

АННОТАЦИЯ

Пояснительная записка к выпускной квалификационной работе на тему «Discord-бот для автоматизированного вывода расписания», представленная на 80 листах.

Во введении обоснована актуальность выбранной темы.

Литературный обзор посвящен составлению перечня используемой литературы.

Первый раздел рассматривает теоретические вопросы по теме и существующую технологию получения информации о расписании занятий. Также в данном разделе формулируется техническое задание и постановка задачи на разработку приложения.

Во втором разделе предлагается технология для автоматизации процесса получения расписания, создается информационная модель разрабатываемой системы и выбираются техническое и программное обеспечение разработки.

В третьем разделе выбираются технологии для разработки бота и описывается сам процесс реализации приложения:

1. Клиентской части.
2. Серверной части.
3. Базы данных.
4. Подключение к сторонним платформам.
5. Тестирование приложения.
6. Составление руководства пользователя.

Четвертый раздел рассматривает экономические аспекты разработки.

Заключение содержит итоги проделанной работы.

В приложении представлены блок-схемы алгоритмов разработанной программы и часть программного кода приложения.

СОДЕРЖАНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ	7
ВВЕДЕНИЕ	8
ЛИТЕРАТУРНЫЙ ОБЗОР	10
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	11
1.1 Обзор предметной области	11
1.1.1 Боты	11
1.1.2 Мессенджеры	12
1.1.3 Мессенджер Discord	13
1.2 Существующие технологии получения информации о расписании занятий	13
1.3 Техническое задание	15
1.3.1 Общие сведения	15
1.3.2 Назначения и цели создания системы	16
1.3.3 Требования к системе	16
1.4 Постановка задачи на разработку	19
2 АНАЛИТИЧЕСКИЙ РАЗДЕЛ	20
2.1 Предлагаемая технология автоматизации информации о расписании занятий	20
2.1.1 Формирование базы данных	20
2.1.2 Предоставление информации о расписании	21
2.2 Информационное обеспечение приложения	21
2.2.1 Информационная модель и её описание	21
2.2.2 Входная и результативная информация	28
2.3 Программное и техническое обеспечения разработки	29
2.3.1 Программное обеспечение	29
2.3.2 Техническое обеспечение	31
3 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ	32
3.1 Обоснование выбора средств разработки приложения	32

3.1.1 База данных.....	32
3.1.2 Язык программирования JavaScript.....	33
3.1.3 Программная платформа “Node.js”	34
3.2 Описание реализации приложения.....	34
3.2.1 Регистрация бота	34
3.2.2 Создание клиентской части бота	35
3.2.3 Реализация серверной части бота.....	36
3.2.4 Разработка базы данных бота	40
3.2.5 Размещение бота на удаленном сервере Heroku	40
3.3 Тестирование приложения	41
3.3.1 Результат тестирования	42
3.4 Инструкция по работе пользователя с приложением.....	45
4 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ	46
4.1 Организация и планирование работ по теме	46
4.1.1 Организация работ	47
4.1.2 График проведения работ.....	48
4.2 Расчёт стоимости проведения работ по теме	48
4.2.1 Расчёты себестоимости.....	48
4.2.2 Расчёты договорной цены	53
ЗАКЛЮЧЕНИЕ	54
СПИСОК ЛИТЕРАТУРЫ.....	56
ПРИЛОЖЕНИЯ.....	59

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

ВУЗ – высшее учебное заведение

ПК – персональный компьютер

ФОТ – фонда оплаты труда

ОЗП – основная заработная плата научного и производственного персонала

ДЗП – дополнительная заработная плата научного и производственного персонала

СВ – страховые отчисления

НР – Накладные расходы

П – прибыль

С – себестоимость продукта

НДС – налог на добавочную стоимость

ВВЕДЕНИЕ

Интернет в современном мире стал одним из самых популярных способов для получения и обмена информацией. Существует огромное количество различных сайтов, приложений и платформ, хранящих необходимую для пользователя информацию, из-за чего возникает потребность в сборе и структуризации этой информации в одном, удобном для человека, месте.

В данной выпускной работы будет рассматриваться получение информации о расписании занятий в ВУЗе РТУ МИРЭА, и передаваться в мессенджер Discord, где проходят дистанционные занятия по некоторым из изучаемым дисциплинам.

Для сбора, обработки и предоставления пользователю информации будет использоваться специальная программа – бот.

Целью выпускной работы является разработка бота для мессенджера Discord, осуществляющий автоматизированный процесс получения и предоставления расписания занятий групп в РТУ МИРЭА.

Для выполнения поставленных целей будут решены следующие задачи:

- обзор предметной области и существующей технологии получения расписания;
- формулирование технического задания и постановки задачи;
- создание информационной системы приложения;
- выбор технологий и среды программирования для разработки бота;
- расчёт экономических аспектов разработки приложения;
- реализация приложения;
- тестирование приложения.

Процесс написания выпускной квалификационной работы будет проходить в соответствии со следующими нормативными актами:

1. «О защите населения и территории от чрезвычайных ситуаций природного и техногенного характера» от 21.12.1994 № 68-ФЗ.

2. «Об основах охраны здоровья граждан в Российской Федерации» от 21.11.2011 № 323-ФЗ
3. «О гражданской обороне» от 12.02.1998 № 28-ФЗ.
4. Приказ Минздравсоцразвития РФ от 04.05.2012 № 477н «Об утверждении перечня состояний, при которых оказывается первая помощь, и перечня мероприятий по оказанию первой помощи».
5. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ
6. СанПин — 2.2.2/542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы»

ЛИТЕРАТУРНЫЙ ОБЗОР

Для получения более глубоких знаний по предметной области создаваемого приложения и проведения самого процесса разработки бота, был составлен следующий перечень используемой литературы:

- Технологии создания и применения чат-ботов. Н. Ю. Матвеева, А. В. Золотарюк [26]. Статья посвящена анализу возможностей по созданию и применению ботов в мессенджерах, что поможет при исследовании предметной области выпускной квалификационной работы;
- Документация Discord для разработчика [10]. Данная документация описывает работу мессенджера Discord и объясняет все доступные возможности при разработке бота для этого мессенджера;
- JavaScript. Полное руководство. Флэнаган Дэвид [12]. Данный учебник по языку программирования JavaScript раскроет основные особенности при разработке на JavaScript и поможет реализовать программную часть приложения, включая создание базы данных;
- Документация Node.js [27]. Документация объяснит принцип работы на платформе Node.js, необходимой для создания серверной части приложения и подключения к проекту сторонних библиотек;
- Экономика предприятия и управление организацией М. Григоренко О.В., Садовнича И.О., Мыльникова А. [26]. Книга объяснит организационный и экономический аспекты при разработке приложения: планирование, организацию и расчёт затрат на сам процесс разработки.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

В исследовательском разделе будет произведен обзор предметной области, где рассматриваются понятия «бот» и «мессенджер», а также обосновывается выбор мессенджера Discord как платформы для создаваемого бота. Помимо предметной области в разделе будет описан существующий способ получения расписания занятий в ВУЗе РТУ МИРЭА и сформулировано техническое задание и постановка задачи на разработку приложения.

1.1 Обзор предметной области

В качестве предметной области выступает создание бота в мессенджере, объектом в которой является бот.

1.1.1 Боты

Бот — это прикладная программа, которая взаимодействует с пользователем через специальный интерфейс согласно определенному алгоритму.

Взаимодействие бота с человеком представляет собой универсальное средство, способное решать обширную область задач, например, консультирование (Виртуальные ассистенты Салют в мобильном приложении Сбербанк [2]), развлечение (музыкальные или игровые боты в Discord [3]), поиск товара или услуги (чат-бот на сайте DNS.ru [4]) и другие.

Структурно бота можно представить, как базу знаний, состоящую из наборов возможных вопросов, которые человек может задать, и ответов на такие вопросы.

Использование ботов дает огромное преимущество. Так как скорость работы таких программ в разы быстрее человеческой, что позволит максимально быстро среагировать и обработать запрос пользователя.

Помимо скорости применять бота можно для повторяющейся и однообразной работы по поиску информации для пользователя, структурирования данных или других подобных действий.

В последнее время появился особый вид ботов – чат-боты, разработанные специально для работы в мессенджерах, чатах или иных платформах для общения.

Чат-бот является специальным аккаунтом в мессенджере, читающий и отвечающий на сообщения пользователя через интерфейс мессенджера.

1.1.2 Мессенджеры

Мессенджер представляет собой интернет-сервис для общения между пользователями. Он является альтернативой социальным сетям, форумам и другим платформам для общения.

Использование мессенджера, как способа коммуникации, в последние годы сильно возрастает. Согласно опросу ВЦИОМ [5] за 2018 год 62% россиян отправляют хотя бы одно сообщение в мессенджере раз в сутки, хотя ещё в 2015 году в схожем по теме опросе [6] только 44% пользователей интернета практиковали общение в социальных сетях, чатах или форумах.

Возросшая популярность обусловлена в первую очередь распространением использования смартфонов среди россиян и развитием мобильного интернета (удешевление интернет-тарифов, повышение скорости Интернет-соединения).

На сегодняшний день мессенджеры способны отправлять текстовые и аудио-сообщения, медиа-файлы, а также реализовывать аудио- и видео-общение. Такой функционал также способствует увеличению популярности использования мессенджеров.

Помимо популярности мессенджеров актуальность темы обусловлена тем, что в связи с мерами по борьбе с новой коронавирусной инфекцией [7] обучение

временно переходило на дистанционный формат и часть занятий проходила в мессенджерах.

Одним из мессенджеров, где начали проходить занятия был Discord.

1.1.3 Мессенджер Discord

Разработчики Discord позиционируют свой продукт как платформа для общения сообществ по интересам. Интерфейс приложения позволяет гибко взаимодействовать с чатами, давая возможность использовать текстовые или голосовые каналы, а поддержка кроссплатформенности (пользователи для общения могут использовать приложение Discord на ПК, смартфоне или подключиться к мессенджеру через веб-браузер) позволяет охватывать большое количество аудитории для сообществ.

Особенно активно внутри Discord развивается использование ботов. У приложения присутствует встроенный менеджер для разработки и управления ботом – Discord API [13], позволяющий создавать ботов с совершенно разнообразными возможностями: бот-калькулятор, бот, воспроизводящий или транслирующий видеоролики из сервиса “youtube” или даже бот-игра, разработанная на движке Unity и управляемая с помощью реакций на сообщения.

Гибкость при работе и неограниченный функционал ботов, разработанных с помощью Discord API, послужил причиной выбора мессенджера Discord для написания бота.

1.2 Существующие технологии получения информации о расписании занятий

В настоящий момент для мессенджера Discord не существует какого-либо бота, выводящего расписания занятий в ВУЗе РТУ МИРЭА. Поэтому пользователем приходится получать информацию самостоятельно.

Способом узнать расписания для пользователя является получение информации с официального сайта РТУ МИРЭА [16], где в разделе “Расписание – РТУ МИРЭА” [15] находятся файлы с расписанием для каждого курса всех институтов учебного заведения.

Такие файлы имеют формат “xlsx” (формат книг Excel), которые пользователь может скачать себе на устройство и открыть для просмотра в программе “Excel” (Рисунок 1.1).

РАСПИСАНИЕ занятий Института перспективных технологий и индустриального									
День недели	Группа				ЭОСО-01-20	оэпс	12.05.01	19	
	№ пары	Нач. занятия	Оконч. занятия	Неделя			ОЭИППиС		
					Предмет	Вид занятия	ФИО преподавателя	№ ауд.	Ссылка
ПОНЕДЕЛЬНИК	1	9-00	10-30	I					
				II					
	2	10-40	12-10	I	3-17 н. Квантовая механика	лк	Гладышев И.В.	А-3	
				II					
	3	12-40	14-10	I	3-17 н. Физические основы лазерной техники	лк	Величко А.П.	А-5	
				II					
	4	14-20	15-50	I	3-17 н. Методы математической физики	лк	Приходько В.Ю.	А-9	
				II	Теория вероятностей и математическая статистика	лк	Шамин Р.В.	А-9	
				I	Философия	лк	Гладышева Е.В. с ЭСБО-21	А-11	
	5	16-20	17-50	II	Психология и педагогика	лк	Талалуева Т.А.	В-86* К-8	
				I	3-17 н. Метрология, стандартизация и сертификация	лк	Русанов К.Е.	А-11	
	6	18-00	19-30	II	10,12,14 н. Электроника	лк	Брысин А.Н.	В-86* К-8	
ВТОРНИК	1	9-00	10-30	I					
				II	2,6,10,14 н. Электроника	пр	Брысин А.Н.	С-20 333	
	2	10-40	12-10	I					
				II	2,6,10,14 н. Электроника	пр	Брысин А.Н.	С-20 333	
СРЕДА				I					
				II					

Рисунок 1.1 – Файл с расписанием занятий осеннего семестра 3 курса института перспективных технологий и индустриального программирования на 2022-2023 уч. г., открытый в программе “Microsoft Excel”.

Внутри файл предоставляет собой таблицу с информацией о занятиях всех групп для выбранного курса и института.

При этом поиск информации о проводимых занятиях какого-либо преподавателя затруднителен, так как расписание показывается только для

отдельных учебных групп, и пользователю придётся скачивать и просматривать множество файлов с расписанием групп различных курсов и институтов, чтобы получить информацию о занятиях преподавателя.

1.3 Техническое задание

Для проведения разработки составим техническое задание, содержащее сведения о будущем приложении и требования к его разработке.

1.3.1 Общие сведения

Наименование разрабатываемой системы – «Discord-бот для автоматизированного вывода расписания».

Разработчик – Вишневский Никита Александрович.

Плановые сроки начала и окончания работы по созданию системы

26 февраля 2022 – Начало разработки.

27 мая 2022 – Сдача проекта.

Порядок оформления и предъявления результатов работ по созданию системы

Работы по созданию системы делятся на следующие этапы:

1. Создание информационной системы приложения.
2. Выбор технического и программного обеспечения для разработки системы.
3. Выбор технологий разработки для реализации приложения.
4. Написание кода приложения.
5. Тестирование системы.

1.3.2 Назначения и цели создания системы

Система будет предназначена для предоставления пользователю информации о расписании занятий учебных групп или преподавателей в ВУЗе РТУ МИРЭА.

В качестве цели системы будет выступать автоматизация процесса получения расписания, которая заключается:

- в сборе информации о расписании занятий учебных групп и преподавателей из файлов с расписанием, находящихся на официальном сайте РТУ МИРЭА;
- в предоставлении пользователю информации о собранной информации.

1.3.3 Требования к системе

Требования к системе включают в себя:

- общие требования к разрабатываемому приложению;
- частные требования к процессу разработки бота.

Общие требования

Ключевыми требованиями к создаваемому боту являются:

1. Реагирование на сообщения пользователя.
2. Обработка полученной от пользователя информации.
3. Формирование ответа на запрос пользователя.
4. Возможность получения информации из Интернет-ресурсов.
5. Произведение операций по созданию, удалению и изменению файлов внутри проекта.
6. Подключение и взаимодействие с базой данных.

Бот, получив от пользователя запрос на вывод расписания, должен предоставить информацию о расписании определенной группы.

В случае если пользователь посчитает, что предоставляемая информация некорректна или устарела, и отправляет соответствующий запрос на обновление расписания, чат-боту необходимо скачать с официального сайта РТУ МИРЭА [16] файлы с новым расписанием, обработать их и обновить свою базу данных.

Частные требования

Частные требования затрагивают все те требования, которые непосредственно с самим процессом разработки.

Требования к методу разработки

Discord-бота, как программу, можно написать на любом языке программирования или создать с помощью специального конструктора. Но в данной работе вариант с использованием конструктора не подходит, так как созданный бот будет работать только в соответствии с алгоритмами, прописанными в конструкторе, что сильно ограничит функционал проекта и не позволит выполнить все требования к нему.

Требования к языку программирования

Язык программирования для бота должен поддерживать все необходимые библиотеки и методы для выполнения поставленных требований проекта.

Требования к среде разработки

У среды разработки должно быть наличие поддержки выбранного языка программирования, а также возможность подключения всех необходимых библиотек для разрабатываемой системы.

Требования к базе данных

База данных, используемая в проекте, должна удовлетворять следующим требованиям:

1. Возможность многократного использования данных.
2. Простота изменения или обновления информации внутри базы данных.
3. Поиск и получение информации по запросу.
4. Данные в базе данных должны быть целостными и непротиворечащими друг другу.

Требования к серверам для размещения бота

Сервер, на котором будет размещен проект, должен поддерживать язык программирования, на котором будет написан проект, и функционировать в режиме 24/7.

Требования к интерфейсу для пользователя

Интерфейс для разрабатываемого проекта должен быть интуитивно понятным для пользователя и реализовывать обмен информацией между человеком и ботом.

Требования к системе пользователя

Согласно официальной спецификации “Discord” [8] минимальными системными требованиями для пользователя являются:

- для персональных компьютеров – Windows 7, MacOS 10.11;
- для мобильных телефонов – IOS 11.0, Android 5;
- для браузера – Google Chrome, Firefox 80+, Opera, Microsoft Edge 17+, Safari 11+.

При этом бот не требует использование дополнительных мощностей со стороны пользователя, поэтому данных требований к системе будет достаточно.

1.4 Постановка задачи на разработку

Требуется разработать программу, реализующую следующие функции:

- взаимодействие с пользователем в мессенджере Discord;
- получение информации о расписании занятий с сайта РТУ МИРЭА[16] и её сохранение в базе данных;
- предоставление пользователю руководства пользователя или информации о расписании занятий одной из групп или преподавателя, находящихся в базе данных.

При самом процессе необходимо:

- создать информационную модель системы;
- выбрать используемые технологии для разработки;
- реализовать взаимодействия программы с мессенджером Discord;
- создать клиентскую и серверную части приложения;
- выбрать сервер для размещения приложения;
- протестировать приложение.

Также при создании приложения должны учитываться требования из технического задания, п.1.3.3.

2 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

В аналитическом разделе будет предложена технология по автоматизации информации о расписании занятий, построена информационная модель разрабатываемой системы в виде диаграмм, и выбрано программное и техническое обеспечения для разработки приложения.

2.1 Предлагаемая технология автоматизации информации о расписании занятий

Для автоматизации процесса получения информации о расписании занятий в ВУЗе РТУ МИРЭА будет написан бот для мессенджера Discord, собирающий и предоставляющий пользователю расписание занятий для всех групп и преподавателей.

Технология автоматизации позволит структурировать информацию о расписании занятий и реализовать её вывод в более удобной форме.

Процесс автоматизации будет состоять из 2 действий приложения:

1. Формирования базы данных, хранящих расписание для учебных групп и преподавателей.
2. Предоставление пользователю информации о расписании группы или преподавателя из базы данных.

Сами действия пользователь сможет вызвать, взаимодействуя с ботом через мессенджер Discord, с помощью соответствующих запросов.

2.1.1 Формирование базы данных

Формирование базы данных, содержащей информацию о расписании занятий групп и преподавателей будет заключаться в следующих действиях бота:

1. Получения запроса от пользователя на обновление расписания, хранящейся в базе данных.

2. Посещения раздела “Расписание – РТУ МИРЭА” официальной интернет-страницы ВУЗа РТУ МИРЭА и скачивание оттуда файлов с расписанием учебных групп.
3. Сохранение файлов внутри проекта.
4. Чтение файлов и сбора информации о занятиях групп и преподавателей.
5. Сохранения собранной информации в базе данных.

2.1.2 Предоставление информации о расписании

Пользователь сможет запросить информацию о расписании учебной группы или преподавателя, отправив приложению запрос и указав названия требуемой группы.

Бот, получив команду, начнёт поиск необходимой группы или преподавателя, и найдя информацию, выведет пользователю расписание в виде сообщений внутри мессенджера.

2.2 Информационное обеспечение приложения

Информационное обеспечение бота представляет собой всю информацию о системе, её составных частях, и входную и выходную информацию.

2.2.1 Информационная модель и её описание

Первым этапом разработки бота является построение информационной модели приложения.

Информационная модель представляет собой совокупность всей информации о параметрах, свойствах и возможных состояний приложения, а также связи между ними и внешней средой (мессенджером, пользователем и т.п.).

Для информационной модели бота были построены следующие UML-диаграммы:

1. Диаграмма вариантов использования.
2. Диаграмма состояний.
3. Диаграммы последовательностей для действий, при запросе пользователя.
4. Диаграмма сущностей.
5. Контекстная диаграмма.

Диаграмма вариантов использования

Диаграмма вариантов использования показывает все возможные варианты взаимодействия пользователя с разрабатываемым приложением.

Пользователь с помощью бота имеет возможность:

- получить информацию о расписании занятий для отдельной группы или преподавателя;
- обновить расписание внутри базы данных бота;
- ознакомиться с руководством пользователя.

Диаграмма представлена на *Рисунке 2.1*.

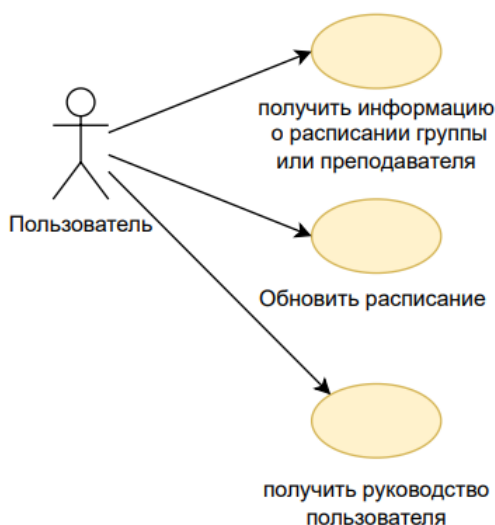


Рисунок 2.1 – Диаграмма вариантов использования разрабатываемой системы.

Диаграмма последовательностей

Диаграмма последовательностей подробно описывает отношения между пользователем и объектами системы в рамках одного из взаимодействий.

Объектами системы является сама программа-бот, а также дополнительные подпрограммы, реализующие определенное действие (например, работу с базой данных).

Все представленные диаграммы последовательностей показывают варианты взаимодействия, в которых не произошло ошибки. В случае неудачи на каком-либо этапе система возвращается в начальное состояние, а пользователю выводится сообщение о ошибке при выполнении запроса.

Запрос на получение руководства пользователя

Данная диаграмма (*Рисунок 2.2*) описывает взаимодействия пользователя и бота, при запросе на вывод руководства пользователя.

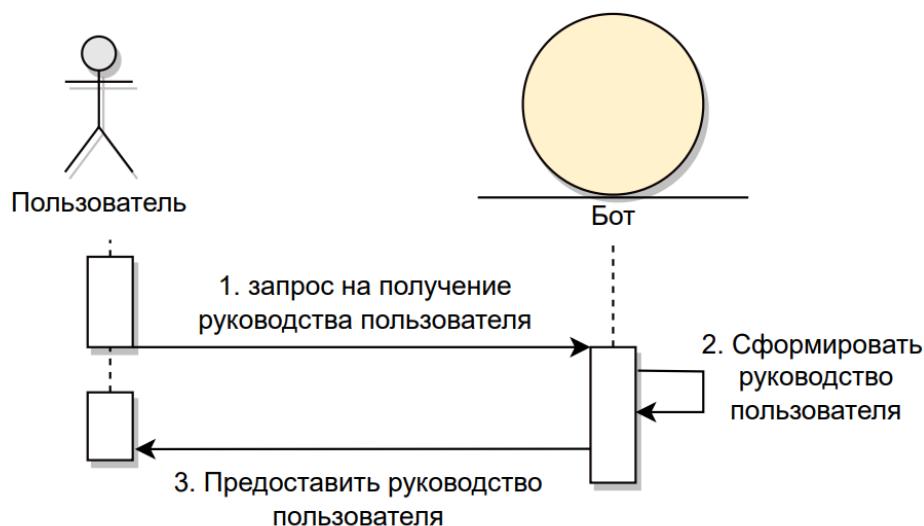


Рисунок 2.2 – Диаграмма последовательностей разрабатываемой системы при запросе на получение руководства пользователя.

Получив запрос, система формирует руководство и отображает его в мессенджере.

Запрос на обновление расписания

При таком запросе объектами внутри системы, помимо бота, будут подпрограммы, реализующие:

- получение из Интернет-ресурсов файлов, хранящих информацию о расписании занятий;
- чтение файлов с расписанием и сбору информации из них;
- взаимодействие с базой данных.

Диаграмма последовательности представлена на *Рисунке 2.3*.

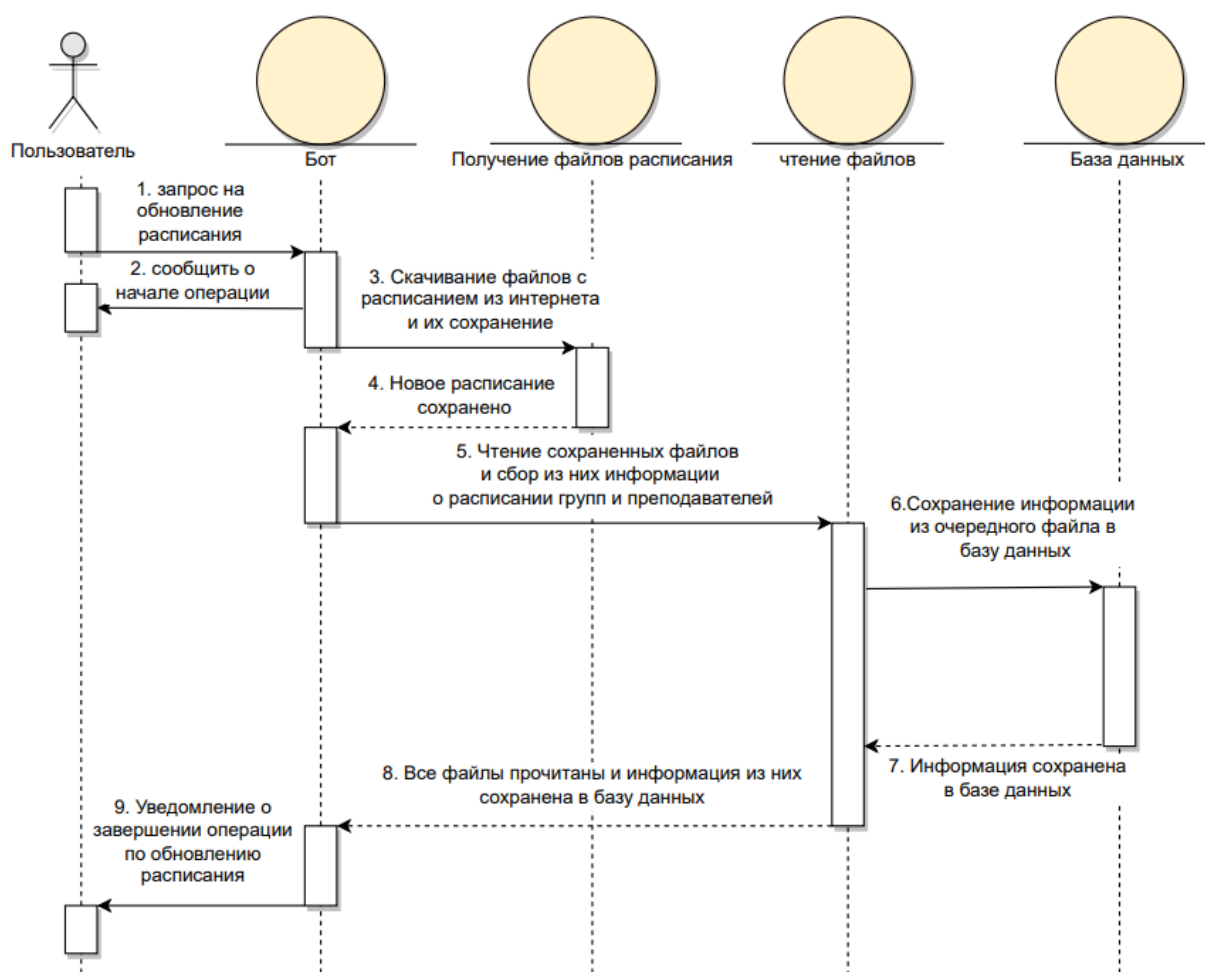


Рисунок 2.3 – Диаграмма последовательностей разрабатываемой системы при запросе на обновление расписания.

Сам процесс обновления заключается в следующем:

Бот, получив запрос, обращается к подпрограмме «получение файлов расписания», которая последовательно скачивает файлы с расписанием со страницы официального сайта РТУ МИРЭА и сохраняет их внутри проекта.

Завершив операцию, бот получает соответствующее уведомление и вызывает подпрограмму «чтение файлов», последовательно читающую скаченные файлы и собирающую из них информацию, которая затем сохраняется в базе данных.

После завершения процесса чтения файлов, бот выводит пользователю сообщение о завершении операции обновления.

Запрос на вывод расписания

При запросе на вывод расписания в качестве объектов системы выступает бот и подпрограммы по работе с базой данных и обработки входной и выходной информации для пользователя.

Диаграмма показана на *Рисунке 2.4*.

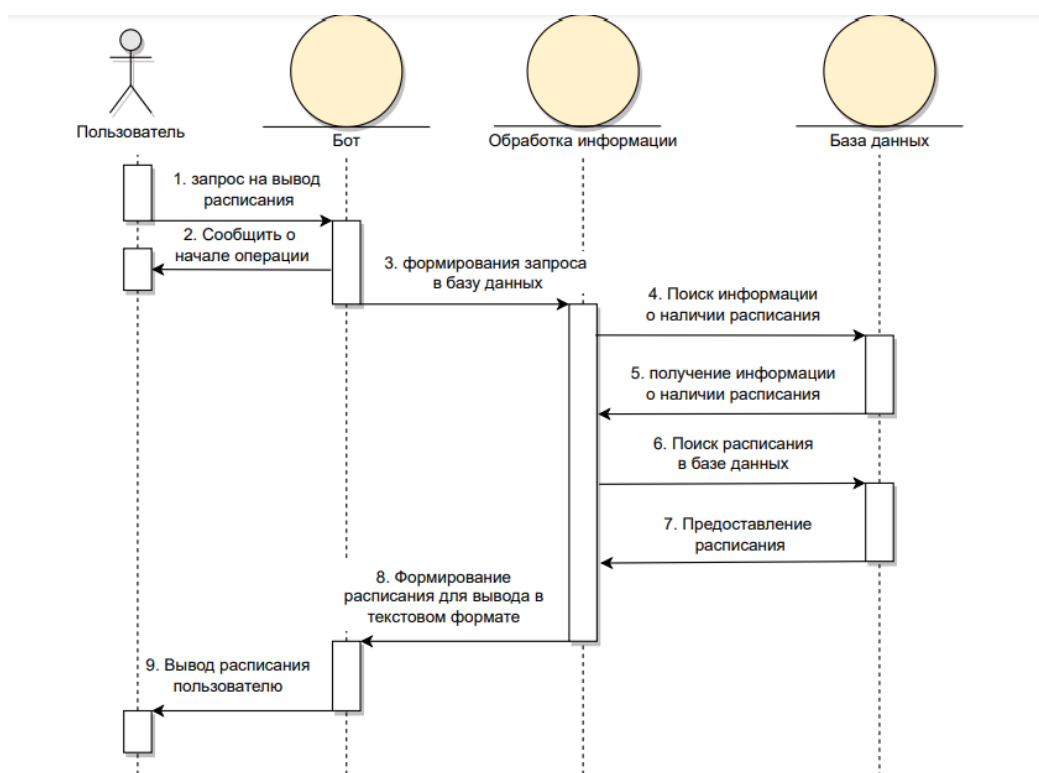


Рисунок 2.4 –Диаграмма последовательностей разрабатываемой системы при запросе на вывод расписания.

При получении запроса на вывод расписания с указанием названия группы или ФИО преподавателя, бот сообщает о начале операции и вызывает

подпрограмму «Обработка информации», отправляющая в базу данных запрос на поиск группы.

Получив информацию о наличии в базе расписания группы, подпрограмма снова обращается к базе данных для получения из неё самого расписания.

База данных передает требуемую информацию, и подпрограмма формирует на основе полученных данных расписание в текстовом формате и передает его боту, который, в свою очередь, выводит расписание пользователю.

Диаграмма состояний

Диаграмма состояний описывает все возможные состояния всей системы при взаимодействии с пользователем (Рисунок 2.5).

Внутри диаграммы в блоках указываются состояние системы, а переходы между блоками являются действиями, из-за которых состояние меняется.

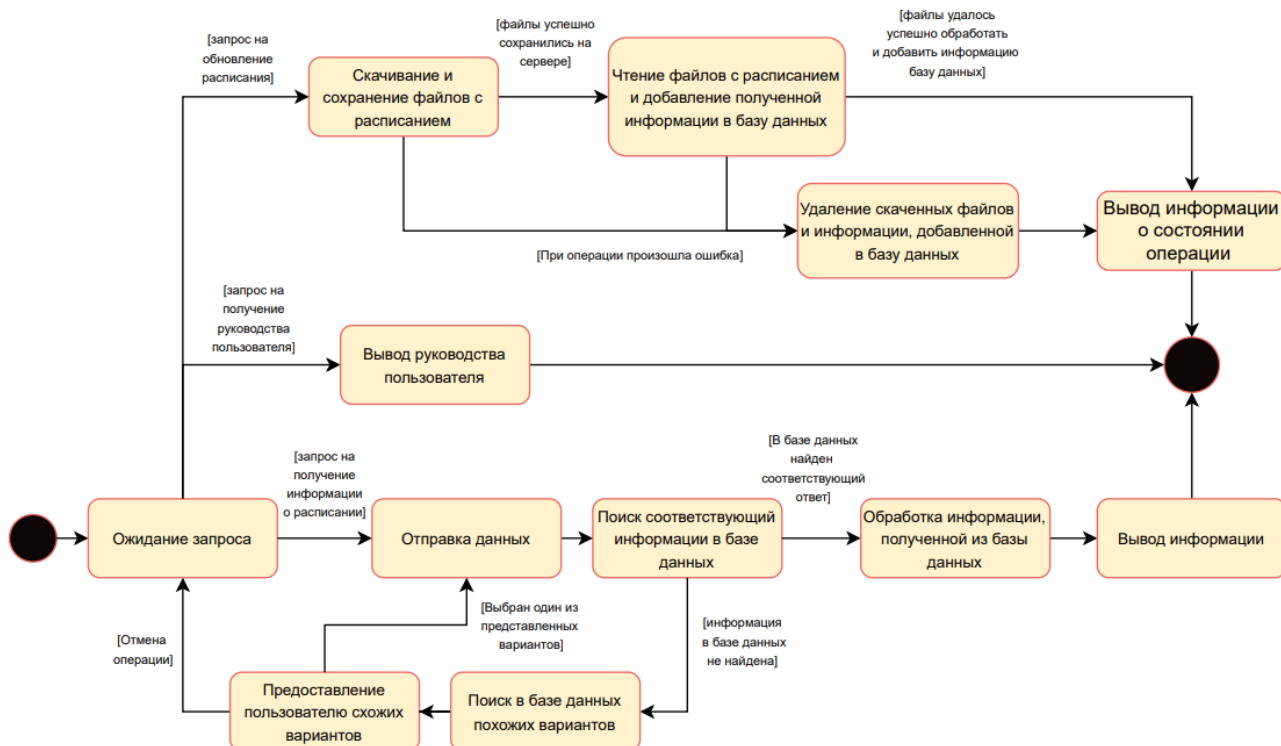


Рисунок 2.5 –Диаграмма состояний разрабатываемой системы.

Диаграмма сущностей

Данная диаграмма описывает сущности, находящиеся внутри базы данных, и их взаимосвязь с друг другом.

Под сущностями подразумевается множество абстрактных или реальных объектов, обладающих общими характеристиками и имеющие одни и те же свойства или атрибуты.

Взаимосвязь является отношением, установленным между несколькими сущностями, и бывает следующих видов: «многие ко многим», «один ко многим», «один к одному». Вид связи характеризует количественное отношение сущностей друг к другу.

На *Рисунке 2.6* представлена диаграмма сущностей, используемая в базе данных внутри приложения.

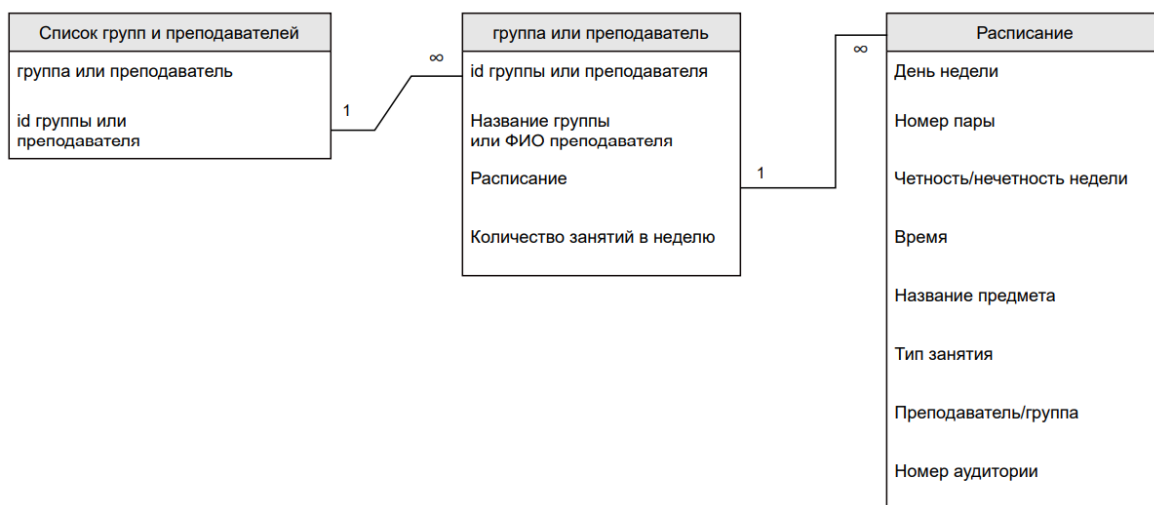


Рисунок 2.6 – Диаграмма сущностей базы данных разрабатываемой системы.

В разрабатываемой системе присутствуют 3 сущности:

1. Сущность «Список групп и преподаватель», описывающая список всех групп и преподавателей, находящихся в базе данных.
2. Сущность «Группа или преподаватель», входящая в сущность «Список групп и преподаватель» и хранящая все характеристики группы или преподавателя.
3. Сущность «Расписание», входящая в сущность «Группа или преподаватель» и описывающая информацию об одном из занятий.

Контекстная диаграмма

Контекстная диаграмма (Рисунок 2.7) отображает основную функцию разрабатываемой информационной системы – получения информации о расписании занятий.

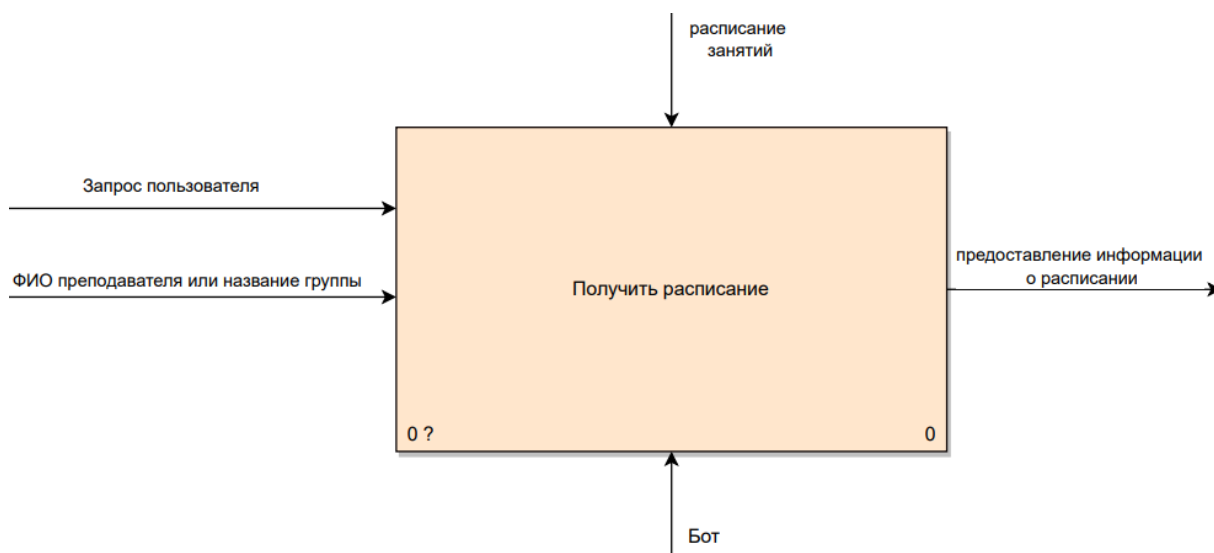


Рисунок 2.7 – Контекстная диаграмма функции получения расписания разрабатываемой системы.

2.2.2 Входная и результирующая информация

В качестве входной информации выступает один из вариантов запроса пользователя, и в случае если это запрос на вывод расписание, помимо самого запроса пользователю необходимо указать наименование группы или ФИО преподавателя, расписание которого требуется вывести.

Результативная информация зависит от варианта взаимодействия:

1. Запрос «вывод расписания» – расписание занятий.
2. Запрос «обновить расписание» – состояние после завершения операции.
3. Запрос «получить руководство пользователя» – руководство пользователя.

2.3 Программное и техническое обеспечения разработки

Перед процессом программной реализации бота необходимо выбрать все средства, которые будут задействованы в процессе разработке.

2.3.1 Программное обеспечение

Программным обеспечением при разработке будут являться среда программирования, где приложение будет создаваться, и интерфейс для взаимодействия бота и мессенджера.

Интерфейс управления ботом Discord API

Discord API является официальным веб-интерфейсом для работы и управления ботом в мессенджере Discord.

Интерфейс позволит создать для бота специальный аккаунт в мессенджере, настроить его название, внешний вид и поведение внутри “Discord”, а также обеспечит защиту приложения в случае несанкционированного вмешательства во внутреннюю структуру программы.

Среда программирования Visual Studio Code

В качестве среды программирования был выбран Visual studio code, представляющий собой редактор исходного кода для разработки, оптимизации и отладки облачных и веб-приложений.

Является продуктом компании Microsoft и позиционируется как “легкая” версия другой среды разработки этой компании – Visual Studio.

Основными плюсами данного редактора является наличие:

- отладчика для разрабатываемых приложений;
- поддержки большинства языков программирования, включая языки, на которых возможна разработка бота;

- технологии автодополнения кода, дописывающее название функций при вводе первых букв;
- широких возможностей для кастомизации приложения, включая изменение пользовательских тем, сочетания клавиш и файлов конфигурации;
- модели бесплатного распространения продукта.

С помощью Visual Studio Code возможно реализовать все требования для разрабатываемого проекта, а пользовательский интерфейс продукта позволит установить все необходимые библиотеки прямо внутри текстового редактора.

Интерфейс IDE представлен на *Рисунке 2.8*.

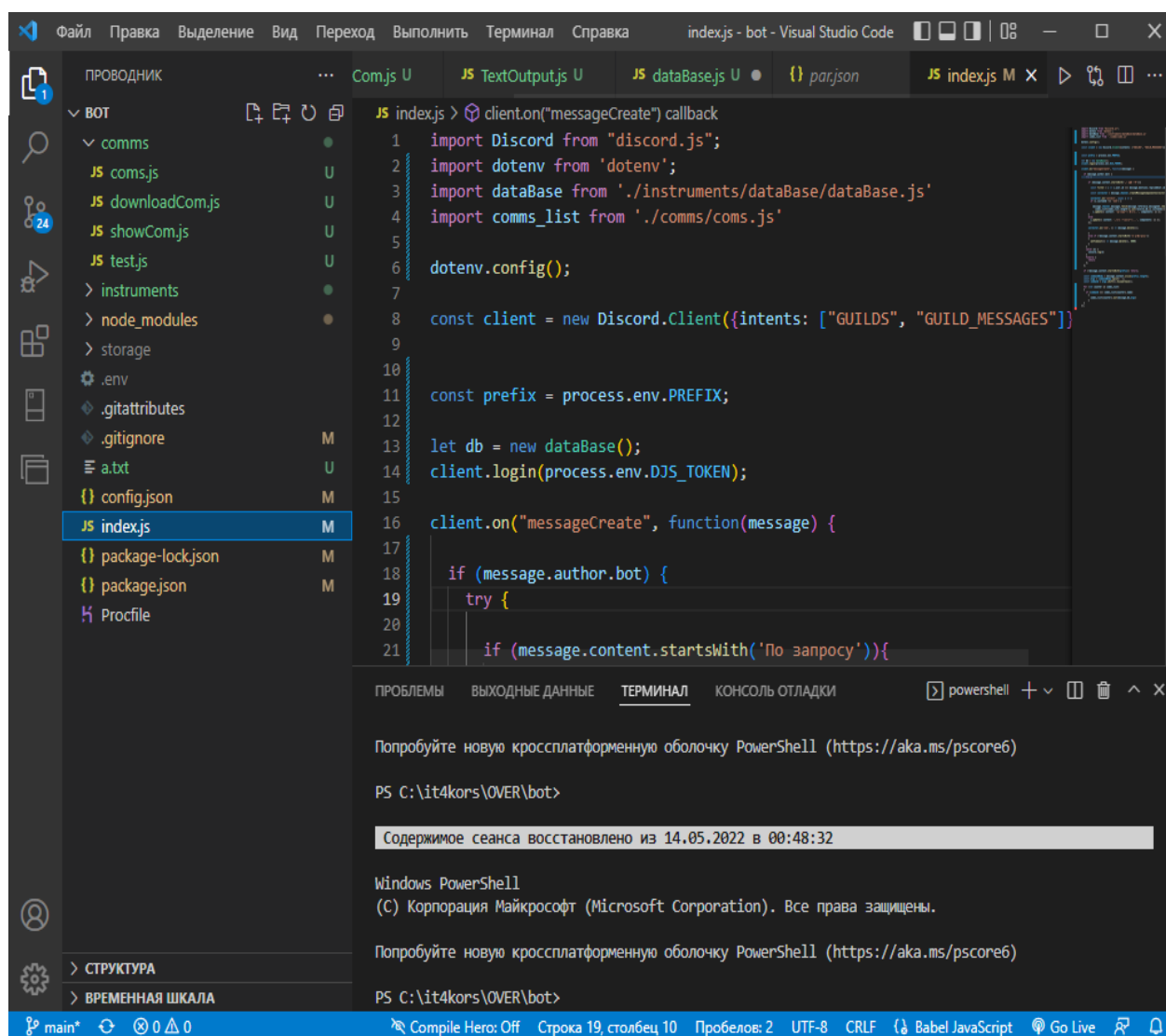


Рисунок 2.8 – Интерфейс среды разработки “Visual studio code”.

2.3.2 Техническое обеспечение

В качестве технического обеспечения для разрабатываемого приложения будет выступать сервер, на котором бот будет размещен.

Облачная платформа Heroku для размещения бота

Heroku – Облачная PaaS-платформа (Platform as a Service, «Платформа как услуга»), позволяющая удаленно размещать на своих серверах разрабатываемые приложения.

Платформа поддерживает большое количество языков программирования (Java, JavaScript, Python, PHP и другие) и баз данных (MongoDB, Redis, PostgreSQL и т.п.), и позволяет подключать любые библиотеки, поддерживаемые используемым языком программирования.

Heroku позволит:

- гибко взаимодействовать с используемыми серверами, изменяя их количество или используемую производительность;
- подключать дополнительные библиотеки для серверов, дополняя их функционал;
- отслеживать состояние приложения через официальный сайт Heroku [22] или клиент.

Также преимуществом Heroku является сотрудничество с GitHub [18] и “DropBox” [19], позволяющее автоматически экспортировать оттуда разрабатываемые проекты на сервера Heroku.

3 ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

В технологическом разделе будут выбраны технологии для разработки бота (язык программирования и база данных) и описана сама реализация приложения: клиентской и серверной частей, базы данных, подключение к облачным платформам для размещения бота.

Также будет произведено тестирование приложения и составлено руководство для пользователя.

3.1 Обоснование выбора средств разработки приложения

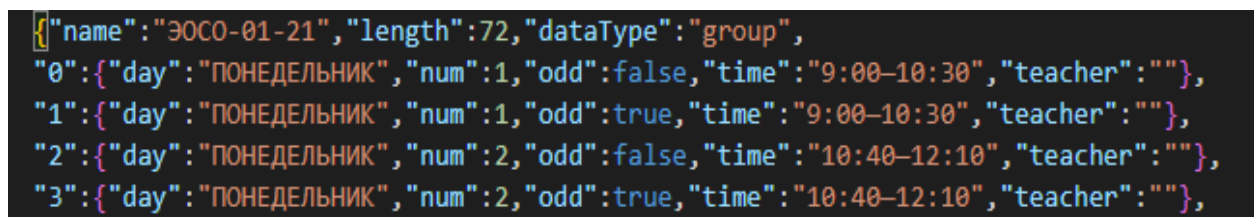
На основе программного и технического обеспечения разработки, необходимо выбрать технологии и методики, позволяющие выполнить поставленные задачи.

3.1.1 База данных

Для выполнения поставленных задач у бота должна быть база данных для хранения информации. Такую базу данных можно создать, используя облачные сервисы (например – MongoDB) или создать локальное хранилище файлов.

В данной работе будет выбран вариант по использованию локальной базы данных для создания более устойчивой и автономной системы.

Локальное хранилище будет представлено как текстовый файл формата “json”, в котором должны храниться JSON-объекты. Пример такого объекта представлен на *Рисунке 3.1*.



```
{
  "name": "ЭОСО-01-21",
  "length": 72,
  "dataType": "group",
  "0": {
    "day": "ПОНЕДЕЛЬНИК",
    "num": 1,
    "odd": false,
    "time": "9:00-10:30",
    "teacher": ""
  },
  "1": {
    "day": "ПОНЕДЕЛЬНИК",
    "num": 1,
    "odd": true,
    "time": "9:00-10:30",
    "teacher": ""
  },
  "2": {
    "day": "ПОНЕДЕЛЬНИК",
    "num": 2,
    "odd": false,
    "time": "10:40-12:10",
    "teacher": ""
  },
  "3": {
    "day": "ПОНЕДЕЛЬНИК",
    "num": 2,
    "odd": true,
    "time": "10:40-12:10",
    "teacher": ""
  }
}
```

Рисунок 3.1 – JSON-файл, хранящий JSON-объект с расписанием занятий учебной группы ЭОСО-01-21.

Информация в JSON-объекте хранится в формате набора пар ключей и их значений. При этом в JSON-файлах поддерживается использование различных типов данных в качестве значений, что позволяет применять вложенность (значения в объекте могут представлять собой вложенные объекты или массивы с собственными значениями).

Такой способ хранения данных поддерживается любым современным языком программирования, и позволяет быстро получать и изменять информацию в базе данных.

3.1.2 Язык программирования JavaScript

Библиотеки для работы с интерфейсом разработчика Discord API, согласно официальной документации [9], существуют для таких языков программирования как: C#, Dart, Go, Java, JavaScript, Lua, PHP, Python, Rubby, Rust.

Существенной разницы в выборе языка при разработке бота не имеется, поэтому был для проекта был выбран язык JavaScript.

JavaScript представляет собой высокоуровневый объектно-ориентированный язык программирования, совмещающий в себе также ряд свойств из функциональных языков (замыкания, анонимные функции и т.п.), что придает языку дополнительные возможности и гибкость при разработки и отладки приложения.

Также в JavaScript имеется динамическая типизация и автоматическое управление памятью, что упрощает работу при написании кода и оптимизирует скорость функционирования приложения.

Помимо архитектурных преимуществ язык имеет крупную документацию на русском языке [11], которая позволит ускорить процесс изучения языка. Кроме документации также будет использована дополнительная литература [12].

3.1.3 Программная платформа “Node.js”

Node.js – платформа с открытым исходным кодом для разработки приложений на языке программирования JavaScript, позволяющая разработчику писать код серверной части программы.

Преимуществом платформы является:

- огромное количество поддерживаемых библиотек для использования разработки;
- универсальность, позволяющая разрабатывать клиентскую и серверную части программы на JavaScript;
- скорость, достигаемая использованием преимуществами архитектуры JavaScript.

3.2 Описание реализации приложения

Реализация приложения состоит из следующих этапов:

1. регистрации бота в мессенджере Discord;
2. создании клиентской и серверной частей приложения;
3. разработка базы данных;
4. размещение бота на облачном сервисе.

3.2.1 Регистрация бота

В первую очередь необходимо зарегистрировать бота на специальном интерфейсе для разработки чат-ботов – Discord API, расположенного на официальном сайте мессенджера [13].

Для этого достаточно зайти на сайт, нажать на кнопку “New Application” и ввести название бота. После этого откроется меню разработчика, позволяющее настраивать поведение и внешний вид бота в мессенджере (*Рисунок 3.2*).

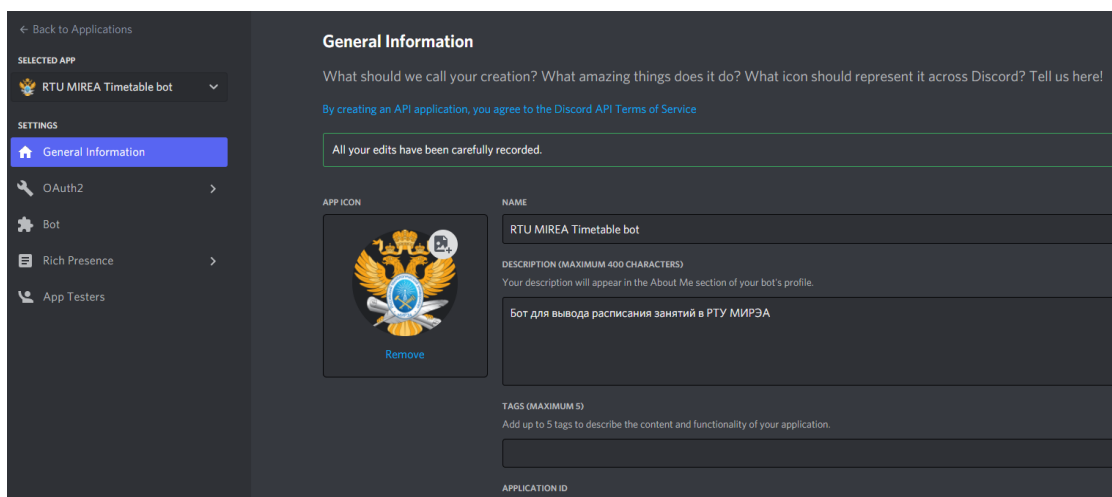


Рисунок 3.2 – Интерфейс разрабатываемого бота в Discord API

Помимо управления ботом, Discord API присваивает аккаунту уникальный идентификатор – токен, позволяющий разработчику связать своё приложение с интерфейсом Discord API и программно настроить поведение бота.

Листинг программной части подключения к мессенджеру Discord представлен в *Приложении Д.1*.

3.2.2 Создание клиентской части бота

Пользователи Discord могут взаимодействовать с ботом с помощью команд (Например, «!обновить»). Перед командой вводится обязательный префикс, позволяющий приложению понять, что сообщение является запросом, который боту необходимо обработать.

В качестве префикса может выступать любой символ или слово, поддерживаемые стандартом «Юникод».

В данной работе таким символом будет «!».

В разрабатываемом приложении будет три команды:

- обновить расписание («!обновить»). Данная команда обновляет расписание занятий, хранимое в базе данных;
- показать расписание («!показать “параметр”»), где в качестве параметра указывается название учебной группы или преподавателя, расписание которого бот попытается найти и вывести в виде сообщения. В

случае если найти расписание не получится, бот предложит пользователю расписание похожих групп (*Рисунок 3.3*), похожие группы будут предложены в виде сообщения с кодом, листинг, которого представлен в *Приложении Д.2*;

– получить руководство пользователя («!справка»). При её вводе бот напишет пользователю личное сообщение в мессенджере, содержащее в себе руководство по использованию приложения.

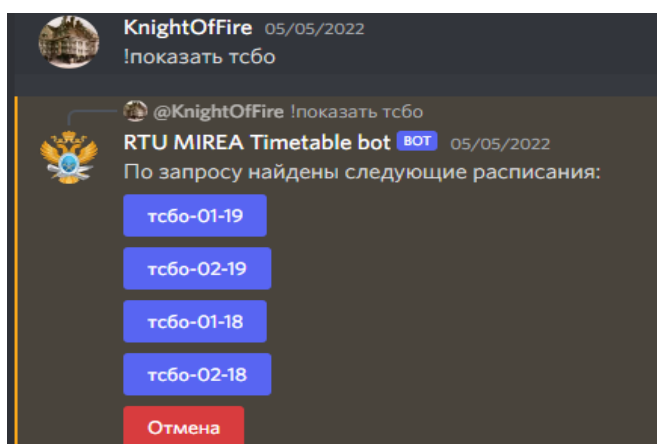


Рисунок 3.3 – Предоставление ботом похожих вариантов запрашиваемого расписания

Все сообщения, выводимые программой, кроме сообщения с расписанием занятий, будут автоматически удаляться через 15 секунд после вывода. Это сделано, чтобы не засорять пользовательское окно с чатом в мессенджере.

Блок-схема алгоритма по работе клиентской части представлена в *Приложении А*.

3.2.3 Реализация серверной части бота

Серверная часть разрабатываемого бота затрагивает обработку следующих запросов:

- обновление расписания;
- вывод расписания.

Выполнение запроса на обновление расписания

Процесс операции обновления расписания состоит из следующих этапов:

1. Скачивание файлов с расписанием.
2. Сбор информации о расписании занятий учебных групп и преподавателей из полученных файлов.
3. Сохранение собранной информации в базе данных.

Блок-схемы алгоритмов по обновлению информации о расписании представлены в *Приложениях Б*.

Получение расписания с официального сайта РТУ МИРЭА

В первую очередь бот получает исходный код веб-страницы с расписанием занятий в РТУ МИРЭА [15], в котором будет происходить поиск гиперссылок на скачивание файлов формата “.xlsx” (формат файлов программы Excel, в котором хранится расписание занятий).

На *Рисунке 3.4* представлен фрагмент исходного кода веб-страницы, содержащий в себе пример ссылки для скачивания расписания.

```
<a class="uk-link-toggle" href="https://webservices.mirea.ru/upload/iblock/cac/99baousulysqdy9xwz4sae9z15d39oke/ИИИ_2_курс_21-22_весна_очка.xlsx"
k"> == $0
<div class="uk-card uk-card-default uk-card-body uk-card-small uk-text-center uk-box-shadow-hover-small">
  
  <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 384 512" class="iconInLink_xbig uk-link-heading uk-svg" width="40" height="40">...</svg>
  <div class="uk-link-heading uk-margin-small-top"> 2 курс </div>
</div>
</a>
```

Рисунок 3.4 – Фрагмент исходного кода сайта расписания РТУ МИРЭА, содержащий файл с расписанием 2 курса института искусственного интеллекта, весеннего семестра 2021/2022 учебного года.

Получив гиперссылки на файлы расписания, Бот скачает расписание и сохранит его внутри проекта.

Листинг кода, который ищет в исходном коде гиперссылки для скачивания представлен в *Приложении Д.3*.

Обработка файлов excel

После скачивания Excel-файлов с расписанием из сети-интернет, бот начнет чтение информации внутри файлов.

На *Рисунке 3.5* представлен графическое изображение алгоритма чтения файла с расписанием.

День недели	Группа				ИВБО-13-21			
	№ пары	Нач. занятий	Оконч. занятий	Неделя	Предмет	Вид занятий	ФИО преподавателя	№ ауд.
ПОНЕДЕЛЬНИК	1	9-00	10-30	I				
				II				
	2	10-40	12-10	I				
				II				
	3	12-40	14-10	I	Схемотехника устройств компьютерных систем	пр	Пономарев А.Н.	Г-112
				II				
ВТОРНИК	4	14-20	15-50	I	Схемотехника устройств компьютерных систем	лаб	Люляева Д.В.	Г-111-6
				II	Схемотехника устройств компьютерных систем	лаб	Люляева Д.В.	Г-111-6
	5	16-20	17-50	I				
				II	2-10 н. Разработка предметно-ориентированных языков программирования	пр	Семенов Р.Э.	Г-109-а
	6	18-00	19-30	I				
				II				
СРЕДА	1	9-00	10-30	I	Математическая логика и теория алгоритмов	лк	Воронцов А.А. с ИВБО-08	А-9
				II	Математическая логика и теория алгоритмов	лк	Воронцов А.А.	А-9
	2	10-40	12-10	I	Проектирование баз данных	лк	Богомольная Г.В. с ИВБО-20	А-17
				II	Проектирование баз данных	лк	Богомольная Г.В.	А-17
				I	Схемотехника устройств компьютерных систем	лк	Смирнов Н.А. с ИВБО-01-20	А-9

Рисунок 3.5 – Графическое изображение алгоритма чтения файла.

Сам алгоритм чтения заключается в следующем:

1. Определении структуры занятий: количества дней в неделю, количеству пар и их времени проведения.
2. Поиску столбцов, содержащих название учебных групп.
3. Сформировав структуру и найдя учебные группы, происходит сбор информации о расписании занятий учебной группы и её сохранению в качестве отдельных переменных.

4. При сборе информации о занятиях группы также создаются отдельные переменные для преподавателей, куда заносятся проводимые ими занятия,
5. Отправке в базу данных переменных, содержащих информацию о занятиях преподавателей и учебных групп.
6. Отправке на клиентскую часть результата операции.

Выполнение запроса на получение информации о расписании

При запросе на вывод расписания, приложение проверит, существует ли в базе данных группа или преподаватель, требуемая пользователем. Удостоверившись, что информация в базе данных существует, бот получит информацию о её расписании занятий.

Полученную информацию приложение отформатирует в формат “string” (строку) и выведет как сообщение с текстом (Рисунок 3.6).

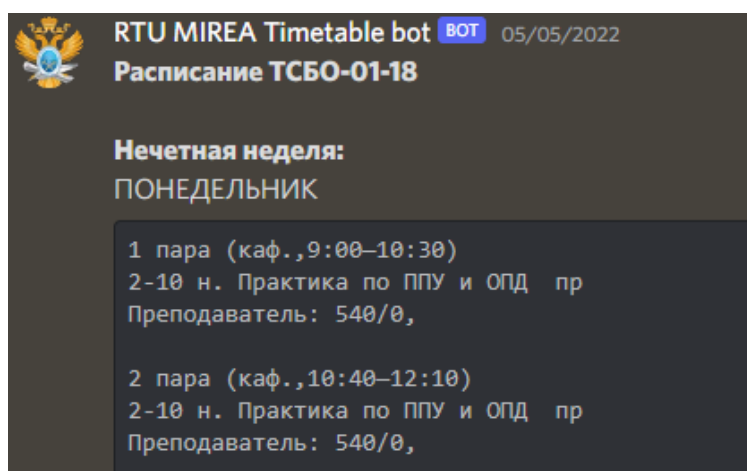


Рисунок 3.6 – Часть вывода ботом расписания группы ТСБО-01-18

При этом, в мессенджере Discord есть ограничение на количество выводимых символов – максимальная длина сообщения равна 2000 символов, поэтому если расписание не вмещается в одно сообщение, оно будет разбито на несколько отдельных сообщений.

В ситуации, когда найти группу или преподавателя в расписании не удалось, бот попытается найти до 4 похожих запросов и отправит их в клиентскую часть для формирования сообщения с выбором для пользователя.

Дождавшись, когда пользователь выберет один из вариантов, приложение повторит операцию для этого варианта.

Блок-схема алгоритмов по выводу расписания пользователю расположена в *Приложениях В*.

3.2.4 Разработка базы данных бота

База данных представляет собой файлы формата “JSON”, которые подразделяются на следующие категории:

- “par.json” – файл, хранящий параметры о состоянии базы данных (дата последнего обновления, число скаченных из интернета «excel-файлов», количество файлов с занятиями групп и преподавателей);
- “names.json” – файл, хранящий информацию о группах и преподавателях, расписания которых хранятся в базе данных и пути к их файлам;
- В папке information находятся «JSON-файлы», каждый из которых хранит отдельное расписание занятий для одной из групп или преподавателей;
- В папке files хранятся «JSON-файлы», являющиеся расписаниями занятий, скаченными из интернета и имеющие формат “xlsx”.

Блок-схемы алгоритмов по работе с базой данных находится в *Приложениях Г*.

3.2.5 Размещение бота на удаленном сервере Heroku

Для постоянного функционирования бота, его следует разместить на удаленном сервере. Для этого будет использована облачная PaaS-платформа “Heroku”.

Размещение возможно тремя способами:

- используя “Heroku CLI”;

- подключение проекта из аккаунта в GitHub с автоматическим развертыванием;
- подключение проекта из Dropbox с автоматическим развертыванием.

В данном проекте будет использован вариант с использованием аккаунта в “Github”, так как Heroku автоматически подключит проект из GitHub к своей платформе (Рисунок 3.7) и останется только запустить проект.

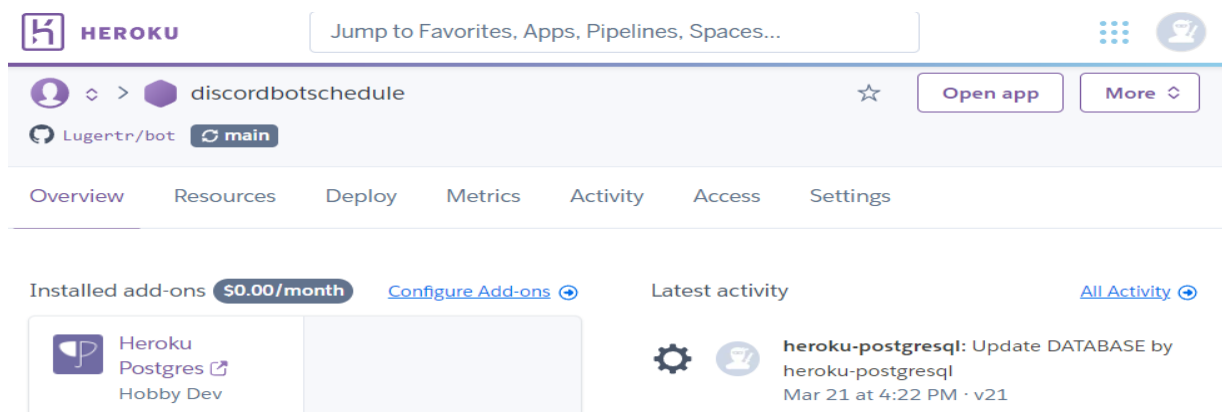


Рисунок 3.7 – Проект бота на платформе Heroku

Запущенный проект станет онлайн в мессенджере (Рисунок 3.8) и будет работать в режиме 24/7 на любом сервере.



Рисунок 3.8 – Включенный бот в мессенджере Discord

3.3 Тестирование приложения

Перед тестированием бота необходимо оценить количество ошибок в приложении с помощью модели Миллса.

Модель Миллса заключается в том, что в программе существует N естественных ошибок, которые не были обнаружены, и чтобы их найти, в программу добавляют M искусственных ошибок (Формула 3.1):

$$C = \begin{cases} 1, & \text{при } n > N \\ \frac{M}{M+N+1}, & \text{при } n \leq N \end{cases} \quad (3.1)$$

где C — мера доверия;

N — количество неизвестных ошибок в программе;

M — количество добавленных искусственных ошибок в программе;

n — количество найденных естественных ошибок в программе.

Количество ненайденных ошибок рассчитывается по *Формуле 3.2*:

$$K = N - n, \quad (3.2)$$

где K — Количество ненайденных ошибок в программе;

N — количество неизвестных ошибок в программе;

n — количество найденных естественных ошибок в программе.

Для проведения испытаний в приложение будет внесено 7 ошибок.

3.3.1 Результат тестирования

Тестирование приложения будет проходить через интерфейс мессенджера Discord.

В первую очередь необходимо проверить работоспособность команды «!обновить», обновляющей расписание занятий внутри базы данных.

Как видно из *Рисунка 3.9* расписание успешно обновилось.

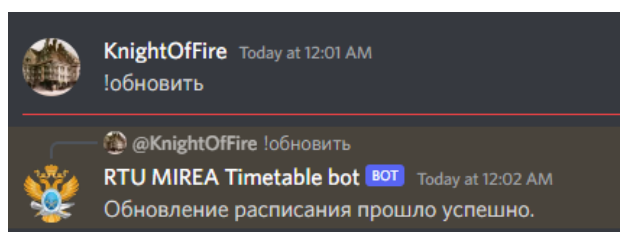


Рисунок 3.9 – Ответ бота на команду «!обновить»

После обновления расписания проверим команду «показать (параметр)», где в качестве параметра укажем одну из учебных групп – «ИББО-01-19» (Рисунок 3.10).



Рисунок 3.10 – Часть ответа бота на команду «!показать ИББО-01-19» с выводом расписания группы ИББО-01-19

Проверка на вывод расписания преподавателя представлена на Рисунке 3.11.

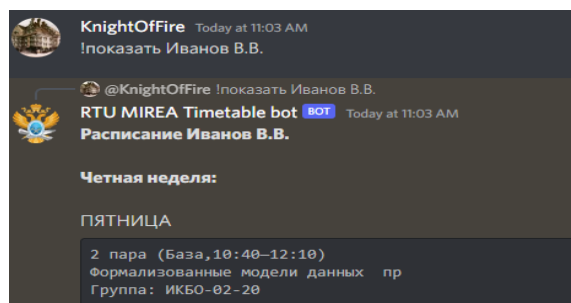


Рисунок 3.11 – Часть ответа бота на команду «!показать Иванов В.В.» с выводом расписания преподавателя Иванова В.В.

Введем несуществующую группу, чтобы проверить работоспособность бота по поиску похожих групп в базе данных (Рисунок 3.12).

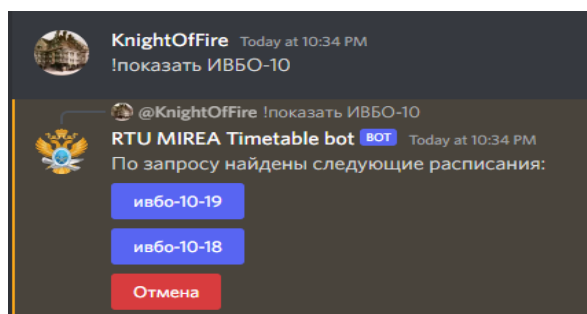


Рисунок 3.12 – Ответ бота на команду «!показать ИВБО-10» с выводом выбора для пользователя

Выберем одного из предложенных ботом вариантов (*Рисунок 3.13*).

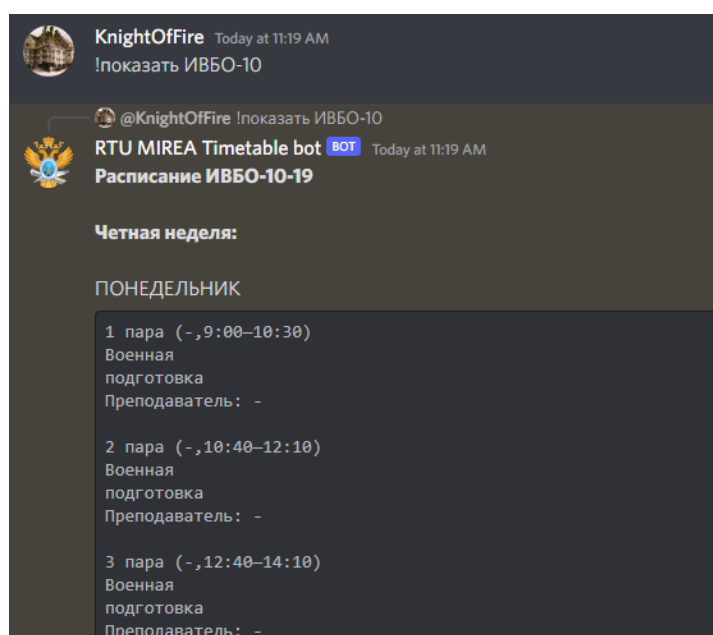


Рисунок 3.13 – Часть ответа бота на выбор варианта ИВБО-10-19 с выводом расписания группы ИВБО-10-19

Попробуем отправить боту сразу 2 запроса на обновление расписания (*Рисунок 3.14*).

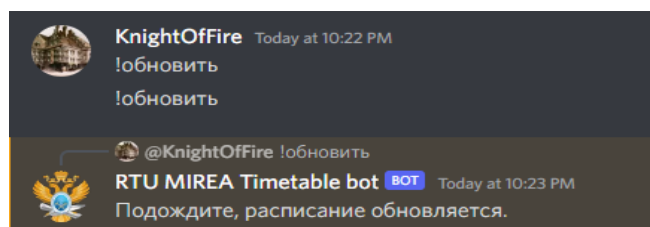


Рисунок 3.14 – Ответ бота на запрос во время обновления расписания

Ответ бота на запрос руководства пользователя изображен на *Рисунке 3.15*.

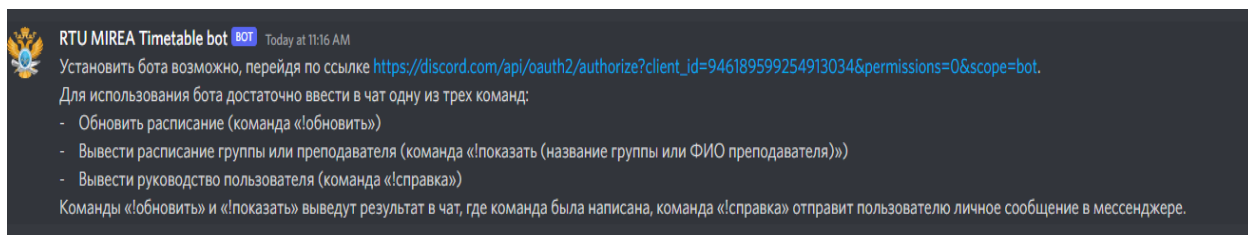


Рисунок 3.15 – Ответ бота на команду «!справка»

В результате тестирования программы было найдено:

- одна естественная ошибка;
- шесть искусственных ошибок.

Тогда коэффициент доверия по модели Миллса будет равен 0,7.

Количество обнаруженных ошибок будет составлять 1.

3.4 Инструкция по работе пользователя с приложением

Установить бота возможно с помощью специальной ссылки [25], перейдя по которой, пользователь сможет добавить приложение на один из своих серверов.

Для использования бота достаточно ввести в чат одну из трех команд:

- обновить расписание (команда «!обновить»);
- вывести расписание группы или преподавателя (команда «!показать (название группы или ФИО преподавателя)»);
- вывести руководство пользователя (команда «!справка»).

Команды «!обновить» и «!показать» выведут результат в чат, где команда была написана, команда «!справка» отправит пользователю личное сообщение в мессенджере.

4 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

Финальным результатом выполнения выпускной квалификационной работы является создание бота в мессенджере “Discord” для автоматизированного вывода расписания занятий в ВУЗе РТУ МИРЭА.

Одним из неотъемлемых этапов создания приложения будет являться расчёт экономических аспектов разработки.

Основными задачами экономической части являются:

- организация и планирования работ по теме;
- расчёт стоимости проведения работ по теме.

4.1 Организация и планирование работ по теме

В процессе разработки приложения задействовано трое человек:

1. Руководитель (Платонова О.В., доцент, к.т.н., зав. каф. ВТ) – отвечает за грамотную постановку задачи, контролирует отдельные этапы работы, вносит необходимые коррективы и оценивает выполненную работу в целом.
2. Консультант (Кондратова С.В., доцент, к.э.н.) – отвечает за консультирование экономической части выпускной квалификационной работы.
3. Разработчик (Студент Вишневский Н.А., группа ИВБО-01-18) – отвечает за реализацию всех поставленных задач, включая тестирование и отладку готового продукта, и подготовку проектной документации.

Состав участников, задействованных в выполняемой работе, представлен на *Рисунке 4.1*.

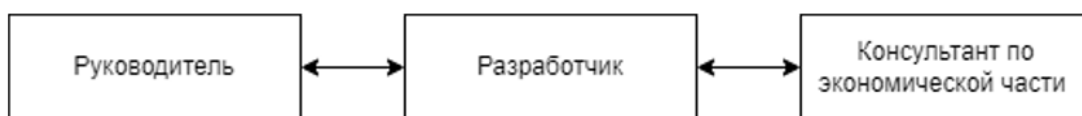


Рисунок 4.1 – Схема состава участников работы

4.1.1 Организация работ

На разработку отводится 90 рабочих дней.

Этапы разработки представлены в *Таблице 4.1*.

Таблица 4.1 Этапы разработки выпускной квалификационной работы

№	Название этапа	Исполнитель	Трудоемкость, чел/дни	Продолжительность работ, дни
1	Разработка и согласование технического задания	Руководитель	5	5
		Разработчик	5	
2	Технико-экономическое обоснование разработки	Руководитель	7	7
		Консультант	1	
		Разработчик	7	
3	Создание информационной модели системы	Руководитель	5	14
		Консультант	1	
		Разработчик	14	
4	Определение технических средств при разработки	Руководитель	3	7
		Консультант	1	
		Разработчик	7	
5	Разработка общего описания алгоритма функционирования	Руководитель	2	10
		Разработчик	10	
6	Программирование приложения	Разработчик	24	24
7	Отладка приложения	Разработчик	4	4
8	Корректировка программы по результатам испытаний	Разработчик	5	5
9	Подготовка технической документации приложения	Консультант	1	7
		Разработчик	7	
10	Сдача готового продукта и его внедрение	Руководитель	2	7
		Консультант	1	
		Разработчик	7	

4.1.2 График проведения работ

Общий срок разработки составляет 90 дней.

На *Рисунке 4.2* изображен календарный график по исполнению работы за весь срок разработки.

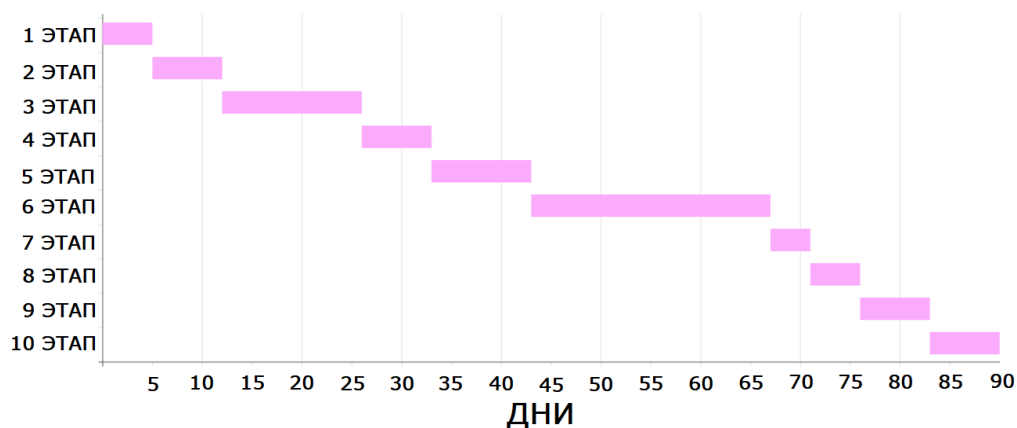


Рисунок 4.2 – график проведения работ

4.2 Расчёт стоимости проведения работ по теме

Стоимость проведения работ включает в себя расчёт:

- полной себестоимости проекта;
- Договорной цены для дальнейшей реализации проекта.

Все расчёты будут проводиться в соответствии с методическими указаниями [22] и учебно-методическими пособиями [23] ВУЗа РТУ МИРЭА.

При этом весь объем затрат на научно-исследовательские и опытно-конструкторские работы будет проведен с помощью метода калькулирования.

4.2.1 Расчёты себестоимости

Для расчёта полной себестоимости при выполнении выпускной квалификационной работы, необходимо посчитать следующие статьи расходов:

1. Материалы, покупные изделия или полуфабрикаты.
2. Специальное оборудование.
3. Основная заработная плата.

4. Дополнительная заработная плата.
5. Страховые отчисления.
6. Командировочные расходы.
7. Контрагентские услуги.
8. Накладные расходы.
9. Прочие расходы.

1 статья «Материалы, покупные изделия и полуфабрикаты».

Эти расходы состоят из стоимости всех комплектующих изделий, полуфабрикатах и других материальных продуктов, затраченных при научно-исследовательских и опытно-конструкторских работах проекта.

Таблица с данными расходами представлена в *Таблице 4.2*.

Таблица 4.2 – расчёты статьи расходов «Материалы, покупные изделия и полуфабрикаты».

№	Наименование материалов	Единицы измерения	Количество	Цена за единицу, рублей	Стоимость, рублей
1	Ручка	шт	10	25	250
2	Карандаш	шт	10	10	100
3	Бумага А4	пачка, 500 листов	1	550	550
4	Картридж для принтера	шт	1	2990	2950
5	USB-Флеш-накопитель 4 ГБ	шт	1	650	650
Итоговая стоимость материалов					4500
Транспортно-заготовительные расходы					920
Итоговая стоимость					5420

2 статья «Специальное оборудование»

Данная статья расходов подразумевает все затраты на покупку нестандартного оборудования, которые в данной работе отсутствуют.

3 статья «Основная заработная плата»

В Таблице 4.3 представлены расчёты основной заработной платы всех участников, задействованных в работе.

Таблица 4.3 – Расчёты основной заработной платы

№	Название этапа	Исполнитель	Месяч- ный оклад, рублей	Трудоем- кость, чел/дни	Оплата за день, руб	Оплата за этап, руб
1	Разработка и согласование технического задания	Руководитель	50000	5	2273	11365
		Разработчик	27000	5	1227	6135
2	Технико-экономическое обоснование разработки	Руководитель	50000	7	2273	15911
		Консультант	45000	1	2045	2045
		Разработчик	27000	7	1227	8589
3	Создание информационной модели системы	Руководитель	50000	5	2273	11365
		Консультант	45000	1	2045	2045
		Разработчик	27000	14	1227	17178
4	Определение технических средств при разработки	Руководитель	50000	3	2273	6819
		Консультант	45000	1	2045	2045
		Разработчик	27000	7	1227	8589
5	Разработка общего описания алгоритма функционирования	Руководитель	50000	2	2273	4546
		Разработчик	27000	10	1227	12270
6	Программирование приложения	Разработчик	27000	24	1227	29448
7	Отладка приложения	Разработчик	27000	4	1227	4908
8	Корректировка программы по результатам испытаний	Разработчик	27000	5	1227	6135

Продолжение Таблицы 4.3

9	Подготовка технической документации приложения	Консультант	45000	1	2045	2045
		Разработчик	27000	7	1227	8589
10	Сдача готового продукта и его внедрение	Руководитель	50000	2	2273	4546
Итоговая сумма						164573

4 статья «Дополнительная заработная плата»

Эта статья включает в себя выплаты, предусмотренные законодательством о труде за неотработанное по уважительным причинам время.

В данном проекте дополнительная заработная плата научного и производственного персонала (ДЗП) представляет собой 20% от основной заработной платы и составляет:

$$\text{ДЗП} = 164573 \times 20\% = 164573 \times 0.2 = 32914,6 \text{ руб.} \quad (4.1)$$

5 статья «Страховые отчисления»

Страховые отчисления составляют 30% от фонда оплаты труда (ФОТ), состоящий из основной (ОЗП) и дополнительной (ДЗП) заработной платы:

$$\text{ФОТ} = \text{ОЗП} + \text{ДЗП} = 164573 + 32914,6 = 197487,6 \text{ руб.,} \quad (4.2)$$

Тогда страховые отчисления (СВ) будут равняться:

$$\text{СВ} = \text{ФОТ} \times 30\% = 197487,6 \times 0,30 = 59246,3 \text{ руб.} \quad (4.3)$$

6 статья «Командировочные расходы»

Командировочные расходы научного и производственного персонала при разработке проекта отсутствуют.

7 статья «Контрагентские услуги»

При разработке проекта услуги сторонних организаций не использовались.

8 статья «Накладные расходы»

Накладные расходы (НР) заключаются в тратах на содержание и ремонт оборудования, инвентаря, помещения или т.п. И в разрабатываемом проекте составляют 200% от основной заработной платы (ОЗП):

$$НР = ОЗП \times 200\% = 164573 \times 2,0 = 329146 \text{ руб.} \quad (4.4)$$

9 статья «Прочие расходы»

Какие-либо прочие расходы в проекте отсутствуют.

Полная себестоимость приложения

В Таблице 4.4 представлена полная себестоимость проекта.

Таблица 4.4 – Таблица полной себестоимости проекта

№	Номенклатура статей расходов	Затраты, руб.
1	Материалы, покупные изделия и полуфабрикаты	5420
2	Специальное оборудование для научных работ	0
3	Основная заработная плата научного и производственного персонала	164573
4	Дополнительная заработная плата научного и производственного персонала	32914,6

Продолжение Таблица 4.4

5	Страховые отчисления в социальные фонды	59246,3
6	Командировочные расходы научного и производственного персонала	0
7	Контрагентские услуги	0
8	Накладные расходы	329146
9	Прочие прямые расходы	0
Итоговая сумма		591299,9

4.2.2 Расчёты договорной цены

Разрабатываемый продукт в дальнейшем должен будет реализован, поэтому необходимо подсчитать его договорную цену, представляю собой сумму полной себестоимости продукта, прибыли и налога на добавочную стоимость.

Для начала необходимо найти прибыль (П), составляющую 30% от себестоимости продукта (С):

$$П = С \times 30\% = 591299,9 \times 0.30 = 177390 \text{ руб.}, \quad (4.6)$$

Далее необходимо найти налог на добавочную стоимость (НДС):

$$\begin{aligned} \text{НДС} &= (С + П) \times 20\% = (591299,9 + 177390) \times 0.2 = \\ &= 153738 \text{ руб.}, \end{aligned} \quad (4.7)$$

Тогда договорная цена готового продукта (ПР) будет равняться:

$$\begin{aligned} \text{ПР} &= С + П + \text{НДС} = 591299,9 + 177390 + 153738 = \\ &= 922427.9 \text{ руб.} \end{aligned} \quad (4.8)$$

ЗАКЛЮЧЕНИЕ

В результате выпускной квалификационной работы был разработан авторский Discord-бот для автоматизированного вывода расписания занятий в ВУЗе РТУ МИРЭА, реализующий следующие функции:

- сбору и сохранению в базе данных информации о расписании занятий учебных групп и преподавателей ВУЗа РТУ МИРЭА;
- предоставлению пользователю информации о занятиях из базы данных приложения.

Для разработанного бота была составлена информационная модель системы, спланирован процесс разработки и выбрано техническое и программное обеспечение приложения.

Были получены навыки по программированию на языке “JavaScript” с использованием платформы “Node.js”, включая подключение и работу со сторонними библиотеками, взаимодействие с веб-страницами, файлами, excel-книгами и реализация базы данных в виде файловой системы;

Был приобретен опыт работы с сервисами “Discord API”, “Heroku” и “GitHub”.

CONCLUSION

As a result of the final qualifying work, the author's Discord bot was developed for automated output of the schedule of classes at the University of RTU MIREA, which implements the following functions:

- collecting and storing in the database information about the schedule of classes of study groups and teachers of the University of RTU MIREA;
- providing the user with information about classes from the application database.

An information model of the system was compiled for the developed bot, the development process was planned and the technical and software applications were selected.

Skills in programming in the “JavaScript” language were obtained using the “Node” platform.js”, including connecting and working with third-party libraries, interacting with web pages, files, excel books, and implementing a database as a file system;

The experience of working with the services “Discord API”, “Heroku” and “GitHub” was acquired.

СПИСОК ЛИТЕРАТУРЫ

1. Официальный сайт мессенджера Discord [электронный ресурс] / Главная страница сайта мессенджера Discord: <https://discord.com/>
2. Официальный сайт компании Сбербанк [электронный ресурс] / виртуальные ассистенты Салют: https://www.sberbank.ru/ru/person/dist_services/salut_bank.
3. Сайт по поиску ботов в Discord [электронный ресурс] / Музыкальные боты для Discord: <https://top.gg/tag/music>.
4. Сайт интернет-магазина DNS [электронный ресурс]: <https://www.dns-shop.ru/>
5. Официальный сайт ВЦИОМ [электронный ресурс] / Опрос ВЦИОМ Каждому возрасту - свои сети от 12 февраля 2018: <https://wciom.ru/analytical-reviews/analiticheskii-obzor/kazhdomu-vozhrastu-svoi-seti>
6. Официальный сайт ВЦИОМ [электронный ресурс] / Опрос ВЦИОМ Интернет: новая эра мобильных устройств от 19 мая 2015 года: <https://wciom.ru/analytical-reviews/analiticheskii-obzor/internet-novaya-era-mobilnykh-ustrojstv>
7. Официальный сайт РТУ МИРЭА [электронный ресурс] / Приказ РТУ МИРЭА о дополнительных мерах по борьбе с новой коронавирусной инфекцией: <https://www.mirea.ru/upload/medialibrary/bda/314.pdf>
8. Официальный сайт мессенджера Discord [электронный ресурс] / Системные требования к ОС у мессенджера Discord: <https://support.discord.com/hc/ru/articles/213491697-%D0%9A%D0%B0%D0%BA%D0%BE%D0%B2%D1%8B-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D1%8B%D0%B5-%D0%B8-%D1%82%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D0%BA-%D0%9E%D0%A1-%D1%83-Discord->
9. Официальный сайт мессенджера Discord [электронный ресурс] /

Официальные библиотеки для использования платформы Discord API:
<https://discord.com/developers/docs/topics/community-resources>

10. Официальный сайт мессенджера Discord [электронный ресурс] / Документация Discord для разработчика: <https://discord.com/developers/docs>

11. Официальный сайт компании Mozilla [электронный ресурс] / Документация языка программирования Javascript: <https://developer.mozilla.org/ru/docs/Web/JavaScript>

12. JavaScript. Полное руководство / Флэнаган Дэвид. – Диалектика-Вильямс, 2021 – 720 с.

13. Официальный сайт интерфейса Discord API [электронный ресурс]: Главная страница: <https://discordapp.com/developers>

14. Официальный сайт мессенджера Discord [электронный ресурс]: Официальная документация для работы в Discord: <https://discord.com/developers/docs/intro>

15. Официальный сайт РТУ МИРЭА [электронный ресурс] / Страница расписания занятий в РТУ МИРЭА: <https://www.mirea.ru/schedule/>

16. Официальный сайт РТУ МИРЭА [электронный ресурс] / Главная страница: <https://www.mirea.ru/>

17. Официальный сайт Node Package Manager [электронный ресурс] / Блог одного из разработчиков Node Package Manager о преодолении отметки 1.3 миллиона зарегистрированных пакетов: <https://blog.npmjs.org/post/615388323067854848/so-long-and-thanks-for-all-the-packages>

18. Сайт сервиса Github [электронный ресурс] / главная страница сайта: <https://github.com/>

19. Сайт сервиса Dropbox [электронный ресурс] / главная страница сайта: <https://www.dropbox.com/>

20. Сайт сервиса Github [электронный ресурс] / страница разрабатываемого проекта: <https://github.com/Lugertr/bot>

21. Сайт сервиса Heroku [электронный ресурс] / главная страница: <https://www.heroku.com/>
22. Методические рекомендации по выполнению организационно-экономической части выпускных квалификационных работ [Электронный ресурс]: метод. указания / Т. Ю. Гавриленко, О. В. Григоренко, Е. К. Ткаченко. — М.: РТУ МИРЭА, 2019. — Электрон. опт. диск (ISO)
23. Экономика предприятия [Электронный ресурс]: учебно-методическое пособие / И.А. Назарова, А.С. Вихрова. — М.: РТУ МИРЭА, 2021. — Электрон. опт. диск (ISO). — 71 с.
24. Григоренко О.В., Садовнича И.О., Мыльникова А. Экономика предприятия и управление организацией М.: РУСАЙНС, 2017-235с.
25. Официальный сайт мессенджера Discord [электронный ресурс]: Страница приглашения разрабатываемого бота на сервер: https://discord.com/api/oauth2/authorize?client_id=946189599254913034&permissions=0&scope=bot
26. Матвеева Н. Ю., Технологии создания и применения чат-ботов [Электронный ресурс] / Н. Ю. Матвеева, А. В Золотарюк. // Научные записки 39 молодых исследователей. — 2018. — №1. — с. 28-30. — Режим доступа: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov>
27. Официальный сайт мессенджера Node.JS [электронный ресурс] / Документация по платформе NodeJS: <https://nodejs.org/ru/docs/>

ПРИЛОЖЕНИЯ

Приложение А — Блок-схема поведения бота в мессенджере и блок-схема вывода руководства пользователя

Приложение Б.1 — Блок-схема команды “обновить”

Приложение Б.2 — Блок-схема алгоритма обновления расписания

Приложение Б.3 — Блок-схема алгоритма скачивания и сохранения файлов с расписанием

Приложение Б.4 — Блок-схема алгоритма чтения скаченных файлов с расписанием

Приложение Б.5 — Блок-схема алгоритма по чтению одного из файлов с расписанием и отправка полученной информации в базу данных

Приложение Б.6 — Блок-схема алгоритма по подготовке к отправке собранной из “excel-файла” информации в базу данных

Приложение В.1 — Блок-схема алгоритма команды “показать”, выводящей расписание

Приложение В.2 — Блок-схема алгоритма для последовательного выполнения операции по выводу расписания

Приложение В.3 — Блок-схема алгоритма взаимодействия приложения с базой данных при поиске расписания группы или преподавателя

Приложение В.4 — Блок-схема алгоритма форматирования информации из базы данных в текстовый формат для вывода в виде сообщения

Приложение В.5 — Блок-схема алгоритма форматирования занятий расписания в текстовый формат для вывода в виде сообщения

Приложение Г.1 — Блок-схема алгоритма инициализации базы данных

Приложение Г.2 — Блок-схема алгоритма очистки базы данных

Приложение Г.3 — Блок-схема алгоритма базы данных по чтению файла с расписанием занятий группы или преподавателя

Приложение Г.4 — Блок-схема алгоритма базы данных по поиску схожей информации требуемого запроса

Приложение Г.5 — Блок-схема алгоритма сохранения информации о расписании занятий в базе данных

Приложение Д.1 — Листинг кода обработки события нажатия на кнопку

Приложение Д.2 — Листинг кода чтения исходной страницы сайта

Приложение Д.3 — Листинг кода обработки события нажатия на кнопку

Приложение А

Блок-схема клиентской части приложения

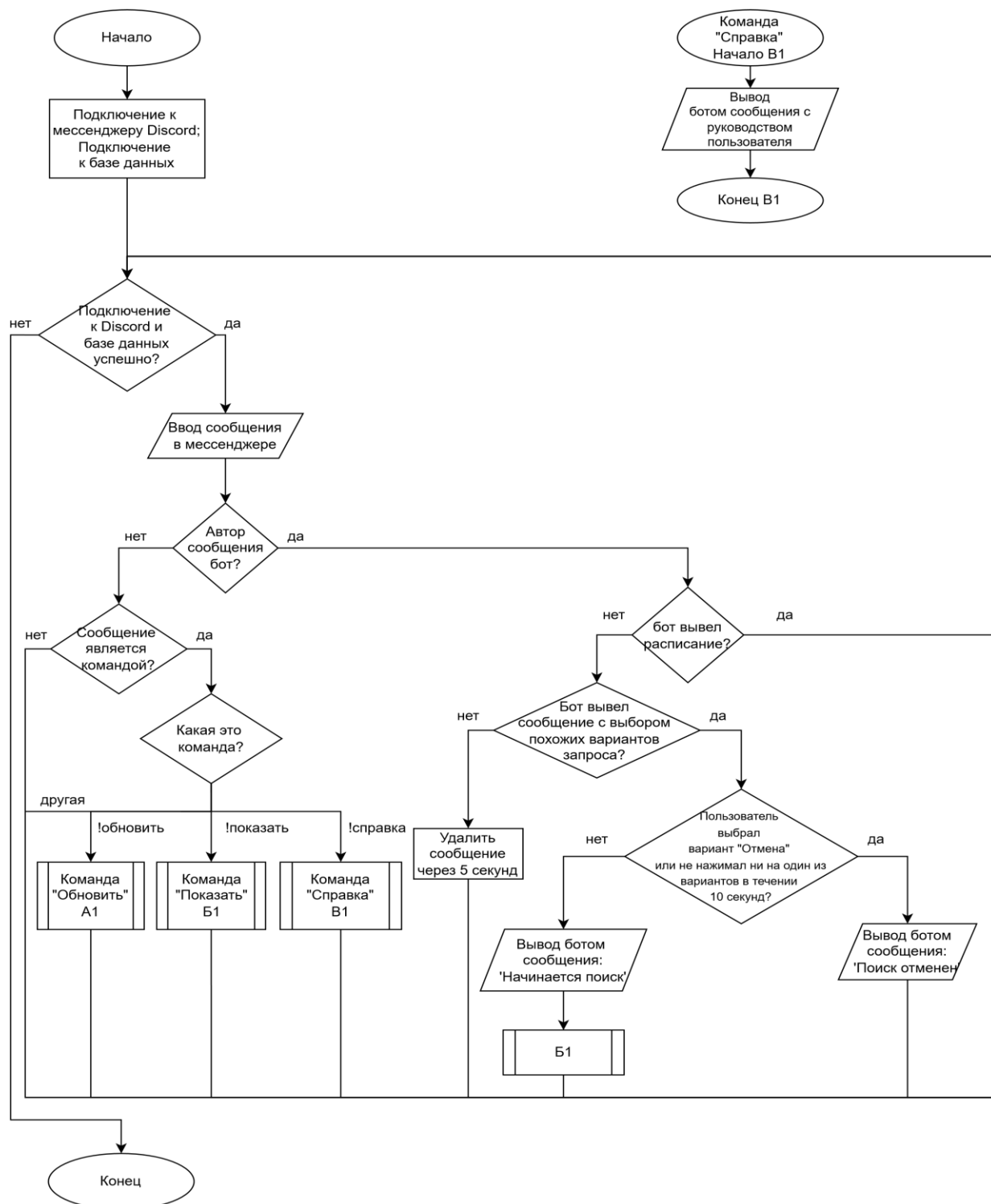


Рисунок А – Блок-схема поведения бота в мессенджере Discord и блок-схема вывода руководства пользователя

Приложение Б.1

Блок-схема команды “обновить”

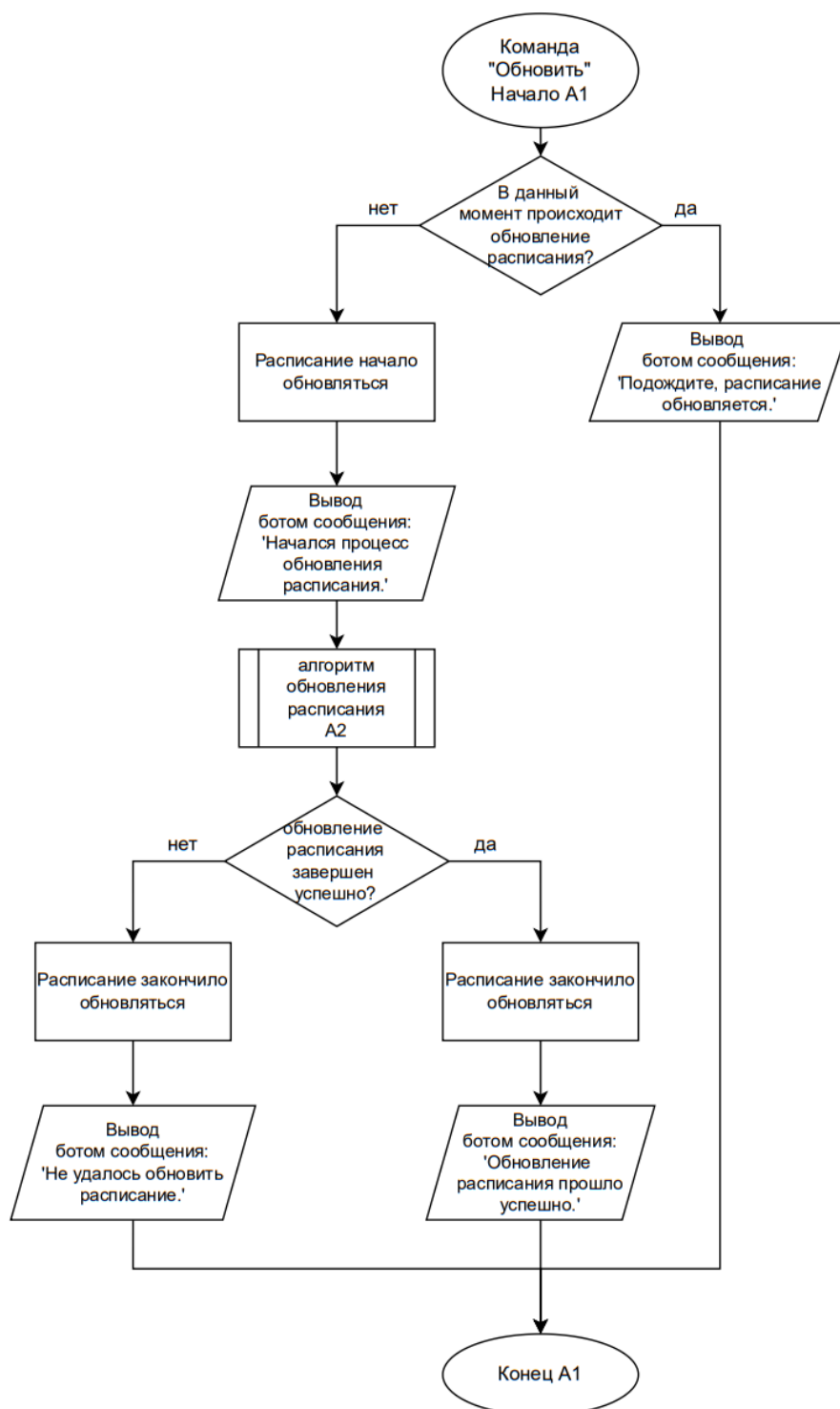


Рисунок Б.1 –Блок-схема команды “обновить”

Приложение Б.2

Блок-схема алгоритма обновления расписания

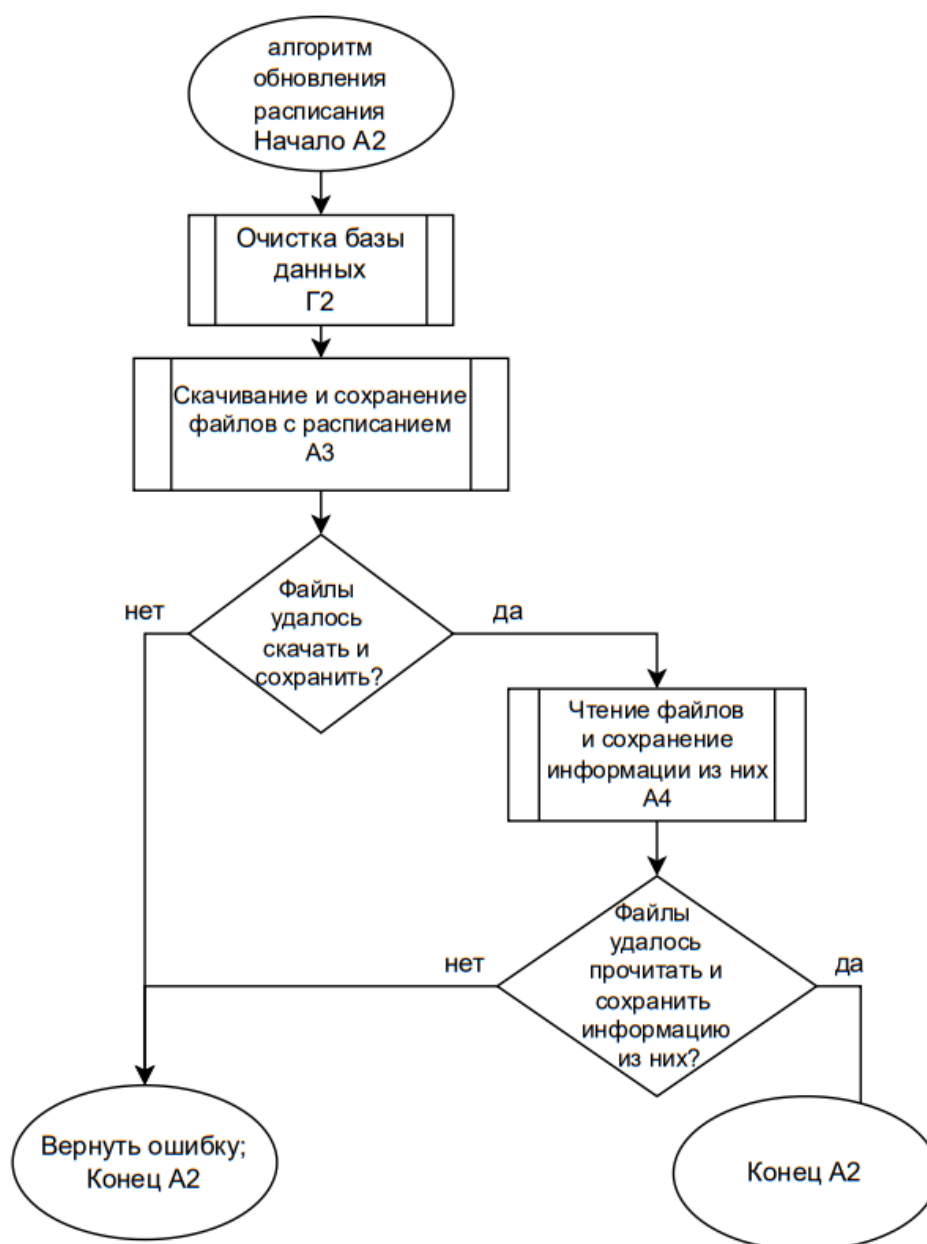


Рисунок Б.2 –Блок-схема алгоритма обновления расписания

Приложение Б.3

Блок-схема алгоритма скачивания и сохранения файлов с расписанием

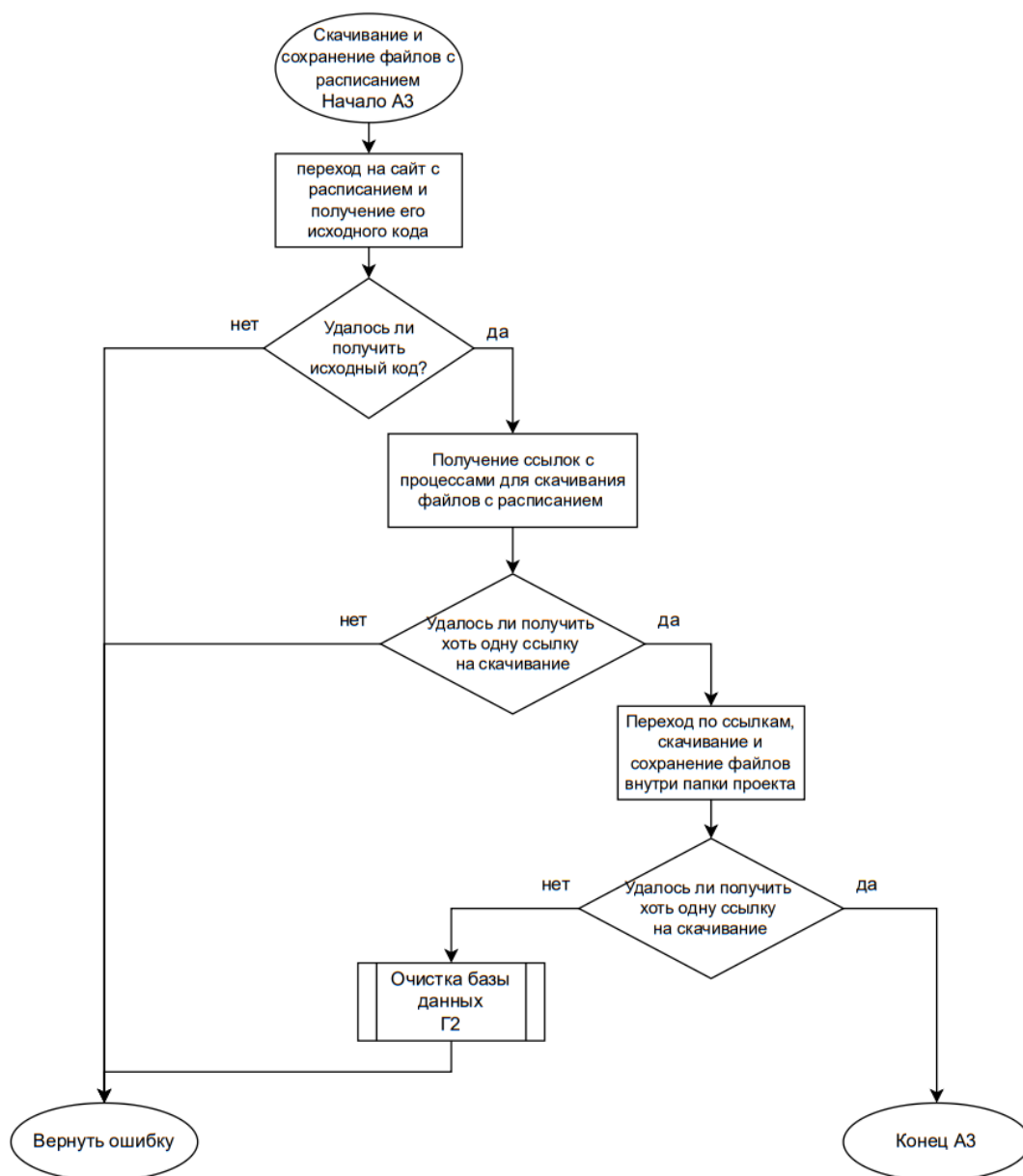


Рисунок Б.3 – Блок-схема алгоритма скачивания и сохранения файлов с расписанием

Приложение Б.4

Блок-схема алгоритма чтения скаченных файлов с расписанием

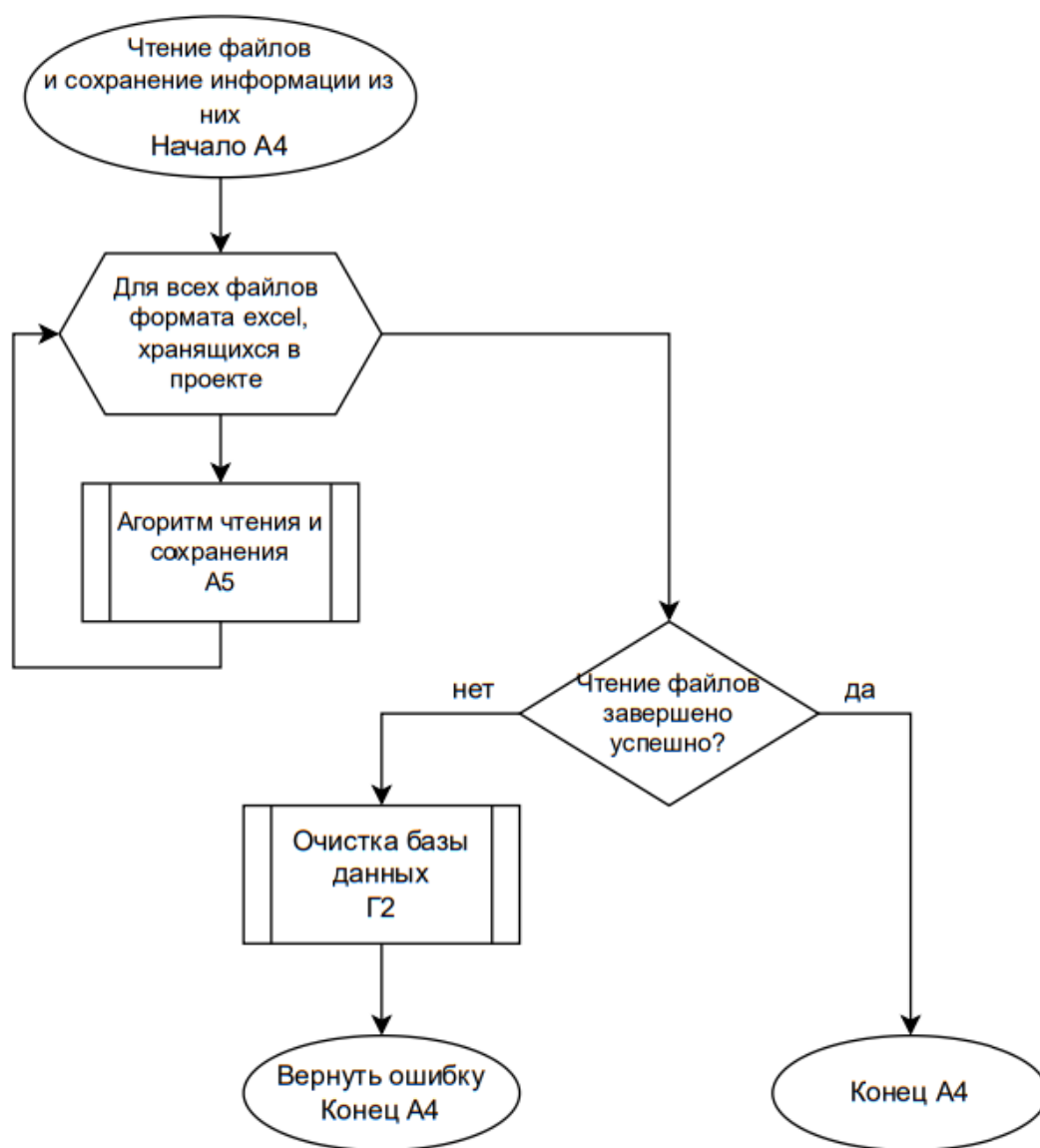


Рисунок Б.4 – Блок-схема алгоритма чтения скаченных файлов с расписанием

Приложение Б.5

Блок-схема алгоритма по чтению одного из файлов с расписанием и
отправка полученной информации в базу данных

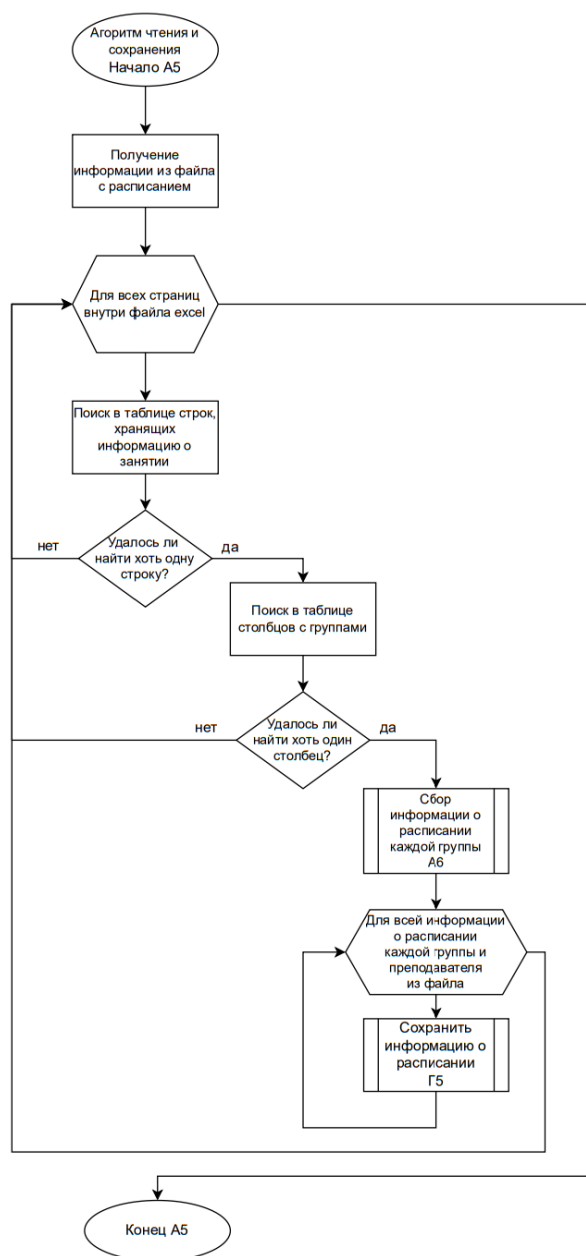


Рисунок Б.5 – Блок-схема алгоритма по чтению одного из файлов с расписанием и
отправка полученной информации в базу данных

Приложение Б.6

Блок-схема алгоритма по подготовке к отправке собранной из “excel-файла” информации в базу данных

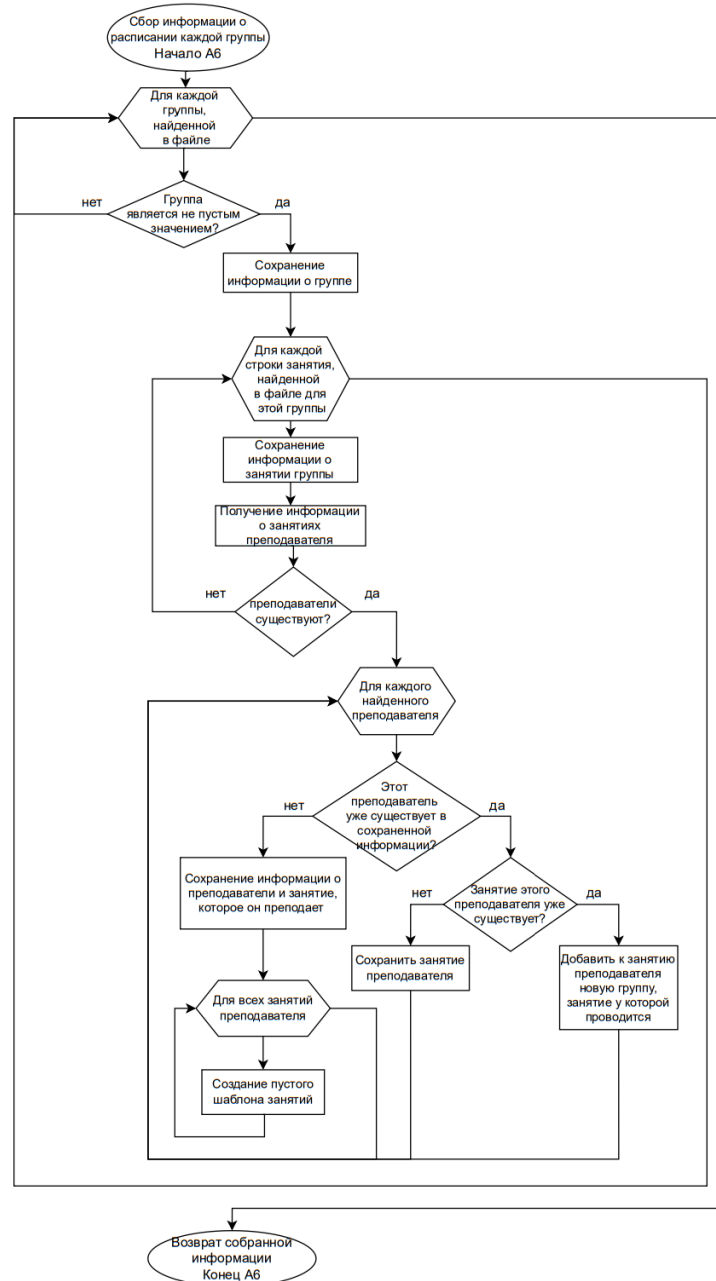


Рисунок Б.6 – Блок-схема алгоритма по подготовке к отправке собранной из “excel-файла” информации в базу данных

Приложение В.1

Блок-схема алгоритма команды “показать”, выводящей расписание

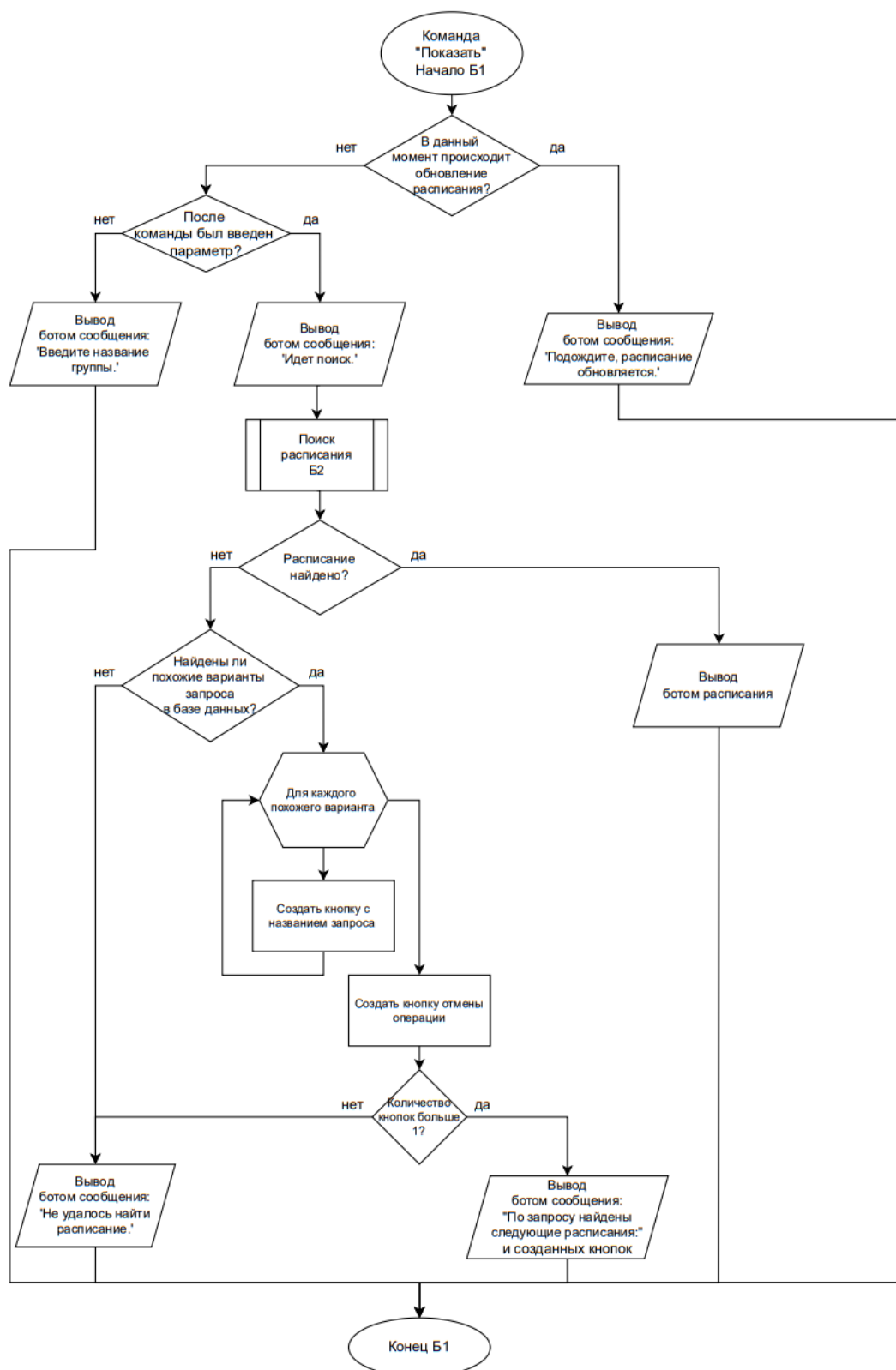


Рисунок В.1 – Блок-схема алгоритма команды “показать”, выводящей расписание

Приложение В.2

Блок-схема алгоритма для последовательного выполнения операции по
выводу расписания

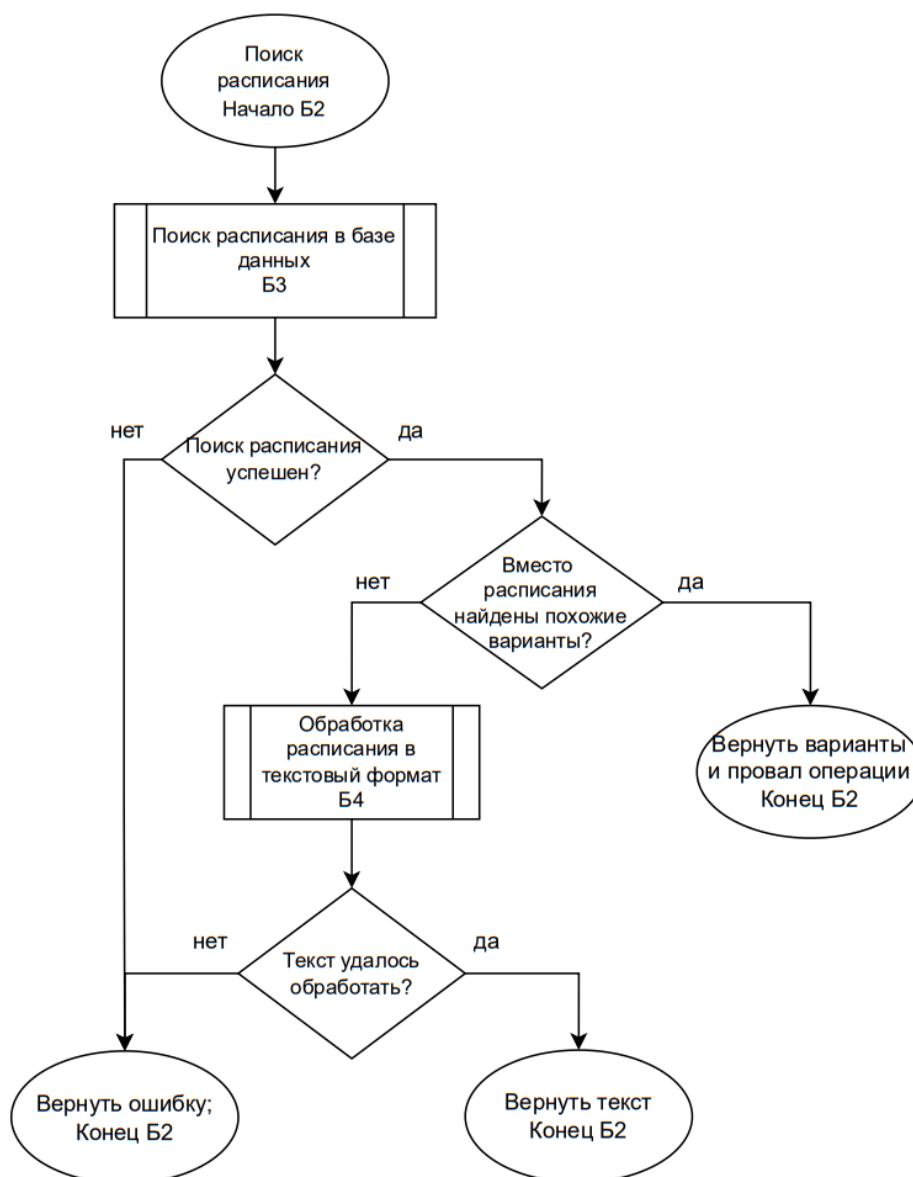


Рисунок В.2 – Блок-схема алгоритма для последовательного выполнения операции по
выводу расписания

Приложение В.3

Блок-схема алгоритма взаимодействия приложения с базой данных при поиске расписания группы или преподавателя

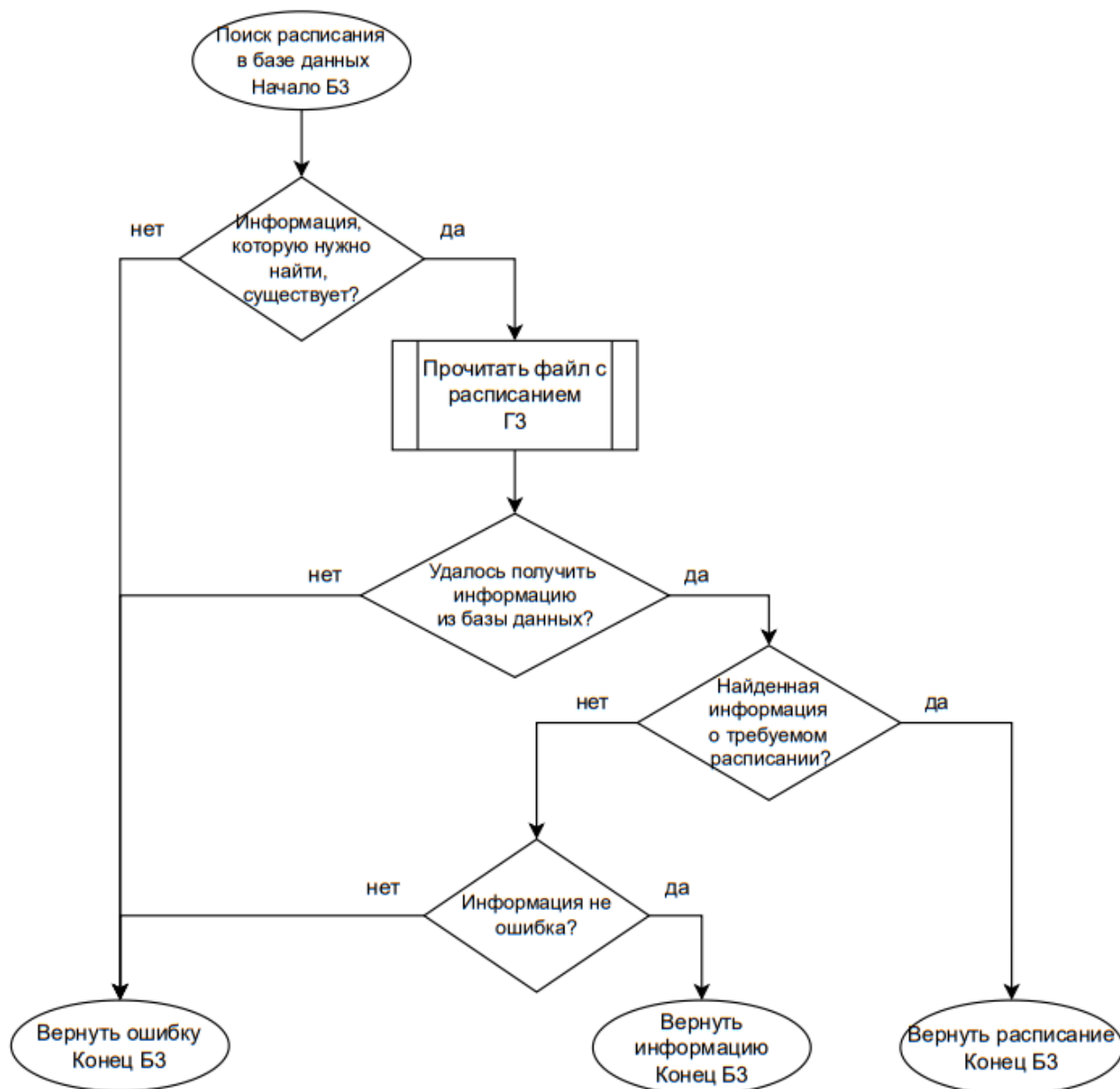


Рисунок В.3 – Блок-схема алгоритма взаимодействия приложения с базой данных при поиске расписания группы или преподавателя

Приложение В.4

Блок-схема алгоритма форматирования информации из базы данных в текстовый формат для вывода в виде сообщения

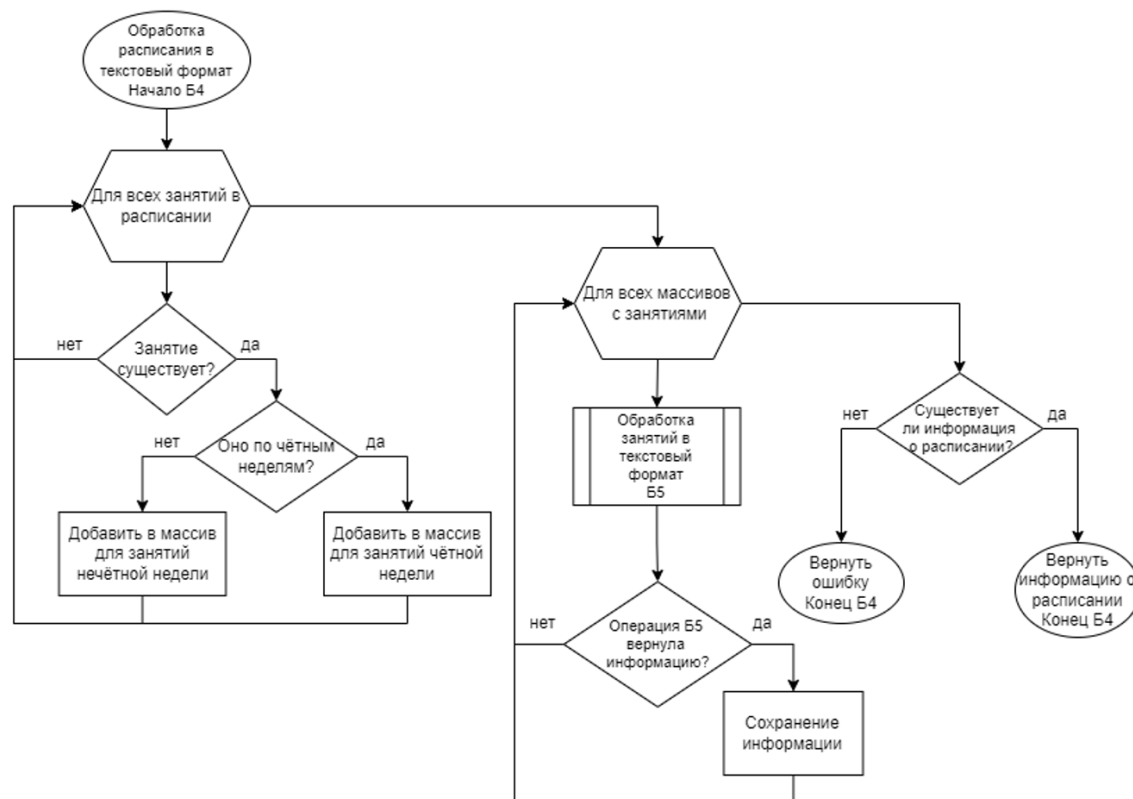


Рисунок В.4 – Блок-схема алгоритма форматирования информации из базы данных в текстовый формат для вывода в виде сообщения

Приложение В.5

Блок-схема алгоритма форматирования занятий расписания в текстовый формат для вывода в виде сообщения

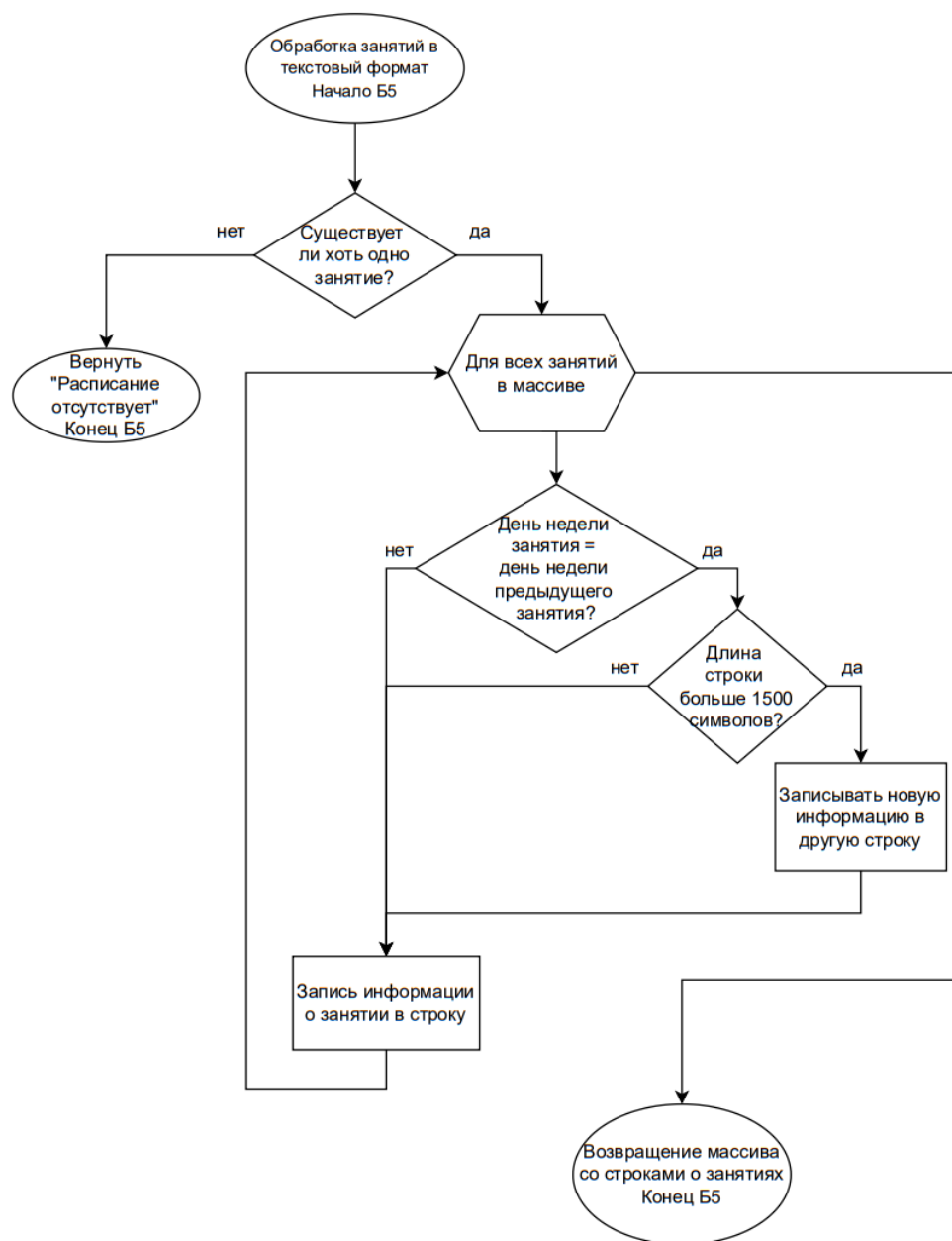


Рисунок В.5 – Блок-схема алгоритма форматирования занятий расписания в текстовый формат для вывода в виде сообщения

Приложение Г.1

Блок-схема алгоритма инициализации базы данных

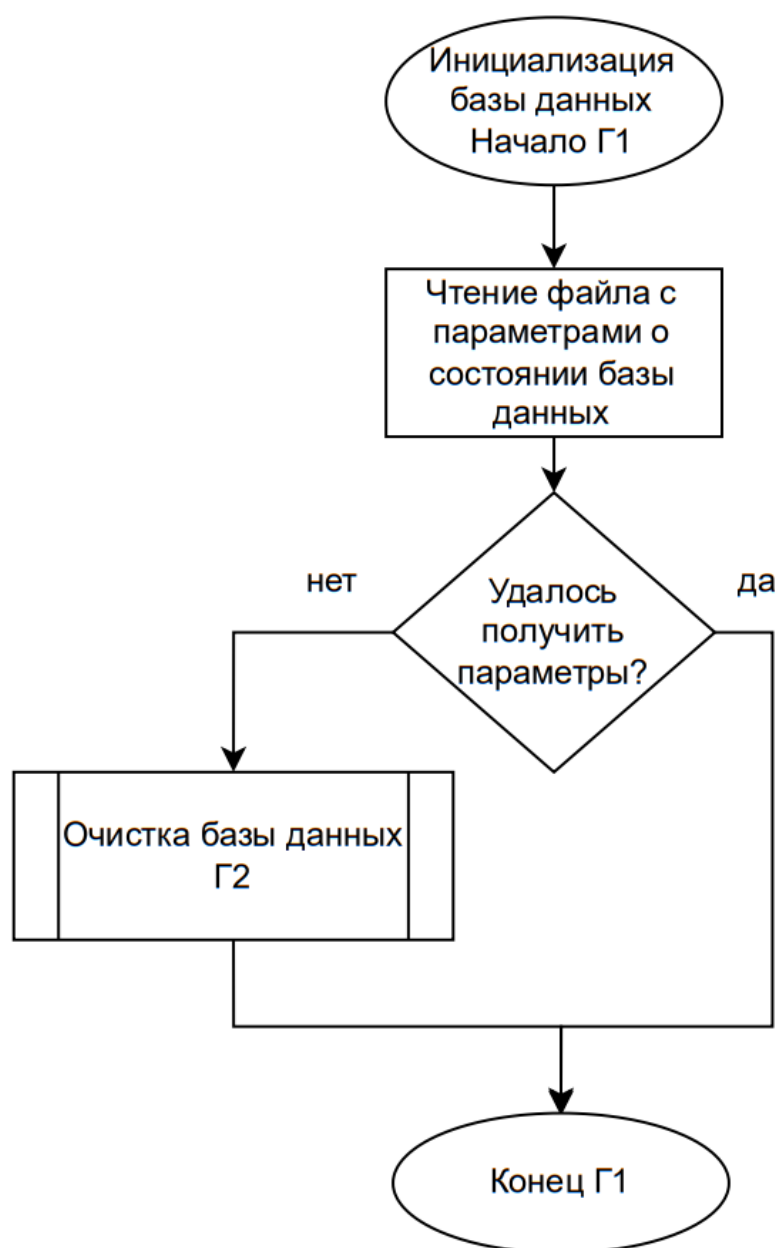


Рисунок Г.1 – Блок-схема алгоритма инициализации базы данных

Приложение Г.2

Блок-схема алгоритма очистки базы данных

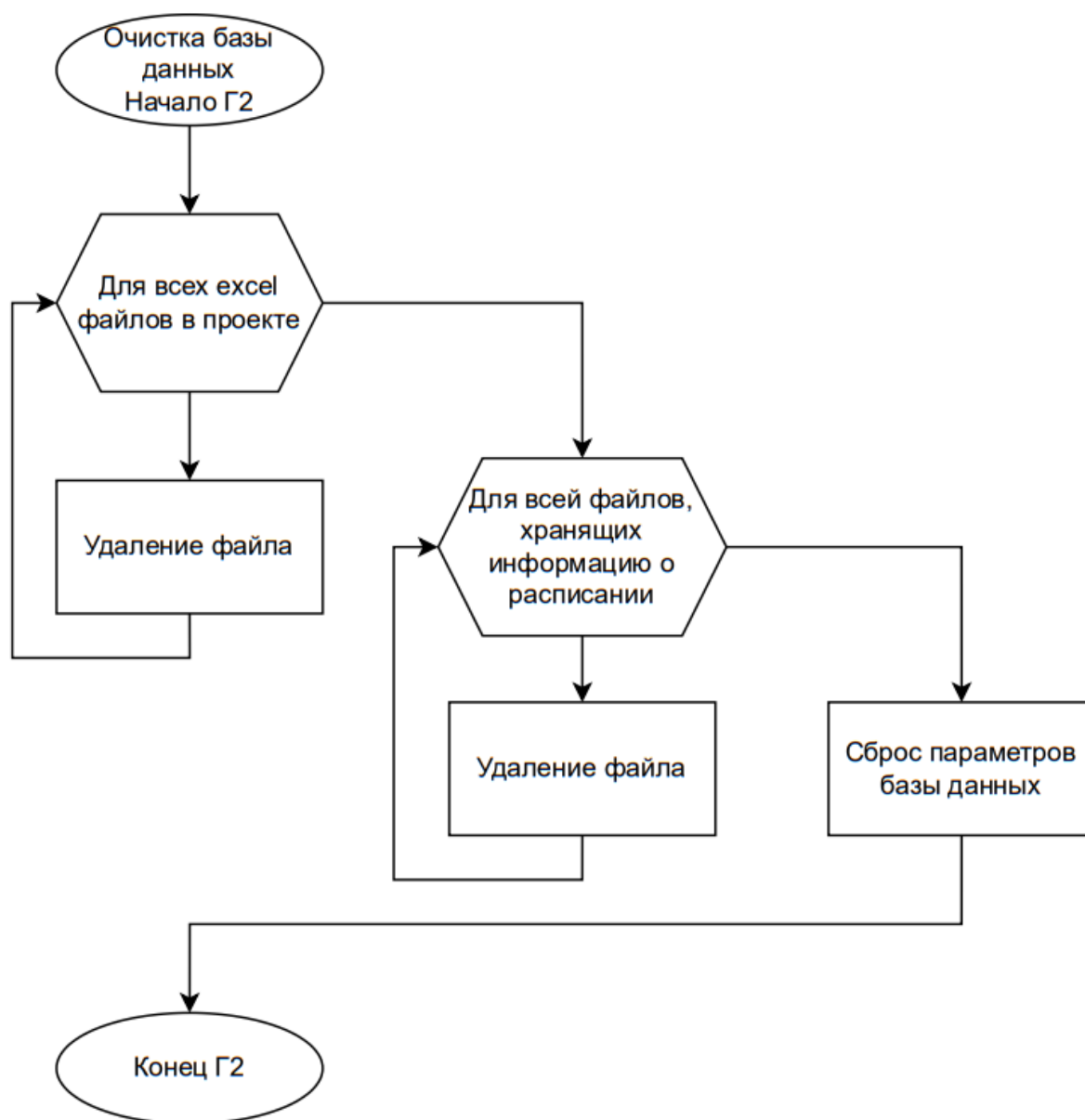


Рисунок Г.2 – Блок-схема алгоритма очистки базы данных

Приложение Г.3

Блок-схема алгоритма базы данных по чтению файла с расписанием занятий группы или преподавателя

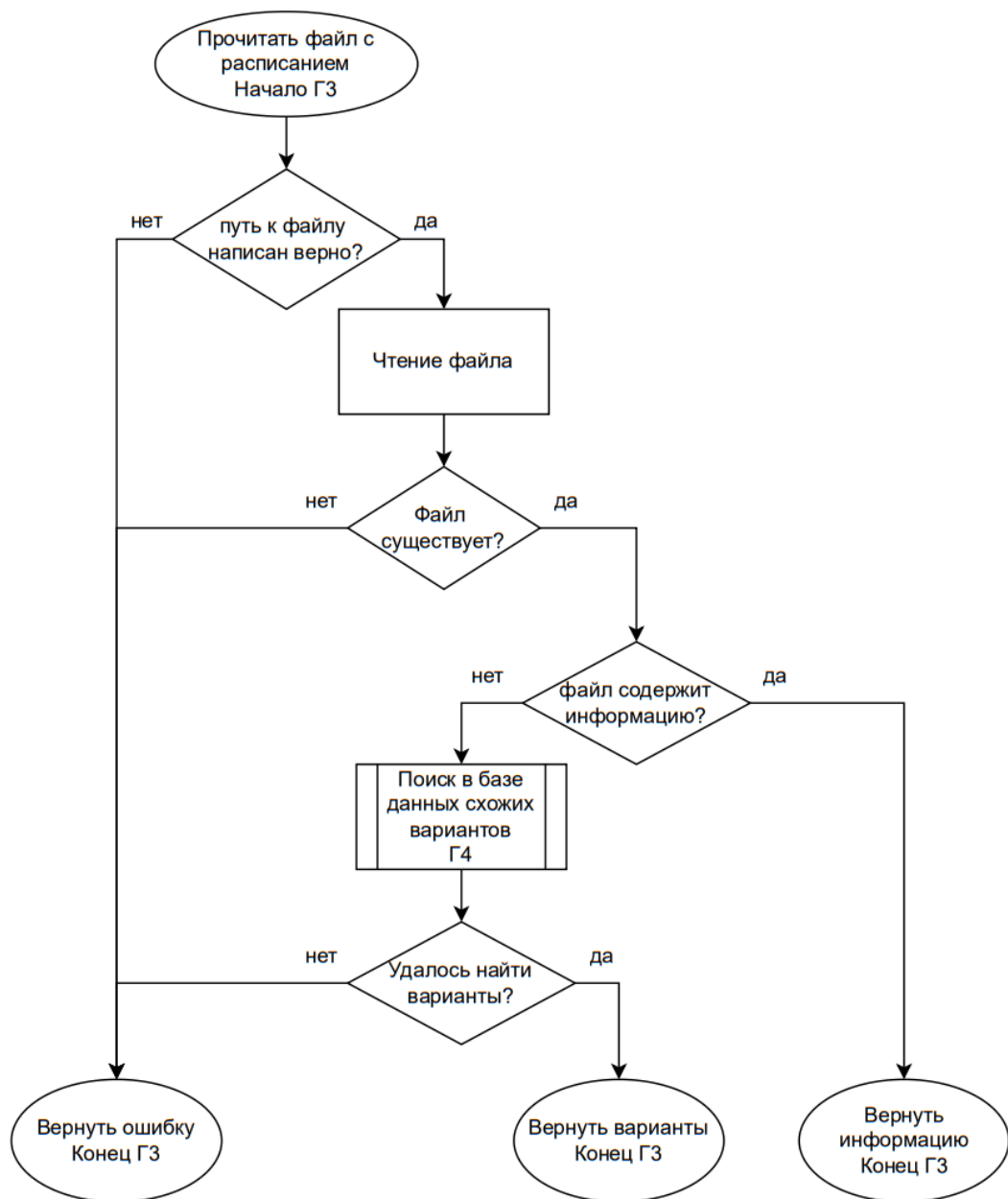


Рисунок Г.3 – Блок-схема алгоритма базы данных по чтению файла с расписанием занятий группы или преподавателя

Приложение Г.4

Блок-схема алгоритма базы данных по поиску схожей информации
требуемого запроса

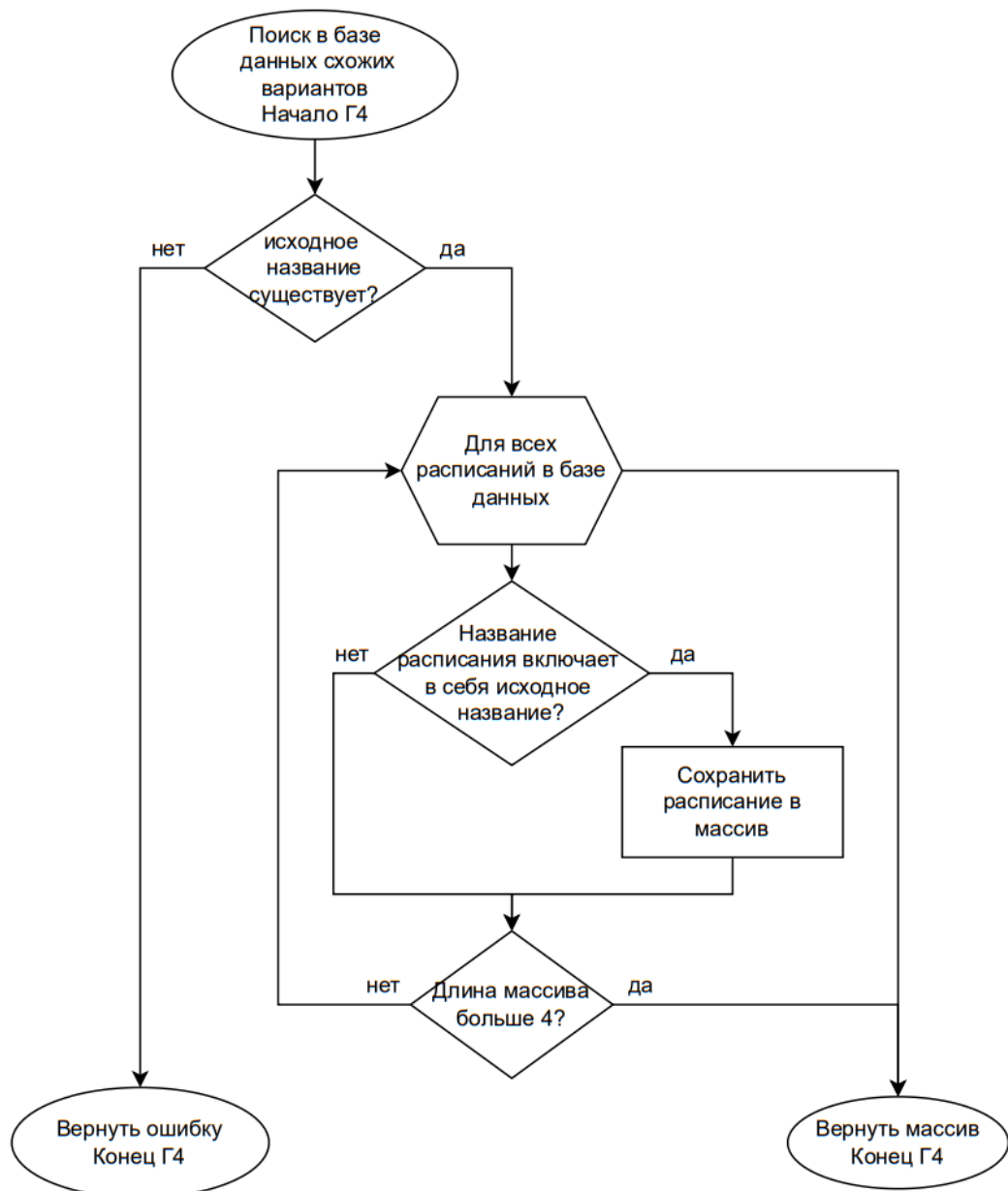


Рисунок Г.4 – Блок-схема алгоритма базы данных по поиску схожей информации
требуемого запроса

Приложение Г.5

Блок-схема алгоритма сохранения информации о расписании занятий в базе данных

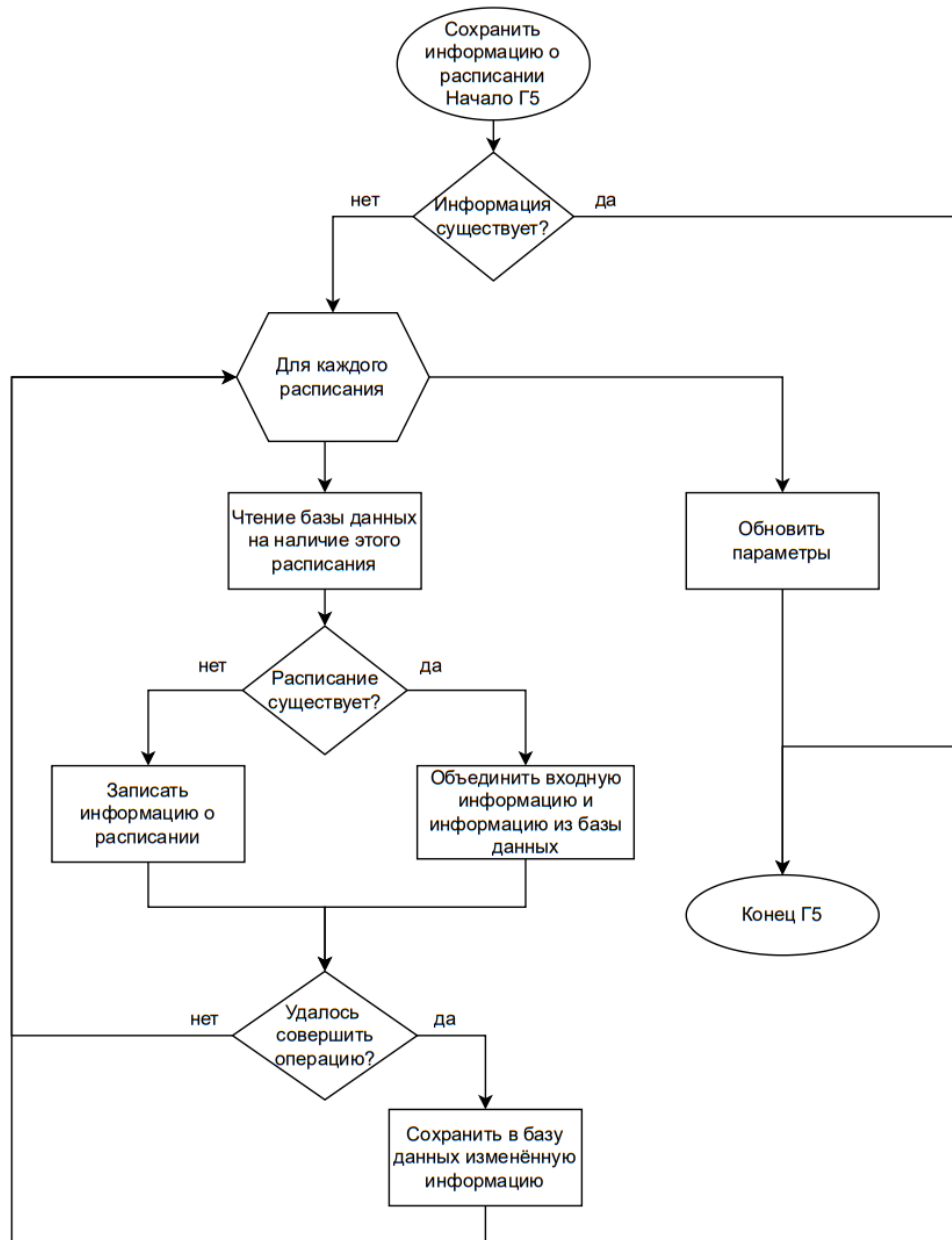


Рисунок Г.5 Блок-схема алгоритма сохранения информации о расписании занятий в базе данных

Приложение Д.1

Листинг кода обработки события нажатия на кнопку

Листинг Д.1 – Код подключения бота к Discord и настройка его поведения в мессенджере

```
const client = //Инициализация объекта-бота
new Discord.Client({intents: ["GUILDS", "GUILD_MESSAGES"]});

const prefix = process.env.PREFIX; // объявление префикса команды

let db = new DataBase(); //инициализация базы данных
client.login(process.env.DJS_TOKEN); //подключение к мессенджеру Discord

client.on("messageCreate", function(message) { //обработка события
                                                    //написания сообщения в чате

    if (!message.author.bot) return                //закончить операцию
                                                    //если автор сообщения бот

    if (!message.content.startsWith(prefix)) return; //прервать операцию если
                                                    //сообщение введено без
                                                    //префикса

    for (let counter in comms_list) //Для всех команд бота
    {
        if (command === comms_list[counter].name) //если сообщение является командой
        {
            comms_list[counter].out(message, db, args) //выполнить эту команду
        }
    }
});
```

Приложение Д.2

Листинг кода обработки события нажатия на кнопку

Листинг Д.2 – Код обработки события нажатия на кнопку

```
const collector =
message.channel.createMessageComponentCollector({filter, time: 15000 });

collector.on('collect', async i => { //Обработчик нажатия кнопок

  if (i.customId !== 'end') { //если нажатая кнопка не "отмена"
message.channel.messages.fetch(message.reference.messageId).then((resolve)=>{
  comms_list[comms_list.length-1].out(resolve,db,[i.customId]))
  message.delete()//Удалить сообщение и начать поиск нового расписания
}
Else //Если нажата кнопка отмена, вывести сообщение "поиск отменяется"
  i.update({ content: 'Поиск отменяется...', components: [] });
});
```

Приложение Д.3

Листинг кода чтения исходной страницы сайта

Листинг Д.3 – Код чтения исходной страницы сайта

```
let count = 0; //объявление счётчика на количество найденных ссылок
let i = 0; //Номер позиции чтения исходного кода страницы
let t = 'https://webservices.mirea.ru/'; //сайт, откуда скачиваются
//гиперссылки

let arrayUrl = []; //массив куда гиперссылки будут сохраняться

while ((i = text.indexOf('href="https://webservices.mirea.ru/',i)) != -1)
{
    //пока на странице есть гиперссылки

    i+=34; //сдвинуть читаемую позицию на 34 символов, равную длине t
    let str = ''; //объявить переменную, куда будет сохранена гиперссылка

    while (text[i+1]!='') //цикл для сбора символов, содержащих ссылку
    {
        i++;
        str +=text[i]; //добавление в переменную символов ссылки
    }
    if (str.indexOf('xlsx') != -1) //если это ссылка на скачивания файла
        //формата xlsx (формат книг excel)

    arrayUrl.push(encodeURIComponent(t+str)); //преобразовать ссылку в формат URI
        //для возможности боту перейти по ней

    count++; //прибавить к счётчику единицу
}
return arrayUrl //Вернуть ссылку на скачивание файла
```