|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**Техническое задание**

**по дисциплине «Разработка** **интернет приложений»**

**Тема предметной области «Электронные уведомления»**

**Выполнил:**

Студент группы ИУ5-53Б

Балюк А.В.

(дата, подпись)

**Проверил:**

Преподаватель

Канев А.И.

(дата, подпись)

Москва, 2023

# **Цель**

Разработка системы электронных уведомлений, состоящей из веб-сервиса, веб-приложения, кроссплатформенного приложения и асинхронного сервиса для обеспечения эффективной работы, и координации системы рассылок уведомлений.

# **Назначение**

Система предназначена для регистрации пользователей и отправки соответствующих электронных уведомлений. Система предоставляет пользователю интерфейс для просмотра и управления отправкой электронных уведомлений получателям. Администраторы могут взаимодействовать с базой данных и выполнять модерацию уведомлений и получателей. Главная задача системы – автоматизация процесса создания уведомлений и обеспечение бесперебойной рассылки получателям.

# **Задачи**

* 1. Разработка базового дизайна системы;
  2. Создание структуры базы данных PostgreSQL;
  3. Реализация REST-API на GO;
  4. Реализации интерфейса гостя на React на React-Vite;
  5. Добавление авторизации, аутентификации и Swagger;
  6. Добавление интерфейса создателя заявки;
  7. Добавление нативного приложения React Native, Redux Toolkit;
  8. Работа асинхронными сервисами доставки электронных уведомлений;
  9. Разработка интерфейса модератора и администратора React;

1.10 Разработка и документирование ТЗ, РПЗ и диаграммы.

1.11 Развернуть веб-приложение React на Github Pages.

# **Методы веб-сервиса**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Метод | URL | Описание | Входные данные | | Выходные данные | |
| 4.1 Методы авторизации и аутентификации | | | | | | | |
| 4.1.1 | POST | api/user/sign\_up | Регистрация | | login: string  password: string | |  |
| 4.1.2 | POST | api/user/login | Вход в аккаунт | | login: string  password: string | | jwt token |
| 4.1.3 | POST | api/user/logout | Выход из аккаунта Доступно только авторизированным пользователям | | jwt token | |  |
| Методы получателей | | | | | | | |
| 4.1.1 | GET | api/recipients/ | Возвращает список получателей, удовлетворяющих переданным критериям и черновую заявку пользователя | jwt token  {  fio: string  } | | {  {  uuid:string recipient\_count:int  }  [  {  uuid:string,  fio: string,  image\_url: varchar,  email: string,  age: int,  adress: string  },  …  {}  ]  } | |
| 4.1.2 | POST | api/recipients/ | Добавление получателя.  Доступно только модератору | jwt token  {  fio: string,  email: string, image\_url: varchar,  age: int,  adress: string  } | | {  uuid: string,  fio: string,  email: string, image\_url: varchar,  age: int,  adress: string  } | |
| 4.1.3 | GET | api/recipients/<id> | Возвращает получателя по уникальному id | id:int | | {  uuid:int,  fio: string,  email: string, “image\_url”: string,  “age”: int,  “adress”: string  } | |
| 4.1.4 | PUT | api/recipients/<id> | Обновляет информацию о получателе.  Доступно только модератору | jwt token  {  id: int,  fio: string,  email: string, image\_url: string,  age: int,  adress: string  } | | {  recipient\_id:int,  fio: string,  email: string, image\_url: varchar,  age: int,  adress: string  } | |
| 4.1.5 | DELETE | api/recipients/<id>/ | Меняет статус получателя на «Удален». Доступно только модератору | id:int | | Отсутствует | |
| 4.1.6 | POST | api/recipients/<id>/add\_to\_notification | Добавляет в черновое уведомление пользователя получателя {id}, возвращает число получателей в черновом уведомлении.  Доступно только авторизированным пользователям | jwt token  id:int | | recipient\_count:int | |
| Заявки - уведомления | | | | | | | |
| 4.2.1 | GET | api/notifications/ | Возвращает список уведомлений. Возможна фильтрация уведомлений.  Доступно только авторизованным пользователям | jwt token  Status: string, formation\_date\_start: datetime,  formation\_date\_end: datetime | | {  notifications:  [  {  uuid: int,  status: string, creation\_date: timestamp, formation\_date: timestamp, completion\_date:timestamp, moderator\_id: varchar,  customer\_id: varchar, notification\_type: string  },  …  {…}  ]  } | |
| 4.2.2 | GET | api/notifications/<id>/ | Возвращает информацию об уведомлении и его получателях. Доступно авторизованным пользователям | jwt token  id:int | | {  notification:  {  uuid: int,  status: string, creation\_date: timestamp, formation\_date: timestamp, completion\_date:timestamp, moderator\_id: varchar,  customer\_id: varchar, notification\_type: string  }  recipients:  [  {  recipient\_id:int,  fio: string,  email: string, image\_url: varchar,  age: int,  adress: string  },  …  {…}  ]  } | |
| 4.2.2 | PUT | api/notifications/<id>/ | Обновление информации об уведомлении. Доступно авторизованным пользователям | { notification\_type:string  } | | {  uuid: int,  status: string, creation\_date: timestamp, formation\_date: timestamp, completion\_date:timestamp, moderator\_id: varchar,  customer\_id: varchar, notification\_type: string  } | |
| 4.2.2 | DELETE | api/notifications/<id>/ | Меняет статус заявки на «Удален». Доступно авторизованным пользователям | jwt token  id:int | |  | |
| 4.2.2 | PUT | api/ notifications /<notification\_id>/moderator\_confirm | Меняет статус с «Сформирован» на «Отклонен» или «Завершен». Доступно модератору | {  “confirm”: boolean  } | | Отсутствует | |
| 4.2.2 | PUT | api/ notifications /<notification\_id>/user\_confirm | Меняет статус с «Черновик» на «Сформирован».  Доступно авторизованному пользователю | Отсутствует | | Отсутствует | |
| Получатели уведомлений – вспомогательная таблица (М-М) | | | | | | | |
| 4.3.1 | DELETE | api/notifications/<notification\_id>/delete\_recipient/<recipient\_id>/ | Удаляет получателя из уведомления. Доступно авторизованному пользователю, модератору | Отсутствует | | Отсутствует | |
| 4.3.2 | POST | api/ recipients/<recipient\_id>/add\_to\_notification | Добавляет получателя в заявку-черновик. Доступно сотруднику. | Отсутствует | | Отсутствует | |
| Методы пользователей | | | | | | | |
| 4.4.1 | POST | api/user/login | Авторизация пользователя. Необходимо ввести логин и пароль. Доступно любому неавторизованному пользователю | Login, Password | |  | |
| 4.4.2 | POST | api/user/sign\_up | Регистрация нового пользователя. Необходимо ввести логин и пароль. Доступно любому неавторизованному пользователю | Login, Password | |  | |
| 4.4.3 | POST | api/user/logout | Выход из аккаунта пользователя. Доступно авторизованному пользователю, модератору | Отсутствует | |  | |
| 4.5 Асинхронный сервис | | | | | | | |
|  |  |  |  |  | |  | |
|  |  |  |  |  | |  | |

# **Описание функциональных требований**

## Главное меню

### Главная – переход на главную страницу

### Получатели – переход на страницу с получателями. Пункт 5.4

### Заявки – переход на страницу с заявками создателя. Пункт 5.6. *Доступно авторизированным пользователям*

### Вход – переход на страницу авторизации. Пункт 5.3. *Доступно неавторизированным пользователям*

### Регистрация – переход на страницу регистрации. Пункт 5.2. *Доступно неавторизированным пользователям*

## Страница Регистрации

### Доступна неавторизированным пользователям

### Форма с полями:

1. Логин
2. Почта
3. Фамилия
4. Имя
5. Отчество
6. Модератор (да/нет)
7. Пароль
8. Повторить пароль

### Кнопка «Зарегистрироваться». Если данные верны, то вызов метода 4.4.2, нет – вывод информации об ошибке и просьбы ввести верные данные. Переход на страницу 5.4

## Страница Авторизации

### Доступна неавторизированным пользователям

### Форма с полями:

1. Логин
2. Пароль

### Кнопка «Войти». Если данные верны, то вызов метода 4.4.1, нет - вывод информации об ошибке и просьбы ввести верные данные. Переход на страницу 5.4

## Страница со списком получателей

### Метод 4.1.1.

### Карточка для каждого получателя включает картинку, ФИО, кнопки:

1. «Подробнее». Переход на страницу 5.5
2. «Удалить». Метод 4.1.5

### Поле фильтрации состоит из:

1. Поля ввода ФИО получателя
2. Кнопки «Поиск». После нажатия на кнопку вызов метода 4.1.1.

Если получатель найден, то должна быть отображена карточка найденного получателя, если не найден, то сообщение «Получатель не найден».

## Страница получателя

### Метод 4.1.3

### Карточка с подробной информацией о получателе. Она включает картинку, ФИО, почту, возраст, адрес, кнопку «Назад» - переход на страницу 5.4

## Страница списка заявок

### Метод 4.2.1

## Страница конструктора и просмотра заявки

## Страница списка получателей для модератора

## Страница добавления/редактирования получателя

# **Требования к аппаратному обеспечению**

## Серверная часть

### Процессор:

### ОП: от 8 Гб

### Жесткий диск:

## Клиентская часть

### Процессор:

### ОП: от 2 Гб

# **Требования к программному обеспечению**

## Серверная часть

### ОС: Linux/Windows

### СУБД: PostgreSQL 16.0

### Go 0.40.0

### Node.js

## Клиентская часть

### OC: Windows, Linux, MacOs, Android, ios

### Браузер: GoogleChrome, Firefox, Safari, Yandex