfoResponse

Свойства			
Название	Модификатор доступа	Тип	Назначение
Name	public	string	Название команды
Picture	public	string	Аватарка команды
Subjects	public	List<SubjectInTeam>	Список предметов в команде

Таблица 7.12.1 – Список полей, методов и свойств класса SubjectInTeam

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	string	Идентификатор предмета в таблице Subjects

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Name	public	string	Название предмета
Description	public	string	Описание предмета

Таблица 7.13 – Список полей, методов и свойств класса *GetTeamMembersResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Members	public	List<MemberInTeam>	Список участников команды

Таблица 7.13.1 – Список полей, методов и свойств класса *MemberInTeam*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор пользователя в таблице Users
Username	public	string	Имя пользователя
Email	public	string	Адрес электронной почты пользователя
Role	public	Role	Роль пользователя в команде

Таблица 7.14 – Список полей, методов и свойств класса *SelectTeamResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Role	public	Role	Роль пользователя в команде

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 7.15 – Список полей, методов и свойств класса *SaveTestResultsRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
UserId	public	long	Идентификатор пользователя в таблице Users
TopicId	public	long	Идентификатор темы в таблице Topics
Result	public	int	Результат прохождения темы

Таблица 7.16 – Список полей, методов и свойств класса *AddTopicRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
SubjectId	public	long	Идентификатор предмета в таблице Subjects, в который нужно добавить тему
TopicName	public	string	Название темы
TopicDescription	public	string	Описание темы

Таблица 7.17 – Список полей, методов и свойств класса *EditTopicRequest*

Свойства			
Название	Модификатор доступа	Тип	Назначение
TopicName	public	string	Новое название темы
TopicDescription	public	string	Новое описание темы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 7.18 – Список полей, методов и свойств класса *GetTopicInfoResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Name	public	string	Название темы
Description	public	string	Описание темы
Questions	public	List<QuestionInTopicInfo>	Список вопросов в теме

Таблица 7.18.1 – Список полей, методов и свойств класса *QuestionInTopicInfo*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Id	public	long	Идентификатор вопроса в таблице Questions

Таблица 7.19 – Список полей, методов и свойств класса *GetUserTeamsResponse*

Свойства			
Название	Модификатор доступа	Тип	Назначение
Teams	public	List<UserTeam>	Список пользователей в команде

Таблица 7.19.1 – Список полей, методов и свойств класса *UserTeam*

Свойства			
Название	Модификатор доступа	Тип	Назначение
TeamId	public	long	Идентификатор команды в таблице Teams, в которой состоит пользователь
Role	public	Role	Роль пользователя в команде

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

TeamName	public	string	Название команды
TeamAvatar	public	string	Аватар команды

Таблица 7.20 – Список полей, методов и свойств класса RegisterUserRequest

Свойства			
Название	Модификатор доступа	Тип	Назначение
Username	public	string	Имя нового пользователя
Email	public	string	Адрес электронной почты нового пользователя
Password	public	string	Пароль нового пользователя
AvatarId	public	int	Идентификатор аватара нового пользователя

Таблица 7.21 – Список полей, методов и свойств класса RegisterUserResponse

Свойства			
Название	Модификатор доступа	Тип	Назначение
UserId	public	long	Идентификатор нового пользователя в таблице Users

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



## ПРИЛОЖЕНИЕ 8

## ПРИМЕРЫ CURL-ЗАПРОСОВ К API

## 1) Получение информации о предмете

- Запрос:

```
curl -X 'GET' \
'https://stoady.herokuapp.com/subjects/1' \
-H 'accept: text/plain'
```

- Ответ:

```
{
  "name": "Algebra",
  "description": "\nThis is about algebra.",
  "topics": [
    {
      "id": 1,
      "name": "Matrices"
    },
    {
      "id": 2,
      "name": "Vectors"
    }
  ]
}
```

## 2) Регистрация нового пользователя

- Запрос:

```
curl -X 'POST' \
'https://stoady.herokuapp.com/users/register' \
-H 'accept: text/plain' \
-H 'Content-Type: application/json' \
-d '{
  "username": "Ivan Dedov",
  "email": "ivan.dedov@mail.com",
  "password": "VerySecretPassword",
  "avatarId": 12
}'
```

- Ответ:

```
{
  "userId": 27
}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 3) Обновление аватара пользователя

- Запрос:

```
curl -X 'PUT' \
'https://stoady.herokuapp.com/users/27/avatar/set?avatarId=10' \
-H 'accept: */*'
```

- Ответ:

```
200 Success
```

## 4) Удаление темы

- Запрос:

```
curl -X 'DELETE' \
'https://stoady.herokuapp.com/topics/remove/27' \
-H 'accept: */*'
```

- Ответ:

```
200 Success
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 9

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ДРУГИХ КЛАССОВ

Таблица 9.1 – Список других классов, требуемых для нормального функционирования программы

Класс	Назначение
Role	Перечисление – роль пользователя в команде: «Creator», «Admin» или «Member»
ExceptionHandlerMiddleware	Позволяет универсально обрабатывать ошибки и возвращать объект с информацией о возникнувшем исключении
ValidationBehavior	Содержит инструкции, позволяющие валидировать приходящие в серверную часть запросы
PasswordValidatorService	Содержит функциональность, позволяющую хэшировать строки, а также валидировать правильность пароля
RightsValidatorService	Содержит функциональность, позволяющую проверять наличие у пользователя прав на совершение различных действий
QuestionRepository	Репозиторий, содержащий методы, позволяющие обращаться к таблице Questions базы данных и получать требуемые объекты
RoleRepository	Репозиторий, содержащий методы, позволяющие обращаться к таблице Roles базы данных и получать требуемые объекты
StatisticsRepository	Репозиторий, содержащий методы, позволяющие обращаться к таблице Statistics базы данных и получать требуемые объекты
SubjectRepository	Репозиторий, содержащий методы, позволяющие обращаться к таблице Subjects базы данных и получать требуемые объекты

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

*Продолжение таблицы 9.1 – Список других классов, требуемых для нормального функционирования программы*

TeamRepository	Репозиторий, содержащий методы, позволяющие обращаться к таблице Teams базы данных и получать требуемые объекты
TopicRepository	Репозиторий, содержащий методы, позволяющие обращаться к таблице Topics базы данных и получать требуемые объекты
UserRepository	Репозиторий, содержащий методы, позволяющие обращаться к таблице Users базы данных и получать требуемые объекты
RepositorySettings	Содержит константы, определяющие общие для всех репозиториев свойства

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## ПРИЛОЖЕНИЕ 10

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ ДРУГИХ КЛАССОВ

Таблица 10.1 – Список полей, методов и свойств класса *ExceptionHandlerMiddleware*

Методы			
Название	Модификатор доступа	Тип	Назначение
InvokeAsync	public	Task	Асинхронно вызывает метод обработки исключения
HandleExceptionAsync	private	Task	Собирает информацию о произошедшем исключении и возвращает в качестве HTTP-ответа объект с информацией об ошибке
GetMessage	private	string	Возвращает сообщение о произошедшей ошибке: валидации или исключение в ходе выполнения кода
GetTitle	private	string	Возвращает название ресурса, в котором произошла ошибка
GetStatusCode	private	int	Возвращает код ошибки

Таблица 10.2 – Список полей, методов и свойств класса *ValidationBehavior*

Поля			
Название	Модификатор доступа	Тип	Назначение
_validators	private	IEnumerable< IValidator< TRequest>>	Содержит валидаторы, необходимые для проверки определенного класса

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Методы			
Название	Модификатор доступа	Тип	Назначение
Handle	public	Task<TResponse>	Осуществляет валидацию объектов, в случае ошибки – бросает исключение с соответствующим сообщением

Таблица 10.3 – Список полей, методов и свойств класса PasswordValidatorService

Поля			
Название	Модификатор доступа	Тип	Назначение
LocalSalt	private	string	«Соль», приписываемая к паролю, общая для всех пользователей
SaltLength	private	int	Длина сгенерированной соли, своей для каждого пользователя
Методы			
Название	Модификатор доступа	Тип	Назначение
GetHashedPassword	public	(string, string)	Хэширует введенную строку
ValidatePassword	public	bool	Проверяет соответствие введенного пароля тому, который лежит в базе данных
GenerateSaltBytes	private	byte[]	Генерирует массив случайных байт фиксированной длины

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Таблица 10.4 – Список полей, методов и свойств класса *RightsValidatorService*

Поля			
Название	Модификатор доступа	Тип	Назначение
_roleRepository	private	IRoleRepository	Репозиторий ролей, в который из методов поступают необходимые запросы
Методы			
Название	Модификатор доступа	Тип	Назначение
ValidateRights	public	Task<bool>	Проверяет, что у пользователя есть права администратора в данной команде

Все методы в репозиториях выполняются асинхронно.

Таблица 10.5 – Список полей, методов и свойств класса *QuestionRepository*

Методы			
Название	Модификатор доступа	Тип	Назначение
GetQuestionsByTopicId	public	Task<IEnumerable<QuestionDao>>	Получает все вопросы по идентификатору темы
AddQuestion	public	Task<int>	Добавляет новый запрос в базу данных
EditQuestion	public	Task<int>	Редактирует вопрос по идентификатору

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RemoveQuestion	public	Task<int>	Удаляет вопрос из базы данных по идентификатору
GetSavedQuestions	public	Task<IEnumerable<QuestionDao>>	Возвращает сохраненные вопросы пользователя по идентификатору
SaveQuestion	public	Task<int>	Добавляет выбранный вопрос в сохраненные
RemoveQuestionFromSaved	public	Task<int>	Удаляет вопрос из сохраненных вопросов пользователя

Таблица 10.6 – Список полей, методов и свойств класса RoleRepository

Методы			
Название	Модификатор доступа	Тип	Назначение
GetRoleByName	public	Task<RoleDao>	Получает информацию о роли по ее названию
GetUserRoleByTeamId	public	Task<RoleDao>	Получает роль пользователя в команде по ее идентификатору

Таблица 10.7 – Список полей, методов и свойств класса StatisticsRepository

Методы			
Название	Модификатор доступа	Тип	Назначение
GetStatisticsByUserId	public	Task<IEnumerable<StatisticsDao>>	Получает статистику пользователя по его идентификатору

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



AddStatistics	public	Task<int>	Добавляет новую статистику по пройденной теме
EditStatistics	public	Task<int>	Редактирует существующую статистику пользователя по теме

Таблица 10.8 – Список полей, методов и свойств класса *SubjectRepository*

Методы			
Название	Модификатор доступа	Тип	Назначение
GetSubjectById	public	Task<SubjectDao>	Получает информацию о дисциплине по ее идентификатору
GetSubjectsByTeamId	public	Task<IEnumerable<SubjectDao>>	Получает информацию о предметах в команде по идентификатору команды
AddSubject	public	Task<int>	Добавляет новую дисциплину в команду
EditSubject	public	Task<int>	Редактирует дисциплину
RemoveSubject	public	Task<int>	Удаляет дисциплину по ее идентификатору

Таблица 10.9 – Список полей, методов и свойств класса *TeamRepository*

Методы			
Название	Модификатор доступа	Тип	Назначение
GetTeamById	public	Task<TeamDao>	Получает информацию о команде по ее идентификатору

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

GetTeamByNameAndCreator	public	Task<TeamDao>	Получает информацию о команде по ее названию и идентификатору создателя
GetTeamMembersByTeamId	public	Task<IEnumerable<UserDao>>	Получает информацию об участниках команды по идентификатору команды
CreateTeam	public	Task<int>	Создает новую команду
AddMember	public	Task<int>	Добавляет нового участника в команду
ChangeMemberStatus	public	Task<int>	Изменяет роль участника в команде
EditTeam	public	Task<int>	Редактирует информацию о команде
RemoveMember	public	Task<int>	Удаляет участника из команды

Таблица 10.10 – Список полей, методов и свойств класса TopicRepository

Методы			
Название	Модификатор доступа	Тип	Назначение
GetTopicById	public	Task<TopicDao>	Получает информацию о теме по ее идентификатору

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

GetTopicsBySubjectId	public	Task<IEnumerable<TopicDao >>	Получает информацию о темах в предмете по идентификатору предмета
AddSubject	public	Task<int>	Добавляет новую тему в команду
EditSubject	public	Task<int>	Редактирует тему
RemoveSubject	public	Task<int>	Удаляет тему по ее идентификатору

Таблица 10.11 – Список полей, методов и свойств класса UserRepository

Методы			
Название	Мод-р доступа	Тип	Назначение
GetUserById	public	Task<UserDao>	Получает информацию о пользователе по его идентификатору
GetUserWithPasswordByEmail	public	Task<UserWithPasswordDao>	Получает информацию о пользователе, включая его захешированный пароль, по адресу электронной почты
GetUserByEmail	public	Task<UserDao>	Получает основную информацию о пользователе по

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

			адресу электронной почты
GetTeamsByUserId	public	Task< IEnumerable< TeamDao>	Получает команды пользователя по его идентификатору
AddUser	public	Task<int>	Создает нового пользователя
ChangeUserAvatarById	public	Task<int>	Изменяет аватар пользователя

Таблица 10.12 – Список полей, методов и свойств класса RepositorySettings

Поля			
Название	Модификатор доступа	Тип	Назначение
ConnectionString	public	string	Строка, используемая для подключения к базе данных
TimeoutSeconds	public	int	Значение времени таймута для любого запроса в базу данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 ПЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[illegible]