

## 1. Название продукта

Рабочий блокнот

## 2. User story

User story (пользовательская история) — это простой и короткий текст, который описывает потребности и ожидания заказчика от продукта или услуги. Основная цель — понять, что именно нужно создать, чтобы удовлетворить потребности конечного пользователя.

Формула составления US: кто + что + зачем.

User story:

Я как пользователь хочу, получить уведомление о степени актуальности задачи, чтобы я мог эффективнее выставлять приоритеты в работе.

## 3. Макет

Макеты или прототипы системы нужны для того, чтобы дизайнер мог понять, что именно ему необходимо изобразить, а также для того, чтобы понять пользовательский путь.

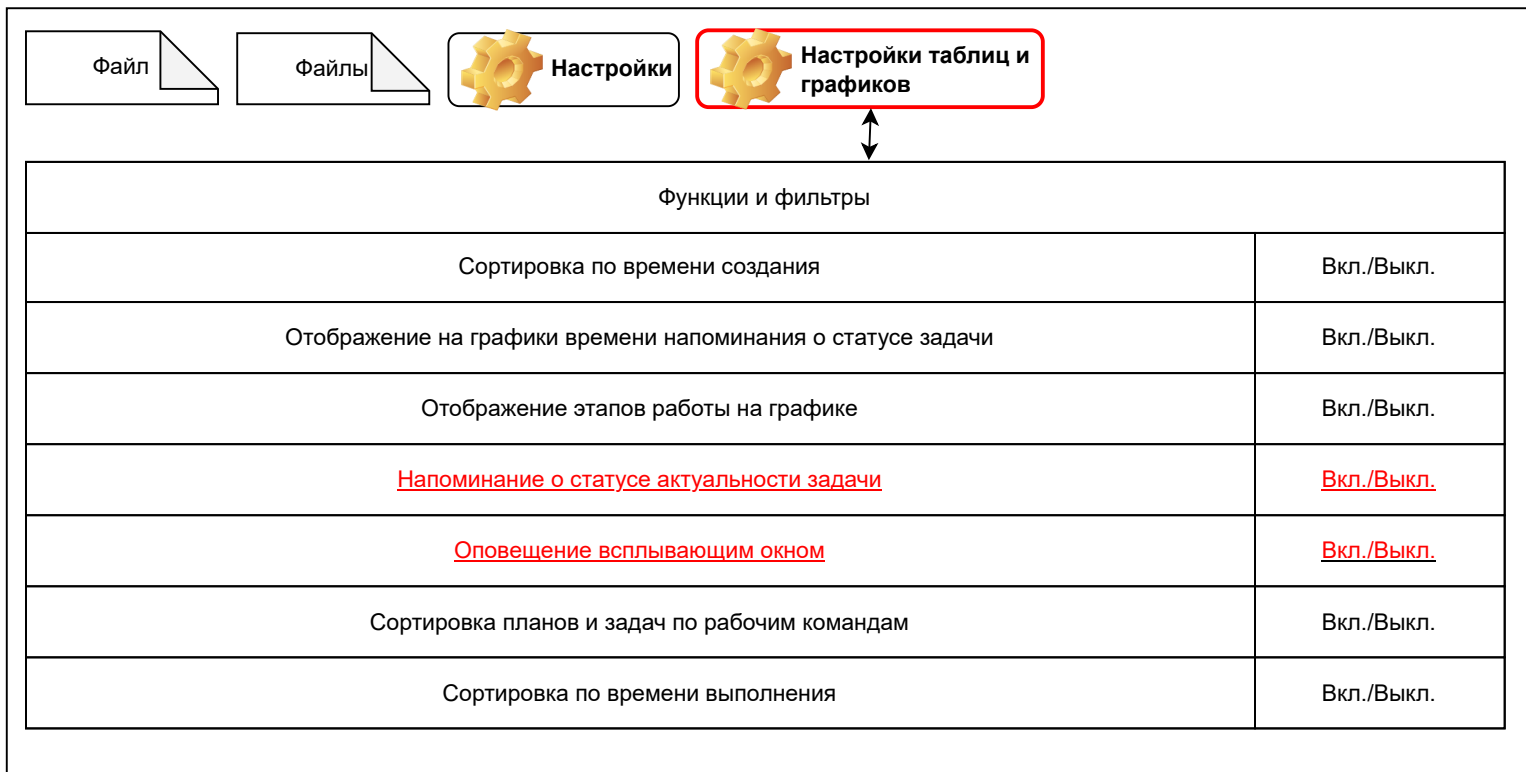
Стандартные функции, такие как: создать, открыть, сохранить как и т.д.

Списки, таблицы, заметки, тексты, задачи не вошедшие в "список и перечень задач"

Настройки. Иные настройки. Обратная связь.


Настройки таблиц и графиков.

Функции, фильтры и возможность сортировки по цвету, имени, времени создания, времени изменения, времени напоминания о статусе актуальности задачи и возможность "включения" данной функции.



Файл

Файлы

 **Настройки**

 **Настройки таблиц и графиков**

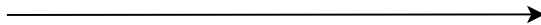
План работ на январь 2022

Задача	Пристапуть к выполнению	Продолжительность (в днях)	Назначенное время уведомления	Исполнитель	Статус
Согласовать акт приемки 1-го этапа работ	10.01.22	2	---	ФИО	Выполнена
Составить протокол принятия 2-го этапа работ	13.01.22	3	18.01.22	ФИО	В работе
Отправить документы на проверку куратору	19.01.22	1	14:00 19.01.22	ФИО	В ожидании
Подготовить техническую документацию для внутреннего аудита	20.01.22	8	24.01.22 26.01.22 13:00 31.01.22	ФИО	В ожидании
+ (Добавить задачу)					

Создать график

Список задач

Календарь



Файл

Файлы



Настройки



Настройки таблиц и графиков

### План работ на январь 2022

Задача 1 - "Согласовать акт приемки 1-го этапа работ".

Задача 2 - "Составить протокол принятия 2-го этапа работ"

Задача 3 - "Отправить документы на проверку куратору"

Задача 4- "Подготовить НТД для внутреннего аудита".

Задачи  
(Список задач)

### Календарь (Сроки)

Задача	Напоминание	Дедлайн
Задача 1	- - -	12.01.22
Задача 2	18.01.22	18.01.22
Задача 3	19.01.22	19.01.22
Задача 4	24.01.22...	31.01.22

### План работ

Начало  
работ

Задача 1

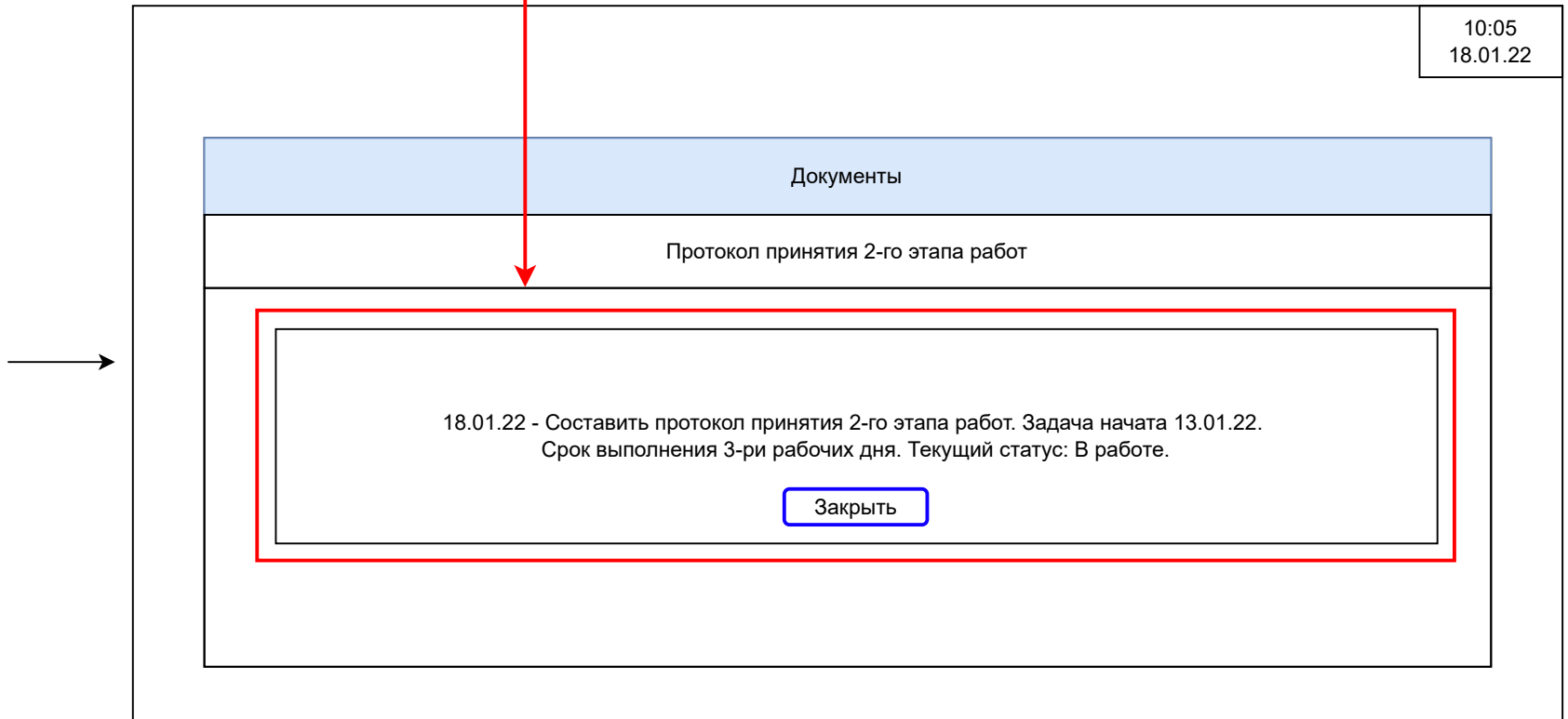
Задача 2

Задача 3

Подведение  
итогов

Конец

В ходе работы в назначенное для напоминания время появляется уведомление о текущем статусе работы, напоминающее так же о скором дедлайне.



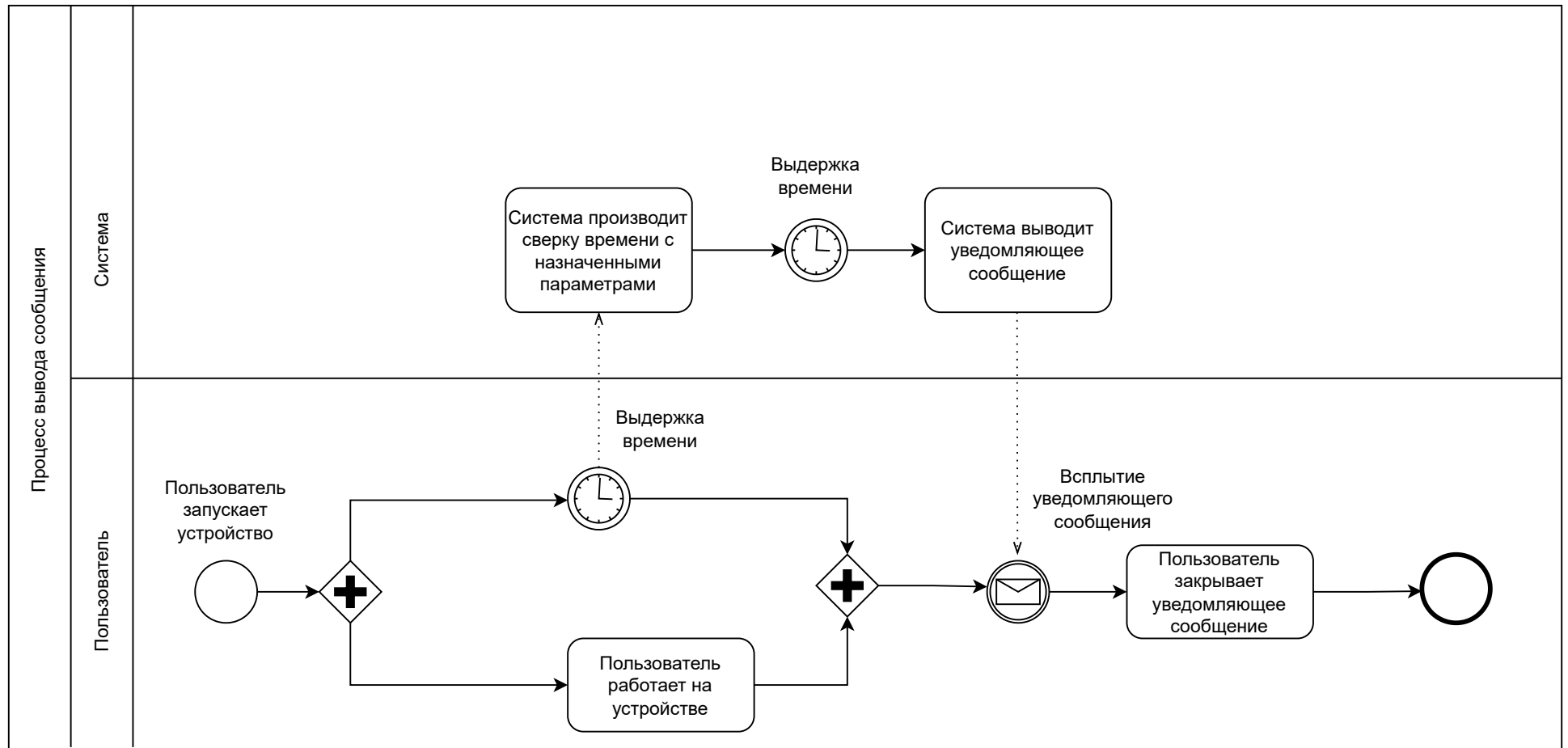
#### 4. Use case

Use case (вариант использования) — это описание того, как пользователь взаимодействует с системой, чтобы достичь определенной цели. Варианты использования помогают понять требования к системе и то, как она будет работать в разных сценариях.

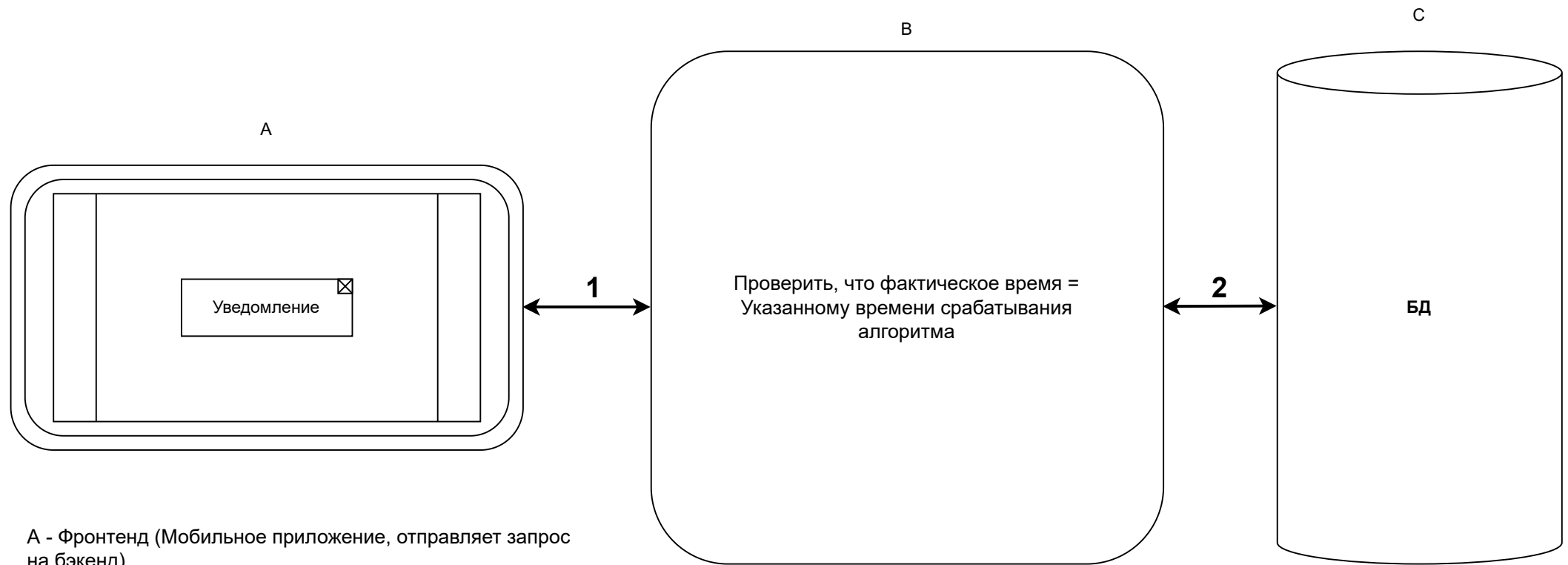
<b>Заголовок</b>	Установка и получение уведомления о степени актуальности задачи.
<b>Акторы</b>	Пользователь (клиент) - желающий систематизировать и оптимизировать свою работу.
<b>Предусловие</b>	Работе приложения, устройства и пользователя ничто не мешает и все данные для формирования уведомления заполнены.
<b>Ограничения</b>	Приложение выдаст уведомление только при включенном устройстве. Приложение выдаст уведомление с учетом установленной пользователем задержки в настройках.
<b>Триггер</b>	Пользователь приложения активирует функцию уведомлений и их вывода на экран.
<b>Основной сценарий</b>	<p>1-Пользователь выполняет действия на устройстве. 2-Система выдает уведомление согласно заданным настройкам/параметрам. 3-Пользователь закрывает уведомление. 4-Система переходит в режим ожидания. 5-Пользователь продолжает свою работу.</p> <p><b>Критерий успеха:</b> Пользователь получил необходимое ему уведомление, систематизировав и оптимизировав свой рабочий процесс.</p>
<b>Альтернативный сценарий</b>	Не предусмотрен.
<b>Исключительный сценарий 1</b>	<p>2а. Система не выдает уведомление согласно заданным параметрам/настройкам.</p> <p><b>Результат:</b> Пользователь не получил системное уведомление.</p>
<b>Исключительный сценарий 2</b>	<p>3а. Пользователь не может закрыть уведомление.</p> <p><b>Результат:</b> Пользователь не может продолжить работу.</p>

## 5. BPMN

BPMN (Business Process Model and Notation) 2.0 — это стандартная нотация для моделирования бизнес-процессов. Она предоставляет графический язык для описания процессов организации, который понятен как бизнес-аналитикам, так и техническим специалистам. Основная цель BPMN — сделать бизнес-процессы более прозрачными и понятными для всех заинтересованных сторон, что упрощает их анализ и улучшение.



## 6. Модель базовой архитектуры приложения



А - Фронтенд (Мобильное приложение, отправляет запрос на бэкенд)

В - Бэкенд (Проверяет фактическое время с установленным в качестве триггера на срабатывание алгоритма, и обращается к БД для получения информации в сообщении-уведомлении отображаемом на экране пользователя)

С - База данных возвращает результат (Информацию содержащуюся в сообщении-уведомлении)

US: Я как пользователь хочу, получить уведомление о степени актуальности задачи, чтобы я мог эффективнее выставлять приоритеты в работе.

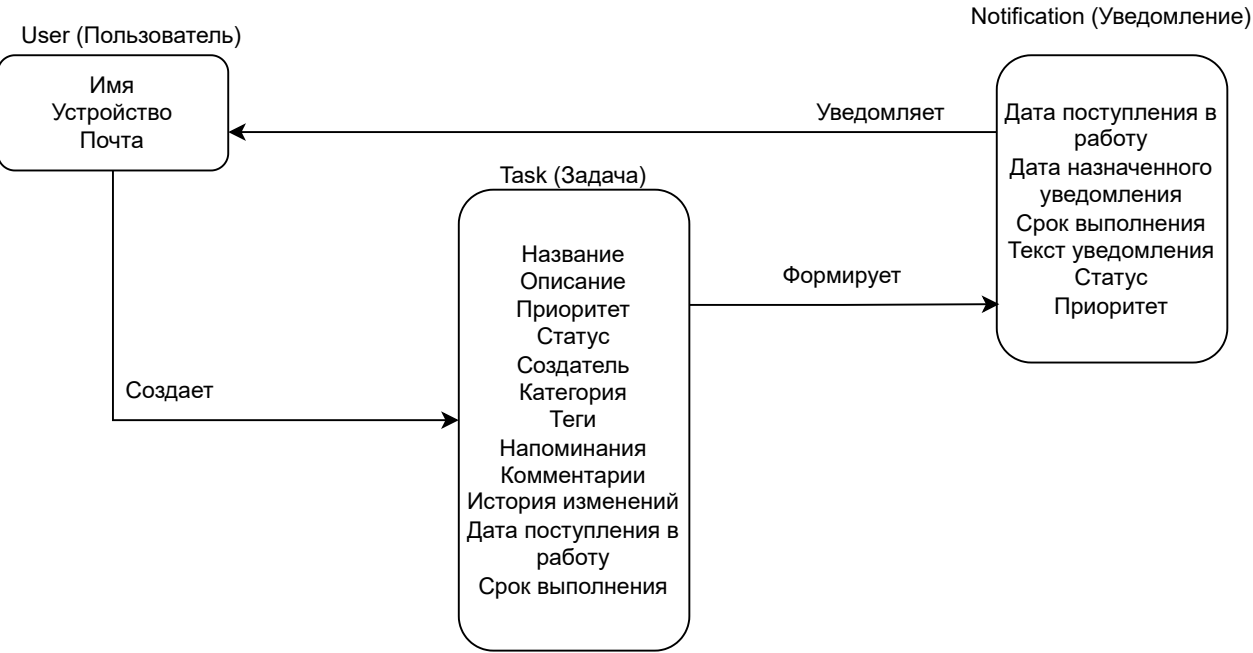
Описание процесса вывода уведомления.

В приложении задействован алгоритм вывода созданного сообщения в назначенное время. Как только фактическое время будет равно указанному времени срабатывания алгоритма, то из БД будет извлечена необходимая запись и выведена на экране пользователя.



7. Модель данных

Модель данных – это представление данных, атрибутного состава сущностей, как сущности связаны друг с другом.



Объект User

Родительская сущность	Атрибут	Описание
User	---	Объект пользователя, который имеет атрибуты и ссылки на другие объекты
	UserName	Имя пользователя (в системе)
	device	Устройство пользователя
	email	Почта пользователя

Объект Task

Родительская сущность	Атрибут	Описание
Task	---	Объект задач пользователя
	TaskName	Название задачи
	Description	Описание задачи
	Priority	Приоритет
	Status	Статус
	Creator	Создатель
	Category	Категория
	Tag	Теги
	Reminder	Напоминания

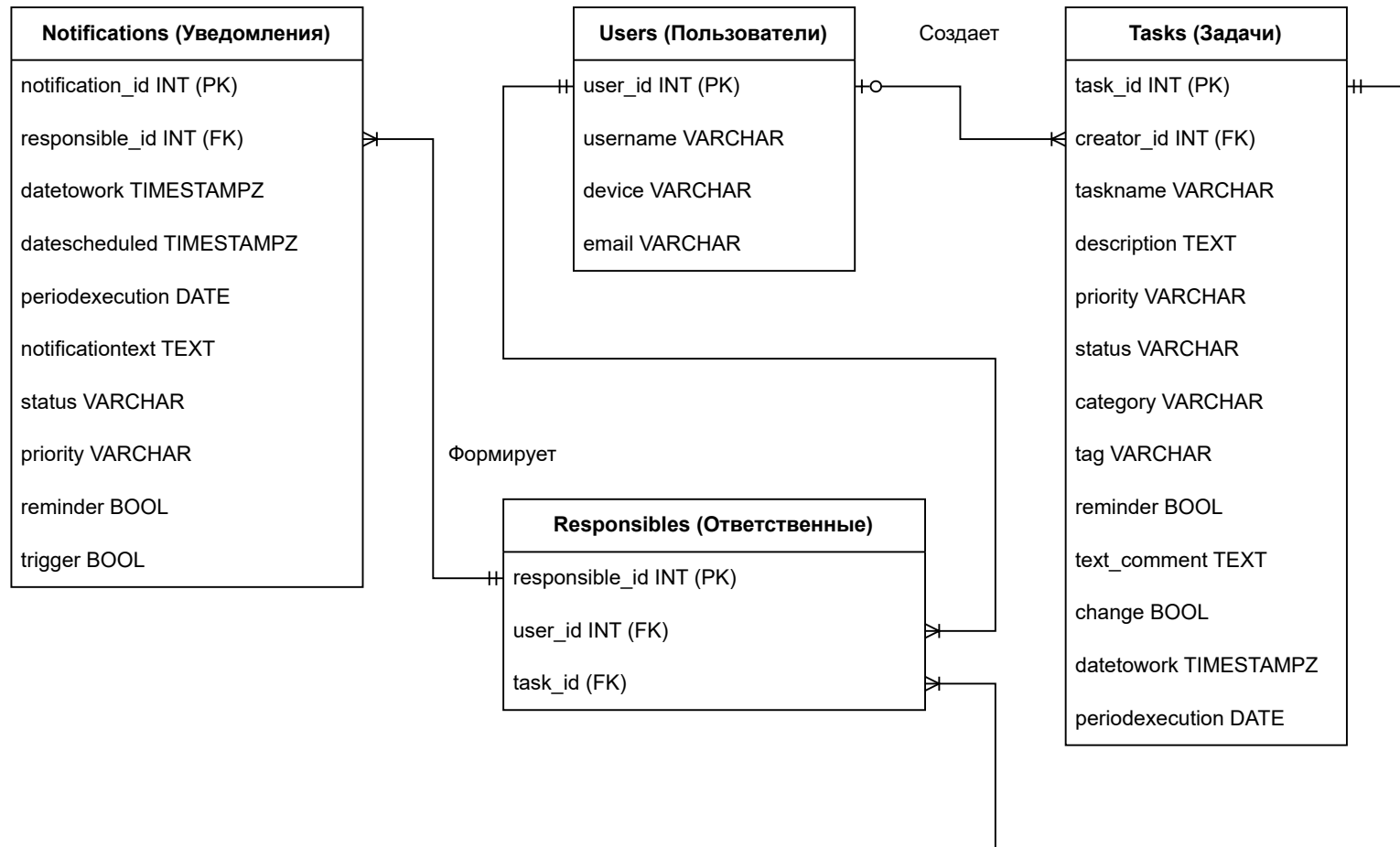
	TextComment	Комментарии
	Change	История изменений
	DateToWork	Дата поступления в работу
	PeriodExecution	Срок выполнения

**Объект Notification**

Родительская сущность	Атрибут	Описание
Notification	- - -	Объект связывающий пользователя и его задачи в созданном списке
	DateToWork	Дата поступления в работу задачи
	DateScheduled	Дата назначенного уведомления
	PeriodExecution	Срок выполнения указываемый в уведомлении
	NotificationText	Текст уведомления
	Status	Статус задачи
	Priority	Приоритетность задачи
	Reminder	Ответственный
	Trigger	Триггер

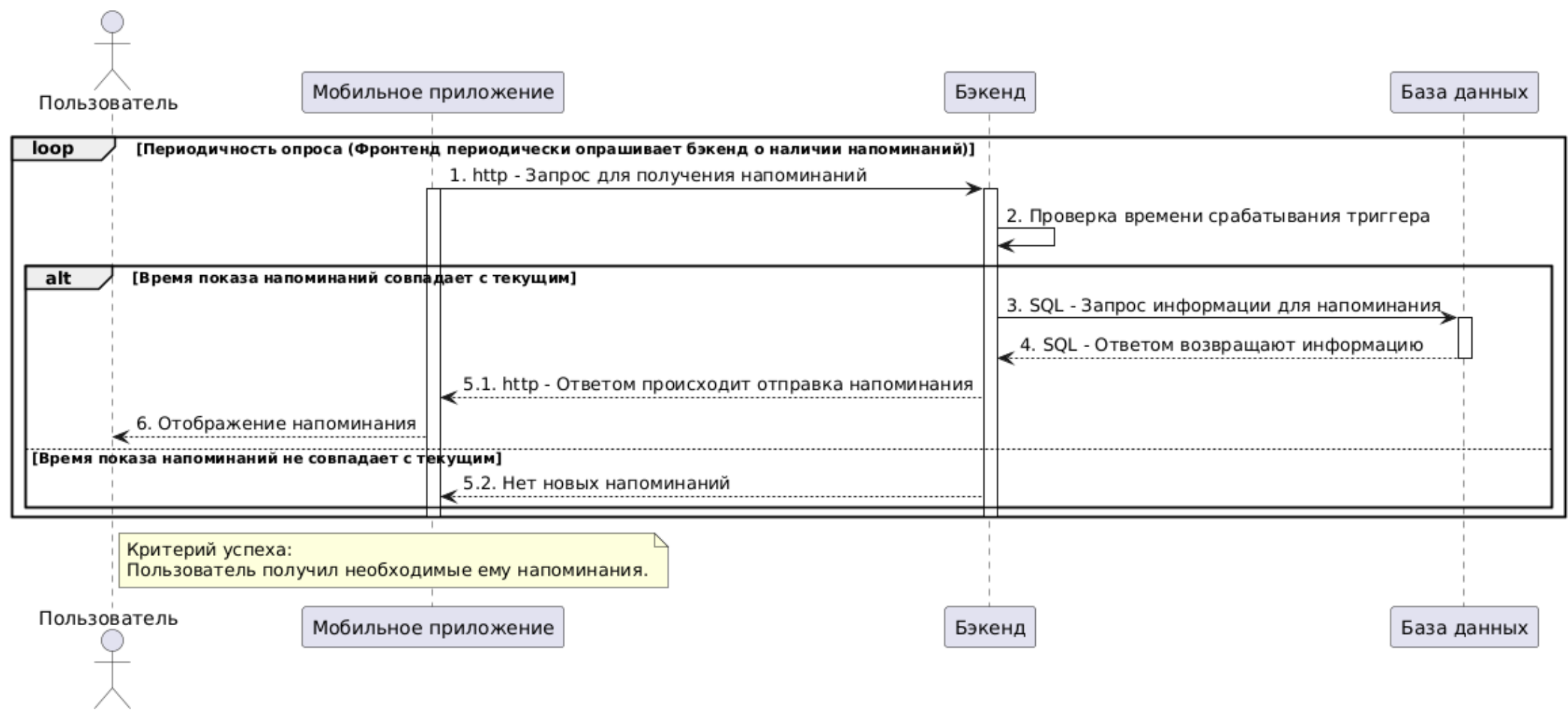
## 8. ERD

ERD (Entity-Relationship Diagram) - диаграмма, которая используется для моделирования данных и отображения взаимосвязей между сущностями в базе данных.



9. Sequence diagram

Диаграмма последовательности (Sequence Diagram) — это тип диаграммы, используемой в UML (Unified Modeling Language) для отображения взаимодействий между объектами в системе в виде последовательности сообщений.



№ Шага	Описание
1	Мобильное приложение отправляет http протоколом запрос для получения напоминаний на бэкенд.
2	Бэкенд проверяет время срабатывания триггера с текущим временем.
3	Бэкенд запрашивает информацию SLQ - запросом для напоминания из базы данных.
4	База данных возвращает SQL - ответом необходимую информацию для напоминания.
5.1	Бэкенд отправляет напоминание http протоколом на мобильное приложение.
5.2	Нет новых напоминаний
6	Мобильное приложение отображает напоминание пользователю.

10. REST Табличный вид

Запрос на получение конкретного уведомления - напоминания.

Request

GET /notifications/{responsible\_id}

GET /notifications/1

Название (Query) параметра	Тип параметра	Тип данных	Описание	Обязательность параметра
responsible_id	path	int	Уникальный идентификатор ответственного за задание	да

Response

Code 200 Ok

Название параметра	Тип параметра	Тип данных	Описание	Обязательность параметра
notification_id	body	int	Уникальный идентификатор напоминания	да
responsible_id	body	int	Уникальный идентификатор ответственного за задание	да
datework	body	date-time (В БД: timestamp with time zone)	Дата и время, когда задание должно быть начато и находиться в процессе выполнения.	нет
datesheduled	body	date-time (В БД: timestamp with time zone)	Дата и время напоминания	да
periodexecution	body	date	Срок исполнения задания	нет
notificationtext	body	string (В БД: text)	Текст уведомления	да
status	body	string (В БД: varchar)	Текущий статус задачи	нет
priority	body	string (В БД: varchar)	Приоритет задачи	нет
reminder	body	boolean	Указывает, является ли уведомление напоминанием.	нет
trigger	body	boolean	Указывает, зависит ли уведомление от триггера.	нет

# Notifications API

7.0.0

OAS 3.0

Запрос на получение конкретного уведомления-напоминания.

Servers

https://notification.swagger.io/api/v1

▼

Notification for responsible

Метод для получения уведомления для ответственного.

^

GET

/notifications/{responsible\_id}

Получить конкретное уведомление.

▼

# Responses

Code	Description	Links
200	OK.	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "notification_id": 1,
  "responsible_id": 1,
  "datework": "2023-10-15T10:00:00-00",
  "datescheduled": "2023-10-14T09:00:00Z",
  "periodexecution": "2023-10-20",
  "notificationtext": "Reminder: Complete Project Report is due tomorrow.",
  "status": "Pending",
  "priority": "High",
  "reminder": true,
  "trigger": false
}
```

## 11. Критерии приемки

Критерии приемки — это формализованные требования, которые описывают, каким образом система должна работать, чтобы пользователь или заказчик признали функциональность выполненной.

Критерии приемки состоят из кейсов "Дано - Когда – Тогда".

### Шаблон описания кейса:

Функциональность: формулировка US.

Номер кейса: N.

Дано: Предварительные условия или начальный контекст.

Когда: Событие или триггер.

Тогда: Ожидаемый результат.

### Требования предъявленные к работе:

КП (критерии приемки):

- Кейсы в рамках ваших фичей, связаны с US, UC и макетами
- Кейсы не должны включать в себя основной сценарий
- Описать от 2 до 4 кейсов включительно

НФТ (нефункциональные требования):

- Скопипастить информацию из шаблона 1к1

**US:** Я как пользователь хочу, получить уведомление о степени актуальности задачи, чтобы я мог эффективнее выставлять приоритеты в работе.

**UC** (Основной сценарий):

- 1 - Пользователь выполняет действия на устройстве.
- 2 - Система выдает уведомление согласно заданным настройкам/параметрам.
- 3 - Пользователь закрывает уведомление.
- 4 - Система переходит в режим ожидания.
- 5 - Пользователь продолжает свою работу.

Критерий успеха UC: Пользователь получил необходимое ему уведомление, систематизировав и оптимизировав свой рабочий процесс.

### Кейс №1:

В ходе работы в назначенное для напоминания время появляется уведомление о текущем статусе работы, напоминающее так же о скором дедлайне.

10:05  
18.01.22

Документы

Протокол принятия 2-го этапа работ

18.01.22 - Составить протокол принятия 2-го этапа работ. Задача начата 13.01.22.  
Срок выполнения 3-ри рабочих дня. Текущий статус: В работе.

Закреть



**Функциональность:** Пользователь получает уведомление о степени актуальности задачи.

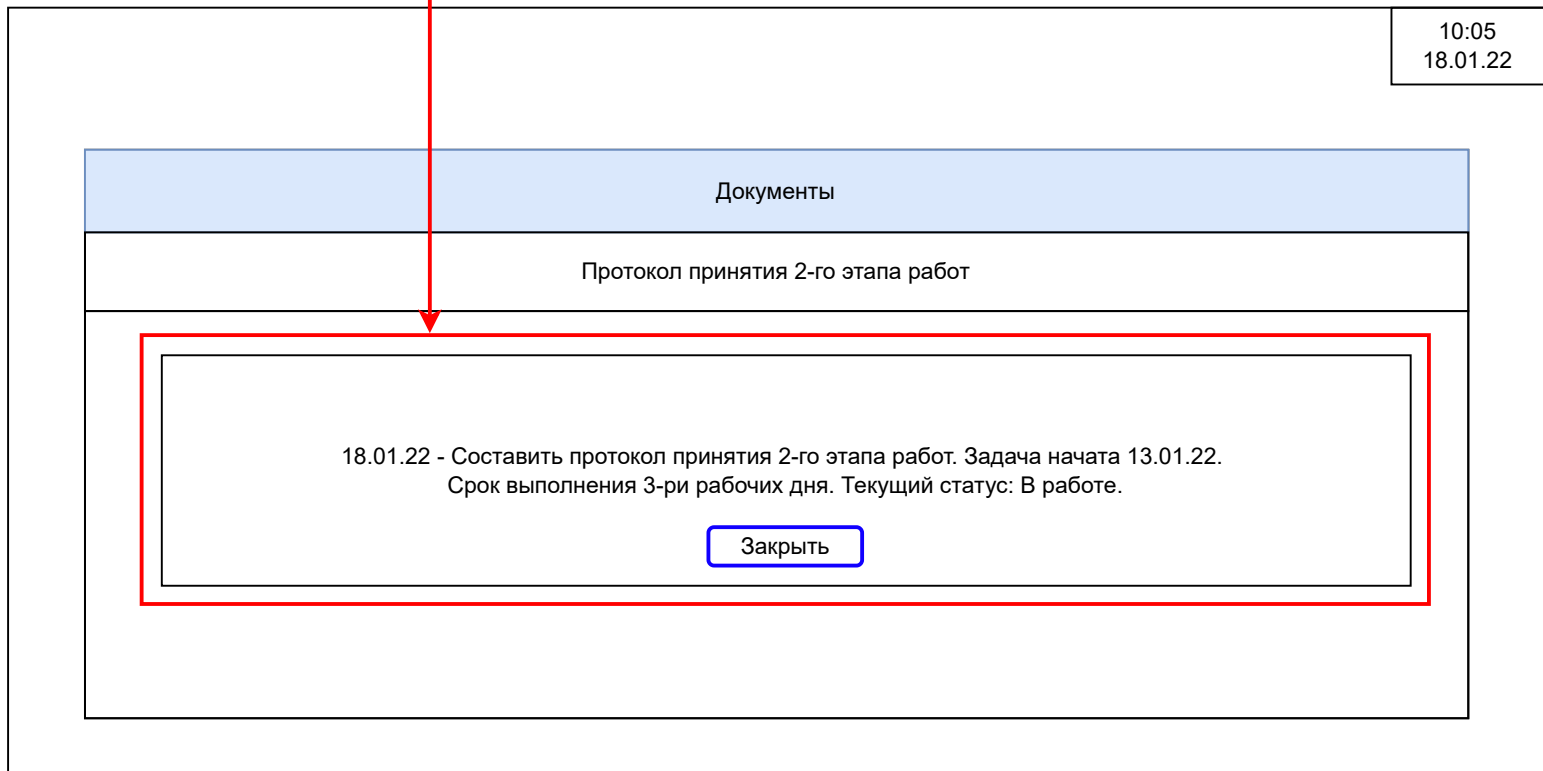
**Дано:** Устройство работает, а все необходимые настройки уведомления включены.

**Когда:** Приложение закрылось без действия пользователя отметившего принятие уведомления.

**Тогда:** Система сохраняет уведомление и отображает его при следующем запуске устройства (приложения).

#### Кейс №2:

В ходе работы в назначенное для напоминания время появляется уведомление о текущем статусе работы, напоминающее так же о скором дедлайне.



**Функциональность:** Пользователь получает уведомление о степени актуальности задачи.

**Дано:** Устройство работает, а все необходимые настройки уведомления включены.

**Когда:** Системой отправлялось уведомление, произошла ошибка не позволившая получить пользователю вовремя необходимое ему сообщение.

**Тогда:** Система не получившая от пользователя подтверждение о получении уведомления, повторно отправляет необходимое ему уведомление.

## 12. Нефункциональные требования

НФТ относятся к атрибутам качества системы, которые определяют, как она работает, а не что она делает.

Требования надежности:

1. Система должна быть доступна 99% времени.

Требования производительности:

1. Страница перечня списка задач, должна открываться не более 2 секунд.

2. Запрос получения уведомлений пользователя GET/notifications/{responsible\_id} должен выдерживать нагрузку 1 rps.