

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Вятский государственный университет»
(ВятГУ)

УТВЕРЖДАЮ
Руководитель практики от Университета

(должность, ФИО)

МП

(Подпись)

Дата «__» _____ 2022 год

**ОТЧЕТ
ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.07 Сoadминистрирование и автоматизация баз данных и серверов

Иванов Иван Иванович

(Ф.И.О. обучающегося)

09.02.07 Информационные системы и программирование

(специальность)

Учебная группа

ИСПк-302-52-00

Место прохождения практики
Колледж ВятГУ

ФГБОУ ВО «Вятский государственный университет»,

(наименование организации, структурного подразделения организации)

Итоговая оценка:

Руководитель
практики от университета

(дата)

(подпись)

(Ф.И.О.)

Киров, 2022 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Вятский государственный университет»
(ВятГУ)

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

Ф.И.О. _____
обучающегося Иванов Иван Иванович
Специальность 09.02.07 Информационные системы и программирование
Учебная группа ИСПк-302-52-00
Вид практики учебная практика
Сроки прохождения практики с 17.01.2022 по 01.05.2022
Место прохождения практики ФГБОУ ВО «Вятский государственный университет»,
Колледж ВятГУ
(наименование организации, структурного подразделения организации)

№	Виды работ, выполняемых обучающимися во время практики	Объем работ (час)	Формируемые компетенции
1	Пройти инструктаж по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте	2	ОК-7
2	Анализ заданий на разработку, проведение исследований в заданной предметной области, поиск и анализ готовых технических решений	6	ОК-1-4
3	Проектирование структуры продукта, в том числе всех необходимых вспомогательных структур, включая базы данных	12	ОК-1, ОК-2, ОК-4, ПК-7.3
4	Реализация программного продукта	36	ОК-1, ОК-4, ОК-8, ОК-10, ПК-7.2, ПК-7.4
5	Настройка окружения, в том числе серверной части, для бесперебойного функционирования продукта	12	ОК-9-11, ПК-7.1, ПК-7.2
6	Верификация полученного программного продукта	12	ОК-10, ПК-7.1, ПК-7.5, ПК-7.6
7	Подготовка отчетной документации, включая детальное описание использованных инфраструктурных решений	8	ОК-5
8	Подготовка к сдаче и сдача промежуточной аттестации	2	ОК-6

Индивидуальное задание на практику разработано в соответствии с рабочей программой практики.

Руководитель
практики от университета

_____ (дата)

_____ (подпись)

_____ (Ф.И.О.)

С индивидуальным заданием
ознакомлен(а)

_____ (дата, подпись обучающегося)

ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ

Ф.И.О. обучающегося	Иванов Иван Иванович		
Специальность	09.02.07 Информационные системы и программирование		
Учебная группа	ИСПк-302-52-00		
Вид практики	учебная практика		
Сроки прохождения практики с	17.01.2022	по	01.05.2022
Место прохождения практики	ФГБОУ ВО «Вятский государственный университет», Колледж ВятГУ		
	(наименование организации, структурного подразделения организации)		

ВИДЫ И КАЧЕСТВО ВЫПОЛНЕНИЯ РАБОТ

Вид работ	Критерий выполнения работ		
	Выполнены полностью самостоятельно	Выполнены с незначительной помощью наставника	Выполнены с помощью наставника
Анализ заданий на разработку, проведение исследований в заданной предметной области, поиск и анализ готовых технических решений	V		
Проектирование, в том числе проектирование всех необходимых вспомогательных структур, включая базы данных	V		
Реализация программного продукта	V		
Настройка окружения, в том числе серверной части, для бесперебойного функционирования продукта	V		
Верификация полученного программного продукта	V		
Подготовка отчетной документации, включая детальное описание использованных инфраструктурных решений	V		
Подготовка к сдаче и сдача промежуточной аттестации	V		

Обучающийся ознакомлен с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также прошел вводный инструктаж и инструктаж на рабочем месте.

Во время прохождения учебной практики обучающимся освоены следующие профессиональные и общие компетенции:

Наименование компетенции	Показатели оценки	Оценка	
		Освоена	Не освоена
ПК 7.1. Выявлять технические проблемы, возникающие в процессе эксплуатации баз данных и серверов.	Способен анализировать работоспособность инфраструктуры, предназначенной для поддержки работоспособности программного продукта	V	
ПК 7.2. Осуществлять	Способен интегрировать	V	

администрирование отдельных компонент серверов.	программный продукт в инфраструктуру и обеспечивать его работоспособность		
ПК 7.3. Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов.	Способен вырабатывать проектные решения, принимая во внимание ресурсные составляющие, необходимые для реализации данных решений	✓	
ПК 7.4. Осуществлять администрирование баз данных в рамках своей компетенции.	Способен обеспечивать связь программного продукта с базой данных	✓	
ПК 7.5. Проводить аудит систем безопасности баз данных и серверов с использованием регламентов по защите информации.	Способен определять корректность и уровень безопасности функционирования инфраструктурных решений	✓	
ПК 7.6. Производить оценку баз данных для выявления возможности их модернизации.	Способен оценивать соответствие структур хранения данных заданным спецификациям и осуществлять их модернизацию в случае необходимости	✓	
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности	✓	
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.	Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности	✓	
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.	Способен определять спектр программных инструментов, позволяющих повысить личную, командную и профессиональную результативность	✓	
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.	Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разъясняющей информации	✓	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	Способен разрабатывать проектную, техническую и пользовательскую документации	✓	

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.	Способен конструктивно обмениваться информацией с коллегами, грамотно формулировать запросы в целях получения разьясняющей информации	✓	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.	Способен прогнозировать эффективность и ресурсозатратность используемых средств	✓	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.	Способен соблюдать требования внутреннего трудового распорядка организации, охраны труда и техники безопасности в целях сохранения собственного здоровья	✓	
ОК 09. Использовать информационные технологии в профессиональной деятельности.	Способен применять стек современных средств разработки ПО для решения задач профессиональной деятельности в заданном контексте	✓	
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.	Способен использовать в своей работе специализированную документацию	✓	
ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере.	Способен выбирать решения, принимая во внимание имеющиеся ресурсные ограничения	✓	

Краткая характеристика работы обучающегося

Программа практики выполнена в полном объеме. Все виды работ выполнялись в срок, без существенных замечаний. В достаточной степени была проявлена самостоятельность и умение грамотно пользоваться источниками информации, находящимися в свободном доступе. Принятые архитектурно-структурные решения в целом адекватны. Процесс работы был выстроен методологически грамотно.

Руководитель практики

_____/_____
Подпись ФИО

(должность)

Дата «__» _____ 2022 год

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ.....	6
2. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ.....	7
3. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПРОГРАММНОГО РЕШЕНИЯ	17
4. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА	22
5. ИСПОЛЬЗУЕМЫЙ ИНСТРУМЕНТАРИЙ И РАБОЧЕЕ ОКРУЖЕНИЕ.....	31
6. СИСТЕМА РАЗДЕЛЕНИЯ ТРУДА.....	34
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	40
ПРИЛОЖЕНИЕ А.....	41

ВВЕДЕНИЕ

Учебная практика ПМ.07 проходила на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 17.01.2022 г. по 01.05.2022 г. по понедельникам.

Цель практики: сформировать у обучающихся навыки коллективной разработки программных продуктов, включающих подсистемы хранения данных (базы данных), а также предполагающих использование инфраструктурных решений для их бесперебойного функционирования.

Задачи практики:

- закрепить полученные в ходе освоения предшествующих дисциплин навыки и умения в области создания программных продуктов;
- закрепить навыки создания и администрирования подсистем хранения данных;
- закрепить навыки организации, использования и администрирования инфраструктурных решений, предназначенных для обеспечения бесперебойного функционирования разрабатываемых программных продуктов.

1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ

В период с 17.01.2022 по 01.05.2022 при прохождении учебной практики ПМ.07 на базе Колледжа ФГОБУ ВО «Вятский государственный университет» был выполнен следующий перечень работ, представленный в таблице 1.

Таблица 1 – Сведения о работе, выполненной в период практики

Дата	Краткое содержание выполненных работ
21.01.2022	Пройти инструктаж по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте
21.01.2022-27.01.2022	Анализ заданий на разработку, проведение исследований в заданной предметной области, поиск и анализ готовых технических решений
28.01.2022-10.02.2022	Проектирование структуры продукта, в том числе всех необходимых вспомогательных структур, включая базы данных
04.02.2022-18.03.2022	Реализация программного продукта
19.03.2022-01.04.2022	Настройка окружения, в том числе серверной части, для бесперебойного функционирования продукта
02.04.2022-15.04.2022	Верификация полученного программного продукта
16.04.2022-29.04.2022	Подготовка отчетной документации, включая детальное описание использованных инфраструктурных решений
30.04.2022	Подготовка к сдаче и сдача промежуточной аттестации

(дата)

(подпись)

2. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ

Описание поставленной задачи.

Поставленной задачей была разработка системы рассылки сообщений в мессенджерах и посредством электронной почты. Данная задача включала в себя ряд подзадач.

Знакомство с особенностями рассылок в мессенджерах и электронной почте. Для решения данной задачи использовались источники информации в интернете, что привело к более ясному представлению дальнейших действий разработки и постановки подзадач.

Следующей подзадачей было описание функциональных требований для будущего приложения. На данном этапе рассматривались аналогичные приложения для рассылки, чтобы взять некоторые особенности. Составлялись требования, которые были необходимы для реализации поставленной задачи.

Визуализация готового приложения, а именно описание концепции физической реализации приложения. В данной подзадаче были рассмотрены аналогичные приложения, которые направлены на рассылку, для того чтобы определить плюсы и минусы внешнего вида аналогов и выбрать оптимальное решение для внешнего вида разрабатываемого программного обеспечения.

Следующей подзадачей было определения языка разработки и разбор программных кодов аналогичных приложений. Данная задача позволила более детально посмотреть ход выполнения рассылки в мессенджеры и на почту, что позволило сложить более ясное понимание дальнейших действий в разработке.

Дальнейшим ходом выполнения подзадач было определение экранных форм приложения и выбор цветовой палитры для готового приложения. Данная подзадача являлась переходной от изучения теории и аналогов, к физической разработке приложения, что позволило более наглядно понимать визуальные аспекты готового приложения.

Просмотр аналогов.

В ходе работы были просмотрены некоторые похожие проекты. Просматривались они и в плане работы, и в плане внешнего вида, и в плане реализации (программного кода).

- [Проект mail-send](#),

Особенностью проекта были хорошие комментарии к коду, адреса получателей сообщались программе единым файлом, текст сообщения заранее был запрограммирован в программе, те программа не является универсальной.

Для избежание блокировки отсылки почтовым сервером из-за подозрения на спам, после отсылки каждого второго сообщения делалась пауза в 5 секунд. Это также сообщалось в инструкции к приложению.

Выводом из оценки программы стало то, что для нашего приложения не подходит метод сообщение адресов файлом. Пользователь может забыть, запутаться, сделать опечатку, и отправка станет не возможной.

Также для удобства и понятности программы как пользователям, так и разработчикам необходимо создать подробные и понятные комментарии к коду и доступную инструкцию пользователю.

- [Проект vk_sender](#)

Особенностью проекта были консольная реализация, понятная инструкция для пользователя, текст сообщения должен был сообщаться программе файлом, что не очень удобно, а также возможность удаления сообщений

Для спасения от подозрений в спаме, программа полностью переделывала сообщение, каждый раз создавая уникальный текст. Меняла русские буквы на английские (аналогичные), добавляла в случайные места лишние пробелы, а также между отправкой сообщений стоял таймер.

Случаи ошибки с отправкой регистрировались и сообщались пользователю.

Выводом из анализа программы стало то, что при наличии достаточного времени на разработку, возможно тоже перенять метод переделки сообщения и таймера, чтобы сервер не блокировал отправку сообщения. А также при

наличии времени на разработку реализовать возможность отчета об ошибках. Но вот консольная реализация показала себя крайне нежизнеспособной для использования в текущем проекте.

- [Проект TDiscordMessage](#)

Чтобы программа запустилась, нужно было скачать Synapse и скомпилировать его в Lazarus.

Также для корректной работы в Windows необходимо было скачать и поместить в директорию проекта две динамически подключаемых библиотеки с расширением .dll (libeay32.dll и libssl32.dll).

Программа предназначалась для отправки сообщений на канал в Discord от имени бота, и являлась демоверсией программы, потому что в самой программе нельзя было изменить текст сообщений, имя бота, изображения и т.п.

Для этого на канале необходимо было создать вебхук.

Встроенная Функция Webhooks – это простой способ получать автоматические сообщения и обновления данных, которые были посланы на текстовый канал Вашего сервера.

Программа состояла из одной формы, где есть поле для ввода вебхука и две кнопки: “Simple” и “Advanced”.

При первоначальном запуске программы, при вводе правильного вебхука и при поочередном нажатии кнопок на канале появляются следующие сообщения:

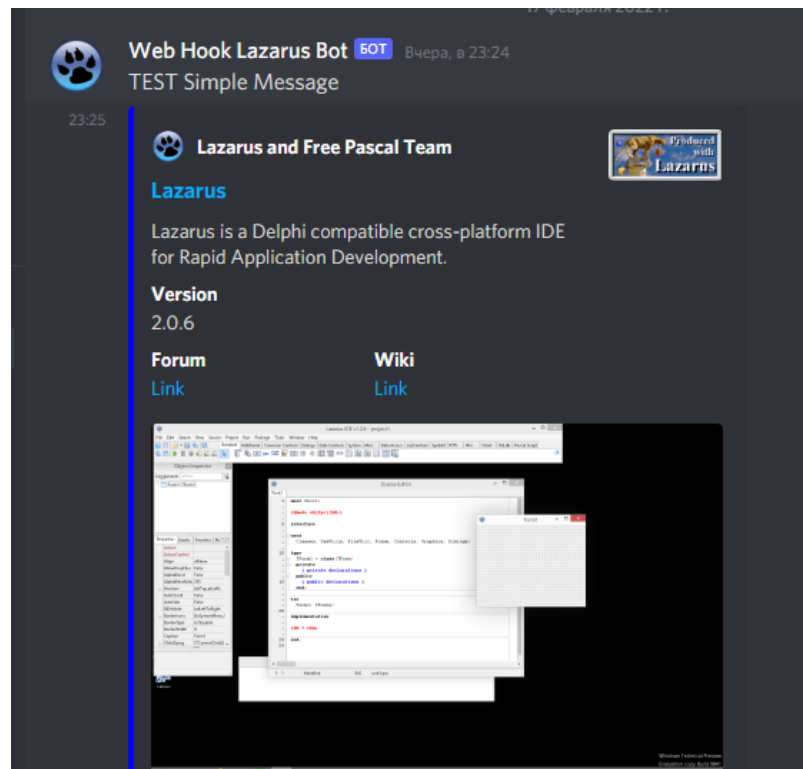


Рисунок 1 — Результат работы программы.

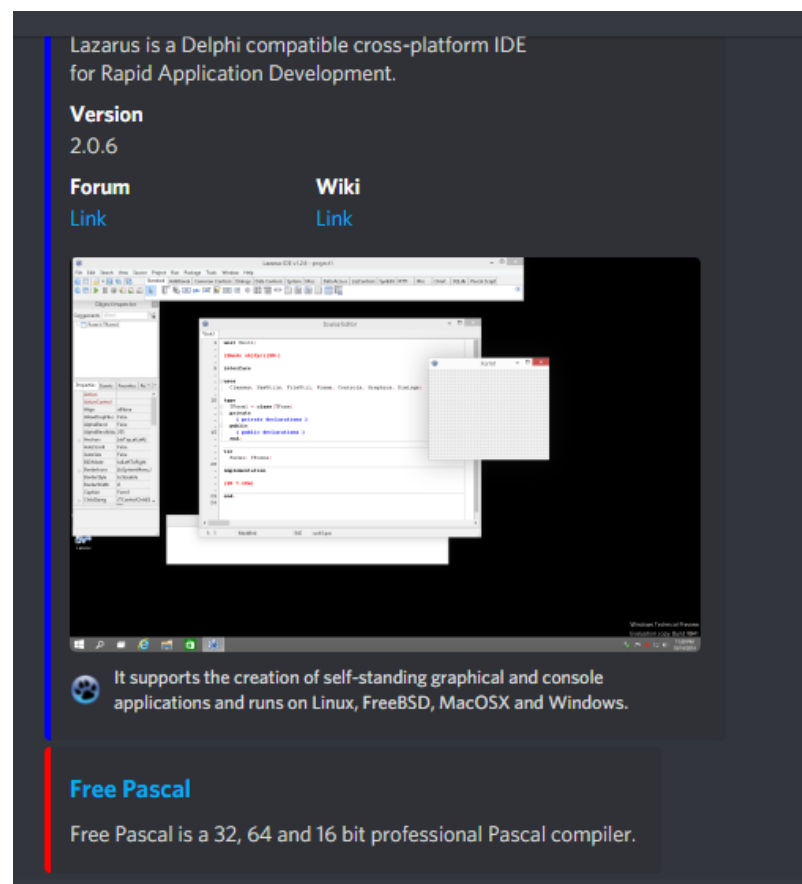


Рисунок 2 — Результат работы программы.

Посмотрев в код, можно было найти код нажатия на первую кнопку “Simple” (простое сообщение):

```

. procedure TForm1.Button2Click(Sender: TObject);
. var
.   Discord: TDiscordMessage;
. begin
40   Discord := TDiscordMessage.Create(Edit1.Text);
.   try
.     Discord.Content := 'TEST Simple Message';
.     Discord.UserName := 'Web Hook Lazarus Bot';
.     Discord.AvatarURL := 'https://a.fsdn.com/allura/p/lazarus/icon?1552026857?&w=90';
45   Discord.SendMessage;
.   finally
.     FreeAndNil(Discord);
.   end;
. end;
. end;

```

Рисунок 3 — Код нажатия кнопки “Simple”.

Тут можем было увидеть, каким переменным присваиваются значения, которые выводятся в боте – текст сообщения, имя бота и изображение бота.

Точно такие же используются и в коде нажатия второй кнопки, но там добавляются новые переменные. Вот некоторые из них:

```

procedure TForm1.Button1Click(Sender: TObject);
Embeds := TDiscordEmbeds.Create;
Embeds.Title := 'Lazarus';

Embeds.URL := 'https://www.lazarus-ide.org/';
Embeds.Description := 'Lazarus is a Delphi compatible cross-platform IDE for Rapid
Embeds.Color := $0000FF;
// Image
Image := TDiscordEmbedsImage.Create;
Image.URL := 'https://wiki.lazarus.freepascal.org/images/5/5d/Windows_10.png';
Embeds.SetImage(Image);
FreeAndNil(Image);

// thumb
Thumb := TDiscordEmbedsThumbNail.Create;
Thumb.URL := 'https://www.freepascal.org/pic/lazarus_produced_logo.gif';
Embeds.SetThumbNail(Thumb);
FreeAndNil(Thumb);

// Author
Author := TDiscordEmbedsAuthor.Create;

```

Рисунок 4 — Переменные использующиеся в боте.

Таким образом, можно было поменять данные в коде на любые корректные значения, чтобы вывести именно то, что необходимо. Причем можно было бы использовать не все поля, а только те, что необходимы. В качестве примера получилось такое сообщение:

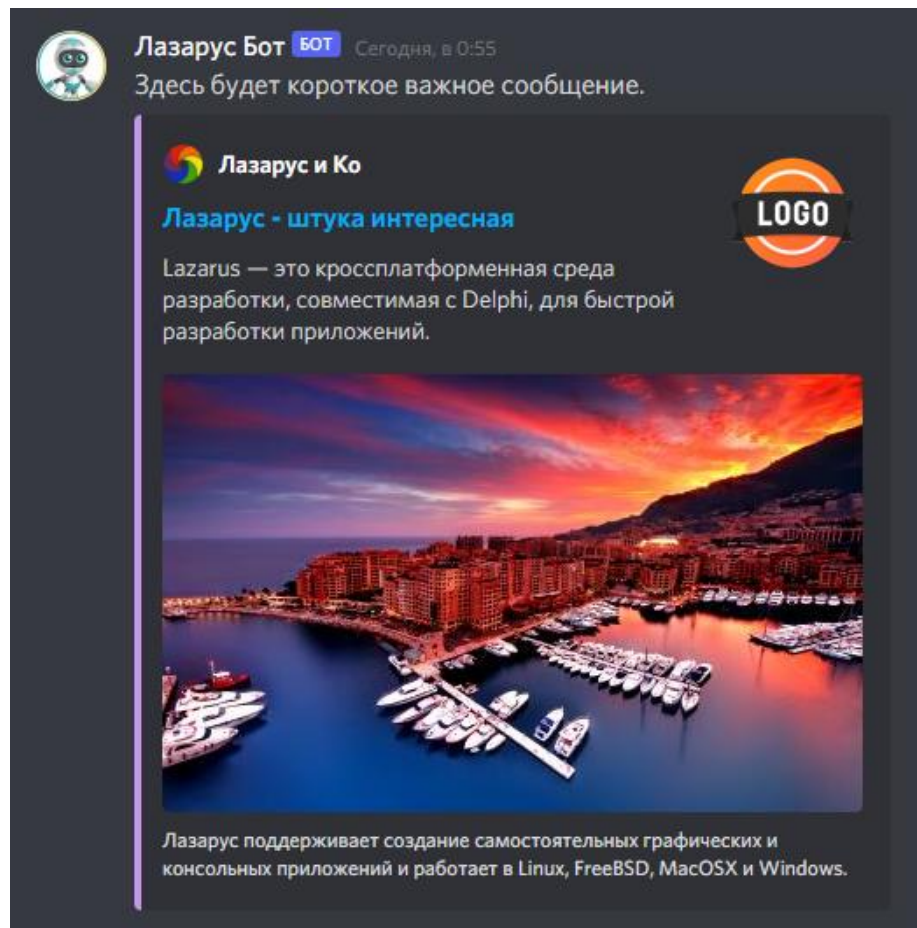


Рисунок 5 — Измененное сообщение.

- Chat 2 Desk (дизайн)

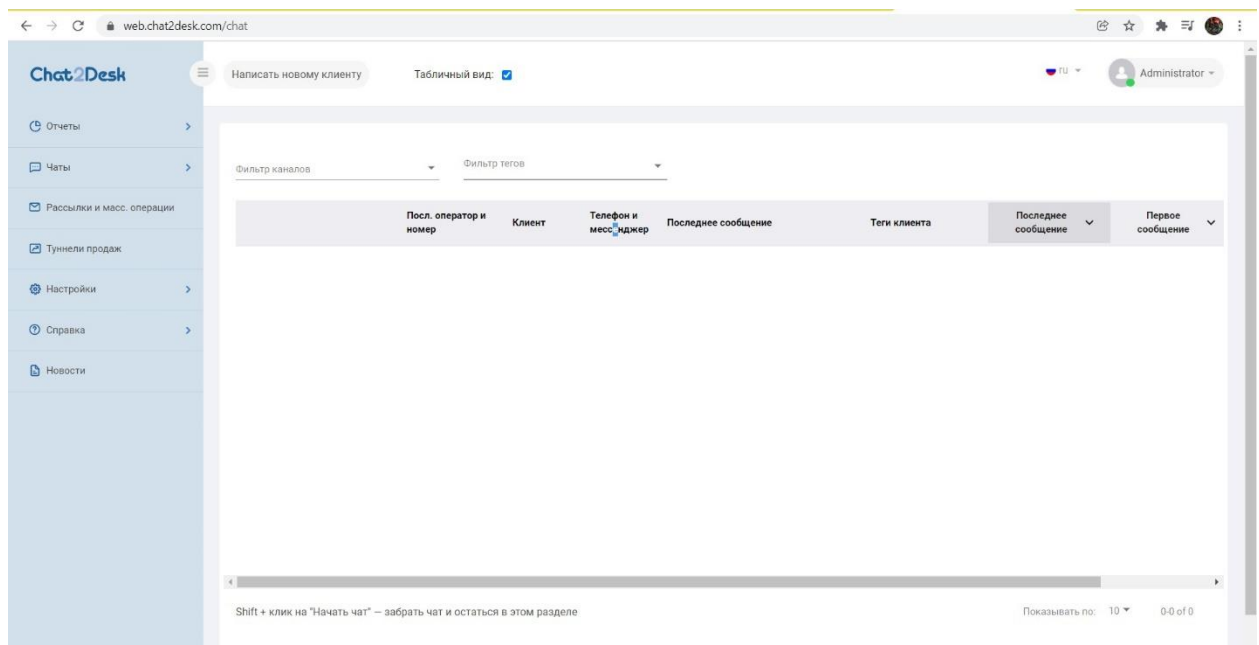


Рисунок 6 — Chat 2 Desk.

Плюсы:

- Приятная цветовая гамма сайта.

- Шрифт подобран грамотно, с соответствием моды.

Минусы:

- Слишком много различных вкладок.
- Wings (дизайн)

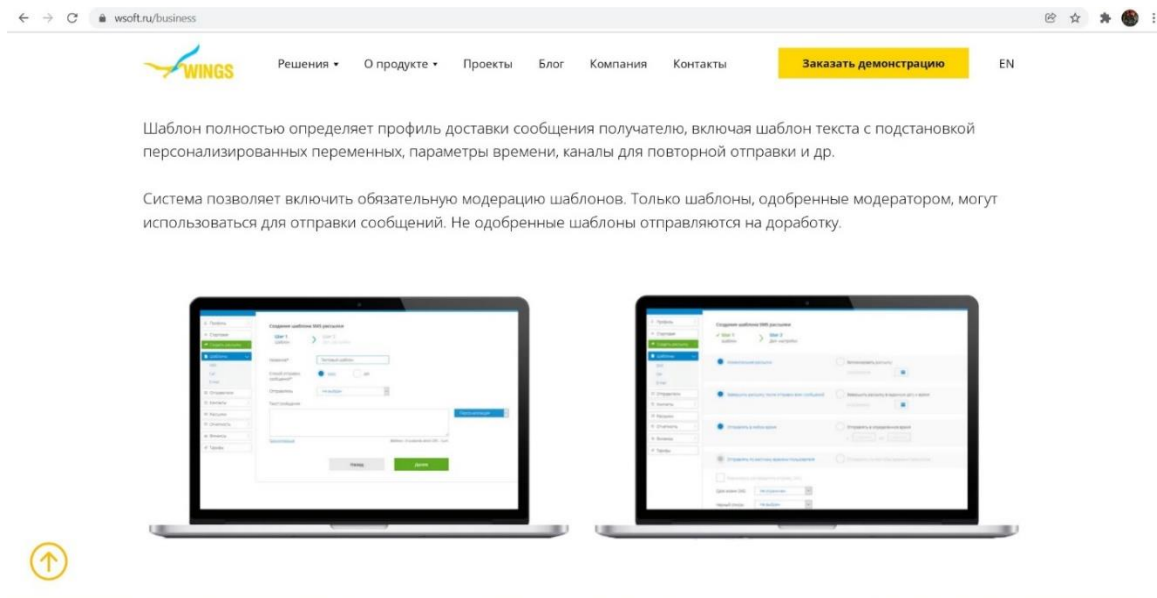
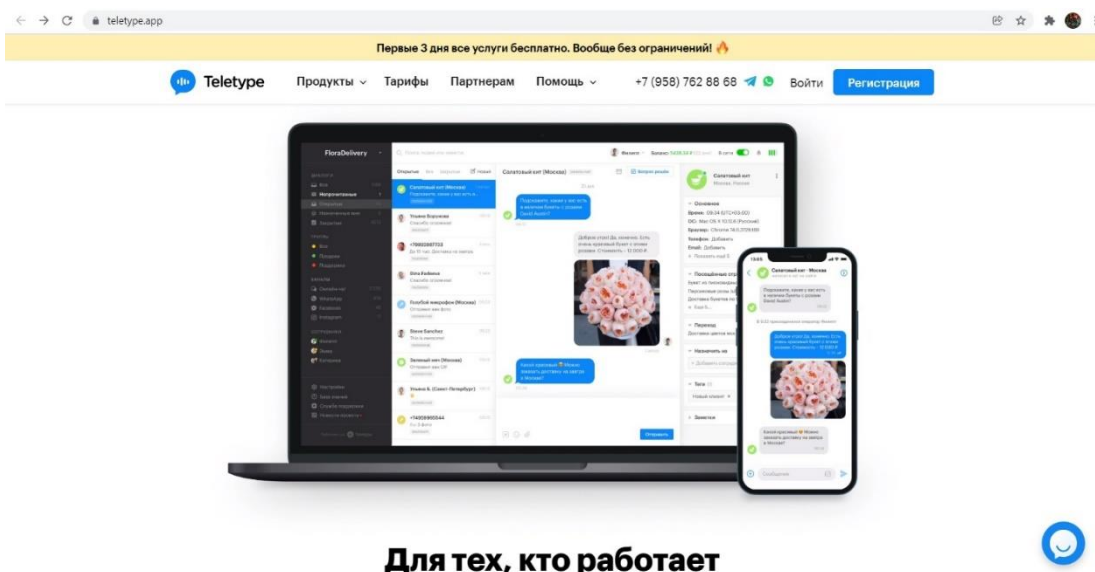


Рисунок 7 — Wings.

Плюсы:

- Использовано много разных цветов, для упрощения нахождения информации.
- Отличительная черта данного сервиса, это цвет его логотипа.
- Teletype (дизайн)



Для тех, кто работает

Рисунок 8 — Teletype.

Плюсы:

- Огромное количество, свежих и полезных цветовых решений.
- Многофункциональность всей платформы.
- Логотип довольно прогрессивный и выполнен в минимализме.
- I2CRM (дизайн)

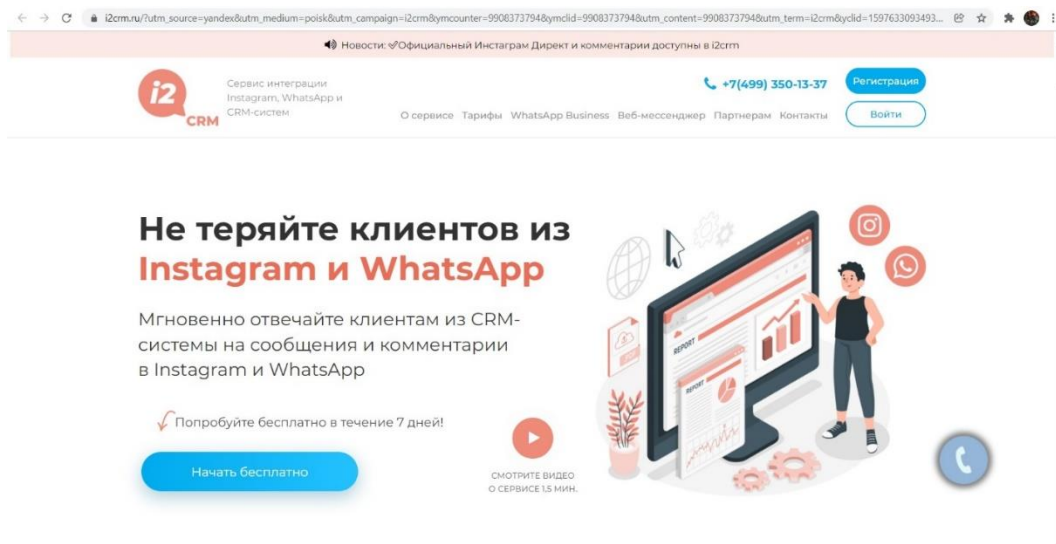


Рисунок 9 — I2CRM.

Плюсы:

- Отличительные цветовые решения, очень постельные тона.
- Большое количество стикеров и скриптов.
- W-Sender (дизайн)

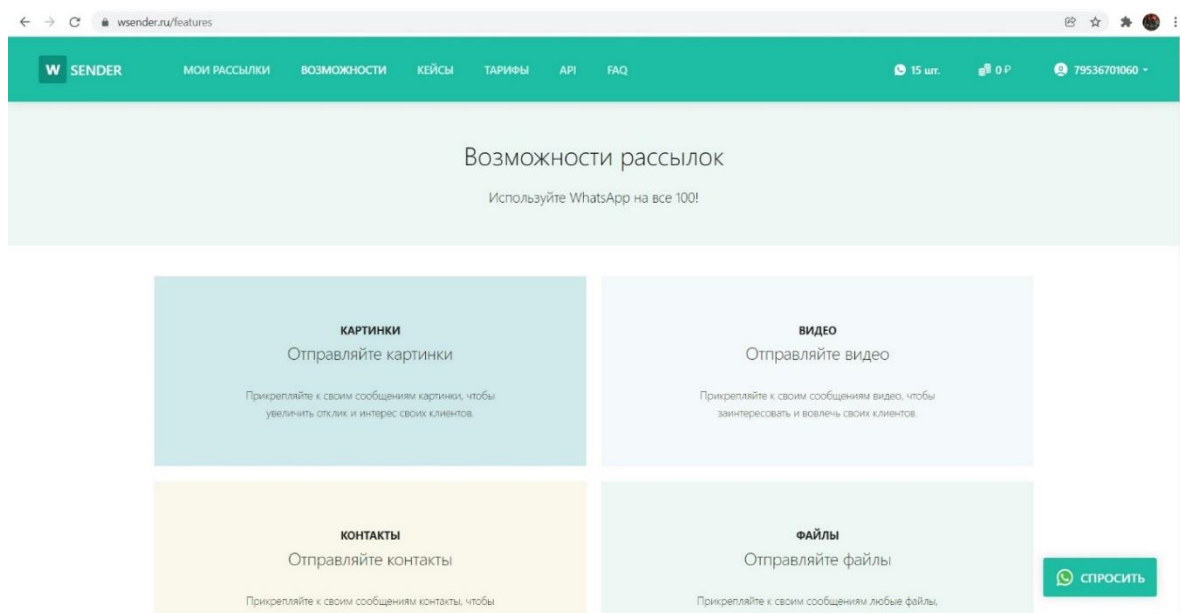


Рисунок 10 — W-Sender.

Плюсы:

- Цветовая гамма подобрана под главную цель данной платформы

Минусы:

- Непонятный интерфейс с точки зрения читабельности и функциональности.

- Е-Sputnik (дизайн)

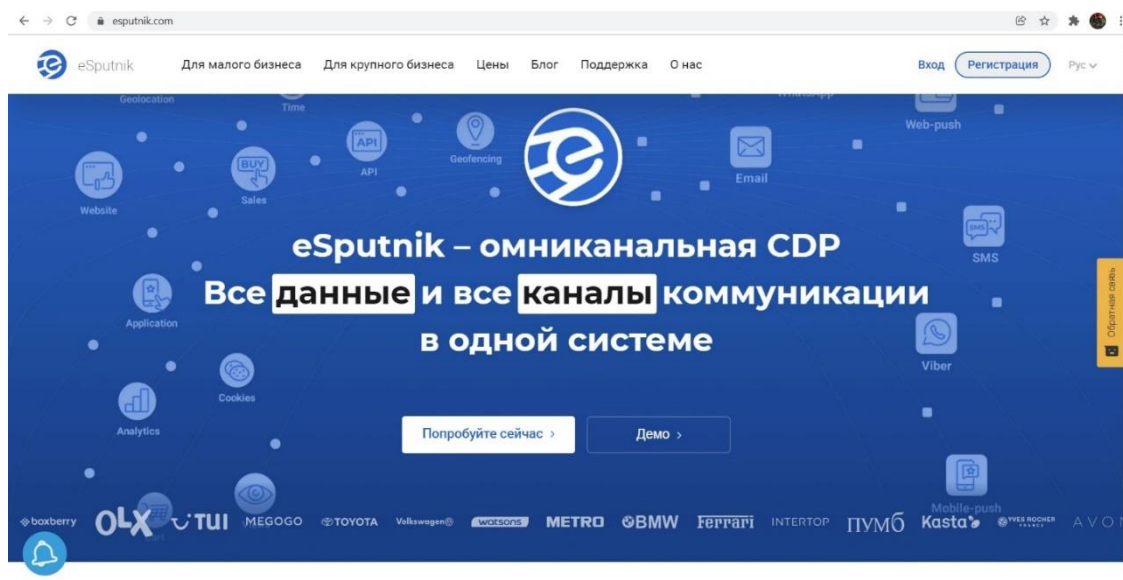


Рисунок 11 — Е-Sputnik.

Плюсы:

- Очень раскрученная и узнаваемая платформа.
- Цветовая гамма и кнопки взяты у топовых компаний, что даёт платформе большую лояльность клиента.

Из просмотренных приложений относительно дизайна можно было сделать следующие выводы.

У некоторых компаний очень много недочётов (W-Sender, I2CRM, E-sputnik, Wings), кажется, что данные платформы интересные, но до лидеров рынка им очень далеко, из-за слабой функциональности и странных цветовых решений.

Лидерами по дизайну стали компании: Chat2Desk и Teletype) – потому что данные платформы имеют обширный функционал и востребованные цветовые решения. Данные две компании, а точнее их принцип работы и дизайнерское решение, стоило бы использовать для опоры в текущем проекте.

Техническое задание.

После просмотра аналогов и коллективного обсуждения было создано техническое задание к проекту. Увидеть его можно в Приложении А.

Проблемная область.

Приложение для рассылки New Message создано для облегчения продвижения своего бизнеса, и поддержание связи с клиентами.

Большая организация может совершать E-mail рассылки внутри компании, чтобы оповестить всех сотрудников о каких-либо изменениях, рассказать последние новости или просто поздравить с наступающим праздником.

Тем, кто ведёт тренинги или любые обучающие программы, данное приложение поможет с рассылкой заданий или уроков в ВК, ВК от e-mail отличается тем, что у e-mail не всегда всё очевидно с успешной доставкой письма, и во избежание плохих отзывов для этого лучше использовать ВК.

Введение бизнеса, которое направленно на аудиторию геймеров. New Message поможет провести рассылку в дискорде, чтобы привлечь аудиторию на свой сервер. Поскольку дискорд является довольно популярной социальной сетью, в ней общаются, в основном геймеры. Потому продать им соответствующие товары будет не проблемой. Вернее, процент тех, кто может заинтересоваться, в данном мессенджере выше. И это позволяет продавать свои товары и услуги, связанные с данной тематикой, и не только.

New Message приложение решающее многие проблемы с продвижением бизнеса, в таких сервисах как, Discord, VK и E-mail, что является большим отличием от других приложений, которые узко направлены, на какой то один сервис.

3. ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ПРОГРАММНОГО РЕШЕНИЯ

Идея.

Каждый день тысячи мобильных приложений появляются в Google Play и Apple App Store. Соцсети, мессенджеры, игры и многие другие – все они делаются профессионалами по одному алгоритму разработки. Как бы тривиально это не звучало, но хорошее приложение начинается с хорошей идеи. Наша команда использовала одну из изначального списка идей, а именно создание приложения для рассылки сообщений на нескольких платформах одновременно.

Ключевым является понимание того, что людям иногда нужно написать одно сообщение, но отправить его не только нескольким людям, но и на разные платформы одновременно. После просмотра многочисленных аналогов, о которых упоминалось ранее, сбора данных, обсуждение вариантов реализации и программных средств для этого, мы пришли к такому выводу. Для программы, из всех платформ-мессенджеров лучше всего будет использовать E-mail, Discord и Telegram. По срокам реализации и техническому оснащению, мы могли осуществить задумку и взяли проект в работу.

Функционал приложения.

Данный шаг, позволил понять, что именно нами и будущими пользователями ожидается от приложения?

Таким образом после коллективного обсуждения и опроса, для аналитика возникла задача — прописать функциональные требования к приложению. В тот момент команда имела слабое представление о будущем виде проекта и даже о его функциональности.

После возник первый вариант функциональных требований. От него и стала отталкиваться работа. После функциональные требования изменялись и дополнялись несколько раз. А в середине проекта возникла ситуация, из-за которой пришлось внести в ФТ достаточно серьезные изменения.

Дело было в том, что не у всех участников проекта было понимание конечной идеи. А именно, одновременной рассылки на несколько платформ. А изначальная концепция предполагала, как бы, несколько под-приложений рассылки никак не связанных друг с другом.

Подробнее итоговый вариант можно прочесть в Техническом задании в Приложении А.

Дизайн

Далее необходимо было определиться с внешним видом приложения. После анализа аналогов стало понятно, что консольный вариант нам не подходит. Значит следующей задачей стало разработка интерфейса.

Для проработки дизайна использовалось приложение Figma. Его выбрали из-за удобного функционала, широкого набора инструмента для редактирования, а самое главное, возможности создать прототип приложения: вариант с работающими переходами, но без функционала.

Первый вариант дизайна выглядел так:

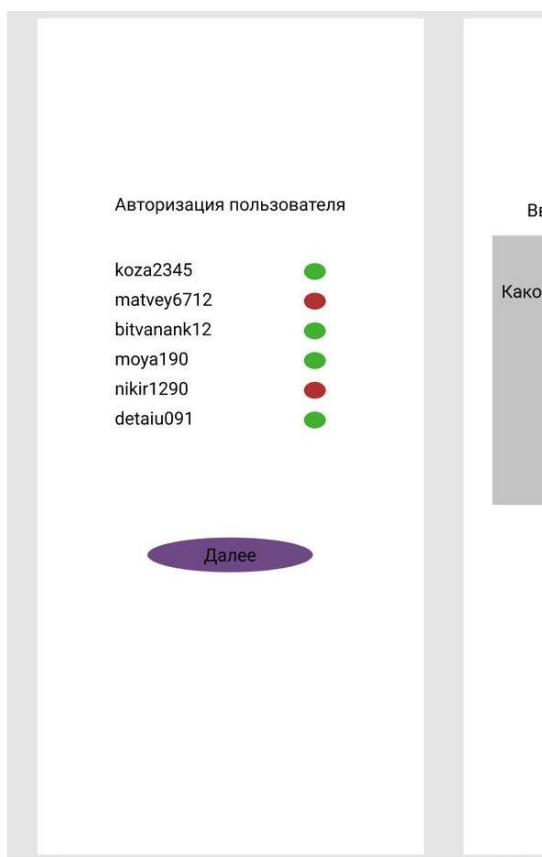


Рисунок 12 — Изначальный вариант дизайна.



Рисунок 13 — Изначальный вариант дизайна.

Но в последующем дизайн перетерпел значительные изменения. Изменился формат кнопок подстраиваясь под изначальные инструменты компилятора. Например, в Лазарус невозможно поменять цвет или форму кнопки (только если сделать её с помощью других элементов или создать свой пользовательский тип). Далее добавилась цветовая гамма в осенних тонах и фотоподложка для фона.

Итоговый дизайн выглядел таким образом:

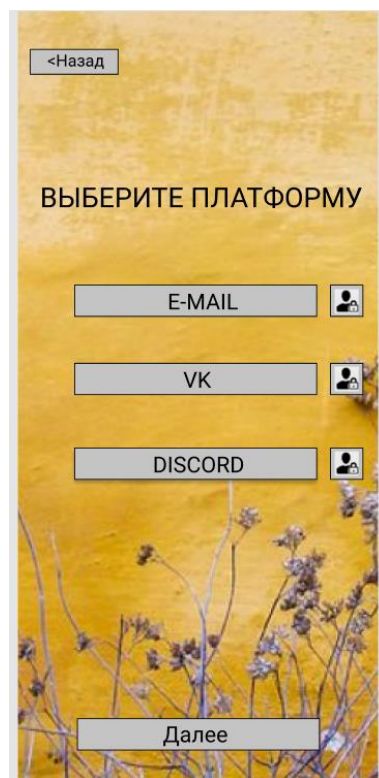


Рисунок 14 — Итоговый дизайн.

Подробнее все требования к дизайну указаны в Техническом задании в Приложении А.

Изучение теории

Для реализации приложения необходимо было подготовиться и изучить теорию по следующим темам: работа с базами данных и работа с сетью. Работа с базами данных была более легкой задачей, так как на предыдущем семестре уже изучался SQL язык и подключение баз данных к приложению. По этой теме сдавалась курсовая работа. А изучение работы с сетью стало тяжелой работой, из-за отсутствия внятных источников и непопулярности выбранного языка программирования. Существовали решения и инструкции на других языках, таких как Python

В итоге разработки концепции приложения получилось следующее: дизайн в «Осеннем» стиле, логотип в жёлтом цвете, рассылка сообщения на несколько платформ одновременно, регистрация пользователей (собственная база аккаунтов), и адреса для рассылки и данные для авторизации на платформах сохраняются в базе данных (повторно вводить их не требуется).

Спорным вопросом также была кнопка вложения файлов или картинок. Для подобного приложения это безусловно обязательное требование. Но в данном случае вставал вопрос, возможно ли реализовать подобное в наших условиях. В итоге пришли к выводу, что можно реализовать подобное в ограниченном варианте. То есть пользователь может прикреплять файл, но только изображение и только одно.

Распределение ролей

К данной ступени, мы уже: нашли идею, прописали функционал, оценили рынок конкурентов. Осталось раздать роли, каждому члену команды. После коллективного обсуждения мы пришли к такому выводу:

Роли разработчиков - Широкова Виктория и Максимов Дмитрий.

Роль дизайнера – Верещагин Матвей.

Роль аналитика – Наговицын Александр.

4. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

Для реализации программы был выбран язык программирования Delphi и компилятор Lazarus. Подробнее об используемом окружении можно прочитать ниже.

Разработка двигалась с внешней части приложения – интерфейса, к внутренней – коду обработчиков.

Создание форм

Первые появившийся вариант форм был еще черновым, так как дизайн не был до конца закончен, а ответственный за него участник проекта отсутствовал по причине болезни.

Так выглядели формы изначально:

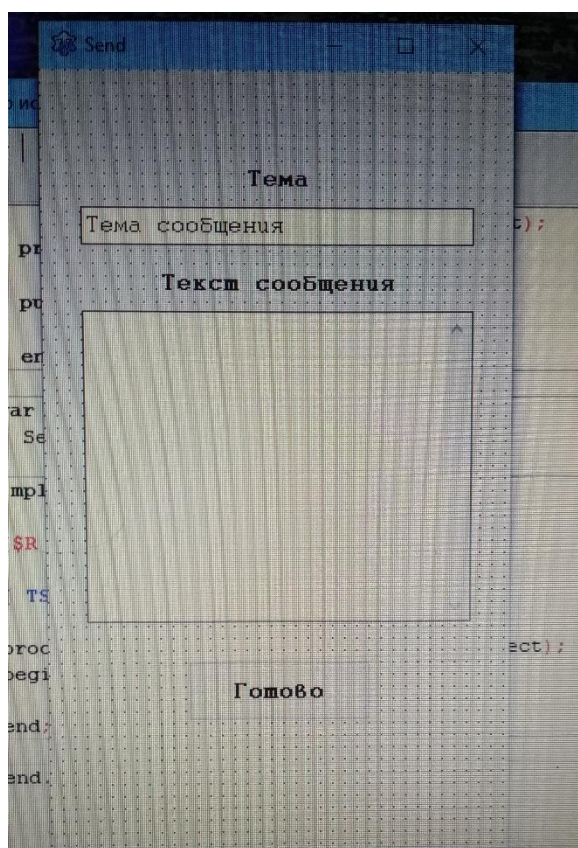


Рисунок 15 — Черновик формы отправки сообщений.

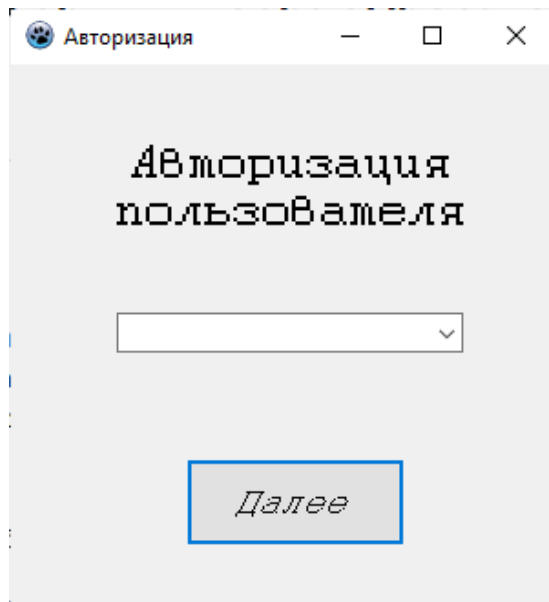


Рисунок 16 — Черновик авторизации пользователя.

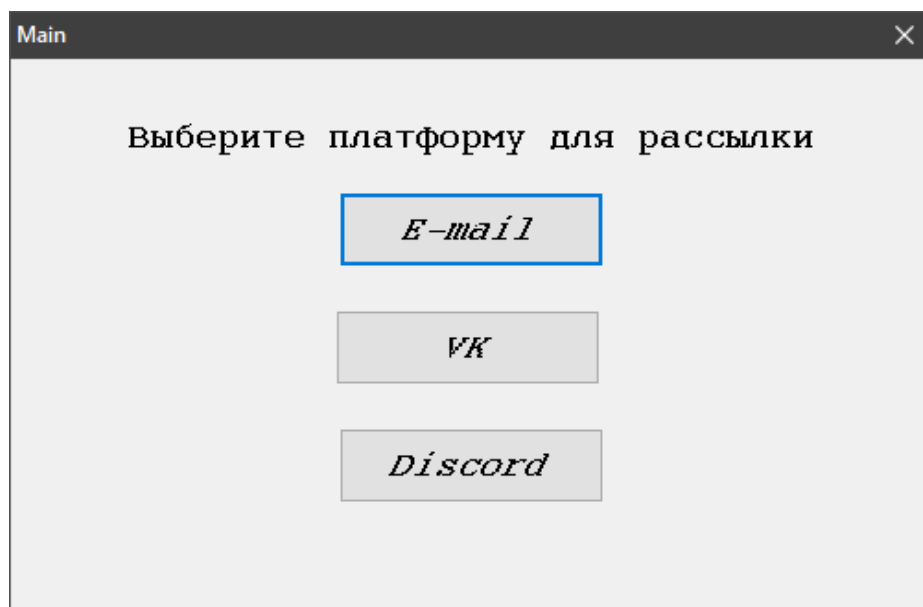


Рисунок 17 — Черновик главной формы.

В дальнейшем, когда дизайн был полностью доработан, указанные выше формы приобрели следующий вид (см. рисунки 25 и 26).

Разработка базы данных

Для дальнейшей работы нужно было разработать базу данных. Она тоже подвергалась значительным изменениям. Изначально база данных строилась по следующей схеме:

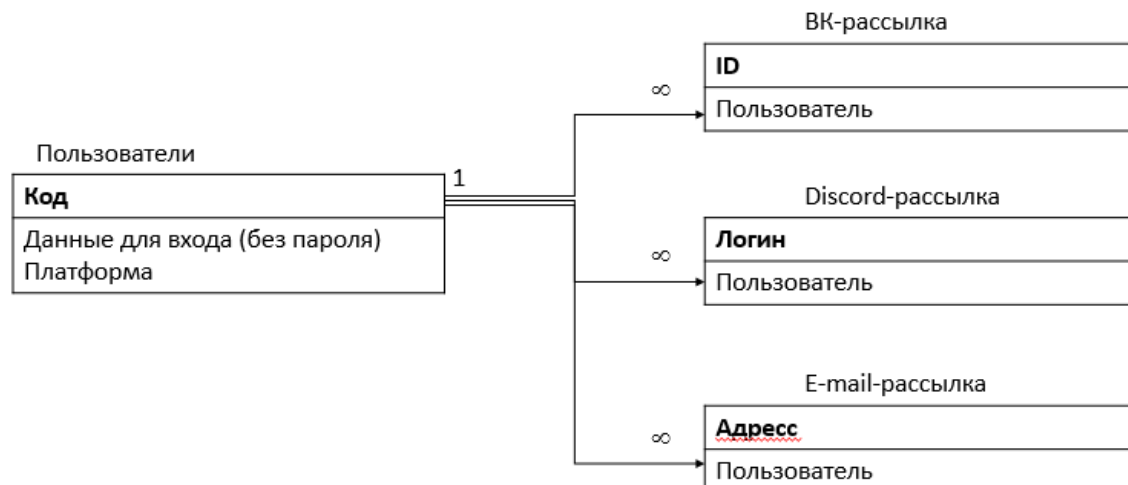


Рисунок 18 — Черновик схемы базы данных.

Но после изменения концепции работы, о которой упоминалось ранее пришлось изменить базу данных:

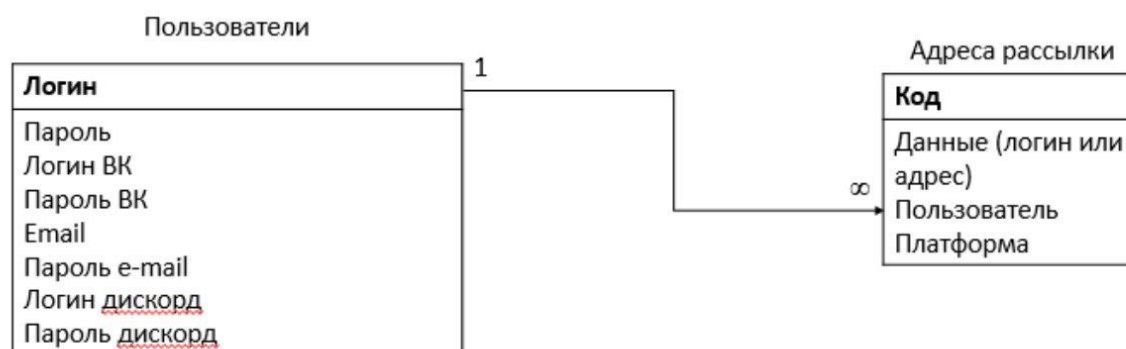


Рисунок 19 — Итоговая схема базы данных.

Но и она в итоге в самом конце разработки приобрела незначительные изменения, для документации которых уже не хватало ресурсов.

Разработка происходила в программе SQLite Studio, подробнее в следующем пункте. Там же на рисунке 27 можно увидеть разработанное отношение Пользователи.

Подключение базы данных к программе.

Подключение базы данных к программе было значительной частью работы. Данные об авторизованных пользователях, а также их получателях хранилась в базе данных и нуждалась в обработке.

Также нужно было написать динамический вывод пользователей с разделением их по платформам. А значит – процедурно создавать объекты форм с помощью динамических массивов.

Вот отрезок кода для выполнения этой задачи:

```
//Заполняем адреса с разделением по платформам

KolOnPlat[1]:=0;KolOnPlat[2]:=0;KolOnPlat[3]:=0;
for PL:=1 to 3 do
  if platsel[PL]=true then begin
    with fUsers do begin
      SQLQ.Close;
      SQLQ.SQL.Text:='select Count(*) as Количество from Адреса where Пользователь = :L and Платформа=:p';
      SQLQ.ParamByName('L').AsString :=login_name;
      SQLQ.ParamByName('p').AsInteger :=PL;
      SQLQ.Open;
      KolOnPlat[PL]:=SQLQ.Fields.FieldByName('Количество').AsInteger;
      SQLQ.Close;
    end;
    SetLength(resip[PL],KolOnPlat[PL]);

    platname[PL]:=TLabel.Create(self); //Название платформы
    platname[PL].Parent:=Field;
    platname[PL].Top:=x;
    platname[PL].Left:=yn;
    platname[PL].Caption:=n[PL];
    platname[PL].Font.Style:=[fsBold];
    x:=x+16;

  //Заполнение адресов
  i:=0;
  with fUsers do begin
    x:=x+5; //мое дополне
    SQLQ.Close;
    SQLQ.SQL.Text:='select Данные from Адреса where Пользователь = :L and Платформа= :p';
```

```

SQLQ.ParamByName('L').AsString :=login_name;
SQLQ.ParamByName('p').AsInteger :=PL;
SQLQ.Open;
SQLQ.First;
while not SQLQ.EOF do begin
  with FSelectU do begin
    resip[PL][i]:=TCheckBox.Create(self);
    resip[PL][i].Parent:=Field;
    resip[PL][i].Top:=x;
    resip[PL][i].Caption:=fUsers.SQLQ.Fields.FieldByName('Данные').AsString;
    end;
    inc(i);SQLQ.Next; x:=x+20;
  end;
end;
x:=x+16; //Пустая строка
end;

```

Создание шифровки для данных из БД.

Следующим этапом работы было создание процедур шифровки, дешифровки и хеширования паролей, для сохранения конфиденциальности пользователей.

В последствии, при шифровании вебхуков вскрылся баг. В некоторых ситуациях зашифрованные буквы превращались в программные символы. А конкретнее в символ #130 «'».

Пришлось изучить подробную информацию о ASCII кодировке. И ограничить символы до промежутка с #32 до #126:

```
T:=T-126+32;
```

Код работы с email.

По изначальной концепции мы планировали дать возможность пользователям делать отправку сообщений со своих почт. Но по факту это оказалось довольно проблематично для пользователя (код был уже готов, когда пришлось вносить правки).

Современные почтовые серверы сильно беспокоятся о защите почтовых ящиков и не позволяют подключаться к ним, даже при наличии логина и пароля, не дав в настройках разрешение и выбрав правильный порт соединения. А в некоторых сервисах и вовсе отсутствует эта возможность, как например в gmail.

Поэтому мы пришли к выводу, что для пользователей будет проще если рассылка будет проходить с нашего аккаунта. Хотя, возможно, в дальнейшем можно реализовать возможность выбрать, делать рассылку со своего аккаунта или с общего.

Второй проблемой стали переходы между абзацами. В ascii кодировке для этого используется символ #13, но для html разметки он не читаем. В ней надо использовать
. Решением стало написание обработчика для замены СИМВОЛОВ:

```
for i:=1 to length(textm) do  
  if textm[i]=#13 then begin  
    delete(textm,i,1);  
    insert('<br />',textm,i);
```

Вот примеры письма до и после:

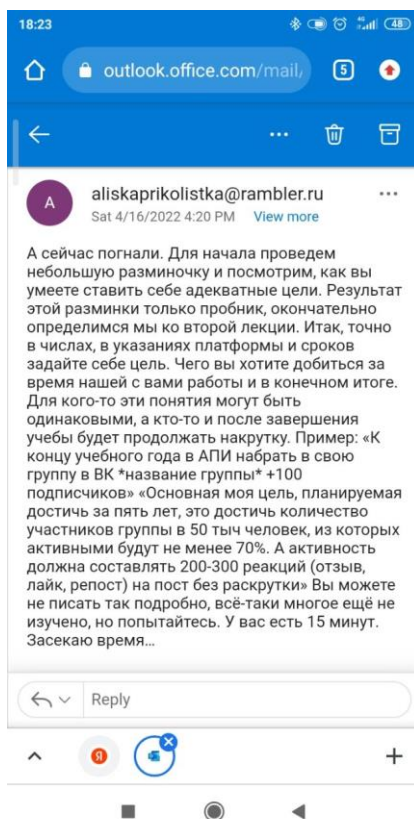


Рисунок 20 — До.

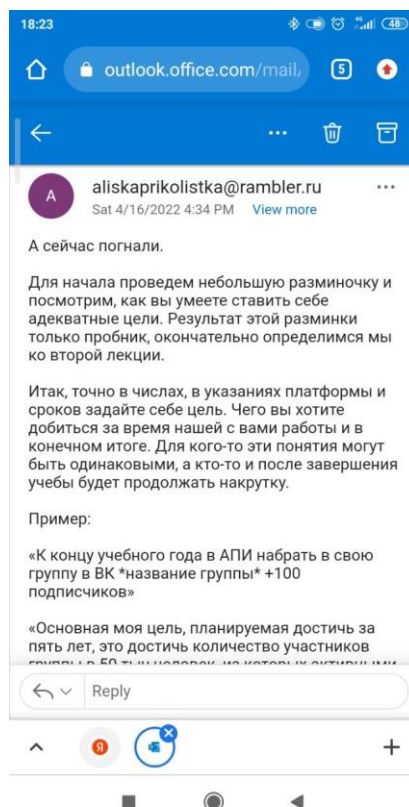


Рисунок 21 — После.

Код для Discord.

Для работы с дискордом были использованы вебхуки. Что значительно облегчало проблемы с доступом. А в рассмотренных аналогах была похожая работа, с примерами написания обработчиков и подходящими библиотеками для работы.

Наблюдались сложности с отправкой картинки загруженной с компьютера, тогда мы попробовали заменить её на картинку с интернет сервера по URL и все заработало.

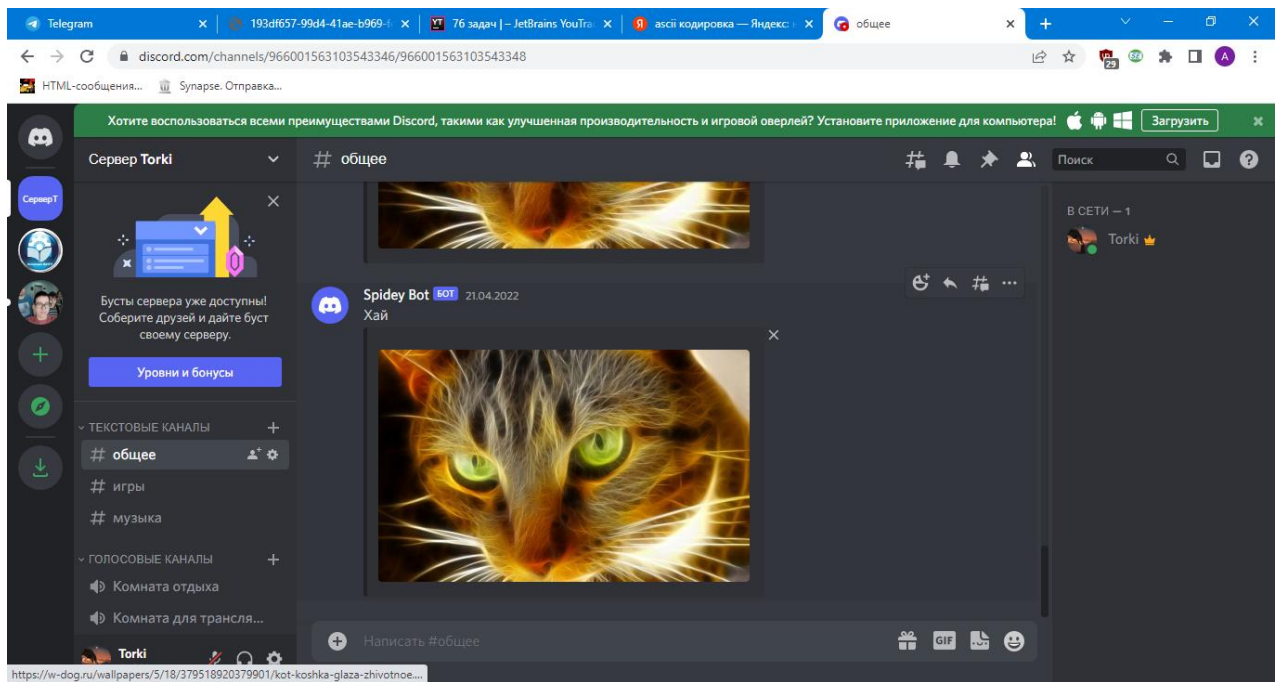


Рисунок 22 — Тест на работу с Дискордом.

Код для Telegram

Телеграм рассылка тоже происходит через бот, других возможностей для доступа через программу платформа не представляет.

Главная загвоздка в том, что для отсылки получатель должен хоть раз написать боту первым. К сожалению, обойти этот запрет никак не удалось.

Сначала была попытка реализовать работу напрямую через доступ к API, с использованием примерно такого кода:

```
try
    IdHttp:=TIdHTTP.Create;
    SSL:=TIdSSLIOHandlerSocketOpenSSL.Create;
    IdHttp.IOHandler:=SSL;
    JsObject:=SO(IdHttp.Get('https://api.telegram.org/bot'+API+'/GetMe'));
    if JsObject.B['ok']=True then //Ауторизация успешна
        ShowMessage('Бот успешно стартовал');
finally
    IdHttp.Free;
    SSL.Free;
end;
```

Но при подключении появлялась ошибка «error:1409442E:SLL routines^ssl3_read_bytes:tlsv1 alert protocol version». Смена версии не помогала, замена библиотек тоже. В итоге решение так и не нашлось, но нашлась

библиотека, с помощью которой можно было делать эти запросы без обращения напрямую к http.

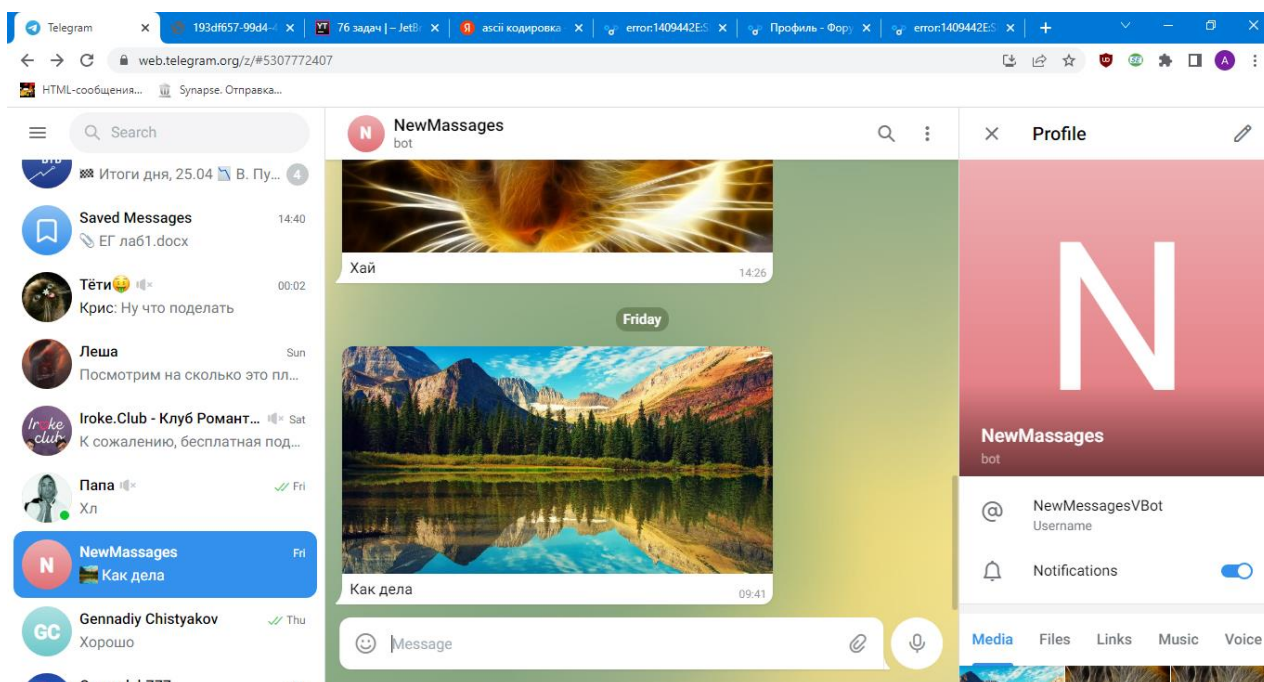


Рисунок 23 — Результат работы Телеграм.

Обработчик исключений.

Для защиты от возможных необдуманных действий пользователя необходимо было прописать обработчик исключений. Изначальные требования к нему указаны в Техническом задании в Приложении А.

Но в итоге работы пришлось переработать некоторые из них и создать новые, которые оказались необходимы при тестировании приложения.

Пример кода:

```
If KolRes[1]+KolRes[2]+KolRes[3]=0 then begin
  ShowMessage('Выберите хотя бы одного получателя!');
  exit;
end;
```

Тестирование и итоговая шлифовка продукта

В заключительном этапе реализации мы отправили приложение на тестирование, разрешили замеченный баг с именем пользователя (так как переменная называлась Name, компьютер путал её со свойством формы Fusers.Name и при вводе служебных символов выводил ошибку).

Дальше были убраны предупреждения и написаны комментарии к сложным моментам кодам. На этом разработка завершилась.

5. ИСПОЛЬЗУЕМЫЙ ИНСТРУМЕНТАРИЙ И РАБОЧЕЕ ОКРУЖЕНИЕ

Для разработки программы был выбран язык программирования Delphi. Компилятор поддерживающий язык и обладающим достаточным набором инструментов был Lazarus ADI v.2.2.0

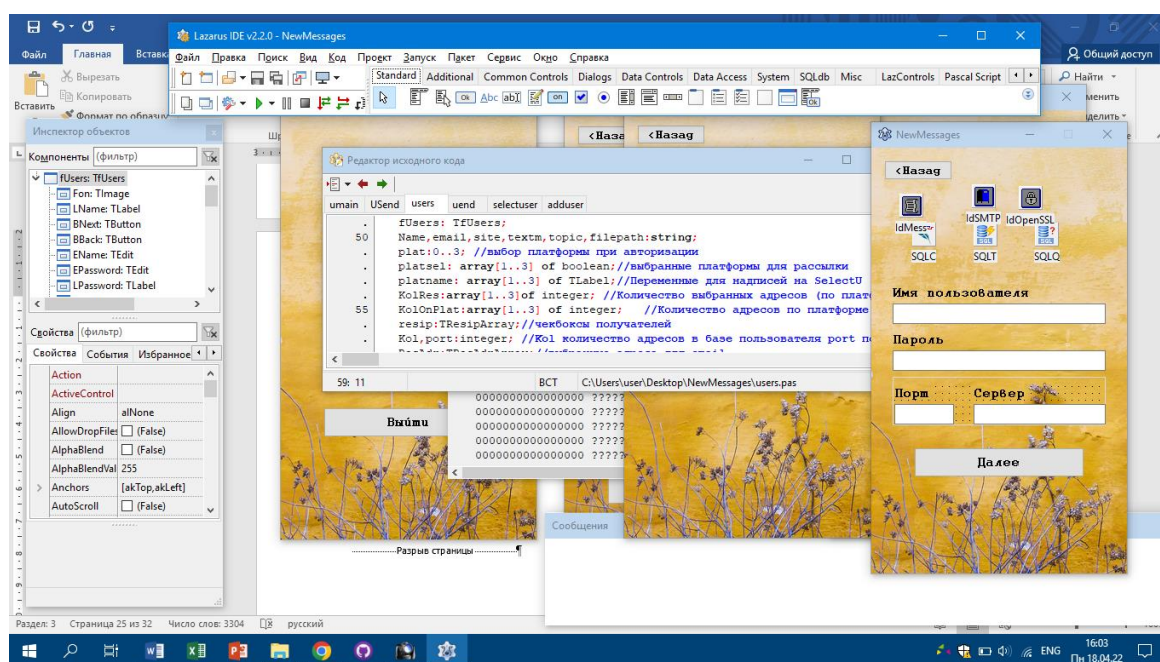


Рисунок 24 — Lazarus ADI v.2.2.0.

В Lazarus ADI v.2.2.0 с помощью стандартного набора инструментов были созданы рабочие формы приложения и написаны обработчики для переходов, а также неспецифические исключения:

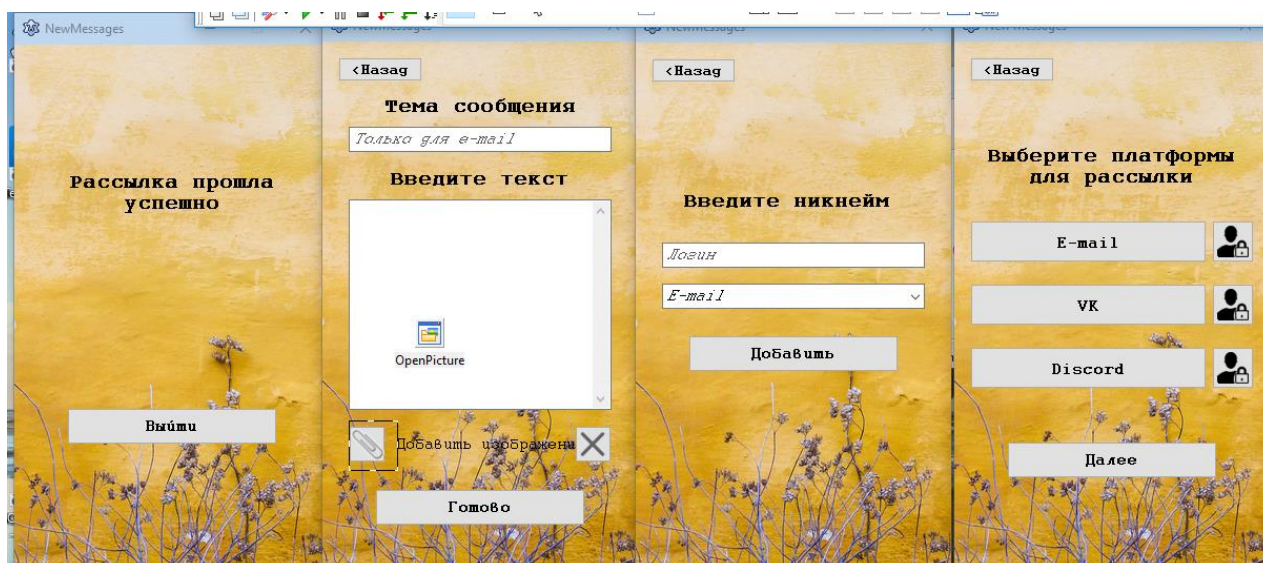


Рисунок 25 — Формы приложения.

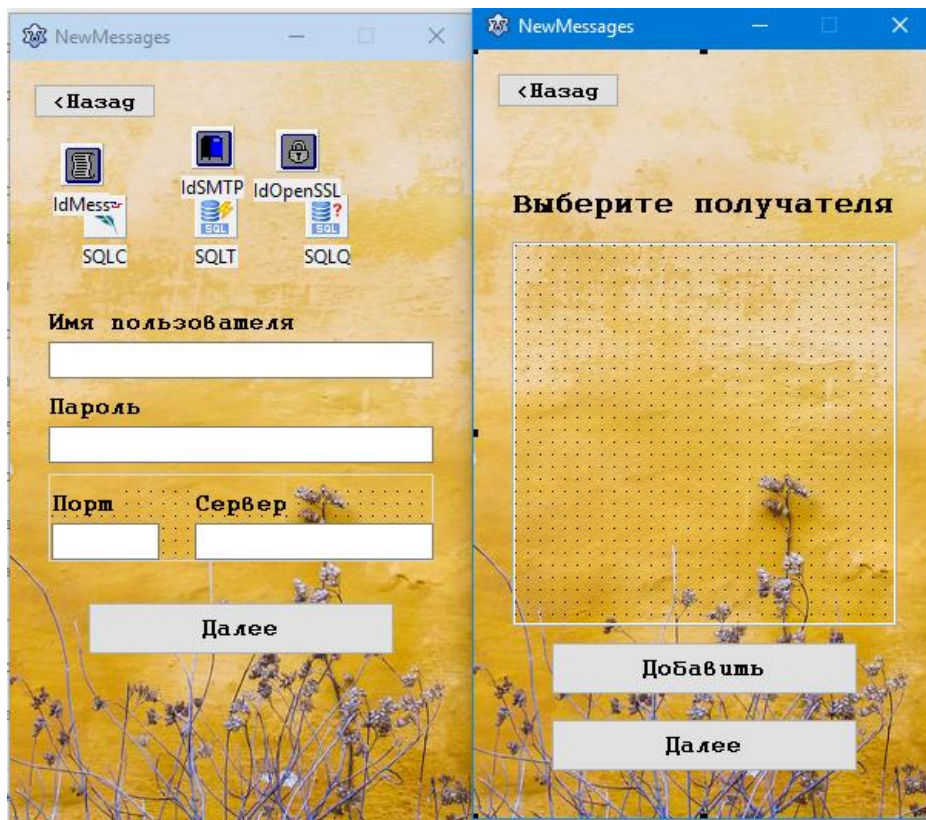


Рисунок 26 — Формы приложения.

Для создания базы данных было использовано приложение SQLite Studio 3.3.3.

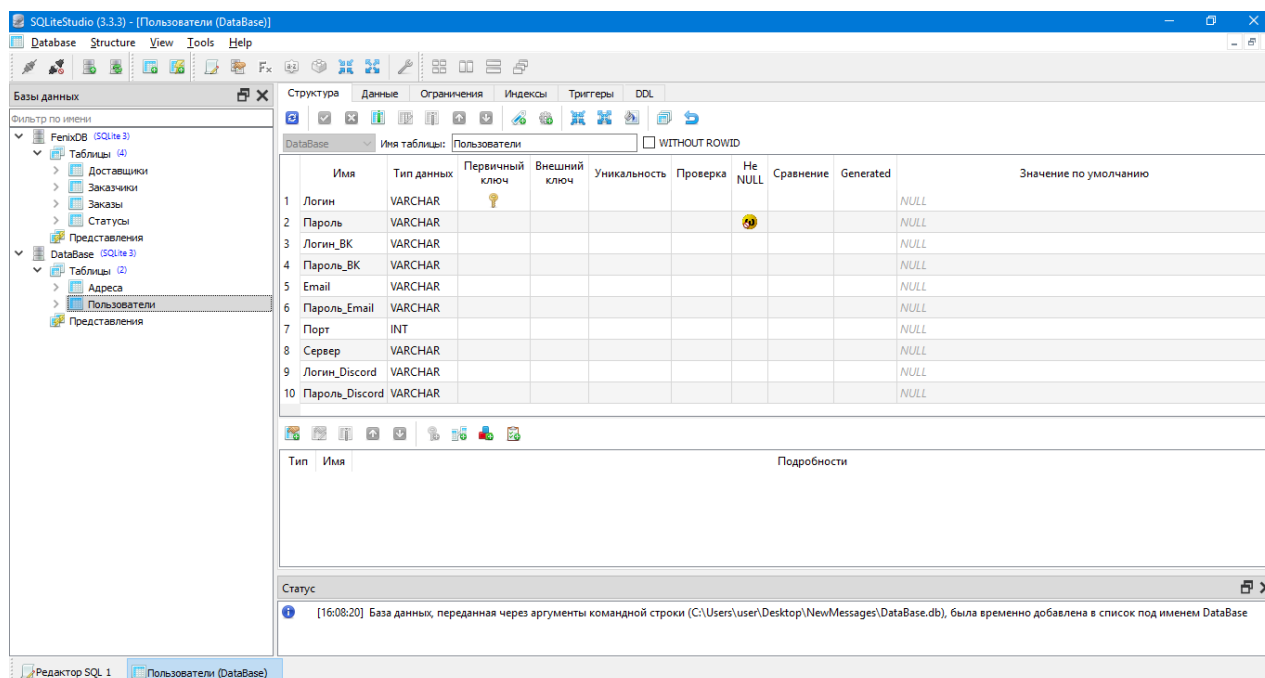


Рисунок 27 — SQLite Studio.

Так же для использования SQL запросов к проекту была подключена библиотека sqlite3.dll

Для отправки сообщений на почту был загружен пакет Indy10, а также в проекте использовались библиотеки libeay32.dll и ssleay32.dll.

Для отправки сообщений в Discord использовалась библиотека Synapse.

Synapse предоставляет простой в использовании последовательный порт и возможность подключения по TCP/IP.

Для отправки сообщений в Telegram использовалась библиотека fptelegram.

6. СИСТЕМА РАЗДЕЛЕНИЯ ТРУДА

Задачи и обязанности в команде распределялись по способностям, интересам и по сложности. В начале разработки капитан команды распределил основные направления деятельности каждого участника. В некоторых случаях участники выполняли задачи, которые не предусматривались в направлении их деятельности, в таких ситуациях при назначении задачи капитан полностью был уверен в способности участника команды выполнить представленную им задачу.

Капитан команды придумывал задачи для каждого участника и расписывал особенности каждой задачи, участники могли просмотреть все назначенные задачи в YouTrack, там же участники отчитывались о завершении задачи и прикрепляли отчёт о ходе выполнения назначенной задачи, а также указывали время, затраченное на выполнение каждой задачи.

YouTrack — это инструмент управления проектами.

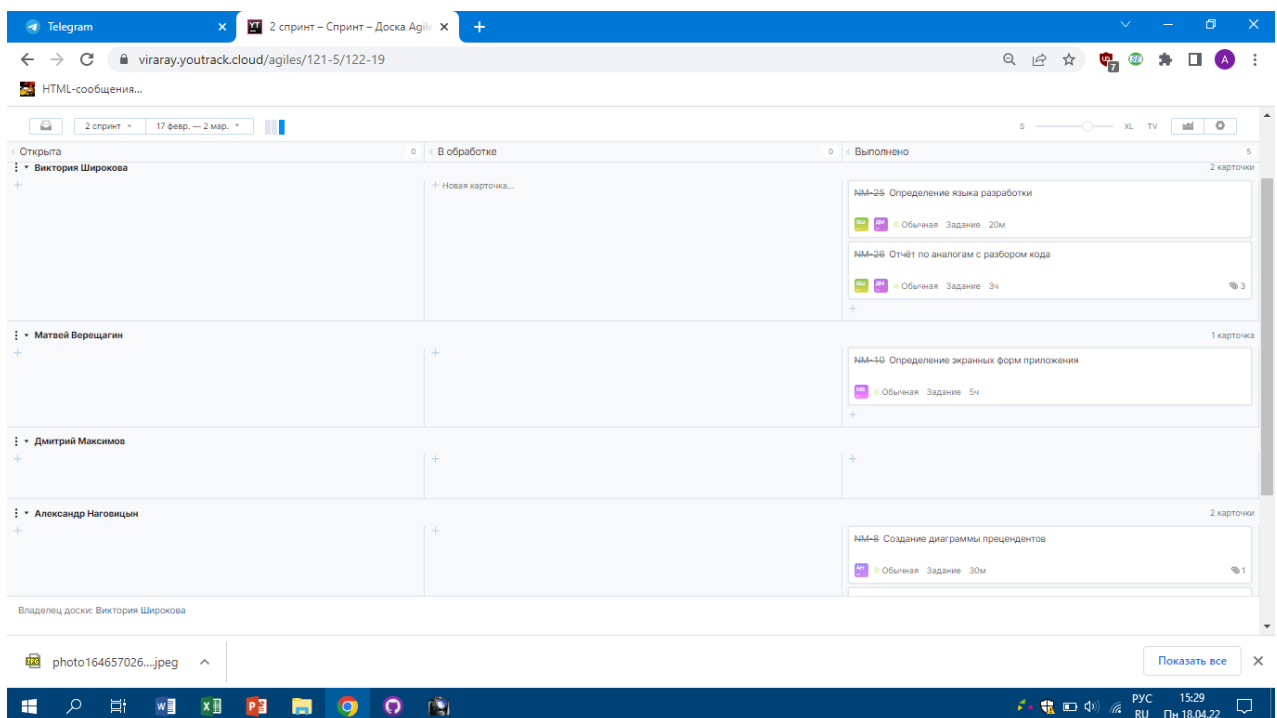


Рисунок 28 — YouTrack.

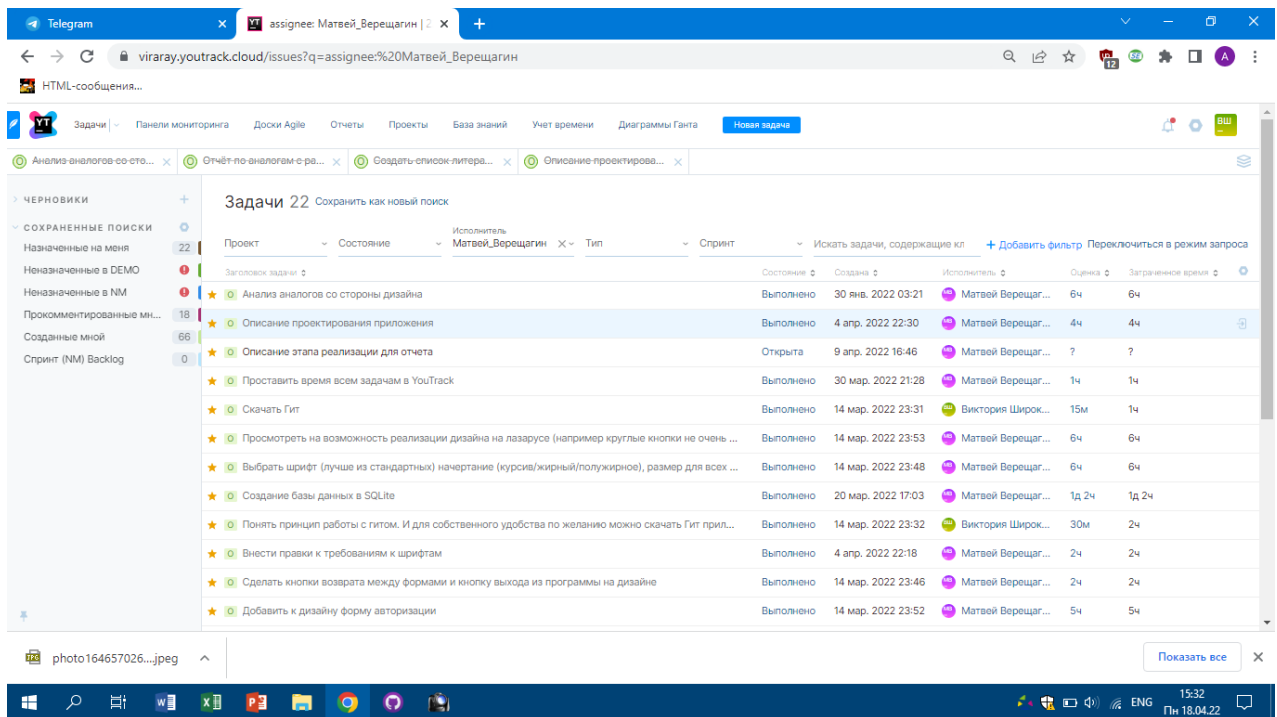


Рисунок 29 — Поставленные задачи в YouTrack.

Капитан команды составлял отчёт после каждого третьего спринта, чтобы было наглядно видно, как выполняются задачи каждого участника за данные спринты.

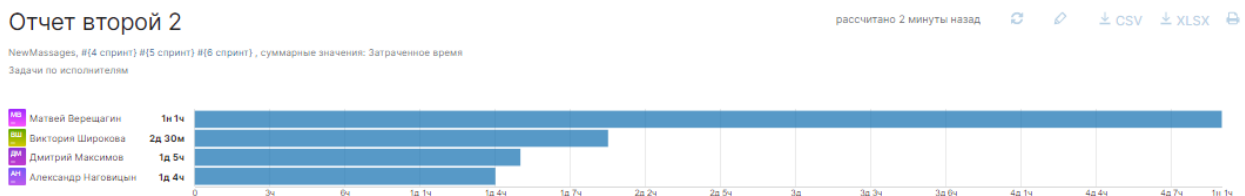


Рисунок 30 — Отчет за 4-6 спринты.

Капитан был ответственен за проведение брифингов, а также за оповещение команды о предстоящем брифинге. Они проводились в формате онлайн, в программе Microsoft Teams, где также происходила запись каждой встречи:

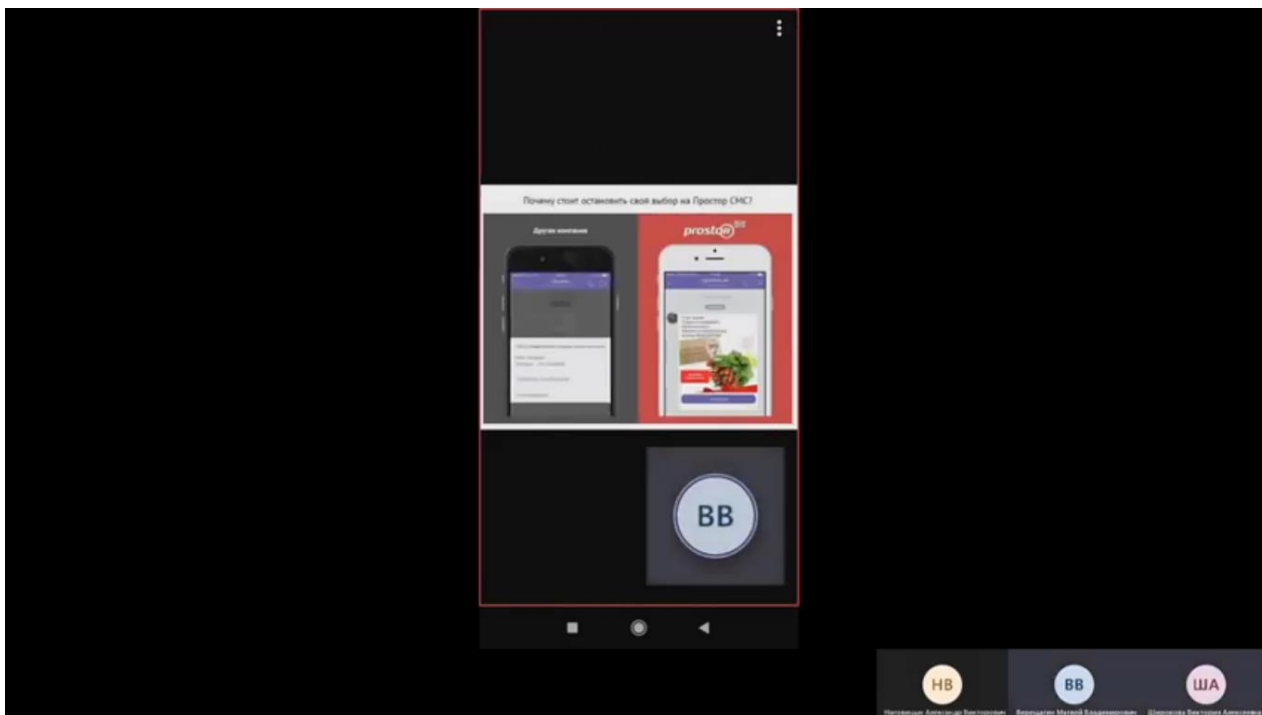


Рисунок 31 — Брифинг в Microsoft Teams.

После чего запись публиковалась на канале YouTube для дальнейшего просмотра преподавателем практики:

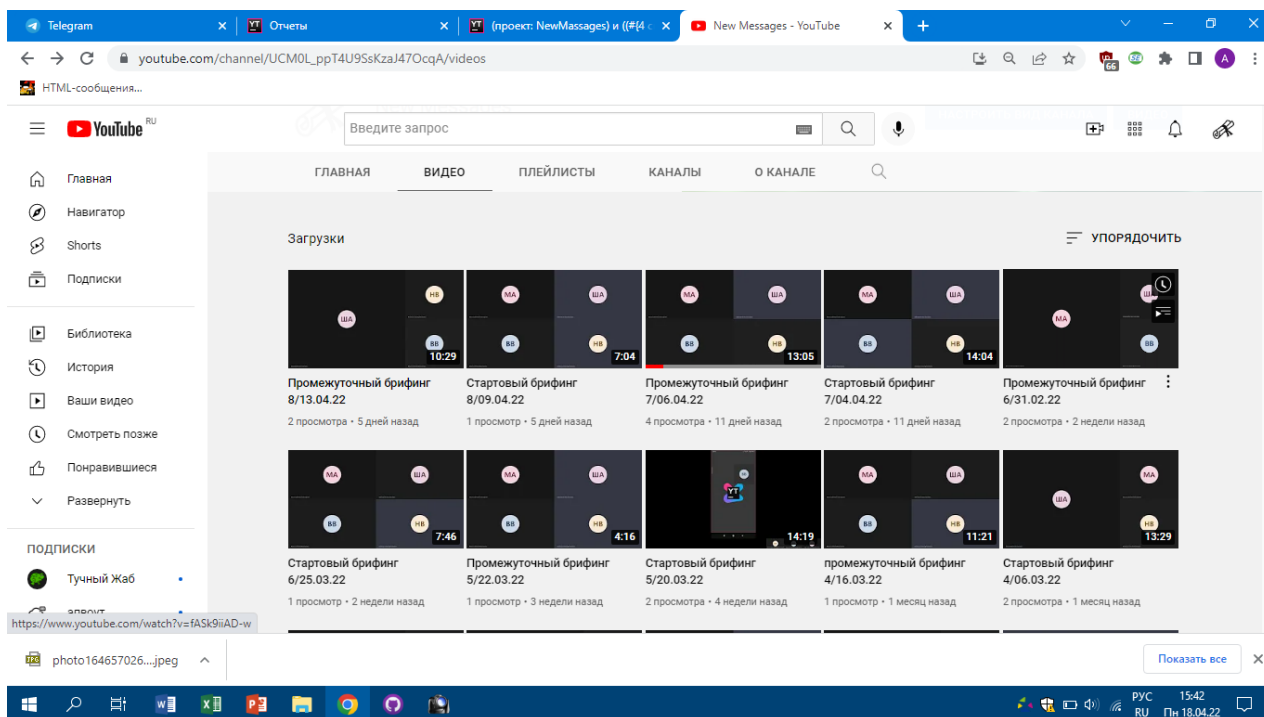


Рисунок 32 — Канал с записями брифингов.

Спринт длился неделю. В каждом спринте команда должна была провести установочный брифинг, на котором назначались задачи на неделю, промежуточный, на котором происходило обсуждение текущих результатов и

корректировка задач, и заключительный, на котором команда должна была отчитаться о проделанной работе руководителю практики. Заключительный брифинг проводился в очной форме, в назначенный день, в присутствии преподавателя.

На брифингах участники команды задавали вопросы по назначенным задачам и отчитывались за уже выполненные задачи, обсуждали дальнейший ход работы и спорные моменты проекта.

После этапа проектирования наступил этап совместной разработки программы. Для облегчения совместной работы необходим был инструмент объединяющий разные версии работы от разных участников в одну, с возможностью откатить изменения. Чтобы каждый участник на своём ПК мог изменить или добавить файлы. Для этого командой была использована систему управления версий GitHub и клиент GitHub Desktop для загрузки изменений на сайт с локального компьютера.

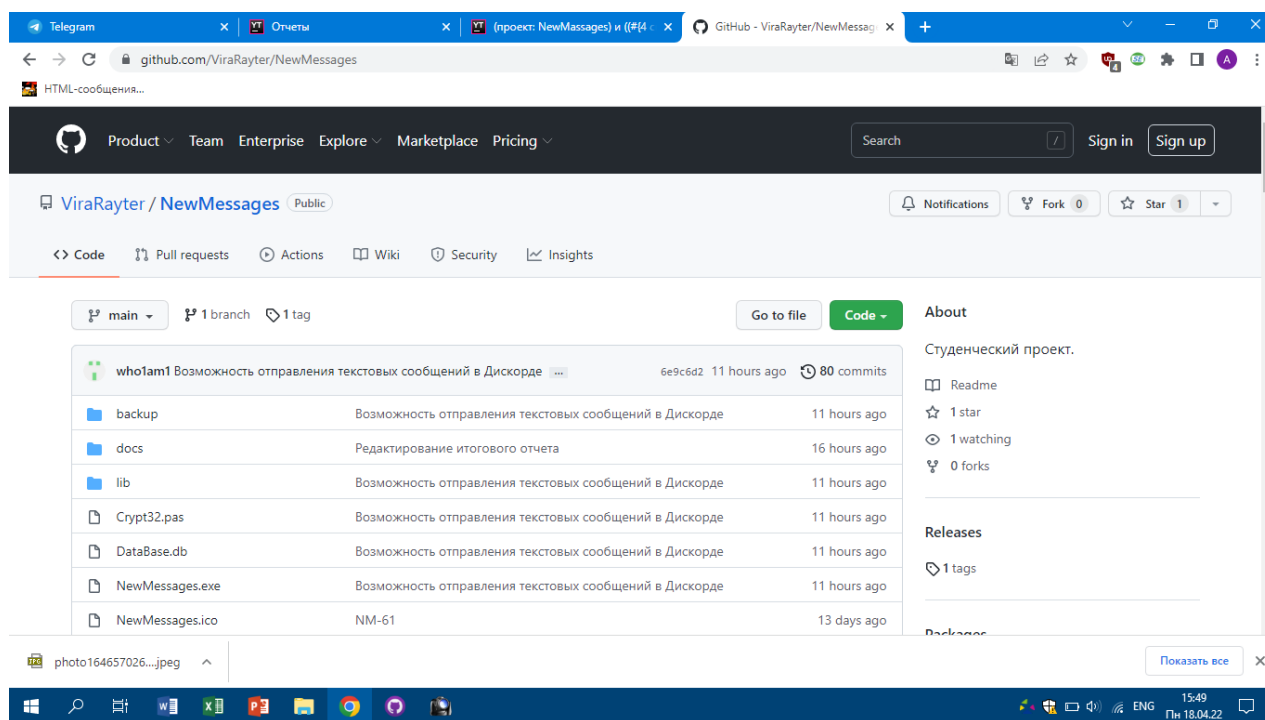


Рисунок 33 — GitHub.

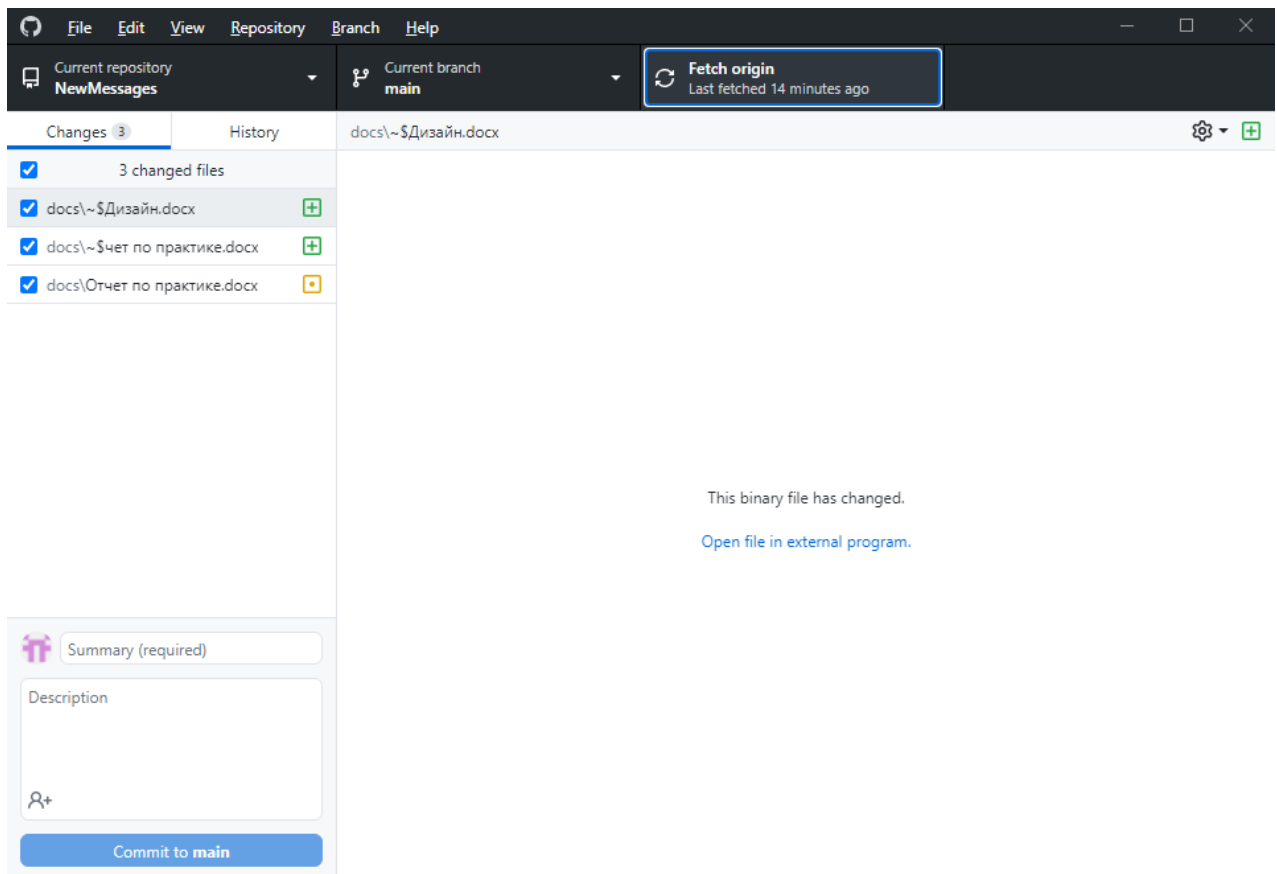


Рисунок 34 — GitHub Desktop.

ЗАКЛЮЧЕНИЕ

За время прохождения учебной практики в команде были закреплены теоретические знания и практические навыки работы в Lazarus'е, а также приобретены навыки использования Git Hub'а, Youtrack'а. Также при работе в команде улучшились коммуникативные способности и приобретён опыт командной работы. Командой приобретён навык разработки ПО спринтами. Участники поняли, что цель команды лучше всего достигать спринтами, то есть короткими забегами, это помогает оценивать и корректировать направление движения в ходе разработки.

В ходе учебной практики, участники осознали, что командная работа включает в себя ответственность каждого члена команды за свою работу и за работу команды, и что невыполнение поставленной задачи может принести вред всем участникам команды.

Участники команды считают, что учебная практика необходима для закрепления теоретических знаний, полученных в рамках учебной программы, способствует приобретению необходимых компетенций.

Таким образом, программа учебной практики ПМ.07 в период с 17.01.2022 по 01.05.2022 была выполнена в полном объеме.

При выполнении работ, предусмотренных заданием на практику, в организации ФГБОУ ВО «Вятский государственный университет» мной, **Ивановым Иваном Ивановичем**, соблюдались правила внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологические правила и гигиенические нормативы. Пройден вводный инструктаж и инструктаж на рабочем месте. Замечаний не имел.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архангельский, А.Я. Программирование в Delphi. Учебник по классическим версиям Delphi (+ дискета); Бином - М., 2006. - 439 с.
2. Культин, Никита Основы программирования в Delphi 7; СПб: БХВ - М., 2003. - 608 с.
3. Марков, Е.П.; Никифоров, В.В. Delphi 2005 для .NET; БХВ-Петербург - М., 2005. - 896 с.
4. Понамарев, В. Базы данных в Delphi 7. Самоучитель; СПб: Питер - М., 2003. - 224 с.
5. Установка GitHub Desktop и его базовое использование [Электронный ресурс] — (<https://frontips.ru/ustanovka-github-desktop-i-ego-bazovoe-ispolzovanie/>)
6. Скотт Чакон и Бен Штрауб. Книга. Pro Git [Текст] / С. Чакон, Б Штрауб — 2-е изд., 2014.
7. Сухарев, М.В. Основы Delphi. Профессиональный подход; Наука и техника - М., 2004. - 600 с.
8. Шумаков, П.В. Delphi 3 и разработка приложений баз данных; Нолидж - М., 1998. - 704 с.

ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку программы

«NewMessages»

Введение

Наименование программы

«NewMessages».

Программа, предназначенная для рассылки сообщений и писем, в таких сервисах, как телеграм, дискорд и почта.

Краткая характеристика области применения программы

Программа предназначена для проведения рассылки различного наполнения в вконтакте, дискорде и почты. С помощью данной программы пользователь, без труда разослать большому количеству людей информацию, которую он посчитает нужной.

Данная программа повышает навыки исполнителя в сферах:

Разработки технического задания;

Реализации оконного приложения на Windows;

Применение данного приложения может распространяться в сфере маркетинга, в информационной цели для привлечь внимание к компании, напомнить о себе, создать положительный образ в глазах клиентов, выстроить доверительные отношения с ними, или в коммерческой цели рассылки, чтобы побудить читателя к действию: от перехода на нужную вам страницу до заказа или покупки. Этот тип рассылки включает себя уведомления об акциях и новых поступлениях, описания новых возможностей сервиса, письма с эксклюзивными промо-кодами, тематические подборки продуктов (например, к праздникам) с прямыми ссылками для заказа. А также в других сферах.

Основания для разработки**Сроки исполнения работ**

Начало разработки – 28.01.2022.

Конец разработки - 30.04.2021.

Заказчик.

Преподаватель колледжа ВятГУ Чистяков Геннадий Андреевич.

Исполнители

Студенты колледжа ВятГУ группы ИСПк-302-52-00: Широкова Виктория Алексеевна, Максимов Дмитрий Александрович, Верещагин Матвей Владимирович, Наговицын Александр Викторович.

Назначение разработки

Функциональное назначение программы

Данная работа способствует приобретению и улучшению следующих навыков исполнителя, приобретаемых в рамках дисциплин третьего курса:

- Разработки технического задания;
- Составления алгоритмов программы;
- Написание кодов программ на языке программирования Lazarus.
- Подключение базы данных и её использование в выбранной среде разработки ПО.
- Работа в коллективе.

Эксплуатационное назначение программы

Программа должна эксплуатироваться как приложение на персональном компьютере под операционной системой Windows.

Требования к программе

Требования к функциональным характеристикам

Система должна обеспечивать выполнение следующих функций:

1. Отправлять электронные письма пользователям.
2. Возможность производить рассылку в социальной сети Телеграм, в сервисе Дискорд и через электронную почту.
3. Возможность одновременной рассылки сообщения, в выбранных сервисах.
4. Выбор получателя из списка по идентификационным данным.
5. Возможность добавить адреса пользователей сервисов для рассылки сообщений.
6. Разделение получателей рассылок, на группы соответствующих сервисов (Дискорд, Email, Телеграм).
7. Возможность отправить письмо нескольким получателям.
8. Ведение базы данных получателей рассылок.
9. Каждый пользователь должен быть идентифицирован данными от своего аккаунта на сервисе, при выборе сервиса, на котором происходит рассылка.
10. Проверка данных заносимых в базу данных адресов получателя в соответствии с форматом, который используется в сервисе данной таблицы (Правильный формат почтового адреса и т.п.)

Требования к надежности

1.1.1 Требования к обработчику исключений

Пользовательские исключения:

1. При добавлении данных аккаунтов в базу данных, должно появляться окно, в котором будет показан результат проверки правильного ввода данных, проверка на пустое поле и проверка на наличие данного адреса в базе данных, а именно в базу данных добавлено столько-то адресов, не добавлено столько-то, можно будет посмотреть данные, которые не прошли проверку и не были добавлены в базу данных.
2. При достижении лимита рассылки, который является индивидуальным для разных сервисов, должно выводиться предупреждающее окно, что превышен лимит, а также кнопка отправки стаёт не действующей, до момента пока лимит не будет сброшен.
3. При превышении символов в сообщении рассылки, невозможно начать рассылку, а также появляется предупреждение, на котором написано сколько символов нужно удалить.
4. Невозможно начать рассылку, если сообщение пустое, также появляется предупреждение при наведении на кнопку начала рассылки.
5. При начале рассылки в которой нет адресов получателей, также будет показано предупреждение.

При этих исключениях должно выводиться сообщение с пояснением возникшего исключения. В остальных, менее распространённых случаях должно выводиться сообщение об ошибке без пояснения. После сообщения об ошибке пользователя возвращают к предыдущему действию, если это возможно.

Требования к обеспечению надёжного функционирования системы

Надёжное функционирование программы должно быть обеспечено выполнением пользователем совокупности организационно-технических мероприятий:

Организацией бесперебойного питания технических средств;

Использованием лицензионного программного обеспечения.

Контроль входной и выходной информации

Необходима проверка наличия файла с исходными данными на устройстве для того, чтобы программа не завершалась аварийно.

Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не должно превышать времени, требуемого для перезапуска системы после устранения всех неисправностей, при условии соблюдения условий эксплуатации технических и программных средств.

Допустимые потери данных при отказе

Возможна частичная потеря данных находящихся в базе данных.

Требования к пользователю

Пользователь должен иметь следующие навыки:

Навыки использования операционной системы Windows;

Навыки чтения;

Умение пользоваться клавиатурой и мышью.

Иметь понятие рассылки и её целей

Требования к составу и параметрам технических средств

К техническим средствам предъявляются следующие требования:

- IBM-совместимая электронная вычислительная машина;
- Микропроцессор с архитектурой x86 и частотой от 1,8 ГГц;
- Манипулятор типа «мышь»;
- Операционная система Windows, начиная с 10 версии.

Требования к исходным кодам и языкам программирования

Для реализации системы должен использоваться язык Free Pascal.

Требования к программным средствам, используемым программой

Разрабатываемая программа должна исполняться в среде операционной системы Microsoft Windows версии не ниже 10.

Требования к программным средствам, используемым приложением не предъявляются.

Требования к защите информации и программ

Шифрование конфиденциальных данных (вебхуков и id), хеширование паролей.

Требования к программной документации

Система должна сопровождаться комплектом документации согласно ГОСТ ЕСПД:

описание программы;

текст программы;

описание применения.

Технико-экономические показатели

Ориентировочная экономическая эффективность не рассчитывается и аналогии программных изделий не проводятся.

Стадии и этапы разработки

Стадии разработки

Разработка должна быть проведена в три стадии:

разработка технического задания;

рабочее проектирование;

тестирование приложения.

Этапы разработки

согласование и утверждение ТЗ (до 28.02.2022);

разработка программы (до 21.04.2022);

испытание программы (до 30.04.2021).

Порядок контроля и приемки

Виды испытаний

Приемо-сдаточные испытания должны проводиться в оговоренные сроки.

Приемо-сдаточные испытания программы должны проводиться с использованием разработанных Исполнителем и согласованных Заказчиком тестовых примеров.

8 Рабочий проект

8.1 Структура базы данных

База данных хранит данные аккаунта пользователя в приложении, а также адреса пользователей, которые участвуют в рассылке. Данная база данных представлена на рисунке 1.

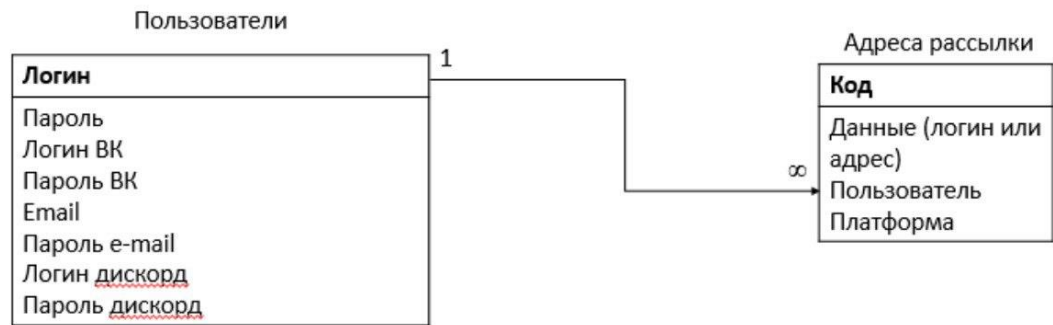


Рисунок 1 – База данных приложения NewMessage.

8.2 Построение диаграммы прецедентов

На начальном этапе проектирования важно построить укрупненную диаграмму деятельности программы в виде диаграммы вариантов использования. Диаграмма прецедентов (вариантов использования) – диаграмма поведения, на которой показаны множество прецедентов, которые представляют собой законченную последовательность действий и субъектов (исполнителей) и отношения между ними. Исполнители могут быть внешними или внутренними и представлять собой личность, организацию или систему, которая взаимодействует с ИС. Строится на этапе создания концептуальной модели в виде общей диаграммы деятельности и для описания бизнес-деятельности в виде моделей отдельных бизнес-прецедентов. Подобно инструментарию BPWin, можно с помощью нескольких диаграмм представить модель на нескольких уровнях функционирования. Детальное описание прецедента представлено на рисунке 2.

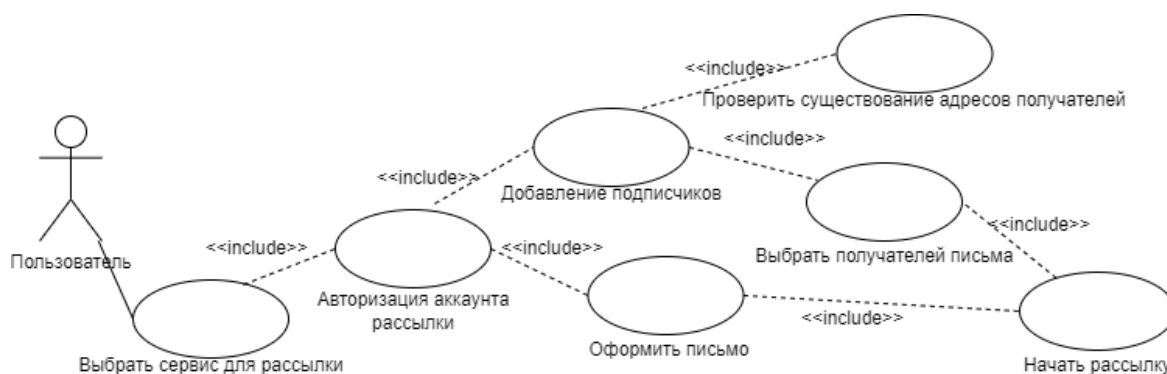


Рисунок 2 – Диаграмма вариантов использования (Use Case Diagram).

8.3 Разработка интерфейса

Для работы с программным продуктом используется простой, интуитивно понятный интерфейс. Последовательность работы с объектами формы определяется доступностью командных кнопок, целостность данных определяется набором используемых в программе проверок.

Переход от одного объекта формы к другому осуществляется при нажатии мыши по соответствующему объекту.

Для дизайна была выбрана палитра в осенних тонах:



Рисунок 3 – Палитра.

На основе данной палитры был разработан логотип:



Рисунок 4 – Логотип.

Прототип программы разрабатывался на сайте Figma. Прототипы форм представлены на следующих рисунках.

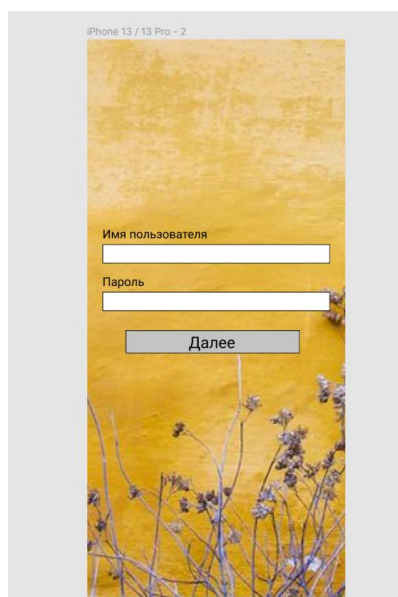


Рисунок 5 - Авторизация



Рисунок 6 - Выберите платформу

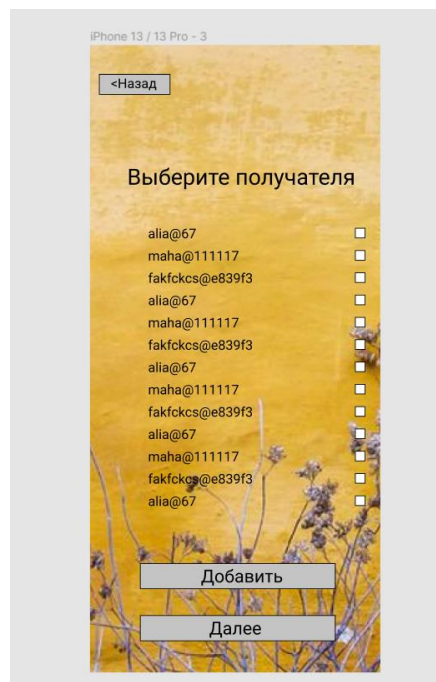


Рисунок 7 - Выберите получателя

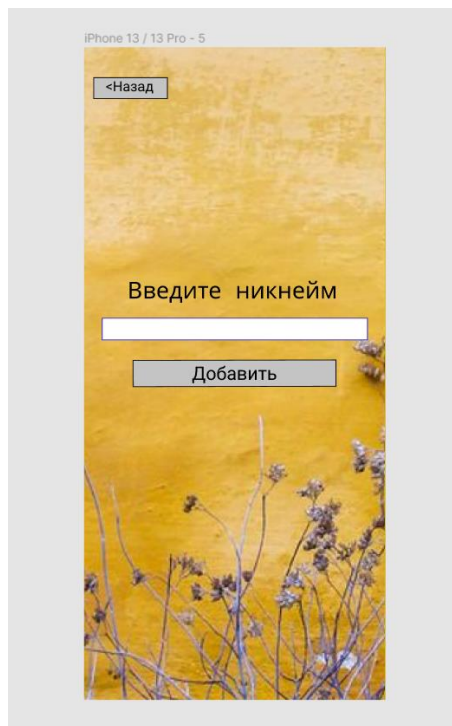


Рисунок 8 - Введите никнейм

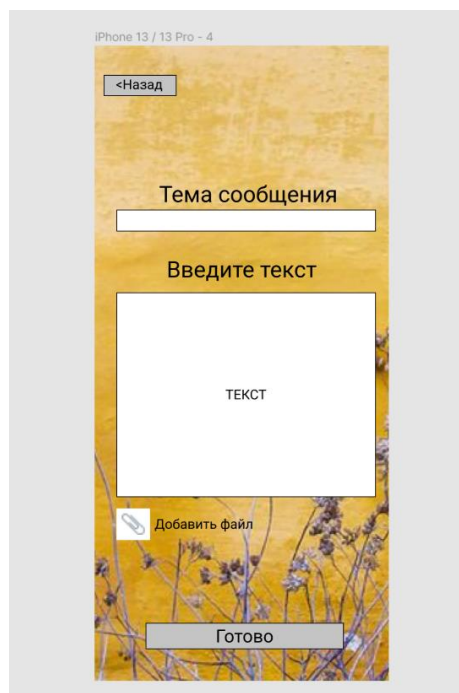


Рисунок 9 - Введите текст

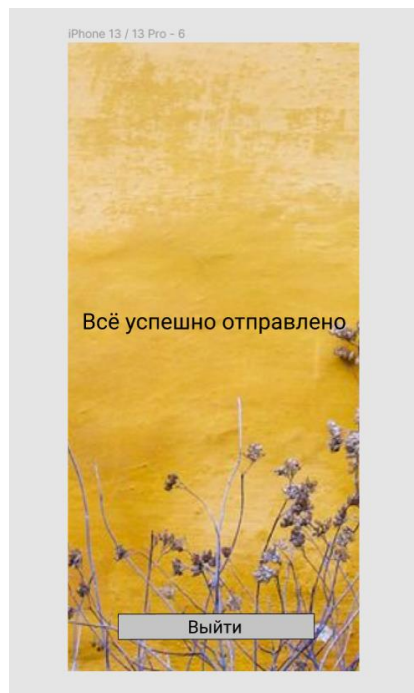


Рисунок 10 - Всё успешно отправлено

Шрифт для проекта был выбран Courier с данными характеристиками

1. Заголовки 15 Полужирный
2. Edit 12 Наклонный
3. Кнопки 12 Жирный
4. Кнопка «Назад» 10 Полужирный
5. Имя пользователя и Пароль 12 Полужирный
6. Добавить файл 12 Обычный

8.4 Инструкция пользователя

При запуске приложения пользователя встречает форма, которая представлена на рисунке 11. На этой форме пользователь должен ввести данные от своего аккаунта, если аккаунта ещё нет, придумать имя и пароль, ввести в соответствующие поля и нажать кнопку далее. Каждый пользователь имеет свой аккаунт, на котором хранятся ранее добавленные участники рассылок.



Рисунок 11 – Начальная форма.

После авторизации, пользователь попадает на форму, представленную на рисунке 12. Эта форма отвечает за выбор сервисов, на которых пользователь будет осуществлять рассылку. На форме есть три кнопки, на которых написаны названия сервисов и три кнопки, которые располагаются правее кнопок с сервисами. Для выбора сервиса, нужно сначала нажать на кнопку, которая находится правее выбираемого сервиса и авторизоваться для рассылки на данном сервисе. Можно одновременно выбирать несколько сервисов.



Рисунок 12 – Форма выбора сервиса.

После прохождения авторизации, пользователя встречает форма (рисунок 13), на которой он должен выбрать участников рассылки, тех кто получит его сообщение. При первой авторизации, тут будет пусто, нужно сначала добавить адреса этих участников, для этого пользователю нужно нажать на кнопку добавить, после чего он перейдет на форму (рисунок 14), на которой он введет адрес, ник или id пользователя, который будет участвовать в рассылке. После добавления пользователь вернется на форму выбора участников рассылки. Выберет участников, путем проставления галочек на против выбранного адреса пользователя и нажмет далее.

The screenshot shows a mobile application window titled 'NewMessages'. At the top left is a back button labeled '<Назад'. Below it is the instruction 'Выберите получателя' (Select a recipient). A list of 12 items, labeled 'Пример1' through 'Пример12', is displayed, each with an unchecked checkbox to its left. The list is contained within a scrollable container. At the bottom of the screen are two buttons: 'Добавить' (Add) and 'Далее' (Next).

Рисунок 13 – Форма выбора участников рассылки.

The screenshot shows the same 'NewMessages' application window. The instruction 'Введите никнейм' (Enter nickname) is displayed. Below it are two input fields: the first is labeled 'Логин' (Login) and the second is labeled 'E-mail' with a dropdown arrow. At the bottom is a single 'Добавить' (Add) button.

Рисунок 14 – форма добавления участника для рассылки.

Как только пользователь выбрал участников рассылки и нажал далее он попадает на форму (рисунок 15), на которой он должен заполнить поля тема сообщения и поле самого

сообщения, также он может прикрепить файл (чтобы удалить прикрепленный файл нужно нажать на квадрат с крестиком). После необходимо нажать кнопку готово, которая отвечает за начало рассылки. Когда рассылка закончилась пользователь попадает на форму (рисунок 16), которая оповещает об успешной отсылке его сообщений.

The image shows a mobile application window titled 'NewMessages'. At the top left is a back arrow icon and the text '< Назад'. Below this is the label 'Тема сообщения' followed by a text input field containing the word 'Тема'. Underneath is the label 'Введите текст' followed by a large text area containing the word 'Привет'. At the bottom left is a paperclip icon and the text 'Добавить файл'. At the bottom right is a square icon with an 'X' inside. At the very bottom is a wide button labeled 'Готово'. The background of the form is a yellowish-orange textured surface with some dried plant stems at the bottom.

Рисунок 15 – Форма составления сообщения для рассылки.

The image shows a mobile application window with a yellowish-orange textured background and dried plant stems at the bottom. In the center, the text 'Рассылка прошла успешно' is displayed in bold. At the bottom, there is a wide button labeled 'Выйти'.

Рисунок 16 – Форма, которая показывает успешное выполнение рассылки.