**СПЕЦИФИКАЦИЯ ТРЕБОВАНИЙ**

**К ПРОЕКТУ «Seriy\_Gus»**

**СОДЕРЖАНИЕ**

[**1. ВВЕДЕНИЕ** 4](#_Toc169649646)

[**1.1 Назначение** 4](#_Toc169649647)

[**1.2 Соглашения, принятые в документах** 4](#_Toc169649648)

[**1.3 Границы проекта** 4](#_Toc169649649)

[**2. ОБЩЕЕ ОПИСАНИЕ** 5](#_Toc169649650)

[**2.1 Общий взгляд на продукт** 5](#_Toc169649651)

[**2.2 Классы и характеристики пользователей** 5](#_Toc169649652)

[**2.3 Операционная среда** 5](#_Toc169649653)

[**2.4 Ограничение дизайна и реализации** 5](#_Toc169649654)

[**2.5 Предположения и зависимости** 6](#_Toc169649655)

[**3. ФУНКЦИИ СИСТЕМЫ** 7](#_Toc169649656)

[**3.1 Регистрация пользователя** 7](#_Toc169649657)

[**3.1.1 Описание** 7](#_Toc169649658)

[**3.1.2 Функциональные требования** 7](#_Toc169649659)

[**3.2 Авторизация пользователя** 7](#_Toc169649660)

[**3.2.1 Описание** 7](#_Toc169649661)

[**3.2.2 Функциональные требования** 7](#_Toc169649662)

[**3.3 Выбор задания** 7](#_Toc169649663)

[**3.3.1 Описание** 7](#_Toc169649664)

[**3.3.2 Функциональные требования** 7](#_Toc169649665)

[**3.4 Выполнение задания** 8](#_Toc169649666)

[**3.4.2 Функциональные требования** 8](#_Toc169649667)

[**3.5 Отображение правильно решённых задач** 8](#_Toc169649668)

[**3.5.2 Функциональные требования** 8](#_Toc169649669)

[**3.6 Вывод лучших пользователей** 8](#_Toc169649670)

[**3.6.1 Описание** 8](#_Toc169649671)

[**3.6.2 Функциональные требования** 8](#_Toc169649672)

[**4. ТРЕБОВАНИЯ К ДАННЫМ** 9](#_Toc169649673)

[**4.2 Словарь данных** 9](#_Toc169649674)

[**4.4 Целостность, сохранение и утилизация данных** 10](#_Toc169649675)

[**5. ТРЕБОВАНИЯ К ВНЕШНИМ ИНТЕРФЕЙСАМ** 11](#_Toc169649676)

[**5.1 Пользовательские интерфейсы** 11](#_Toc169649677)

[**5.2** **Интерфейсы ПО** 11](#_Toc169649678)

[**6. АТРИБУТЫ КАЧЕСТВА** 12](#_Toc169649679)

[**6.1 Требования по удобству использования** 12](#_Toc169649680)

[**6.2 Требования к производительности** 12](#_Toc169649681)

[**6.3 Требования безопасности** 12](#_Toc169649682)

[**6.4 Требования к доступности** 13](#_Toc169649683)

# **1. ВВЕДЕНИЕ**

## **1.1 Назначение**

Этот документ представляет собой спецификацию требований к приложению "Seriy\_Gus". Он предназначен для разработчиков, менеджеров проектов, маркетологов, пользователей, тестировщиков и составителей документации.

## **1.2 Соглашения, принятые в документах**

В этой спецификации все заголовки разделов и подразделов выделены жирным шрифтом.

## **1.3 Границы проекта**

Seriy\_Gus (серый гусь) - клиент-серверное приложение, которое позволяет эффективно производить проверку следующих заданий:

* Шифр Виженера
* Шифр sha-512
* Метод деления пополам
* Поиск кратчайшего расстояния между вершинами графа
* Метод хорд

# **2. ОБЩЕЕ ОПИСАНИЕ**

## **2.1 Общий взгляд на продукт**

Приложение предоставляет учителям удобную платформу для проверки знаний своих учеников с возможностью просмотра статистики и выбора лучших.

## **2.2 Классы и характеристики пользователей**

* Ученики: выполняют тесты и видят правильно ли они выполнили задачу
* Учителя: могут видеть статистику учеников, выполненные правильно ими задания и количество выполненных заданий

## **2.3 Операционная среда**

Приложение будет работать на персональных компьютерах и не будет требовать доступа в Интернет.

## **2.4 Ограничение дизайна и реализации**

1. Безопасность данных: необходимо обеспечить безопасность передаваемых данных между клиентом и сервером, а также сохраняемых данных в базе данных. Рекомендуется использовать шифрование при передаче данных по сети и защитить базу данных от несанкционированного доступа.

2. Масштабируемость: при увеличении числа пользователей и объема данных необходимо обеспечить масштабируемость системы, чтобы она могла обслуживать все запросы эффективно. Это может потребовать наличия кластеризации, кеширования, распределенной базы данных и других подходов.

3. Пользовательский интерфейс: интерфейс приложения должен быть интуитивно понятным для пользователей, удобным для работы и соответствовать требованиям задач и ответов

4. Оптимизация производительности: необходимо обеспечить быструю обработку запросов, генерацию задач, ввод ответов и сохранение данных.

## **2.5 Предположения и зависимости**

* У приложения будет механизм авторизации и аутентификации для входа в систему
* Система будет обладать двумя ролями – администраторами и обычными пользователями, у каждой роли свои права
* Задачи будут генерироваться автоматически

**3. ФУНКЦИИ СИСТЕМЫ**

## **3.1 Регистрация пользователя**

### **3.1.1 Описание**

Эта функция запрашивает у пользователя email, имя пользователя и пароль, затем сохраняет их в базу данных для того, чтобы пользователь смог авторизоваться в системе.

### **3.1.2 Функциональные требования**

* Система должна сохранять данные в базу данных
* Система должна выдавать сообщение успешно ли пройдена регистрация

## **3.2 Авторизация пользователя**

### **3.2.1 Описание**

Эта функция допускает зарегистрированных пользователей в систему.

### **3.2.2 Функциональные требования**

* Система должна сравнивать введённые данные с данными, которые хранятся в базе данных
* Система должна выводить сообщение об успешной/неуспешной авторизации

## **3.3 Выбор задания**

### **3.3.1 Описание**

Пользователю предоставляется выбор заданий из 5 тем (шифр Виженера, шифр sha-512, метод деления пополам, метод хорд, поиск кратчайшего расстояния между вершинами графа).

### **3.3.2 Функциональные требования**

* Каждое окошко должно быть подписано именем задачи

## **3.4 Выполнение задания**

**3.4.1 Описание**

Генерируется задача, пользователь должен ввести ответ на неё, затем нажать кнопку проверки и узнать верный ли его ответ.

### **3.4.2 Функциональные требования**

* Система должна рандомно генерировать определённую задачу и выводить текст задачи на экран
* Система должна иметь окошко для ввода ответа пользователем
* Система должна выводить сообщение, верный ли ответ

## **3.5 Отображение правильно решённых задач**

**3.5.1 Описание**

Эта функция показывает пользователю количество верно выполненных им заданий каждого типа.

### **3.5.2 Функциональные требования**

* Система должна брать из базы данных информацию о количестве правильно выполненных тестов
* Система должна показывать название теста и количество правильных выполнений

## **3.6 Вывод лучших пользователей**

### **3.6.1 Описание**

Эта функция предназначена для администраторов, чтобы

просматривать статистику пользователей, которые выполнили наибольшее количество заданий верно.

### **3.6.2 Функциональные требования**

* Система должна иметь окно для ввода администратором количества пользователей, выводящихся на экран
* Система должна автоматически выводить пользователей в порядке убывания
* Система должна показывать id пользователя, логин, роль, количество правильных выполнений каждой задачи и общую сумму баллов за все задачи

# **4. ТРЕБОВАНИЯ К ДАННЫМ**

**4.1 Логическая модель данных**

Сущности:

- Ученик: содержит информацию об учениках, включая их уникальный идентификатор, роль, имя пользователя, пароль.

- Задача: представляет собой сгенерированную задачу, которую ученик должен решить. Включает в себя уникальный идентификатор, текст задачи.

- Ответ: хранит информацию об ответе конкретного ученика на конкретную задачу.

- Учитель: сущность, которая может иметь доступ к статистике учеников. Включает в себя уникальный идентификатор, имя, роль, логин, пароль.

Связи:

-Ученик дает множество ответов на различные задачи.

- У учителя есть доступ к статистике ответов учеников на различные задачи.

- Задача может быть привязана к множеству ответов, где каждый ответ относится к конкретной задаче.

- Ответ связан с конкретным учеником, который дал этот ответ.

- Ответ связан с конкретной задачей, на которую был данный ответ.

## **4.2 Словарь данных**

1. Ученик:

- Идентификатор ученика

- ФИО ученика

- Адрес электронной почты ученика

2. Учитель:

- Идентификатор учителя

- ФИО учителя

- Адрес электронной почты учителя

3. Задача:

- Идентификатор задачи

- Описание задачи

- Ученик, которому назначена задача

- Статус задачи (выполнена/не выполнена)

4. Ответ:

- Идентификатор ответа

- Текст ответа

- Результат (правильный/неправильный)

- Ученик, который дал ответ

- Задача, на которую был дан ответ

## **4.4 Целостность, сохранение и утилизация данных**

Целостность данных:

- Гарантировать, что все данные, включая задачи, ответы пользователей и статистику учеников, сохраняются и обновляются корректно.

- Использовать ограничения целостности, такие как уникальные ключи, внешние ключи и правила проверки, чтобы предотвратить ошибки ввода данных.

Сохранение данных:

- Регулярно создавать резервные копии базы данных, чтобы минимизировать потерю данных в случае сбоя системы.

- Хранить данные в надежном и защищенном сетевом хранилище данных, чтобы предотвратить потерю или утечку данных.

Утилизация данных:

- Каждые полгода проводить чистку данных о выполненных тестах

- Удалять из базы данных пользователей, не пользовавшихся программой больше 5 лет

# **5. ТРЕБОВАНИЯ К ВНЕШНИМ ИНТЕРФЕЙСАМ**

## **5.1 Пользовательские интерфейсы**

- Приложение должно иметь простой и интуитивно понятный способ генерации задач для пользователей.

- Приложение должно иметь возможность выбора типа задачи.

- Приложение должно иметь удобное поле для ввода ответа пользователем.

- Приложение должно иметь валидацию введенного ответа, чтобы предотвратить некорректные данные.

- Приложение должно отображать статистику лучших учеников в виде таблицы с данными.

- Приложение должно обеспечить четкость и наглядность дизайна, чтобы пользователи могли легко ориентироваться.

- Приложение должно иметь несколько ролей для управления доступом и правами пользователей (администраторы, учителя, ученики).

## **Интерфейсы ПО**

-Приложение должно взаимодействовать с базой всех пользователей.

- Приложение должно взаимодействовать с сервером авторизации.

# **6. АТРИБУТЫ КАЧЕСТВА**

## **6.1 Требования по удобству использования**

* Система должна ясно и наглядно генерировать задачи, пользователь должен легко понять, что от него требуется.
* Система должна иметь понятный и легко доступный интерфейс для ввода ответов.
* Система должна сохранять ответы пользователя автоматически и мгновенно после ввода.
* Система должна предоставлять учителям легкий доступ к статистике учеников, возможно даже с графическим представлением результатов.

## **6.2 Требования к производительности**

* Система должна обеспечивать быстрое отображение задач, возможность ввода ответа без задержек.
* Время ожидания ответа от сервера должно быть минимальным, чтобы предотвратить задержки при взаимодействии пользователя с приложением.
* Система должна сохранять ответы пользователей в базу данных моментально.

## **6.3 Требования безопасности**

* Пользователи должны проходить аутентификацию для доступа к приложению.
* Разграничение прав доступа между учениками и учителями, чтобы каждый видел только соответствующую информацию.
* Данные пользователя, включая ответы на задачи, должны быть зашифрованы при хранении в базе данных.
* Все коммуникации между клиентом и сервером должны быть зашифрованы.
* Система должна регулярно создавать резервные копии данных для защиты от потери информации

## **6.4 Требования к доступности**

* Система должна обеспечивать возможность доступа к приложению с разных устройств (компьютеры, планшеты, мобильные устройства) для удобства пользователей.
* Система должна поддерживать высокую доступность сервера для обеспечения непрерывной работы приложения.