

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук  
Кафедра программирования и информационных технологий

Техническое задание  
на разработку автоматизированной системы  
«Система для дистанционного управления учебным процессом Tagger»

Исполнители

\_\_\_\_\_ Д.Г. Шлыков

Воронеж 2024

# СОДЕРЖАНИЕ

1 Введение.....	4
1.1 Цели создания АС.....	4
1.2 Краткая сводка возможностей.....	4
2 Обзор системы.....	6
2.1 Взаимодействие продукта (с другими продуктами и компонентами) ..	6
2.2 Функции продукта (краткое описание) .....	6
2.3 Характеристики пользователя.....	7
2.4 Ограничения.....	7
2.5 Допущения и зависимости.....	7
3 Детальные требования .....	9
3.1 Требования к внешним интерфейсам .....	9
3.1.1 Интерфейсы пользователя .....	9
3.1.2 Интерфейсы аппаратного обеспечения .....	9
3.1.3 Интерфейсы программного обеспечения .....	10
3.1.4 Интерфейсы взаимодействия.....	10
3.2 Функциональные требования .....	10
3.2.1 Главное меню .....	10
3.2.2 Просмотр расписания .....	10
3.2.3 Просмотр домашних заданий .....	11
3.2.4 Просмотр оценок.....	11
3.2.5 Управление расписанием .....	12
3.2.6 Управление домашними заданиями.....	12
3.2.7 Управление оценками.....	13
3.2.8 Оповещения и напоминания .....	14
3.2.9 Логирование и история действий.....	14
3.2.10 Ошибки и обработка исключений.....	15
3.3 Проектные ограничения.....	15
3.4 Нефункциональные требования .....	15
3.5 Другие требования.....	16
4 Тестирование и проверка.....	17
4.1 Тестирование пользовательских функций .....	17
4.1.1 Главное меню .....	17

4.1.2 Просмотр расписания .....	17
4.1.3 Просмотр домашних заданий .....	17
4.1.4 Просмотр оценок.....	17
4.2 Тестирование административных функций.....	18
4.2.5 Управление расписанием .....	18
4.2.6 Управление домашними заданиями.....	18
4.2.7 Управление оценками.....	18
4.3 Тестирование уведомлений и оповещений: .....	18
4.3.8 Оповещения о начале занятий .....	18
4.3.9 Оповещения о новых и изменённых домашних заданиях .....	18
4.3.10 Напоминания о сроках сдачи.....	19

## **1 Введение**

### **1.1 Цели создания АС**

Целью создания автоматизированной системы «Tagger» является автоматизация процесса уведомления учебной группы о сроках выполнения домашних заданий, расписании занятий и своевременного оповещения о начале пары. Система позволит сократить вероятность пропуска важной информации студентами и упростить организацию учебного процесса за счет своевременной и точной передачи данных.

Создание такой системы позволит повысить уровень дисциплины среди студентов, снизить количество ошибок, связанных с пропуском информации, и улучшить управление учебным процессом в группе.

Также будет введена небольшая функциональность для администраторов системы, чтобы проверить насколько удобно будет им управлять учебным процессом в ВК боте.

Для начала бот будет работать только для 3 групп факультета, по итогам тестирования на них, будет предпринята попытка развернуть одного бота для всего факультета.

### **1.2 Краткая сводка возможностей**

Основные задачи, которые решает система:

- Уведомление студентов о текущих домашних заданиях, включая описание, сроки сдачи и изменения.
- Оповещение о времени начала занятий и месте их проведения (ссылки на онлайн-занятия или номер аудитории).
- Быстрое редактирование оценок студентов администраторами.
- Предоставление возможности изменения и просмотра информации через интуитивно понятный интерфейс с кнопками и контекстными меню в чат-боте ВКонтакте. Все действия пользователя (просмотр

расписания, домашних заданий, оценок) осуществляются через простые и понятные элементы управления (кнопки), исключая необходимость ввода текстовых команд. Администраторы получают доступ к дополнительным функциям через цветовую дифференциацию кнопок и контекстные подсказки.

## **2 Обзор системы**

### **2.1 Взаимодействие продукта (с другими продуктами и компонентами)**

Бот Tagger будет взаимодействовать с социальной сетью ВКонтакте через API для получения, отправки и обработки сообщений, а также для управления группами пользователей. Аутентификация реализована через VK ID, что значит – использование бота доступно только для зарегистрированных пользователей ВК. Авторизация реализована по средствам использования базы данных студентов группы и добавления их id в ВК. Основное взаимодействие осуществляется через:

- API ВКонтакте: для работы с отправкой сообщений, получения данных о пользователях, и управления группой.
- База данных: для хранения данных о домашних заданиях, расписании и информации о пользователях.
- Административные инструменты: для предоставления администраторам функционала управления ботом (добавление/удаление заданий, изменение расписания).

### **2.2 Функции продукта (краткое описание)**

Бот Tagger выполняет следующие основные функции:

- Добавление, редактирование и удаление домашних заданий для учебных групп.
- Хранение и управление расписанием занятий.
- Отправка уведомлений о предстоящих занятиях и сроках сдачи домашних заданий.
- Автоматические напоминания о начале пары и месте её проведения (ссылка или аудитория).

- Возможность администраторам управлять заданиями и расписанием через интуитивное кнопочное меню и понятный формат ввода данных просто сообщением в ВК.

## **2.3 Характеристики пользователя**

Основные пользователи системы:

- Студенты: будут получать уведомления о домашних заданиях и расписании, а также запрашивать информацию через команды бота.
- Преподаватели или старосты: будут выполнять административные функции, добавляя и изменяя информацию через команды.
- Администраторы группы ВКонтакте: обладают правами для управления ботом и его настройками.

Пользователи не требуют специальных знаний для работы с системой, взаимодействие происходит через стандартный интерфейс ВКонтакте (чат в группе).

## **2.4 Ограничения**

- Бот работает только в рамках платформы ВКонтакте.
- Максимальная длина сообщения и передаваемой информации ограничена требованиями ВКонтакте (например, 4096 символов на сообщение).
- Производительность и быстродействие бота зависят от скорости работы API ВКонтакте и серверной инфраструктуры.
- Для полноценной работы необходима постоянная связь с интернетом и доступ к серверу, на котором размещён бот.

## **2.5 Допущения и зависимости**

- Предполагается, что учебная группа уже активно использует ВКонтакте для коммуникации, и бот будет интегрирован в эту среду.

- Для корректной работы системы требуется актуальное расписание, которое будет обновляться администратором вручную в соответствии с изменениями (переносы, отмены).
- Бот зависит от сторонних сервисов (API ВКонтакте), и его работоспособность может быть ограничена в случае изменений на стороне этих сервисов.



### **3 Детальные требования**

#### **3.1 Требования к внешним интерфейсам**

##### **3.1.1 Интерфейсы пользователя**

Взаимодействие с ботом происходит через интерфейс кнопок в чат-боте ВКонтакте. Пользователи видят главное меню, содержащее следующие кнопки:

- Расписание
- Оценки
- Домашние задания

При нажатии на кнопку Расписание, меню меняется, предлагая выбор:

- Показать расписание на конкретный день.
- Показать расписание на неделю.

Администраторы получают доступ к дополнительным функциям, которые выделяются кнопками другого цвета:

- Добавить/изменить/удалить занятие из меню Расписания
- Добавить/изменить/удалить оценку студента из меню Оценки
- Добавить/изменить/удалить задание из меню Домашние задания

При выборе нужного пункта меню бот отображает соответствующую информацию в виде сообщений, а администраторы могут вносить изменения с помощью кнопок, не используя текстовые команды.

##### **3.1.2 Интерфейсы аппаратного обеспечения**

Бот не требует специального аппаратного обеспечения на стороне пользователей, работает на любых устройствах с доступом к ВКонтакте (мобильные телефоны, компьютеры).

### **3.1.3 Интерфейсы программного обеспечения**

Основное взаимодействие происходит через API ВКонтакте.

Для хранения данных используется база данных, взаимодействие с которой происходит через серверное приложение бота.

### **3.1.4 Интерфейсы взаимодействия**

Внешние системы: интеграция с ВКонтакте через VK API для работы с сообщениями и пользователями.

Взаимодействие между пользователями и ботом осуществляется через интерфейс меню с кнопками. Действия пользователей зависят от контекста текущего состояния меню. Например, при нажатии на кнопку “Домашние задания” отображаются задания с возможностью управления ими для администраторов через кнопки.

## **3.2 Функциональные требования**

### **3.2.1 Главное меню**

После взаимодействия с ботом пользователю отображается главное меню, содержащее следующие кнопки:

- Расписание
- Оценки
- Домашние задания

Пользователь может выбрать интересующую его опцию для получения или управления информацией.

### **3.2.2 Просмотр расписания**

При нажатии на кнопку Расписание открывается меню с выбором:

Показать расписание на конкретный день – пользователь выбирает день недели, и бот отображает расписание занятий на этот день.

Показать расписание на неделю – бот отображает полное расписание занятий на всю неделю.

Расписание включает информацию о времени начала пары и месте проведения (ссылки для онлайн-занятий или номер аудитории).

Также присутствуют стрелки влево и вправо для просмотра недель, при этом в сообщении бот присылает информацию о датах начала и конца недели и числитель или знаменатель эта неделя.

### **3.2.3 Просмотр домашних заданий**

При нажатии на кнопку Домашние задания бот отображает список всех актуальных домашних заданий с указанием сроков сдачи и описанием.

Пользователь может выбрать домашнее задание для получения подробной информации (например, описание и требования).

### **3.2.4 Просмотр оценок**

При нажатии на кнопку Оценки бот отображает пользователю его личные оценки по доступным дисциплинам.

Информация о каждой оценке включает:

- Название дисциплины.
- Полученная оценка.
- Дата выставления оценки.
- Комментарии от преподавателя или администратора (если такие были добавлены).

Пользователь может выбрать конкретную дисциплину, чтобы получить более подробную информацию.

Если у пользователя нет доступных оценок, бот отправляет сообщение с соответствующей информацией.

### 3.2.5 Управление расписанием

Администраторы могут управлять расписанием через меню, доступное только им (кнопки другого цвета):

- Добавить занятие: позволяет добавить информацию о новом занятии. При нажатии на кнопку Добавить занятие, администратору отображается формат ввода информации о новом занятии: **дата проведения**, время начала, **название предмета, преподаватель**, продолжительность, и место проведения (ссылка для онлайн-занятий или номер аудитории). После ввода данных администратор подтверждает добавление занятия, и бот сохраняет его в базе данных.
- Изменить занятие: позволяет изменить время, **преподавателя** или место существующего занятия. При нажатии на кнопку Изменить занятие администратор видит список всех занятий в текущем расписании. Администратор выбирает конкретное занятие для редактирования, после чего бот предлагает изменить параметры занятия (например, время, место или **дату проведения**). После изменения данных бот сохраняет обновлённую информацию в базе данных.
- Удалить занятие: позволяет удалить занятие из расписания. При нажатии на кнопку Удалить занятие, администратор видит список всех занятий в расписании. Администратор выбирает занятие, которое необходимо удалить, и бот подтверждает удаление, после чего удаляет занятие из базы данных.

### 3.2.6 Управление домашними заданиями

Администраторы могут управлять домашними заданиями через меню, доступное только им (кнопки другого цвета):

- Добавление домашнего задания: При нажатии на кнопку Добавить задание администратор видит формат ввода следующей информации:

название задания, дата сдачи, краткое описание и другие подробности. После ввода данных администратор подтверждает добавление, и бот сохраняет задание в базе данных, а также отправляет уведомление всем пользователям группы о новом домашнем задании.

- Изменение домашнего задания: При нажатии на кнопку Изменить задание администратор видит список всех активных домашних заданий. Администратор выбирает задание для редактирования, после чего бот предлагает изменить параметры задания (например, срок сдачи, описание или название). После изменений бот обновляет информацию в базе данных и уведомляет участников группы о изменениях.
- Удаление домашнего задания: При нажатии на кнопку Удалить задание администратор видит список всех активных домашних заданий. Администратор выбирает задание, которое нужно удалить, подтверждает действие, и бот удаляет задание из базы данных, уведомив об этом всех пользователей.

### **3.2.7 Управление оценками**

Администраторы могут управлять оценками через меню, доступное только им (кнопки другого цвета):

- Добавление оценки: При нажатии на кнопку Добавить оценку, администратор видит список студентов группы и выбирает студента, которому нужно добавить оценку. После выбора студента администратор вводит данные об оценке (дисциплина, оценка, комментарий, дата выставления) и подтверждает действие. Оценка сохраняется в базе данных, а бот уведомляет студента о добавленной оценке.

- Изменение оценки: При нажатии на кнопку Изменить оценку администратор видит список студентов и выбирает конкретного студента, чью оценку необходимо изменить. Администратор выбирает оценку для редактирования, изменяет необходимые параметры (оценку или комментарии) и подтверждает действие. Бот обновляет информацию в базе данных и уведомляет студента об изменении.
- Удаление оценки: При нажатии на кнопку Удалить оценку, администратор видит список студентов и выбирает студента, чья оценка должна быть удалена. После подтверждения бот удаляет оценку из базы данных и уведомляет студента об этом.

### **3.2.8 Оповещения и напоминания**

- Оповещения о новых заданиях: Бот автоматически отправляет уведомление всем пользователям группы при добавлении нового домашнего задания администратором.
- Напоминание о сроках сдачи заданий: Бот отправляет напоминания о приближении срока сдачи домашнего задания за определённое время до дедлайна (например, за 24 часа и за 1 час до срока сдачи).
- Оповещения о начале занятий: Бот автоматически отправляет уведомления о начале пар в соответствии с расписанием за заранее заданный промежуток времени (например, за 10 минут до начала).

### **3.2.9 Логирование и история действий**

Хранение истории действий: Бот сохраняет историю всех действий пользователей и администраторов (например, добавление или изменение задания, просмотр расписания). Администраторы могут запросить историю изменений для анализа. Каждое изменение хранится в формате <ДАТА> <ДЕЙСТВИЕ> <КТО СОВЕРШИЛ ДЕЙСТВИЕ>.

Восстановление предыдущего состояния: Бот поддерживает возможность возврата к предыдущим действиям через контекстные кнопки, позволяя пользователям продолжить взаимодействие с ботом с того места, на котором они остановились.

### **3.2.10 Ошибки и обработка исключений**

Обработка ошибок: При возникновении ошибки (например, неправильный формат данных при добавлении задания) бот возвращает сообщение об ошибке и предлагает пользователю повторить ввод данных с помощью кнопок.

Подтверждения действий: Все критичные действия (например, удаление задания или изменения в расписании) требуют подтверждения от пользователя через дополнительную кнопку "Подтвердить" перед выполнением действия.

## **3.3 Проектные ограничения**

Разработка должна учитывать ограничения API ВКонтакте (ограничение по количеству запросов в единицу времени, ограничение на длину сообщений).

Использование стандартных технологий для ботов (например, Node.js, Express.js) и баз данных.

## **3.4 Нефункциональные требования**

Бот должен работать 99,5% времени в месяц, что позволяет ему иметь до 3 часов 36 минут простоя в течение месяца. Для достижения такого уровня надёжности будут применяться следующие меры: Использование облачных серверов с автоматическим резервированием и масштабированием. Постоянный мониторинг работы бота и VK API, с быстрым реагированием на проблемы. Планирование профилактических работ в ночное время или периоды низкой нагрузки. Поддержка отказоустойчивой архитектуры на стороне серверной инфраструктуры (например, резервные копии баз данных и кластеризация серверов).

Доступность: бот должен быть доступен всем членам группы ВКонтакте в любое время.

Безопасность: доступ к административным функциям бота должен быть защищен и ограничен только для администраторов.

Масштабируемость: система должна поддерживать возможность увеличения числа пользователей и количества данных без значительного изменения архитектуры.

Время отклика бота на команды не должно превышать 2–5 секунд.

Бот должен поддерживать одновременную работу с группами до 200 участников без ухудшения производительности.

Поддержка до 1000 запросов в час при стандартных нагрузках.

### **3.5 Другие требования**

Логи изменений и действий должны сохраняться в базе данных для дальнейшего анализа и улучшения работы системы.

Все действия пользователей и администраторов должны быть зарегистрированы для последующего аудита.



## **4 Тестирование и проверка**

Для проверки корректности работы бота Tagger должны быть проведены следующие приёмочные тесты, соответствующие функциональным требованиям:

### **4.1 Тестирование пользовательских функций**

#### **4.1.1 Главное меню**

Проверка отображения кнопок: Расписание, Оценки, Домашние задания.

Проверка корректности перехода по каждому пункту меню.

**Успешно открывается соответствующее контекстное меню**

#### **4.1.2 Просмотр расписания**

Тестирование вывода расписания на день и на неделю.

Проверка корректности отображения времени и места занятий.

**Даты и места занятий совпадают с расписанием**

#### **4.1.3 Просмотр домашних заданий**

Проверка отображения списка домашних заданий и их деталей (сроки, описание).

Тестирование правильности перехода к подробной информации по конкретным заданиям.

**Переход к занятию переводит к нужному занятию с корректной информацией**

#### **4.1.4 Просмотр оценок**

Проверка корректности отображения оценок для каждого студента.

Тестирование вывода подробной информации по оценкам (дисциплина, дата, комментарии).

**Выведенные оценки соответствуют студенту**

## **4.2 Тестирование административных функций**

### **4.2.5 Управление расписанием**

Добавление, изменение и удаление занятий с проверкой правильности внесения данных и корректного обновления расписания.

Данные успешно добавляются, изменяются и удаляются и все изменения применяются к базе данных.

### **4.2.6 Управление домашними заданиями**

Тестирование добавления, изменения и удаления домашних заданий.

Проверка отправки уведомлений пользователям при изменении информации.

Данные успешно добавляются, изменяются и удаляются и все изменения применяются к базе данных. И уведомления отправляются

### **4.2.7 Управление оценками**

Тестирование добавления, изменения и удаления оценок.

Проверка уведомлений студентов о внесённых изменениях.

Данные успешно добавляются, изменяются и удаляются и все изменения применяются к базе данных. И уведомления отправляются

## **4.3 Тестирование уведомлений и оповещений:**

### **4.3.8 Оповещения о начале занятий**

Проверка автоматической отправки уведомлений за установленное время до начала занятий.

Уведомления успешно отправляются

### **4.3.9 Оповещения о новых и изменённых домашних заданиях**

Тестирование отправки уведомлений при добавлении или изменении домашнего задания.

Уведомления успешно отправляются

#### **4.3.10 Напоминания о сроках сдачи**

Проверка отправки напоминаний о приближающемся дедлайне домашнего задания.

Уведомления успешно отправляются