

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

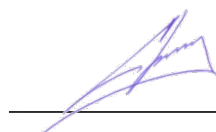
Департамент программной инженерии


СОГЛАСОВАНО

УТВЕРЖДАЮ

Научный руководитель,
доцент департамента
программной инженерии, канд. тех. наук

Академический руководитель
образовательной программы
«Программная инженерия»,
старший преподаватель департамента
программной инженерии


____ С. А. Виденин
« 16 » мая 2025 г.


____ Н. А. Павлов
« 16 » мая 2025 г.

ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА
УПРАВЛЕНИЯ ИНЦИДЕНТАМИ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ
БЕЗОПАСНОСТИ (СЕРВЕРНАЯ ЧАСТЬ: УПРАВЛЕНИЕ УЧАСТНИКАМИ
СИСТЕМЫ, УВЕДОМЛЕНИЯМИ И ДЕЖУРСТВАМИ)

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.05-01 ТЗ 01-1-ЛУ

Исполнители:

Студент группы БПИ-214



____ / Е.К. Фортов /
« 16 » мая 2025 г.

Москва 2025

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

УТВЕРЖДЕН

RU.17701729.05.05-01 ТЗ 01-1-ЛУ

**ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА
УПРАВЛЕНИЯ ИНЦИДЕНТАМИ В ОБЛАСТИ
ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ. СЕРВЕРНАЯ
ЧАСТЬ: УПРАВЛЕНИЕ УЧАСТНИКАМИ СИСТЕМЫ,
УВЕДОМЛЕНИЯМИ И ДЕЖУРСТВАМИ**

Техническое задание

RU.17701729.05.05-01

ТЗ 01-1

Листов 39

АННОТАЦИЯ

Техническое задание – это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы, ее тестирование и приемка.

Настоящее Техническое задание на разработку «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление учатниками системы, уведомлениями и дежурствами» содержит следующие разделы: «Глоссарий», «Введение», «Основание для разработки», «Назначение разработки», «Требования к программе», «Требования к программным документам», «Технико-экономические показатели», «Стадии и этапы разработки», «Порядок контроля и приемки» и «приложения».

В разделе «Глоссарий» содержатся определения терминов и понятий, используемых в настоящем Техническом задании.

В разделе «Введение» указано наименование и краткая характеристика области применения приложения.

В разделе «Основания для разработки» указан документ на основании, которого ведется разработка и наименование темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программного продукта.

Раздел «Требования к программе» содержит основные требования к функциональным характеристикам, к надежности, к условиям эксплуатации, к составу и параметрам технических средств, к информационной и программной совместимости, к маркировке и упаковке, к транспортировке и хранению, а также специальные требования.

Раздел «Требования к программным документам» содержит предварительный состав программной документации и специальные требования к ней.

Раздел «Технико-экономические показатели» содержит ориентировочную экономическую эффективность, предполагаемую годовую потребность, экономические преимущества разработки приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Раздел «Стадии и этапы разработки» содержит стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» указаны общие требования к приемке работы.

Настоящий документ разработан в соответствии с требованиями:

- 1) ГОСТ 19.101-77 Виды программ и программных документов [1];
- 2) ГОСТ 19.102-77 Стадии разработки [2];
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];
- 4) ГОСТ 19.104-78 Основные надписи [4];
- 5) ГОСТ 19.105-78 Общие требования к программным документам [5];
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
- 7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [7].

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ.....	4
ГЛОССАРИЙ.....	6
1. ВВЕДЕНИЕ.....	7
1.1. Наименование программы.....	7
1.2. Краткая характеристика области применения.....	7
2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ.....	8
2.1. Документы, на основании которых ведется разработка.....	8
3. НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	9
3.1. Функциональное назначение.....	9
3.2. Эксплуатационное назначение.....	10
4. ТРЕБОВАНИЯ К ПРОГРАММЕ.....	13
4.1. Требования к функциональным характеристикам.....	13
4.1.1. Состав выполняемых функций.....	13
4.1.2. Организация хранения данных.....	25
4.1.3. Организация входных данных.....	26
4.1.4. Организация выходных данных.....	26
4.2. Требования к временным характеристикам.....	27
4.3. Требования к интерфейсу.....	27
4.4. Требования к надежности.....	27
4.5. Условия эксплуатации.....	28
4.6. Требования к составу и параметрам технических средств.....	28
4.7. Требования к информационной и программной совместимости.....	28
4.8. Требования к маркировке и упаковке.....	28

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4.9. Требования к транспортировке и хранению.....	28
5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	28
5.1. Предварительный состав программной документации.....	29
5.2. Специальные требования к программной документации.....	29
6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....	31
6.1. Ориентировочная экономическая ценность.....	31
6.2. Предполагаемая потребность.....	31
6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами.....	32
6.3.1 Российские аналоги.....	32
6.3.2 Зарубежные аналоги.....	32
7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ.....	33
8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ.....	36
8.1. Виды испытаний.....	36
8.2. Общие требования к приемке работы.....	36
СПИСОК ИСТОЧНИКОВ.....	37
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	39

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ГЛОССАРИЙ

- 1) ТЗ — техническое задание
- 2) БД — база данных
- 3) Пользователь – человек, который использует разрабатываемую систему
- 4) СУБД – система управления базами данных.
- 5) JSON (JavaScript Object Notation) – текстовый формат обмена данными, основанный на языке программирования JavaScript. Но при этом формат независим от JavaScript и может использоваться в любом языке программирования.
- 6) API (Application Programming Interface) – описание способов (набор классов, методов и т. п.), которыми одна компьютерная программа (в данном случае, клиентская часть приложения) может взаимодействовать с другой (в данном случае, с сервером).
- 7) Микросервис — это небольшой атомарный самостоятельный сервис, который отвечает за одну функциональную роль в системе
- 8) Кастомизированный — особый, сделанный, не копируя аналоги
- 9) Тикет — задача в системе планирования задач
- 10) Трекер — система отслеживания
- 11) Бекенд — часть системы, отвечающая за логику работы
- 12) Прокси-слой — промежуточный слой в веб-приложениях, необходимый для различного рода задач (например, шифрование, защита, логирование)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ**1.1. Наименование программы**

Наименование программы – «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.».

Наименование программы на английском языке – «Information and Analytical Incident Management System in the Field of Security. Backend: Users, Notifications and Schedule Handling».

1.2. Краткая характеристика области применения

Программа будет представлять из себя три сервиса и являться частью бекенда для системы по управлению проектами. Первый сервис будет отвечать за профили пользователей системы вместе с группами, второй будет являться сервисом настраиваемых уведомлений, третий будет отвечать за расписание пользователей.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**2.1. Документы, на основании которых ведется разработка**

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем программы тема курсового проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение

Информационно-аналитическая система управления инцидентами в области обеспечения безопасности предназначена для оперативного управления событиями, возникающими на объектах инфраструктуры, где установлены различные датчики безопасности (например, пожарные, водные и другие датчики). Система сочетает в себе функции трекера задач (аналогичного Jira) с интеграцией данных от датчиков для автоматического инициирования предопределённых процессов реагирования на события тревожного характера.

Система предоставляет интуитивно понятный интерфейс для управления датчиками, бизнес-процессами, тикетами и интеграцией с внешними датчиками. Кроме того, она обеспечивает отправку настраиваемых уведомлений в режиме реального времени и обладает гибкостью настройки очередей реагирования, что исключает непредусмотренные временные задержки при угрозах безопасности.

Система предназначена для выполнения следующих функций:

1. Автоматизация тревожных событий: все сигналы от датчиков (например, срабатывание пожарной сигнализации или сигнал о затоплении) автоматически декодируются, привязываются к конкретным аварийным протоколам и запускают заранее определенные процессы реагирования.

2. Автоматизация тревожных событий: все сигналы от датчиков (например, срабатывание пожарной сигнализации или сигнал о затоплении) автоматически декодируются, привязываются к конкретным аварийным протоколам и запускают заранее определенные процессы реагирования.

3. Координация бизнес-процессов безопасности: система управляет процессами и очередями задач для каждого инцидента, автоматически координирует работу ответственных должностных лиц (например, пожарная инспекция, службы охраны) с помощью тикетов.

4. Регулирование прав доступа и безопасность: все запросы — как для мониторинга, так и для управления — проходят через прокси-слой, который проверяет авторизацию и права доступа каждого пользователя в системе.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

5. Уведомления и взаимодействие: уведомления, комментарии и автоматическое назначение статусов задач облегчают взаимодействие между участниками инцидентов и обеспечивают прозрачность управления проектами для обеспечения безопасности.

Серверная часть системы предназначена для реализации внутренней логики приложения, управления безопасностью приложения, в том числе сохранности данных пользователей приложения, обеспечения взаимодействия с базой данных, и, наконец, предоставления своего API для использования клиентской частью веб-сайта, а также административной панелью.

Данное техническое задание распространяется на три микросервиса, а именно на сервис профилей, сервис отправки настраиваемых уведомлений и сервис хранения расписания активных пользователей.

Сервис профилей пользователей хранит всю информацию о пользователях и группах пользователей в БД и предоставляет ее другим частям системы по API.

Сервис отправки настраиваемых уведомлений отвечает за доставку участникам системы уведомлений, шаблоны которых хранятся в БД. Также он может запрашивать информацию у других сервисов для составления уведомлений.

Сервис хранения расписания активных пользователей хранит всю информацию о расписании в БД и предоставляет ее системе по API.

3.2. Эксплуатационное назначение

Эксплуатационное назначение системы заключается в обеспечении работы комплекса программных и аппаратных средств для управления и реагирования на события, связанные с безопасностью на объекте. Как система безопасности, ее основная функция — получение сигналов от различных датчиков, управление событиями и инцидентами, координация участников инцидентных процессов, а также автоматизация обработки бизнес-процессов, связанных с реагированием на события.

Виды эксплуатации:

1. Мониторинг безопасности объектов

Основной эксплуатационной задачей системы является сбор информации с датчиков, установленных на объекте. Каждый датчик связан с конкретной зоной или

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

объектом контроля (например, серверная, склад, производственный цех), при этом типисобытий могут быть различными (пожар, затопление, несанкционированный доступ).

2. Автоматизация работы с инцидентами

После срабатывания датчика система автоматически создает инцидент (тикет), который фиксируется в системе в соответствующей очереди задач. Для каждого тикета задается ответственный сотрудник, параметры и приоритет решения, в зависимости от характера инцидента.

Система координирует выполнение действий по реагированию: отслеживает текущий статус инцидента, назначает исполнителей, уведомляет их о необходимости действий и контролирует выполнение проверок и реагирования.

3. Обработка сигналов и автоматизация процессов реагирования

Эксплуатация системы также включает автоматический запуск заданных бизнес-процессов с привлечением необходимых ресурсов и служб при срабатывании датчиков.

Например, при обнаружении пожара датчиком дыма система автоматически уведомит пожарную службу и сотрудников охраны, создаст инцидент с необходимыми исполнителями.

4. Управление очередями работы (бизнес-процессы)

Операторы системы могут настроить очереди инцидентов различной природы (например, «Пожар», «Протечка», «Взлом») и для каждой очереди регламентировать шаги, которые необходимо предпринять для устранения инцидента.

5. Гибкие уведомления и уведомления на основе ролей

В рамках эксплуатации система позволяет настроить схемы уведомлений в зависимости от типа инцидента, его критичности, а также на основе ролевой модели ответственности.

Например, при наступлении инцидента критического уровня вся цепь командования (дежурный оператор, менеджер службы безопасности, руководитель объекта) должна быть оперативно уведомлена в зависимости от их роли в процессе реагирования.

Серверная часть системы предоставляет API, который будет использоваться клиентской частью веб-приложения. Пользователи будут получать доступ к

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

необходимым им данным из базы данных посредством вызова конечных точек API.

Разрабатываемые в рамках данного технического задания микросервисы, а именно сервис профилей, сервис уведомлений и сервис хранения дежурств активных пользователей, будут использоваться другими микросервисами, а также графической частью приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональным характеристикам

4.1.1. Состав выполняемых функций

Рассматриваемые 3 сервиса в рамках данного ТЗ должны предоставлять API, обеспечивающее выполнение следующих функций:

Группа требований	Формулировка требования	Требования к конечной точке API
Сервис управления профилями пользователей	Создать нового пользователя	<p>Относительный url: /users</p> <p>Метод запроса: POST</p> <p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <ul style="list-style-type: none"> • <i>name</i> – имя пользователя • <i>email</i> — почта • <i>phone</i> — телефон • <i>password</i> — пароль <p>Ограничения на входные данные:</p> <ul style="list-style-type: none"> • name: <ul style="list-style-type: none"> ◦ не пусто ◦ длина: от 2 до 250 символов включительно • email: <ul style="list-style-type: none"> ◦ присутствует ◦ соответствие формату почты ◦ длина: от 6 до 100 символов включительно ◦ уникальность • phone: <ul style="list-style-type: none"> ◦ присутствует ◦ соответствует формату телефона ◦ уникальность • password: <ul style="list-style-type: none"> ◦ не пусто ◦ длина: от 8 до 40 символов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 201 при успешном выполнении запроса, 400/409 при некорректных входных данных, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате JSON: <ul style="list-style-type: none"> • id • name • email • phone
	Редактировать пользователя	<p>Относительный url: /users</p> <p>Метод запроса: PUT</p> <p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <ul style="list-style-type: none"> • <i>id</i> — идентификатор пользователя • <i>name</i> — новое имя • <i>email</i> — новая почта • <i>phone</i> — новый телефон • <i>password</i> — новый пароль пользователя • <i>organization</i> — новая компания, где работает пользователь • <i>avatar</i> — url-адрес новой аватарки пользователя • <i>about</i> — новое описание пользователя <p>При этом каждое поле, кроме id, может отсутствовать — в таком случае данное поле сохранит свое старое значение.</p> <p>Ограничения на входные данные:</p> <ul style="list-style-type: none"> • id: <ul style="list-style-type: none"> ○ присутствует ○ ≥ 0 • email: <ul style="list-style-type: none"> ○ уникальность

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • телефон: <ul style="list-style-type: none"> ◦ уникальность <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном выполнении запроса, 404, если пользователь с текущим id не найден, 400/409 при некорректных входных данных, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате JSON: <ul style="list-style-type: none"> • id
	Получить данные о пользователе	<p>Относительный url: /users/info Метод запроса: GET</p> <p>Параметры: -</p> <p>Переменные пути: <i>id</i> — идентификатор пользователя</p> <p>Тело запроса: Нет</p> <p>Ограничения на входные данные:</p> <ul style="list-style-type: none"> • id: <ul style="list-style-type: none"> ◦ > 0 <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при найденном пользователе, 400 при некорректных входных данных, 404 при отсутствии пользователя, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация о пользователе в формате JSON: <ul style="list-style-type: none"> • id • name • email

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • phone • organization • avatar • about
	Найти пользователей по имени	<p>Относительный url: /users</p> <p>Метод запроса: GET</p> <p>Параметры:</p> <ul style="list-style-type: none"> • <i>name</i> – имя пользователя <p>Тело запроса:</p> <p><i>Нет</i></p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате JSON: <ul style="list-style-type: none"> • массив ids пользователей с таким именем
	По списку id вернуть id пользователей, которых НЕ существует	<p>Относительный url: /users/notexisting</p> <p>Метод запроса: GET</p> <p>Параметры:</p> <p>- <i>ids</i> – id пользователей</p> <p>Переменные пути:</p> <p>-</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<p>JSON:</p> <ul style="list-style-type: none"> • id пользователей, которых не существует (из тех id, которые поступили в запрос)
	Получить id пользователя по email и паролю	<p>Относительный url: /users/userId Метод запроса: GET</p> <p>Параметры:</p> <p>email — имя пользователя password - пароль</p> <p>Тело запроса:</p> <p>Нет</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при найденном пользователе, 404 при отсутствии пользователя с заданными параметрами, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате JSON: <ul style="list-style-type: none"> • id пользователя
	Получить частичную информацию о пользователях	<p>Относительный url: /users/info Метод запроса: GET</p> <p>Параметры:</p> <p>- ids: массив id пользователей</p> <p>Тело запроса:</p> <p>-</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • Response body: <ul style="list-style-type: none"> • массив: <ul style="list-style-type: none"> • id • name • avatar
	Получить информацию обо всех существующих пользователях	<p>Относительный url: /users/all</p> <p>Метод запроса: GET</p> <p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <p>-</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном запросе, 404 — если ни один пользователь не был найден, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: <ul style="list-style-type: none"> • массив: <ul style="list-style-type: none"> • id • name • email • phone • organization • avatar • about
Сервис отправки уведомлений	Создать и зарегистрировать уведомление	<p>Относительный url: /notifications</p> <p>Метод запроса: POST</p> <p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <ul style="list-style-type: none"> • <i>type</i> – тип уведомления (эл.письмо / смс) • <i>content</i> - содержание уведомления • <i>time_to_show</i> – когда уведомление должно показываться • <i>interval_to_repeat</i> — интервал повтора уведомления

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • user_id — на кого назначено уведомление • immediately — флаг мгновенной доставки уведомления <p>Ограничения на входные данные:</p> <ul style="list-style-type: none"> • type: <ul style="list-style-type: none"> ◦ присутствует ◦ допустимые значения: «sms» и «email» • content: <ul style="list-style-type: none"> ◦ присутствует • userId: <ul style="list-style-type: none"> ◦ присутствует • time_to_show (если присутствует): <ul style="list-style-type: none"> ◦ формат: "yyyy-MM-dd'T'HH:mm:ss" • недопустимо, чтобы immediately = false И time_to_show = null • interval_to_repeat (если присутствует): <ul style="list-style-type: none"> ◦ формат: PnYnMnDTnHnMnS <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 201 при успешном выполнении запроса, 400/409 для некорректных входных данных, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате JSON: <ul style="list-style-type: none"> • id созданного уведомления • все, что в теле запроса (см. выше)
	Редактировать уведомление	<p>Относительный url: /notifications</p> <p>Метод запроса: PUT</p> <p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <ul style="list-style-type: none"> • <i>type</i> – тип уведомления (эл.письмо / смс) • <i>content</i> - содержание уведомления • <i>time_to_show</i> – когда уведомление должно показываться • <i>interval_to_repeat</i> — интервал повтора уведомления • user_id — на кого назначено уведомление

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • immediately — нужно ли доставить уведомление мгновенно <p>Ограничения на входные данные:</p> <ul style="list-style-type: none"> • type: <ul style="list-style-type: none"> ◦ допустимые значения: «sms» и «email» (если присутствует) • time_to_show (если присутствует): <ul style="list-style-type: none"> ◦ формат: "yyyy-MM-dd'T'HH:mm:ss" • недопустимо, чтобы immediately = false И time_to_show = null <p>При этом каждое поле, кроме id, может отсутствовать — в таком случае данное поле сохранит свое старое значение.</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном выполнении запроса, 400/409 для некорректных входных данных, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате JSON: все поля обновленного уведомления
	Удалить уведомление	<p>Относительный url: /notifications</p> <p>Метод запроса: DELETE</p> <p>Переменная пути:</p> <ul style="list-style-type: none"> • <i>notificationId</i> – id уведомления <p>Тело запроса:</p> <p>Нет</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 204 при успешном выполнении запроса, 404 — если уведомление не найдено, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: пустое
	Удалить уведомления	<p>Относительный url: /notifications</p> <p>Метод запроса: DELETE</p>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <p>notificationIds — id уведомлений</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 204 при успешном выполнении запроса, 404 — если ни одно из уведомлений не найдено, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: пустое
	Получить информацию по уведомлениям по их id	<p>Относительный url: /notifications</p> <p>Метод запроса: GET</p> <p>Переменная пути:</p> <p>notificationIds — id уведомлений</p> <p>Тело запроса:</p> <p>-</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном запросе, 404 — если ни одно уведомление не было найдено, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: массив с полной информацией об уведомлениях

Сервис дежурств	Создать дежурство для пользователя	<p>Относительный url: /duties</p> <p>Метод запроса: POST</p> <p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <ul style="list-style-type: none"> • <i>start_time</i> — начало дежурства
-----------------	------------------------------------	---

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • <i>name</i> — название дежурства • <i>interval</i> — продолжительность дежурства (1-ой итерации дежурства) • <i>ids</i> — (упорядоченный) массив <i>id</i> дежурных (дежурят по очереди, по завершении дежурства последнего в списке наступает дежурство 1-ого) <p>Ограничения на входные данные:</p> <ul style="list-style-type: none"> • <i>start_time</i>: <ul style="list-style-type: none"> ◦ должно присутствовать ◦ формат: уууу-ММ-дд'T'HH:mm:ss • <i>name</i>: <ul style="list-style-type: none"> ◦ не пусто • <i>interval</i>: <ul style="list-style-type: none"> ◦ должен присутствовать ◦ формат: PnYnMnDTnHnMnS <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 201 при успешном запросе, 400/409 при некорректных входных данных, 500 — любая другая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: <ul style="list-style-type: none"> • <i>id</i> дежурства • <i>name</i> • <i>start_time</i> • <i>interval</i> • <i>ids</i>
	Получить следующее дежурство пользователя	<p>Относительный url: /duties/user</p> <p>Метод запроса: GET</p> <p>Переменная пути:</p> <p><i>userId</i> — идентификатор пользователя</p> <p>Тело запроса:</p> <p>Нет</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном выполнении запроса, 404, если пользователь не найден, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • Response body: информация о дежурстве пользователя в формате JSON: <ul style="list-style-type: none"> • дата его следующего дежурства • продолжительность дежурства • периодичность (через сколько он снова становится дежурным)
	Редактировать дежурство	<p>Относительный url: /duties Метод запроса: PUT</p> <p>Параметры:</p> <p>-</p> <p>Тело запроса:</p> <ul style="list-style-type: none"> • <i>id</i> — идентификатор дежурства • <i>name</i> — название дежурства • <i>start_time</i> — новое начало дежурства • <i>interval</i> – новый конец дежурства • <i>ids</i> – новый массив id дежурных <p>При этом каждое поле, кроме <i>id</i>, может отсутствовать — в таком случае данное поле сохранит свое старое значение.</p> <p>Ограничения на входные данные:</p> <ul style="list-style-type: none"> • <i>id</i>: <ul style="list-style-type: none"> ◦ должен присутствовать • <i>start_time</i> (если присутствует): <ul style="list-style-type: none"> ◦ формат: уууу-ММ-дд"ТНН:мм:сс • <i>interval</i> (если присутствует): <ul style="list-style-type: none"> ◦ формат: PnYnMnDTnHnMnS <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном выполнении запроса, 404, если дежурство не нашлось, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация в формате JSON: все поля обновленного уведомления

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

	Получить информацию о дежурстве	<p>Относительный url: /duties</p> <p>Метод запроса: GET</p> <p>Переменная пути:</p> <p><i>id</i> — идентификатор пользователя</p> <p>Тело запроса:</p> <p>Нет</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном выполнении запроса, 404 — дежурство не найдено, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация о дежурстве в формате JSON: <ul style="list-style-type: none"> • id • name • start_time • interval • ids • currentlyUserDutyId
	Получить информацию обо всех дежурствах	<p>Относительный url: /duties</p> <p>Метод запроса: GET</p> <p>Переменная пути:</p> <p>-</p> <p>Тело запроса:</p> <p>Нет</p> <p>Возвращаемые данные:</p> <ul style="list-style-type: none"> • Status code: 200 при успешном выполнении запроса, 500 — любая ошибка на стороне сервера (в любом из узлов цепочки запросов сервера) • Response body: информация о дежурствах в формате JSON: <ul style="list-style-type: none"> • id

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

		<ul style="list-style-type: none"> • name • start_time • interval • ids • currentlyUserDutyId
--	--	--

При необходимости программист, реализующий данное API, имеет право добавлять новые или редактировать описанные выше конечные точки API.

Для реализации бизнес-логики рекомендуется использовать фреймворк Spring[14] и его экосистему для реализации стандартной функциональности веб-приложения и отправки уведомлений по почте, а также фреймворк SMS Aero[17] - фреймворк для отправки уведомлений по смс.

4.1.2. Организация хранения данных

4.1.2.1. Данные о пользователях и уведомлениях должны храниться в удаленной СУБД PostgreSQL.

4.1.2.2. Данные о профилях пользователей хранятся в таблице users, содержащей следующие поля и ключи:

```

4.1.2.2.1. id          BIGSERIAL PRIMARY KEY,
4.1.2.2.2. name        VARCHAR(100) NOT NULL,
4.1.2.2.3. email       VARCHAR(100) NOT NULL UNIQUE,
4.1.2.2.4. phone       VARCHAR(100) NOT NULL UNIQUE,
4.1.2.2.5. password    VARCHAR(100) NOT NULL,
4.1.2.2.6. organization VARCHAR(100),
4.1.2.2.7. avatar      TEXT DEFAULT 'https://img.icons8.com/?
size=100&id=z-JBA_KtSkxG&format=png&color=000000',
4.1.2.2.8. about       VARCHAR(300) DEFAULT 'hey I'm using
Argus'

```

4.1.2.3. Данные об уведомлениях хранятся в таблице notifications, содержащей следующие поля и ключи:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

4.1.2.3.1. id          BIGSERIAL PRIMARY KEY,
4.1.2.3.2. type        VARCHAR(100) NOT NULL,
4.1.2.3.3. content     TEXT NOT NULL,
4.1.2.3.4. time_to_show TIMESTAMP,
4.1.2.3.5. interval_to_repeat BIGINT -- seconds
4.1.2.3.6. user_id     BIGINT NOT NULL
4.1.2.3.7. immediately BOOLEAN DEFAULT false

```

Примечание. Для таблицы установлены следующие ограничения:

```
CONSTRAINT type_values CHECK (type IN ('sms', 'email')),
```

```
CONSTRAINT check_immediately_timeToShow CHECK (
```

```
NOT (immediately = false AND time_to_show IS NULL) -- иначе
```

уведомление
отправлено)

никогда не будет

4.1.2.4. Данные о дежурствах хранятся в таблице duties, содержащей следующие поля и ключи:

```

4.1.2.4.1. id          BIGSERIAL PRIMARY KEY,
4.1.2.4.2. name        VARCHAR(100) NOT NULL,
4.1.2.4.3. start_time  TIMESTAMP NOT NULL,
4.1.2.4.4. interval    BIGINT NOT NULL, -- seconds
4.1.2.4.5. ids         BIGINT[]

```

4.1.2.5. При необходимости разработчики могут создавать другие необходимые таблицы с любыми необходимыми ограничениями или редактировать описанные выше.

4.1.2.6. Некоторые данные пользователей (например, определенные настройки), а также данные, полученные из серверной части приложения, могут храниться локально, на устройстве пользователя.

4.1.3. Организация входных данных

Входными данными для серверной части приложения являются HTTP-запросы от клиентских частей приложения, которые могут содержать в себе тело в формате JSON.

4.1.4. Организация выходных данных

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Выходными данными серверной части приложения являются HTTP-ответы на запросы от клиентской части приложения, содержащие код ответа, заголовки, а также тело ответа в формате JSON. В число выходных данных также входят логи работы приложения.

4.2. Требования к временным характеристикам

Время отклика сервера при стабильном подключении к сети Интернет не должно превышать 5 секунд.

4.3. Требования к интерфейсу

Серверная часть разрабатываемой системы не предусматривает наличия графического интерфейса.

4.4. Требования к надежности

Серверная часть приложения должна корректно обрабатывать неверные запросы любого вида и выдавать информативные сообщения об ошибках:

Код	Текст	Причина
200	Success	Запрос выполнен успешно
201	Created	Запрос выполнен успешно, запись создана (в базе данных)
204	No content	Запрос выполнен успешно, ничего не возвращено (например, запрос на удаление)
400	Bad Request	В запросе содержится информация, которой нет в базе данных
404	Not Found	По отправленному запросу не нашлось требуемых данных или запрашиваемая конечная точка API не существует
409	Conflict	Запрос на добавление нового ресурса на сервер не был обработан, так как этот ресурс уже существует или нарушены ограничения на данные
500	Internal Server Error	Во время выполнения запроса произошла ошибка, которую невозможно корректно обработать (например, недоступность базы данных, другого сервиса или необработанное исключение в алгоритме)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4.5. Условия эксплуатации

Не требует специального обслуживания.

4.6. Требования к составу и параметрам технических средств

Для нормального функционирования сервера требуется:

1. Операционная система Linux (любой дистрибутив);
2. Установленные Docker, docker-compose.

4.7. Требования к информационной и программной совместимости

Серверная часть системы, а именно рассматриваемые 3 микросервиса (профили, уведомления, расписание) должны быть написаны на языке программирования Java версии 21 (или выше) с использованием фреймворка Spring. Базы данных, используемые для хранения данных пользователей, объявлений и любых других необходимых данных, должны быть реализованы с помощью СУБД PostgreSQL версии 13 (или выше). Функциональность всех конечных точек API серверной части приложений должна быть задокументирована, должны быть описаны методы доступа к этим точкам, принимаемые и возвращаемые ими данные. Также должны быть приведены примеры запросов.

4.8. Требования к маркировке и упаковке

Программа распространяется в виде электронного пакета, содержащего программную документацию, приложение (исполняемые и прочие необходимые для работы файлы).

4.9. Требования к транспортировке и хранению

Программный продукт может храниться и транспортироваться на любом носителе информации или в облачном хранилище.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

5.1. Предварительный состав программной документации

- 1) «Информационно-аналитическая система управления проектами в области обеспечения безопасности (серверная часть)». Техническое задание (ГОСТ 19.201-78) [7];
- 2) «Информационно-аналитическая система управления проектами в области обеспечения безопасности (серверная часть)». Программа и методика испытаний (ГОСТ 19.301-78) [10];
- 3) «Информационно-аналитическая система управления проектами в области обеспечения безопасности (серверная часть)». Пояснительная записка (ГОСТ 19.404-79) [11];
- 4) «Информационно-аналитическая система управления проектами в области обеспечения безопасности (серверная часть)». Руководство оператора (ГОСТ 19.505-79) [12];
- 5) «Информационно-аналитическая система управления проектами в области обеспечения безопасности (серверная часть)». Текст программы (ГОСТ 19.401-78) [13];

5.2. Специальные требования к программной документации

- 1) Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1).
- 2) Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ». Лист, подтверждающий загрузку пояснительной записки, сдается в учебный офис вместе со всеми материалами не позже, чем за день до защиты курсовой работы.
- 3) Вся документация также воспроизводится в печатном виде, она должна быть подписана академическим руководителем образовательной программы 09.03.04 «Программная инженерия», руководителем разработки и исполнителем перед сдачей курсовой работы в учебный офис, не позже одного дня до защиты.
- 4) Документация также сдается в электронном виде в формате .pdf или .docx, а программа – в архиве формата .zip или .rar.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 5) Все документы перед защитой курсовой работы должны быть загружены в информационно-образовательную среду НИУ ВШЭ LMS (Learning Management System) в личном кабинете, дисциплина – «Дипломное проектирование 2024-25», одним архивом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

6.1. Ориентировочная экономическая ценность

Для оценки экономической ценности вашей системы выделим основные аспекты, которые принесут пользу предприятиям, внедряющим систему:

- **Снижение затрат на реагирование на инциденты**

Интеграция с датчиками безопасности (пожарными, водными и т.д.) позволяет оперативно реагировать на угрозы, минимизируя ущерб от пожаров, затоплений или других аварий. Это сокращает расходы на восстановление и снижает риски потерь.

- **Оптимизация трудозатрат**

Благодаря гибким уведомлениям и автоматизации распределения задач между дежурными сотрудниками система уменьшает количество ручных операций, что экономит время и снижает вероятность человеческих ошибок.

- **Предотвращение простоев**

В системах безопасности время реагирования критично. Ваша система способствует предотвращению простоев производств, вызванных авариями.

- **Повышение эффективности управления инцидентами**

Более прозрачное управление процессами безопасности и интеграция с проектным управлением увеличивают общую производительность.

- **Оценочная экономическая выгода**

Внедрение системы может снизить финансовые ущербы, связанный с инцидентами, отслеживаемыми датчиками.

6.2. Предполагаемая потребность

На фоне повышения требований к обеспечению безопасности объектов многие компании, особенно крупные предприятия и промышленные организации, нуждаются в современных решениях для мониторинга и оперативного реагирования.

Такие системы, как Jiga, предназначены в основном для управления инцидентами и не интегрируются с физическими датчиками или системами безопасности. Наша система закрывает этот пробел, предлагая уникальную функциональность.

Рассуждая в реалиях нашей страны, законодательные нормы в России (например, требования пожарной и экологической безопасности) обязывают организации внедрять системы мониторинга и реагирования. Это создаёт базу для потенциального спроса.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Ключевые потребители:

1. Крупные промышленные предприятия.
2. Государственные учреждения (особенно критически важная инфраструктура).
3. Частные компании (торговые и офисные центры).
4. Строительные компании (для обеспечения безопасности на строительных объектах).

6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

6.3.1 Российские аналоги

Многие российские системы сосредоточены либо на управлении проектами (например, 1С:Управление проектами или Яндекс.Трекер), либо на безопасности (например, системы пожарного мониторинга).

Наши преимущества:

1. Комбинация проектного управления и систем безопасности;
2. Лучшая адаптация под требования российского законодательства;
3. Возможность интеграции с датчиками, которые широко используются в России;

6.3.2 Зарубежные аналоги

Зарубежные решения, такие как Microsoft Project или Atlassian Jira, не предназначены для работы с датчиками безопасности и не имеют возможности так гибко кастомизировать уведомления, что критически важно для максимально эффективного взаимодействия системы с дежурными.

Наши преимущества:

1. Низкая стоимость владения (особенно важна для среднего и малого бизнеса);
2. Локализация и поддержка клиентов на русском языке;
3. Упрощённая интеграция с российскими устройствами и датчиками;
4. Низкая стоимость по сравнению с зарубежными решениями за счёт отсутствия валютной составляющей.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Стадии и этапы разработки были выявлены с учетом ГОСТ 19.102-77 [2]:

Таблица 2 – Стадии и этапы разработки

Стадии разработки	Этапы работ	Содержание работ	Сроки	Исполнители
1. Техническое задание	Обоснование необходимости разработки программы	Постановка задачи.	С 15.11.2024 по 07.01.2025	Фортов Е.К.
		Сбор исходных материалов.		
		Проведение сравнительного анализа.		
	Научно-исследовательские работы	Определение структуры входных и выходных данных.	С 08.01.2025 по 31.01.2025	Фортов Е.К.
		Предварительный выбор методов решения задач.		
	Разработка и утверждение технического задания	Определение требований к программе.	С 01.02.2025 по 14.02.2025	Фортов Е.К.
		Разработка технико-экономического обоснования разработки программы. Определение стадий, этапов и сроков разработки программы и документации на неё.		

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2 – Стадии и этапы разработки

		Выбор языков программирования.		
		Согласование и утверждение технического задания.		
2. Рабочий проект	Разработка программы	Программирование и отладка программы.	С 15.02.2025 по 31.03.2025	Фортов Е.К.
	Разработка программной документации	Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.	С 08.01.2025 по 28.02.2025	Фортов Е.К.
	Испытания программы	Разработка, согласование и утверждение программы и методики испытаний.	С 01.03.2025 по 31.03.2025	Фортов Е.К.
		Проведение испытаний программы в соответствии с утверждённой программой и методикой испытаний.	С 01.03.2025 по 12.03.2025	Фортов Е.К.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Продолжение таблицы 2 – Стадии и этапы разработки

		Корректировка программы и программной документации по результатам испытаний.	С 13.03.2025 по 23.03.2025	Фортов Е.К.
3. Внедрение	Подготовка и защита программного продукта, передача программы	Подготовка программы и программной документации для презентации и защиты.	С 24.03.2025 по 04.04.2025	Фортов Е.К.
		Утверждение для защиты программы.	С 05.04.2025 по 10.04.2025	Фортов Е.К.
		Презентация программного продукта.	До 20.04.2025	Фортов Е.К.
		Передача программы и программной документации в архив НИУ ВШЭ.	До 15.04.2025	Фортов Е.К.

Программа и документация к ней разрабатываются к утвержденным приказом декана ФКН срокам.

Исполнитель – студент группы БПИ-214 Фортов Е.К.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1. Виды испытаний

Производится проверка корректного выполнения программой заложенных в нее функций, то есть осуществляется функциональное тестирование программы.

Функциональное тестирование осуществляется в соответствии с документом «Информационно-аналитическая система управления проектами в области обеспечения безопасности (серверная часть: контур ядра)». Программа и методика испытаний (ГОСТ 19.301-79), в котором указывают [14]:

- 1) перечень функций программы, выделенных в программе для испытаний, и перечень требований, которым должны соответствовать эти функции (со ссылкой на пункт 4.1.1. настоящего Технического задания);
- 2) перечень необходимой документации и требования к ней (со ссылкой на пункт 5 настоящего Технического задания);
- 3) методы испытаний и обработки информации;
- 4) технические средства и порядок проведения испытаний.

Сроки проведения испытаний – с 15 марта 2024 г. по 20 марта 2025 г.

8.2. Общие требования к приемке работы

Проверка программного продукта, в том числе и на соответствие техническому заданию, осуществляется исполнителем вместе с заказчиком согласно «Программе и методике испытаний», а также пункту 5.2.

Защита выполненного проекта осуществляется комиссии, состоящей из преподавателей департамента программной инженерии, в утверждённые приказом декана ФКН сроки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.404-79 Программа и методика испытаний. Требования к содержанию и оформлений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. ГОСТ 19.505-79 Руководство оператора. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

13. ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
14. Документация Spring [Электронный ресурс] <https://docs.spring.io/spring-framework/reference/index.html> / Режим доступа: свободный
15. Документация PostgreSQL [Электронный ресурс] <https://postgrespro.ru/docs/postgresql/>, / Режим доступа: свободный
16. Документация Spring Boot [Электронный ресурс] <https://docs.spring.io/spring-boot/index.html>, / Режим доступа: свободный
17. Документация SMS Aero [Электронный ресурс] <https://smsaero.ru/integration/documentation/api/> Режим доступа: свободный

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц в докум.)	№ документа	Входящий № сопроводит ельного докум. и дата	Подп.	Дата
Изм.	Измененных	Замененных	Новых	Аннулированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата