

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Московский политехнический университет»**

**Кафедра «Информационные технологии»
Образовательная программа «Веб-технологии»**

Отчет по дисциплине

«Проектная деятельность»

Тема: «Система оповещений для Московского Политеха»

Выполнил:

Студенты

Гайдучик Александр Валерьевич, гр. 181-321

Моисеенко Илья Андреевич, гр. 181-321

Иванов Григорий Александрович, гр. 181-321

Шубенок Максим Валерьевич, гр. 191-321

Мазаев Роман Романович, гр. 191-321

Канатников Илья Николаевич, гр.191-321

Куратор

Жуплев Антон Сергеевич

Принял:

Старший преподаватель

Даньшина М.В.

Москва 2020

Содержание

Введение.....	3
Распределение ролей.....	4
Глава 1. Формирование требований к проекту	
1.1 Анализ существующей проблемы.....	5
1.2 Формулирование цели, задач по проекту, критериев для приемки работы.....	5
1.3 Проектирование приложения, UML диаграммы, план приёма работы.....	6
1.4 Используемые технологии при создании проекта.....	6
Глава 2. Реализация проекта	
2.1 Проектирование разделов, структура сайта.....	8
2.2 Проектирование структуры базы данных на основе задачи.....	8
2.3 Конкретная реализация задачи на серверной стороне.....	9
2.4 Конкретная реализация задачи на клиентской стороне.....	10
2.5 Разработка дизайна.....	10
Глава 3. Сопровождение проекта	
3.1 Инструкция для пользователя.....	17
3.2 Правовые вопросы.....	22
Глава 4. Результаты.....	23

Введение

Современная жизнь немыслима без эффективного управления и подсчета. Важной категорией являются системы автоматизированной обработки информации, от которых во многом зависит эффективность работы любого предприятия или учреждения. Такая система должна:

- обеспечивать получение общих и/или детализированных отчетов по итогам работы,
- позволять легко определять ошибки в работе и сообщать о них,
- обеспечивать получение и отправку информации, критической по времени, без существенных задержек,
- выполнять точный и полный анализ данных.

Современные системы, которые использует Московский Политех в качестве оповещения студентов, позволяют отправлять изменения на почту из системы lms, но эти системы не осуществляют полного, удобного для студентов функционала и удобства.

Конечно, на данный момент, в университете, используется оповещение через почту который в некоторой степени пользуются студенты оповещения через систему lms и оповещения в личном кабинете, однако часто приходят сообщения , который совсем не нужны и только отталкивают студентов.

Система, которую разрабатывает наша команда, позволит не только учитывать «вход/выход» и «оповещения об изменениях, связанных с учебой студентов», но и многие другие критерии, о которых будет написано далее в документации.

Распределение ролей

Наша команда распределила роли следующим образом:

Гайдучик Александр - team lead, подготовка документации, notion (кейсы, модели)

Моисеенко Илья Андреевич - frontend, верстка, support по бэку

Иванов Григорий Александрович - backend, работа с базой

Шубенок Максим Валерьевич - разработка дизайна (figma), верстка

Мазаев Роман Романович - подготовка отчётов, notion

Канатников Илья Николаевич - backend

Глава 1. Формирование требований к проекту

1.1 Анализ существующей проблемы

При анализе была учтена система оповещений, имеющаяся на данный момент в Московском Политехе:

Изучая систему оповещений Московского Политеха, нами были выяснены следующие методы работы:

В настоящее время вывод оповещений происходит после того как студент получает оценку или комментарий к работе, сообщение от преподавателя в системе lms на почту, однако, часто эти сообщения попадают в спам и студенту в общем не удобно просматривать их. В личном кабинете оповещений вообще почти нет, поэтому, если там приходит какое-то сообщение, студент об этом не узнает. Поэтому нужно объединить эти уведомления и поставить их на площадку, которая будет максимально удобна для студента (vk, telegram).

Вероятно, в анализе не были учтены все ситуации, но в процессе решения и создания сервиса, команда максимально учла все идеи для реализации, в том числе после встречи с представителем кафедры.

1.2 Формулирование цели, задач по проекту, критериев для приемки работы

В ходе изучения анализа существующего решения, анализа востребованности решения и изучения аналогов мы сформулировали следующие задачи и цели:

1. Разработка прототипа
2. Встреча с преподавателями и студентами Московского Политеха
3. Разработка дизайна в Figma
4. Разработка API
5. Разработка frontend части
6. Разработка backend части
7. Проверка на баги/ошибки, а также доработка
8. Презентация готовой реализации в рамках «Проектной деятельности»

Таким образом, нашей основной целью является создание информационной системы для Московского Политеха.

1.3 Проектирование приложения, UML диаграммы, план приема работы

Проектирование приложения выдвинуто в несколько этапов. Для удобства ведения был использован сервис Trello, где каждый участник видел свои задачи, а также сервис Notion, в котором были описано ТЗ по каждому участнику.

На следующей UML-диаграмме расписаны методы взаимодействия студентов и преподавателей с системой:



1.4 Используемые технологии при создании проекта

При создании проекта использовались свободно распространяемые инструменты, фреймворки и язык программирования.

В качестве языка программирования для backend выступает JavaScript (информация о языке - <https://ru.wikipedia.org/wiki/JavaScript>), в качестве платформы для работы сервиса – Node.js (информация - <https://nodejs.org/>), также некоторые дополнительные компоненты, о которых будет описано далее в документации.

В качестве языка программирования для frontend использовался JavaScript, фреймворк React (информация - <https://reactjs.org/>), а также Node.JS

Так же, при работе с проектом использовался «git-flow» - распределенная система контроля версий (version control system – VCS).

Контроль версий означает что мы храним все версии редактируемых документов и можем вернуться к любой сохраненной версии в любой момент




времени, в случае появления каких-либо проблем, либо ошибок.

Команда использовала технологию «CI/CD», что означает концепцию непрерывной интеграции и доставки (CI/CD) — основа наших тестирований. Информация о CI/CD — это концепция, которая реализуется как конвейер, облегчая слияние только что закомиченного кода (то есть, только что написанного кода программы и отправленного в Github) в основную кодовую базу. Концепция позволяет запускать различные типы тестов на каждом этапе и завершать его запуском с развертыванием закомиченного кода в фактический продукт, который видит наша команда, во время тестирования.

В свою очередь, для удобства работы с проектом, мы использовали таск менеджер, о котором было написано ранее – Notion. Это - инструмент для управления проектами, который значительно упрощает рабочий процесс. Благодаря использованию, мы экономим время на переписках и оффлайн-обсуждениях. Кроме этого, такие платформы позволяют узнать, насколько эффективно и слаженно работает команда.

Были проведены встречи со студентами разных групп для обсуждения работы системы – принцип получения уведомлений и новостей, а также дополнения к интерфейсу, такие как личный кабинет и т.д.

Проводилось дополнительно проектирование БД:

 User	 Type	 Example
id	Number	1
Name	String	Aleksandr
Surname	String	Gaiduchik
Login	String	rouvas
Password	String	sha-256
vkToken	String	895153
tgToken	String	228148
vkid	String	null
tgid	String	null
isAdmin	Boolean	False
Messages	Type	Example
id	Number	2368
service	String	moodle
isdeliveredVK	Boolean	True
isdeliveredTG	Boolean	False
text	String	Есть обновления в ЛМС по предмету математика

Глава 2. Реализация проекта

2.1 Проектирование разделов, структура сайта




В результате изучения задачи были выведены следующие разделы сайта (для удобства введем «Администратор», «Пользователь»):

1. «Администратор» - управление системой, базой данных и основным функционалом информационной системы, а также просмотр статистики
2. «Пользователь» - интерфейс просмотра уведомлений, управление привязками к различным мессенджерам

2.2 Проектирование структуры базы данных на основе задачи.

В ходе подготовки проекта, нами было решено использовать базу данных с, минимум, двумя таблицами: «user» - для хранения данных о конкретных пользователях и «messages» - таблица текста сообщений и их направления

Структура базы данных User

 User	 Type	 Example
id	Number	1
Name	String	Aleksandr
Surname	String	Gaiduchik
Login	String	rouvas
Password	String	sha-256
vkToken	String	895153
tgToken	String	228148
vkid	String	null
tgid	String	null
isAdmin	Boolean	False

Структура базы данных сообщений

Messages	Type	Example
id	Number	2368
service	String	moodle
isdeliveredVK	Boolean	True
isdeliveredTG	Boolean	False
text	String	Есть обновления в ЛМС по предмету математика

2.3 Конкретная реализация задачи на серверной стороне

В вопросе реализации серверной стороны, далее backend, нами были использованы следующие технологии:

1. Node.JS – использование этой программной платформы позволяет сделать проект масштабируемым, позволяет обрабатывать сразу много соединений за раз. Для каждого соединения вызывается функция обратного вызова, однако, когда соединений нет Node.js ожидает в состоянии покоя. Следует обратить внимание, что этот подход контрастирует с более распространенной на сегодняшний день моделью параллелизма, в которой используются параллельные OS потоки. Такой подход является относительно неэффективным и очень сложным в использовании. Кроме того, пользователи платформы могут не беспокоиться о блокировках процессов или же зависаний, поскольку их не существует. Почти ни одна из функций в Node.js не работает напрямую с I/O, поэтому поток никогда не блокируется. В следствии этого на Node.js легко разрабатывать масштабируемые системы, чем наша команда и воспользовалась.

2. Mongoose - представляет специальную ODM-библиотеку (Object Data Modelling) для работы с MongoDB (будет описано далее), которая позволяет сопоставлять объекты классов и документы коллекций из базы данных. Грубо говоря, Mongoose работает подобно инструментам ORM. ORM – это технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

3. MongoDB – документо ориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных.

Таким образом, связка всех используемых технологий позволяет создать максимально удобный сервис, как для пользователя, так и для разработчика.

2.4 Конкретная реализация задачи на клиентской стороне

ReactJS - это декларативная, эффективная и гибкая библиотека JavaScript для создания пользовательских интерфейсов. Она позволяет создавать сложные UI из небольших и изолированных частей кода, называемых «компонентами».

NodeJS - Программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через API, подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода

Библиотеки npm - инструмент, управляющий библиотеками программирования Javascript для Node.js, данный инструмент на самом деле необходим для мира открытого исходного кода. Используется для скачивания пакетов из облачного сервера npm, либо для загрузки пакетов на эти сервера.

2.5 Разработка дизайна

Главная страница сервиса:



Poly Alert

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam ac velit lectus. Integer viverra aliquam ex. Mauris nisl turpis, feugiat at tortor at, sollicitudin tempus magna. Praesent aliquam volutpat varius. Fusce in tellus ut erat interdum vehicula non et turpis. Sed nec ex maximus, interdum nisl mollis, laculis magna. Suspendisse potenti.

[Регистрация](#)[Узнать больше](#)

О нас

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam ac velit lectus. Integer viverra aliquam ex. Mauris nisl turpis, feugiat at tortor at, sollicitudin tempus magna. Praesent aliquam volutpat varius. Fusce in tellus ut erat interdum vehicula non et turpis. Sed nec ex maximus, interdum nisl mollis, laculis magna. Suspendisse potenti.



Используемые технологии

[VK API Вконтакте](#)[API Telegram](#)

Socials



Страница регистрации пользователя:

Poly Alert

ВходРегистрация

Регистрация

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Id ut bibendum nulla eu quam. Ac malesuada ultrices ipsum turpis etiam.

Имя

Фамилия

Email

Логин

Пароль

Повторить пароль

Зарегистрироваться

После успешной регистрации, пользователь должен авторизоваться, чтобы привязать аккаунты и получать уведомления:

Poly Alert

ВходРегистрация

Вход

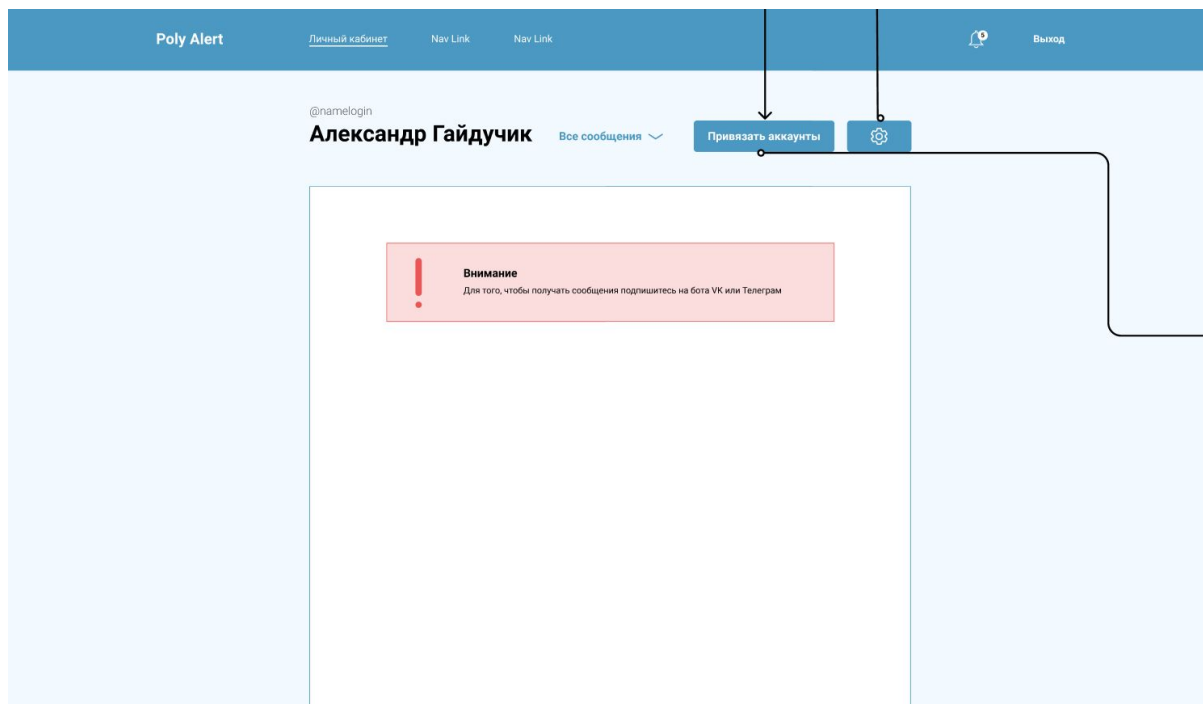
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Id ut bibendum nulla eu quam. Ac malesuada ultrices ipsum turpis etiam.

Email

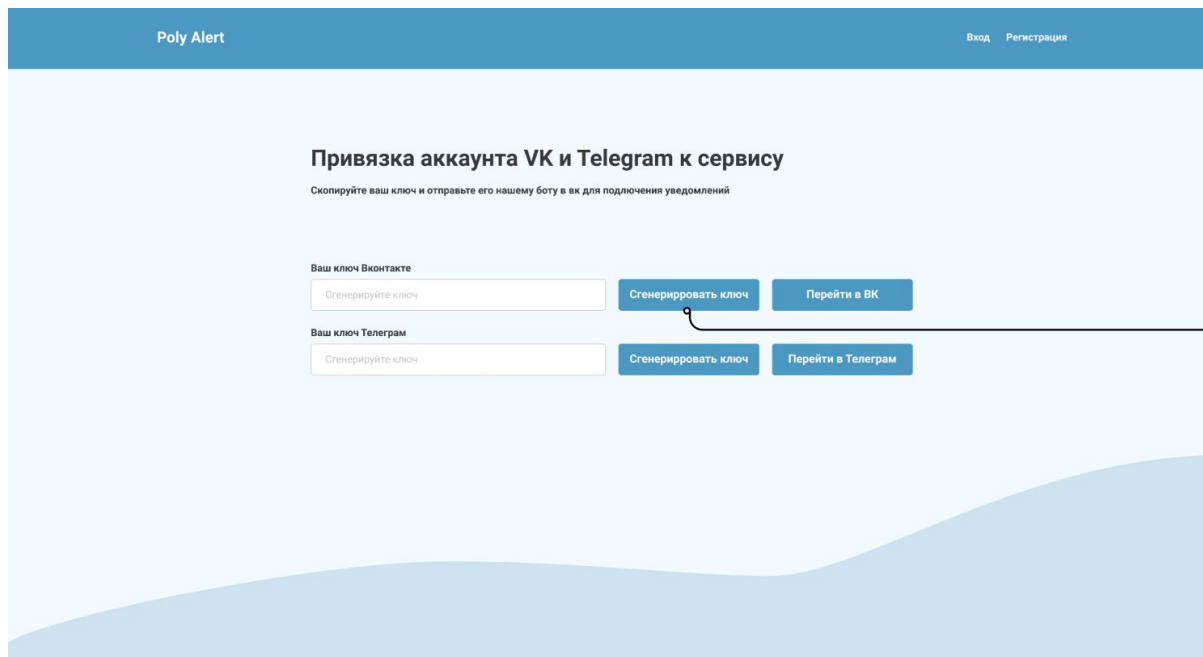
Пароль

Вход

После входа в сервис, пользователь может привязать к аккаунту мессенджеры, удобные ему, либо войти в панель настройки аккаунта:



Страница привязки аккаунтов к различным мессенджерам:



Пользователь получает ключ и после его отправки боту, получает сообщение о том, что аккаунт успешно привязан к мессенджеру:

Poly Alert

ВходРегистрация

Привязка аккаунта VK и Telegram к сервису

Скопируйте ваш ключ и отправьте его нашему боту в вк для подключения уведомлений

Ваш ключ Вконтакте

250154

Перейти в VK

Ваш ключ Telegram

Сгенерируйте ключ

Сгенерировать ключ

Перейти в Telegram

Poly Alert

ВходРегистрация

Привязка аккаунта VK и Telegram к сервису

Скопируйте ваш ключ и отправьте его нашему боту в вк для подключения уведомлений

Аккаунт Вконтакте успешно привязан ✓

Отвязать аккаунт

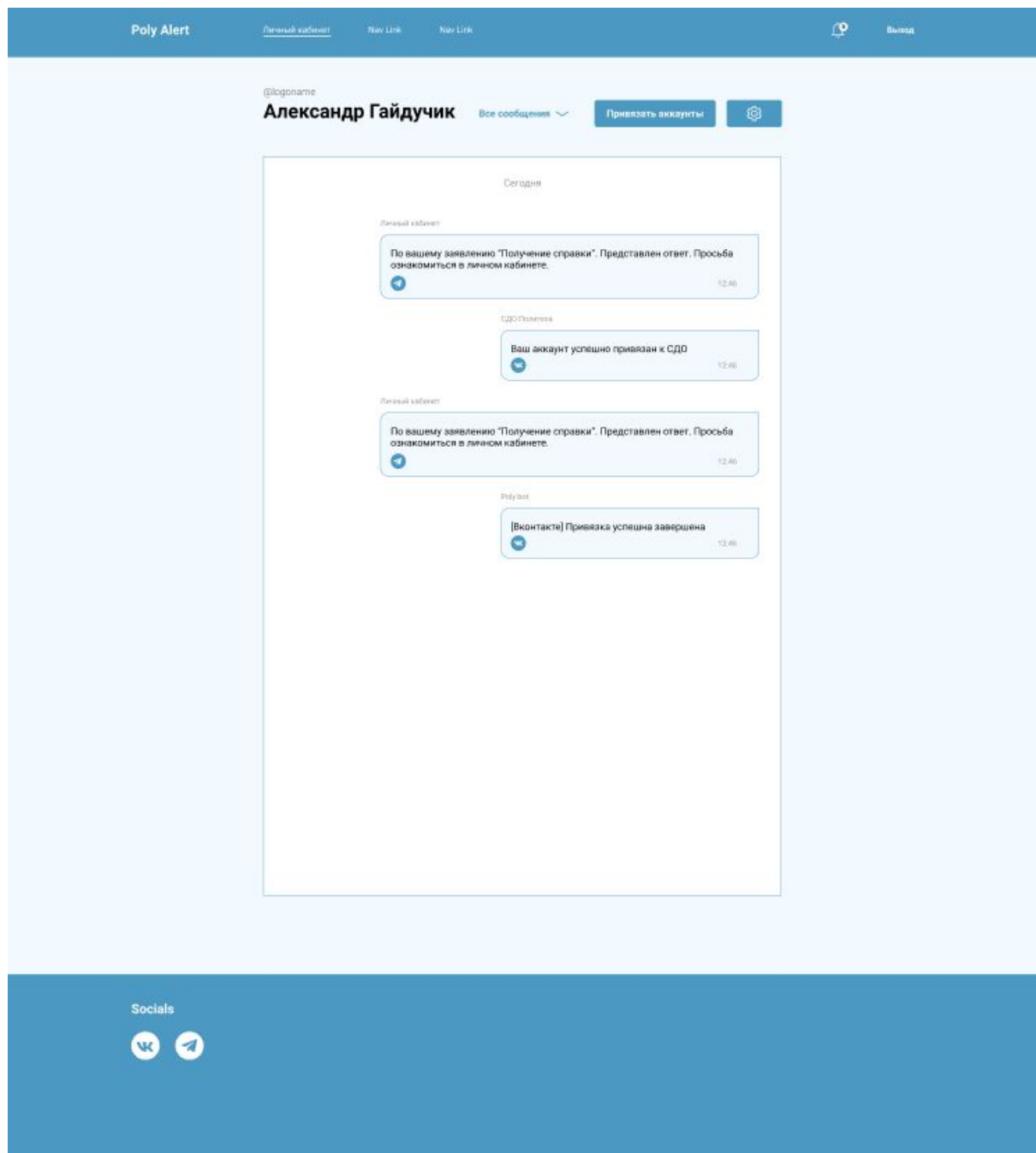
Ваш ключ Telegram

Сгенерируйте ключ

Сгенерировать ключ

Перейти в Telegram

Основной экран пользователя после успешной авторизации: здесь отображаются все уведомления, приходящие в удобный ему мессенджер:



На этой странице пользователь может настроить информацию о себе, а также сменить почту, логин или пароль:

Poly Alert

ВходРегистрация

Настройки аккаунта

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Id ut bibendum nulla eu quam. Ac malesuada ultrices ipsum turpis etiam.

Имя

Александр

Фамилия

Гайдучик

Email

gaidu4ik@mail.tu

Логин

gaidu4ik

Новый пароль

••••••••

Повторить пароль

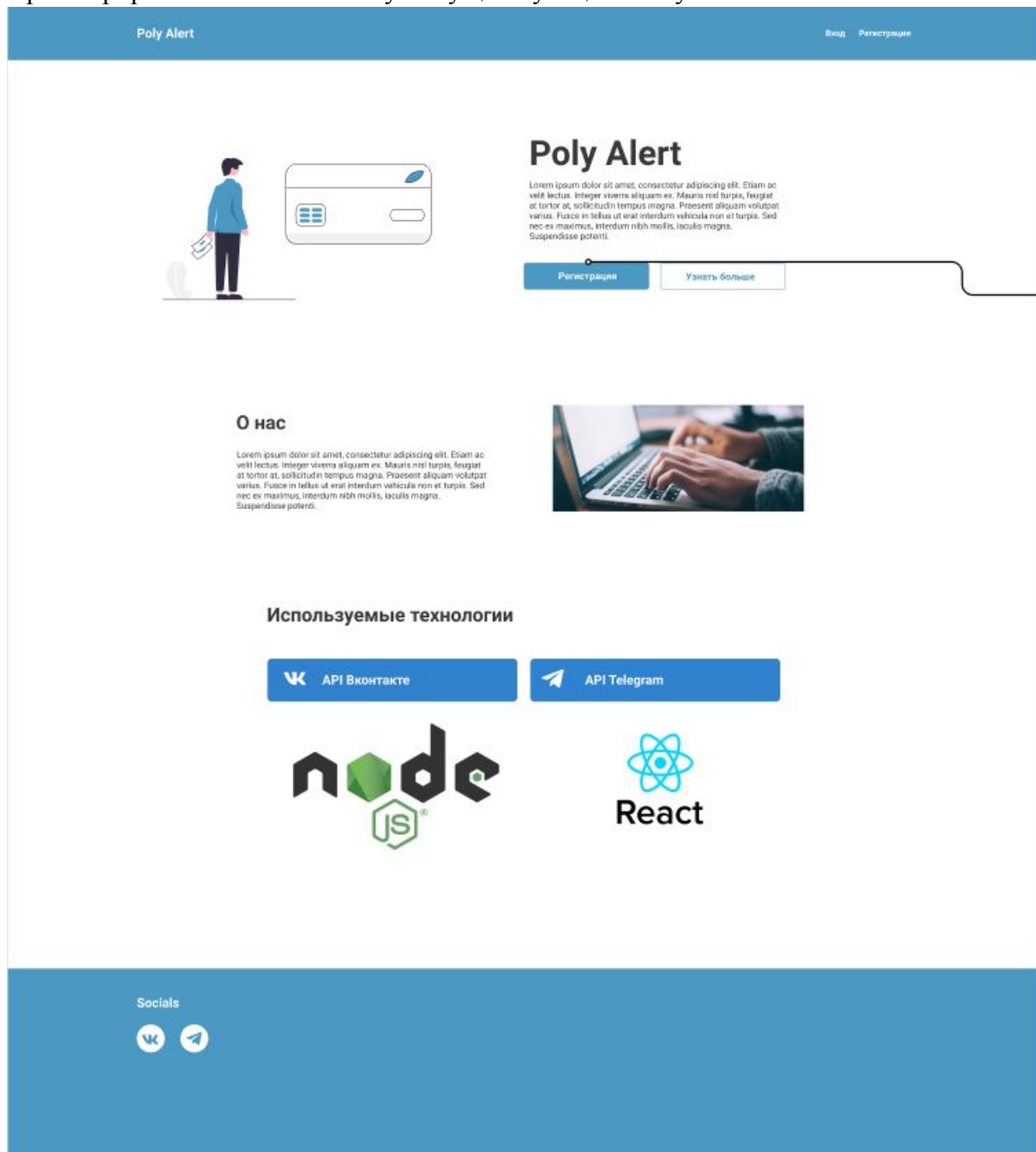
••••••••

Сохранить

Глава 3. Сопровождение проекта

3.1 Инструкция для пользователя

На главной странице можно ознакомиться с базовыми возможностями сервиса, а также зарегистрироваться или войти в уже существующий аккаунт:



Регистрация считается успешно пройденной после корректного заполнения всех полей формы и нажатии на кнопку “Зарегистрироваться”

The screenshot shows the registration page of the Poly Alert system. At the top, a blue header bar contains the text "Poly Alert" on the left and "Вход Регистрация" on the right. The main content area has a light blue background with a subtle wave pattern at the bottom. The title "Регистрация" is centered, followed by a line of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Id ut bibendum nulla eu quam. Ac malesuada ultrices ipsum turpis etiam." Below this, there are six input fields arranged in two columns. The left column contains fields for "Имя" (Name), "Email", and "Пароль" (Password). The right column contains fields for "Фамилия" (Surname), "Логин" (Login), and "Повторить пароль" (Repeat password). A blue button labeled "Зарегистрироваться" (Register) is positioned below the "Пароль" field.

Poly Alert

Вход Регистрация

Регистрация

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Id ut bibendum nulla eu quam. Ac malesuada ultrices ipsum turpis etiam.

Имя

Фамилия

Email

Логин

Пароль

Повторить пароль

Зарегистрироваться

После успешной регистрации, пользователь должен авторизоваться, чтобы привязать аккаунты и получать уведомления.

The screenshot shows the login page of the Poly Alert system. It features the same blue header bar with "Poly Alert" and "Вход Регистрация". The main content area has a light blue background with a wave pattern at the bottom. The title "Вход" (Login) is centered, followed by the same placeholder text as the registration page. Below the text, there are two input fields: "Email" and "Пароль" (Password). A blue button labeled "Вход" (Login) is centered below the input fields.

Poly Alert

Вход Регистрация

Вход

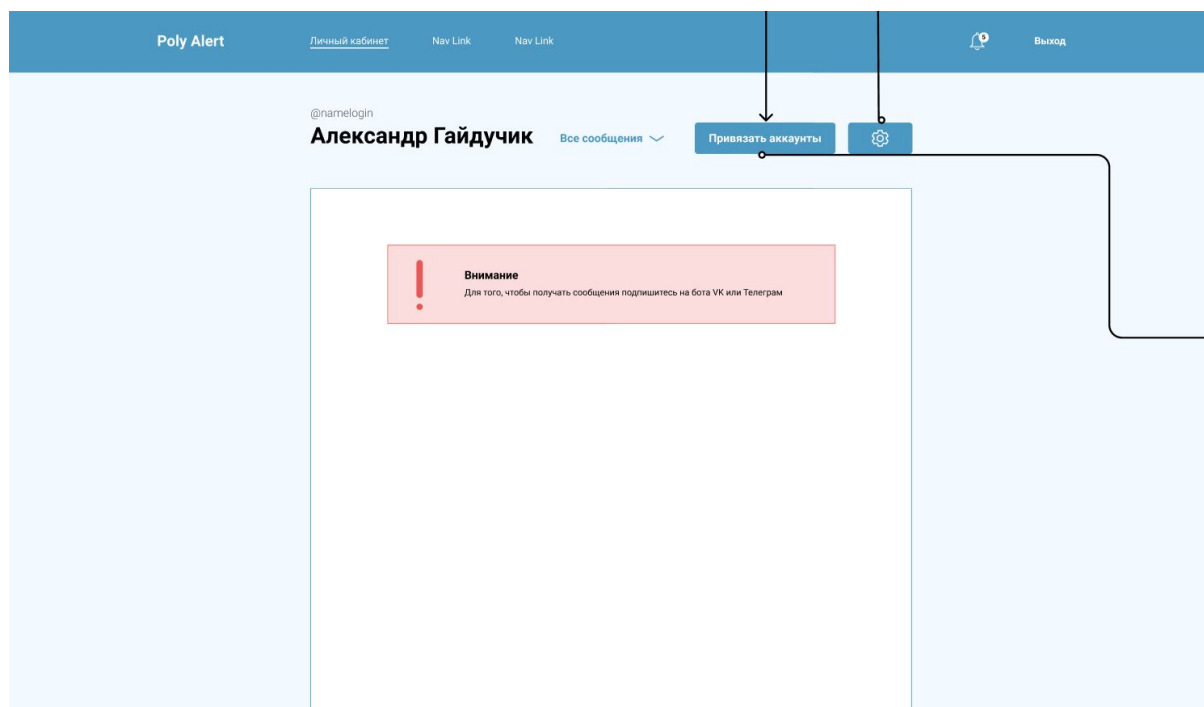
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Id ut bibendum nulla eu quam. Ac malesuada ultrices ipsum turpis etiam.

Email

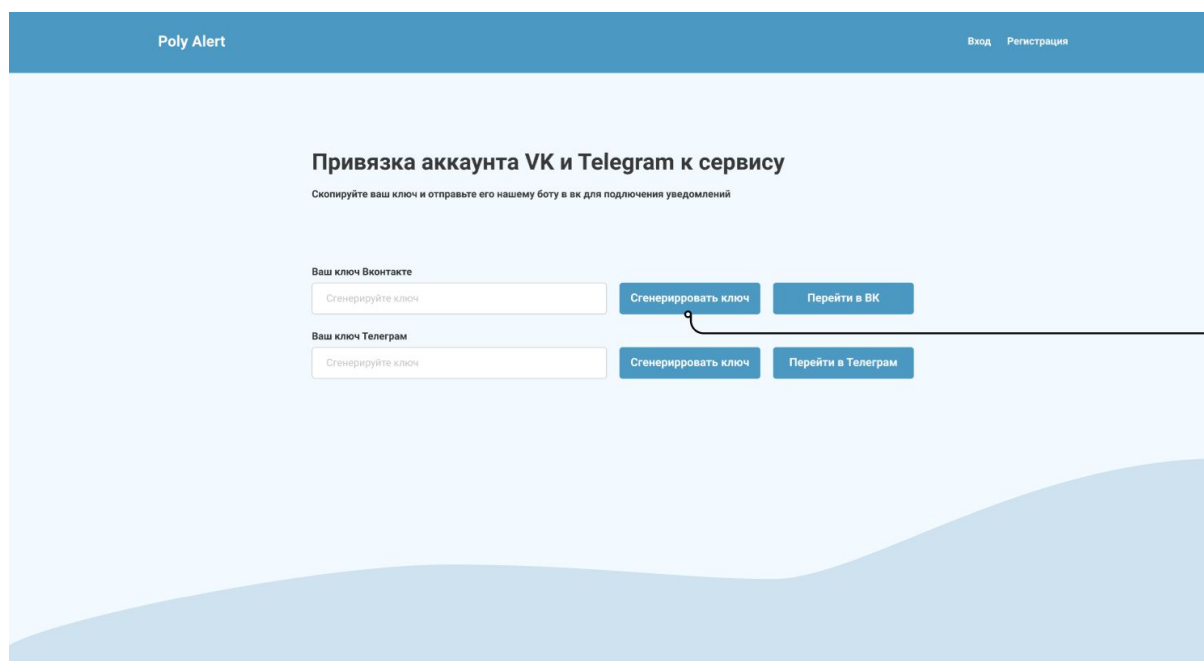
Пароль

Вход

После входа в сервис, пользователь может привязать к аккаунту удобные ему мессенджеры, либо войти в панель настройки аккаунта:



Так выглядит страница привязки аккаунтов к различным мессенджерам:

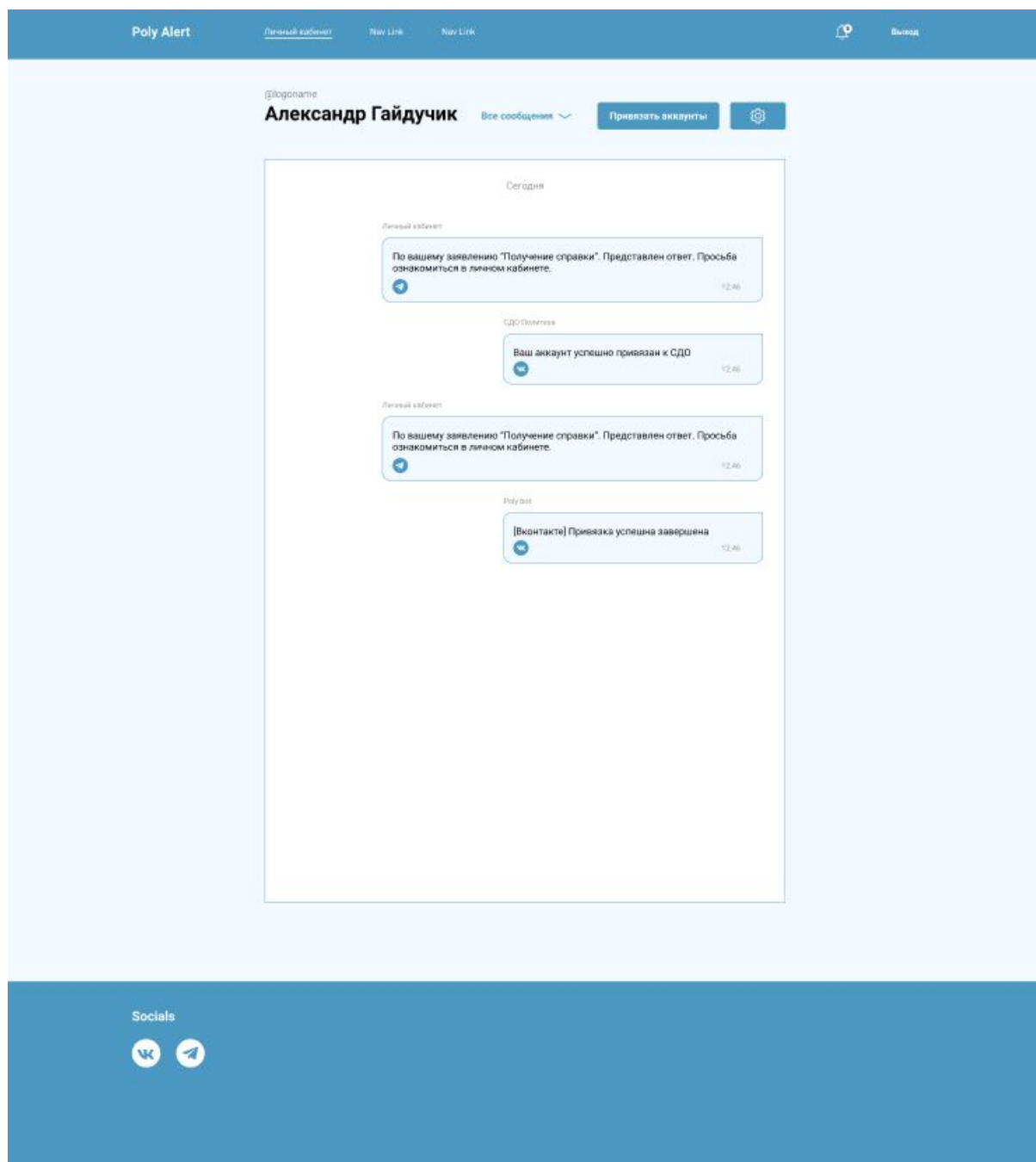


При нажатии на кнопку “Сгенерировать ключ”, пользователю отобразится рандомно сгенерированный код, который он, соответственно, должен отправить боту того мессенджера, к которому выполняется привязка аккаунта

The screenshot shows the 'Poly Alert' website header with 'Вход' and 'Регистрация' links. The main heading is 'Привязка аккаунта VK и Telegram к сервису' with a subtext: 'Скопируйте ваш ключ и отправьте его нашему боту в вк для подключения уведомлений'. Under 'Ваш ключ Вконтакте', there is a text input containing '250154' and a 'Перейти в ВК' button. Under 'Ваш ключ Telegram', there is a 'Сгенерируйте ключ' text input, a 'Сгенерировать ключ' button, and a 'Перейти в Telegram' button. A line connects the 'Перейти в ВК' button to the Telegram section.

The screenshot shows the same 'Poly Alert' website header. The main heading is 'Привязка аккаунта VK и Telegram к сервису' with the same subtext. A green checkmark and the text 'Аккаунт Вконтакте успешно привязан' are displayed. A red 'Отвязать аккаунт' button is located to the right. Under 'Ваш ключ Telegram', there is a 'Сгенерируйте ключ' text input, a 'Сгенерировать ключ' button, and a 'Перейти в Telegram' button.

Основной экран пользователя после успешной авторизации: здесь отображаются все уведомления, приходящие в удобный ему мессенджер:



На странице “Настройки аккаунта” пользователь может настроить информацию о себе, а также сменить почту, логин или пароль:

Poly Alert

Вход Регистрация

Настройки аккаунта

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Id ut bibendum nulla eu quam. Ac malesuada ultrices ipsum turpis etiam.

Имя	Александр	Фамилия	Гайдучик
Email	gaidu4ik@mail.tu	Логин	gaidu4ik
Новый пароль	••••••••	Повторить пароль	••••••••

Сохранить

3.2 Правовые вопросы

В ходе выбора направления проектной деятельности было решено создавать информационную систему для кафедры физической культуры, что означает сотрудничество между университетом и командой по проектной деятельности.

Права на проект принадлежат команде:

Гайдучик Александр

Моисеенко Илья

Иванов Григорий

Шубенок Максим

Мазаев Роман

Канатников Илья

Также права на приложение (в том числе и на исходный код) принадлежат Московскому Политехническому Университету. Далее перечислены лицензии на использование технологий, находящиеся в проекте:

1. React – MIT
5. Node.js – Free/Open Source, MIT
6. Express – MIT
7. Mongoose – MIT
8. MongoDB – MIT

MIT - лицензия открытого программного

обеспечения, разработанная Массачусетским технологическим институтом. Лицензия MIT является одной из самых ранних свободных лицензий, так как она относительно проста и иллюстрирует некоторые из основных принципов свободного лицензирования.

Она является разрешительной лицензией, то есть позволяет программистам использовать лицензируемый код в закрытом программном обеспечении при условии, что текст лицензии предоставляется вместе с этим программным обеспечением. Таким образом, мы можем свободно использовать компоненты.

4. Результаты

Репозиторий на github:

<https://github.com/Rouvas/ProjectActivities2020>

Ссылка на проект:

https://drive.google.com/drive/folders/1S8gwS-WwY596cANCna_yxCfrINUltSRm?usp=sharing

Дизайн (Figma):

<https://www.figma.com/file/7kXRiVOH5ucSiCphoJztDU/PolyAlert?node-id=0%3A1>

Лендинг проекта:

<http://pd-2020-2.std-1026.ist.mospolytech.ru/>

Рабочая документация Notion:

<https://www.notion.so/polyalert/25ec7df7ac3845cc977c016cf4a71bf5>

Иные полезные ссылки по проекту:

https://docs.google.com/document/d/15N0p3wTpaXeF0QgMdz_F6q3mjAi0MZO26DIw1Tfpvpc/edit?usp=sharing