

# ТРЕБОВАНИЯ К ПРОЕКТУ

## 1. ВВЕДЕНИЕ

### 1.1 Название продукта

**ToDoList** – веб-сервис для управления задачами.

### 1.2 Контекст проекта

В современном мире пользователям необходимо удобно организовывать свои задачи и контролировать их выполнение. Сервис ToDoList предназначен для упрощения планирования и управления личными и рабочими делами в онлайн-формате. Система будет доступна через веб-браузер, что позволит использовать её с любого устройства, подключённого к интернету.

### 1.3 Общее описание

ToDoList предоставляет пользователю возможность:

- Создавать, редактировать и удалять задачи.
- Группировать задачи по категориям или проектам.
- Отмечать выполненные задачи.
- Устанавливать сроки выполнения.
- Просматривать список предстоящих дел в удобной форме.
- Создавать объединения пользователей (группы), назначать роли, задачи каждому участнику.

### 1.4 Границы проекта

Продукт **будет** включать:

- Простую и интуитивно понятную веб-интерфейс.
- Функции CRUD для задач.
- Возможность авторизации и работы с личным аккаунтом.
- Хранение данных в базе данных.
- Удобный функционал управления задачами и группами.
- Возможность назначения администраторов для управления пользователями.

Продукт **не будет** включать на начальном этапе:

- интеграцию с внешними сервисами (например, календарями или мессенджерами).
- мобильное приложение (только веб-версия).
- расширенные функции управления проектами.

- офлайн-режим работы.

## **2. Требования пользователя**

### **2.1 Программные интерфейсы**

Продукт ToDoList будет взаимодействовать со следующими внешними системами и библиотеками:

1. Веб-браузеры – доступ к сервису будет осуществляться через современные браузеры (Google Chrome, Mozilla Firefox, Microsoft Edge).
2. База данных – используется PostgreSQL для хранения данных о пользователях и их задачах.
3. Фронтенд – React для построения интерфейса.
4. Бэкенд – Java Spring Boot для реализации бизнес-логики и обработки запросов.
5. Сервисы аутентификации – JWT (JSON Web Token) для реализации авторизации и безопасного доступа к личным аккаунтам.
6. REST API – взаимодействие фронтенда и бэкенда будет осуществляться через REST API.

### **2.2 Интерфейс пользователя**

#### **2.2.1 Интерфейс пользователя**

Система ToDoList будет предоставлять пользователю удобный и интуитивный веб-интерфейс. Доступ осуществляется через браузер. Пользователь после авторизации получает доступ к личному кабинету со списком своих задач и (при необходимости) к пространствам групп, в которых он состоит.

Основные элементы интерфейса:

- Форма входа / регистрации.
- Главная страница со списком задач.
- Элементы управления задачами (добавить, редактировать, удалить, отметить как выполненную).
- Фильтры и поиск для быстрого доступа к нужным задачам.
- Меню профиля пользователя (выход из системы, настройки).
- Модуль управления группами (создание групп, добавление участников, изменение ролей, назначение задач).

Таблица 1 – Действие пользователя - реакция

Вводит логин и пароль, нажимает «Войти»	Система проверяет учетные данные и при успехе открывает личный список задач.
Нажимает «Зарегистрироваться»	Система отображает форму регистрации нового пользователя и сохраняет данные.
Нажимает «Добавить задачу»	Система открывает форму создания задачи. После подтверждения – сохраняет задачу.
Отмечает задачу как выполненную	Система обновляет статус задачи и визуально выделяет её как выполненную.
Нажимает «Редактировать» у задачи	Система отображает форму редактирования с текущими данными задачи.
Нажимает «Удалить» у задачи	Система запрашивает подтверждение и удаляет задачу при согласии.
Вводит текст в поле поиска	Система отображает список задач, удовлетворяющих запросу.
Нажимает «Выход»	Система завершает сессию и возвращает пользователя на экран входа.
Создаёт новую группу	Система формирует пространство группы и назначает пользователя её администратором.
Приглашает участника в группу	Система отправляет приглашение выбранному пользователю.
Назначает роль пользователю в группе	Система обновляет права доступа выбранного участника.
Назначает задачу конкретному участнику группы	Система закрепляет задачу за выбранным пользователем и отображает её в списке.

### 2.2.2 Интерфейс администратора

Администратор имеет все возможности обычного пользователя, а также дополнительные функции управления системой и пользователями. Администратор имеет все возможности обычного пользователя, а также дополнительные функции управления системой и пользователями.

Основные элементы интерфейса администратора:

- Панель управления пользователями (просмотр списка всех

зарегистрированных пользователей, их ролей и активности).

- Панель управления группами (создание, удаление, назначение администраторов групп).
- Управление ролями и доступом (назначение прав: администратор, участник, только чтение и т. д.).
- Мониторинг системы (просмотр статистики использования, активности пользователей, количества задач).  
Панель управления пользователями (просмотр списка всех зарегистрированных пользователей, их ролей и активности).
- Панель управления группами (создание, удаление, назначение администраторов групп).
- Управление ролями и доступом (назначение прав: администратор, участник, только чтение и т. д.).
- Мониторинг системы (просмотр статистики использования, активности пользователей, количества задач).

Таблица 2 – Действие администратора – реакция

Просматривает список пользователей	Система отображает всех зарегистрированных пользователей с их ролями.
Назначает пользователя администратором	Система обновляет роль выбранного пользователя и расширяет его права доступа.
Блокирует пользователя	Система ограничивает вход и действия заблокированного пользователя.
Создаёт/удаляет группу	Система формирует или удаляет пространство группы вместе с её задачами.
Назначает администраторов внутри группы	Система обновляет права доступа выбранных участников группы.
Просматривает статистику по задачам	Система отображает отчёты: количество созданных, выполненных и просроченных задач.
Настраивает глобальные параметры системы	Система обновляет настройки (например, правила паролей, лимиты по задачам и т. д.)

## 2.3 Характеристики пользователей

### 2.3.1 Обычные пользователи

Описание: Основная группа пользователей системы ToDoList. Это люди, которые используют сервис для организации личных и рабочих задач.

Характеристики:

- Возраст: от 16 до 50 лет (студенты, офисные сотрудники, специалисты).
- Уровень образования: среднее или высшее.
- Техническая грамотность: базовый или средний уровень. Пользователи знакомы с веб-браузером, умеют работать с простыми веб-приложениями.
- Опыт работы с подобными системами: может отсутствовать; система должна быть интуитивно понятной и не требовать долгого обучения.

Ожидания: простой интерфейс, удобное создание и редактирование задач, доступ к системе с любого устройства.

### 2.3.2 Администраторы

Описание: Пользователи, обладающие расширенными правами. Отвечают за управление системой, пользователями и группами.

Характеристики:

- Возраст: от 22 до 45 лет (чаще всего IT-специалисты или назначенные ответственные лица в организации).
- Уровень образования: среднее специальное или высшее (желательно техническое).
- Техническая грамотность: выше среднего уровня. Администраторы знакомы с системами администрирования, понимают базовые принципы информационной безопасности и управления доступом.
- Опыт работы с подобными системами: желательно наличие опыта использования таск-менеджеров или корпоративных сервисов.

Ожидания: расширенные возможности по управлению пользователями и группами, настройка ролей и доступов, мониторинг активности.

### 2.3.3 Группы пользователей (команды)

Описание: Несколько пользователей, объединённых в группу (например, рабочая команда или студенческая группа).

Характеристики:

- Состав: от 2 до 20 участников.
- Уровень образования: смешанный (от студентов до специалистов).
- Техническая грамотность: разнородная; система должна быть максимально простой и доступной для понимания.
- Опыт работы с подобными системами: может быть минимальным.

Ожидания: возможность совместной работы над задачами, распределение ролей, прозрачный контроль выполнения задач.

## 2.4 Предположения и зависимости

1. Доступ к интернету – пользователи должны иметь стабильное подключение к сети для работы с веб-сервисом.
2. Серверная инфраструктура – корректная работа приложения зависит

от доступности и стабильности серверов, на которых развернут бэкенд ToDoList.

3. Нагрузка и масштабируемость – предполагается, что количество пользователей и групп будет расти, поэтому система должна поддерживать горизонтальное или вертикальное масштабирование.

4. Обновления и сопровождение – для долгосрочной работы системы требуется регулярное обновление библиотек, фреймворков (Spring Boot, React) и обеспечение технической поддержки.

5. Внешние сервисы – в будущем возможны интеграции (например, с календарями или мессенджерами), что может повлиять на расширение требований.

### **3. Системные требования**

#### **3.1 Функциональные требования**

##### **FR-1. Регистрация и аутентификация пользователей**

- Пользователь должен иметь возможность зарегистрироваться в системе.
- Система должна предоставлять вход по логину и паролю.
- Должна использоваться авторизация на основе JWT.

##### **FR-2. Управление задачами**

- Пользователь должен иметь возможность создавать задачу.
- Пользователь должен иметь возможность редактировать задачу.
- Пользователь должен иметь возможность удалять задачу.
- Пользователь должен иметь возможность отмечать задачу как выполненную.

##### **FR-3. Организация задач**

- Система должна предоставлять возможность группировки задач по категориям или проектам.
- Должен быть реализован поиск и фильтрация задач.

##### **FR-4. Управление группами пользователей**

- Пользователь должен иметь возможность создать группу.
- Пользователь с правами администратора группы должен иметь возможность приглашать новых участников.
- Администратор группы должен иметь возможность назначать роли (администратор, участник).
- Администратор или участник должен иметь возможность назначать задачи конкретным пользователям в группе.

##### **FR-5. Интерфейс администратора системы**

- Администратор должен иметь доступ к панели управления пользователями.
- Администратор должен иметь возможность назначать роли (например, администратор системы).

- Администратор должен иметь возможность блокировать пользователей.
- Администратор должен иметь доступ к статистике использования системы (количество пользователей, задач, активность).

## 3.2 Нефункциональные требования

### 3.2.1 Атрибуты качества

1. Надёжность: важна для обеспечения сохранности пользовательских данных и устойчивой работы системы. Измерение: время безотказной работы (uptime  $\geq$  99%).
2. Безопасность: критична, так как система хранит личные данные пользователей. Использование JWT для аутентификации, шифрование паролей (например, bcrypt). Измерение: успешное прохождение тестов на SQL-инъекции, XSS, CSRF и др.
3. Производительность: система должна обрабатывать действия пользователя без заметных задержек. Измерение: среднее время отклика сервера  $\leq$  1 секунда при стандартной нагрузке.
4. Масштабируемость: важно для поддержки роста числа пользователей и групп. Измерение: возможность увеличения нагрузки (например, +100% пользователей) без существенного снижения производительности.
5. Удобство использования (Usability): интерфейс должен быть простым и интуитивно понятным для пользователей без специальной подготовки. Измерение: не более 5 кликов до выполнения основной операции (создание задачи).
6. Поддерживаемость: важно для долгосрочного развития системы. Код должен быть структурирован, модульный. Измерение: наличие документации, unit-тестов (покрытие кода не менее 70%).