Изображение выглядит как символ, логотип, эмблема, графическая вставка

Содержимое, созданное искусственным интеллектом, может быть неверным.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

**Лабораторная работа № 8**

на тему «Составление SRS-спецификации для программного продукта»

Выполнил обучающийся гр. ВКБ42

Ковалев Д. П.

Проверил:

Доцент Куликова О. В.

Доцент Скляров А. В.

Ростов-на-Дону

2025

Программное обеспечение для автоматизации деятельности отдела вневедомственной охраны

Спецификация программных требований  
к программному средству

Версия 1.1

Изображение выглядит как символ, мультфильм, Графика

Содержимое, созданное искусственным интеллектом, может быть неверным.

Журнал изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 01.09.2025 | 1.0 | Создание спецификации программных требований | Ковалев Данил Петрович |
| 03.09.2025 | 1.1 | Дополнение определений для продукта | Ковалев Данил Петрович |
|  |  |  |  |
|  |  |  |  |

Спецификация программных требований

# Введение

## Назначение

Данный документ представляет собой спецификацию программных требований (SRS) для разработки Программного Обеспечения Автоматизации Деятельности Отдела Вневедомственной Охраны (ОВО).

Целью данного документа — предоставить полное описание требований к функциональности и характеристикам программного средства «Shield». Система – «Shield» – предназначена для централизованного управления ключевыми операционными процессами ОВО, связанными с охраной объектов физических и юридических лиц. Этот документ охватывает все аспекты работы программного средства, включая: интерфейс для конечного пользователя, функциональные требования, базы данных, нефункциональные требования.

Спецификация также описывает проектные ограничения и требования к интерфейсу, которые позволяют управлять базой данных, настраивать параметры анализа и интерпретировать результаты.

Программа «Shield» должна работать под управлением операционной системы Windows. Обновление программы должно выполняться при добавлении или удалении нового функционала.

Пользователями системы являются сотрудники административного персонала ОВО. Программное средство реализовано как клиент-серверное приложение:

* Клиентская часть: Классическое Windows-приложение, предоставляющее интерфейс для работы администрации ОВО со всеми функциями системы.
* Серверная часть: Веб-приложение, обеспечивающее централизованное хранение данных (единая база данных всех контрактов и объектов) и бизнес-логику. Такая архитектура обеспечивает многопользовательскую работу, минимизирует нагрузку на клиентские рабочие места и гарантирует целостность данных через механизмы авторизации и аутентификации пользователей (администраторов ОВО).

## Область применения

Данное Программное Обеспечение (ПО) предназначено для автоматизации основных операционных процессов Отдела Вневедомственной Охраны (ОВО), связанных с охраной объектов физических и юридических лиц.

## Определения, акронимы и сокращения

При описании программы используются следующие сокращения:

* БД – база данных;
* ИС – информационная система;
* ПО – программное обеспечение;
* СУБД – система управления базами данных;
* Гб – гигабайт;
* Тб – терабайт;
* ОЗУ – оперативное запоминающее устройство;
* АРМ – автоматизированное рабочее место;
* ОВО – отдел вневедомственной охраны;
* DTO – Data Transfer Object;
* .NET – .NET Core Platform;
* ОС – операционная система;
* SSD – Solid State Drive.
* MVVM – Model-View-ViewModel
* MVC – Model View Controller
* MVP – Model View Presenter

Документ состоит из нескольких частей. В первой главе приводится общая информация о проекте и его области применения. Во второй главе будут представлены функциональные требования, включая описание интерфейса пользователя, способов ввода и вывода данных, а также основных функций программного средства. В третьей главе будет обсуждаться взаимодействие с базой данных и алгоритмы, используемые для анализа текстов. Четвёртая глава будет посвящена нефункциональным требованиям, таким как безопасность, производительность и удобство использования. Пятая глава будет содержать проектные ограничения.

## Ссылки

* Шустова Л. И., Тараканов О. В., Базы данных: учебник – Москва, НИЦ ИНФРА-М, 2021. – 304 с. [Электронный ресурс] URL: https://znanium.com/catalog/product/1189322 (дата обращения: 01.05.2022).
* Сысоева Л. А., Сатунина А. Е., Управление проектами информационных систем: учебник – Москва, НИЦ ИНФРА-М, 2021. – 345 с. [Электронный ресурс] URL: https://znanium.com/catalog/document?id=362400 (дата обращения: 01.05.2022).
* Федорова Г. Н., Разработка, внедрение и адаптация программного обеспечения: учебник - Москва, НИЦ ИНФРА-М, 2021. – 304 с. [Электронный ресурс] URL: https://znanium.com/catalog/document?id=392321 (дата обращения: 01.05.2022).

# Общее описание

В системе существует три типа пользователей — заказчик, администратор, охрана. Программа предоставляет следующие функции:

* Ведение электронного журнала реестра договоров охраны с детализацией объектов (адрес, планы помещений, точки доступа, ответственные лица).
* Управление информацией об охраняемых объектах и состоянии подключенной сигнализации.
* Прием, регистрация и обработка сигналов тревоги от охраняемых объектов.
* Организация выездов патрульных групп и оповещение ответственных лиц клиентов при срабатывании сигнализации.
* Формирование электронных актов осмотра объектов по результатам выезда патруля.
* Генерация стандартизированных отчетов о деятельности ОВО для вышестоящих милицейских органов.

# Конкретные требования

В состав используемых технических средств должны входить:

Компьютер системного администратора, включающий в себя:

* процессор x64 с тактовой частотой, не менее 3,5 ГГц;
* оперативную память объемом, не менее 16 Гб;
* видеокарту, монитор, мышь, клавиатура;
* ОС Windows 11;
* Платформа .NET 9;
* Диск SSD объемом не менее 1 Тб.

Несколько серверов для СУБД, включающие в себя:

* процессор x86 с тактовой частотой, не менее 6 ГГц;
* оперативную память объемом, не менее 32 Гб;
* диски SSD объемом не менее 3 Тб;
* система контейнеризации Docker.

Программа «Shield» требует загрузки на ПК пользователя, а также авторизации в установленном приложении.

## Функциональность

Система управления «Shield» предназначена для автоматизации ключевых процессов Отдела вневедомственной охраны (ОВО) по охране объектов физических и юридических лиц.

* Для системного администратора (Программиста): Система предоставляет возможность искать, просматривать, добавлять, изменять и удалять любые данные (договоры охраны, объекты, сигнализации, контактные лица, записи о тревогах, акты осмотра, отчеты) как через интерфейс программного обеспечения, так и напрямую в базе данных.
* Для руководителя ОВО: Система позволяет искать и просматривать данные о договорах охраны, охраняемых объектах, истории срабатываний сигнализаций, выездах патрулей, составленных актах и сформированных отчетах исключительно через программное обеспечение.
* Для Дежурного оператора (Администратора): Система предоставляет возможность:
  + Искать и просматривать данные о текущих договорах, объектах и контактных лицах.
  + Регистрировать входящие сигналы тревоги.
  + Назначать патрули на выезд по сигналу тревоги.
  + Фиксировать статусы обработки сигналов (выезд, прибытие, завершение).
  + Инициировать оповещение ответственных лиц клиентов.
  + Все действия выполняются исключительно через программное обеспечение и в рамках их оперативных функций.
* Для Патрульного (Аналог "Заказчика", но как активный участник): Система предоставляет возможность (через специализированный интерфейс, например, мобильное приложение):
  + Получать задания на выезд по сигналу тревоги.
  + Просматривать детали объекта (адрес, план, контакты ответственного лица).
  + Фиксировать факт прибытия на объект и начала осмотра.
  + Заполнять электронный акт осмотра объекта (совместно с ответственным лицом клиента, при его присутствии).
  + Фиксировать результаты осмотра (сохранность имущества, состояние сигнализации).
  + Отправлять завершенный акт в систему.

Функциональные требования включают средства обеспечения безопасности данных. Подразумевается, что каждый пользователь (сотрудник ОВО) создает учетную запись в системе, защищенную логином и паролем. Таким образом, действия пользователей отслеживаются, и в случае утечки данных или некорректных действий есть возможность отследить, из какой учетной записи они произошли.

Описание характеристик качества:

1. Мобильность
2. Надежность
3. Практичность
4. Эффективность
5. Сопровождаемость

|  |  |  |
| --- | --- | --- |
| ***Характеристики качества*** | ***Мера*** | ***Шкала*** |
| *Анализируемость:*  ♦ стройность архитектуры программ;  ♦ унифицированность интерфейсов;  ♦ полнота и корректность документации.  *Изменяемость:*  ♦ трудоемкость подготовки изменений  ♦ длительность подготовки изменений.  *Стабильность:*  ♦ устойчивость к негативным проявлениям при изменениях.  *Тестируемость:*  ♦ трудоемкость тестирования изменений;  ♦ длительность тестирования изменений. | Порядковая  Чел.– часы  Часы  Порядковая  Чел.– часы  Часы | Отл.  Хор.  Отл.  4  40  Отл.  До 2  До 24 |

### **Функциональные возможности**

Функциональная возможность - способность обеспечивать решение задач, удовлетворяющих установленные потребности заказчика и пользователей при применении комплекса программ в заданных условиях.

Функциональная пригодность – это набор и описания атрибутов, определяющих назначение, основные необходимые и достаточные функции ПС, заданные ТЗ и спецификациями требований заказчика или потенциального пользователя.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название характеристики | Разработчик | Заказчик | Пользователь | Персонал |
| Функциональные возможности | 2 | 1 | 2 | 5 |
| Надежность | 4 | 4 | 3 | 4 |
| Практичность | 6 | 2 | 1 | 6 |
| Эффективность | 1 | 3 | 5 | 3 |
| Сопровождаемость | 3 | 6 | 6 | 2 |
| Мобильность | 5 | 5 | 1 | 1 |

*1 – важно*

1. *– не важно*

*\*В столбце не должно быть повторений.*

## Практичность

|  |  |  |
| --- | --- | --- |
| Характеристики качества | Мера | Шкала |
| *Понятность:*  ♦ четкость концепции ПС;  ♦ демонстрационные возможности;  ♦ наглядность и полнота документации.  *Простота использования:*  ♦ простота управления функциями;  ♦ комфортность эксплуатации;  ♦ среднее время ввода заданий;  ♦ среднее время отклика на задание.  *Изучаемость:*  ♦ трудоемкость изучения применения ПС;  ♦ продолжительность изучения;  ♦ объем эксплуатационной документации;  ♦ объем электронных учебников.  *Привлекательность:*  ♦ субъективные или экспертные оценки. | Порядковая  Порядковая  Секунды  Секунды  Чел.– часы  Часы  Страницы  Кбайты  Порядковая | Отл.  Хор.  Хор.  Хор.  Хор.  30  5  1  2  30  1500  Отл. |

## Надежность

|  |  |  |
| --- | --- | --- |
| Характеристики качества | Мера | Шкала |
| *Завершенность:*  ♦ наработка на отказ при отсутствии рестарта.  *Устойчивость:*  ♦ наработка на отказ при наличии автоматического рестарта;  ♦ относительные ресурсы на обеспечение надежности и рестарта.  *Восстанавливаемость:*  ♦ длительность восстановления.  *Доступность–готовность:*  ♦ относительное время работоспособного функционирования.  *Безопасность данных:*  ♦ нарушение целостности конфиденциальность данных  ♦ нарушение доступности данных  ♦ нарушение целостности данных | Часы  Часы  %  Минуты  Вероятность  Вероятность  Вероятность  Вероятность | 10  12  60  15  0,699  0,21  0,44  0,122 |

## Производительность

|  |  |  |
| --- | --- | --- |
| Характеристики качества | Мера | Шкала |
| *Временная эффективность:*  ♦ время отклика – получения результатов на типовое задание;  ♦ пропускная способность – число типовых заданий, исполняемых в единицу времени.  *Используемость ресурсов:*  ♦ относительная величина использования ресурсов ЭВМ при нормальном функционировании ПС. | Секунды  Число в минуту  Вероятность | 5  10  0,8 |
| Характеристики качества | Мера | Шкала |
| *Адаптируемость:*  *♦* трудоемкость адаптации;  *♦* длительность адаптации.  *Простота установки:*  *♦* трудоемкость инсталляции;  *♦* длительность инсталляции.  *Сосуществование – соответствие:*  *♦* стандартизация интерфейсов с аппаратной и операционной средой.  *Замещаемость:*  *♦* трудоемкость замены компонентов;  ♦ длительность замены компонентов. | Чел.– часы  Часы    Чел.– часы  Часы    Порядковая  Чел.– часы  Часы | 24  До 14  10  3  Удовл.  5  30 |

## Возможности поддержки

Для обеспечения долгосрочной поддерживаемости, читаемости кода и простоты внесения изменений система "Shield" должна разрабатываться и сопровождаться в соответствии со следующими стандартами и соглашениями, которые представлены в дальнейших пунктах.

### **Стандарты программирования C#**

Разработчики обязаны следовать официальным рекомендациям по разработке на C# от Microsoft.

#### **Именование:**

* PascalCase для: пространств имен (Namespaces), классов (Classes), интерфейсов (Interfaces), методов (Methods), свойств (Properties), событий (Events), констант (Constants), типов перечислений (Enums) и их значений.
* camelCase для: параметров методов (Method Parameters), локальных переменных (Local Variables), приватных полей класса (можно использовать с префиксом \_ по соглашению: private int \_itemCount;).
* Осмысленные имена, отражающие назначение. Избегать аббревиатур, кроме общепринятых.

#### **Организация кода:**

* Использование регионов (#region ... #endregion) для группировки логически связанных членов класса (поля, свойства, конструкторы, методы).
* Соблюдение порядка объявлений внутри класса (поля -> конструкторы -> свойства -> методы -> события).
* Максимальная декомпозиция: методы должны выполнять одну логическую задачу, быть короткими (предпочтительно < 20-30 строк).
* Использование выражений using для гарантированного освобождения ресурсов (IDisposable).

#### **Комментарии:**

* Обязательное использование XML-документации (///) для публичных классов, интерфейсов, методов, свойств. Краткое описание назначения, параметров, возвращаемого значения.
* Комментарии внутри кода (//, /\* ... \*/) для объяснения *неочевидных* решений, сложной логики или алгоритмов. Избегать комментариев, дублирующих код.

### **Соглашения о присвоении имен:**

#### **WinUI (XAML/C#):**

* Страницы: [FeatureName]Page.xaml / [FeatureName]Page.xaml.cs
* Элементы управления в XAML: использовать осмысленные имена с суффиксом, указывающим тип элемента. Например: ContractsListView, SaveSettingsButton.
* ViewModels: [FeatureName]ViewModel.cs (например, ContractsViewModel). Свойства VM - PascalCase.
* Модели (Data Models): [EntityName].cs (например, Contract.cs, AlarmEvent.cs UserProfile.cs).

#### **ASP.NET Core (Backend):**

* Контроллеры: [EntityName/Feature]Controller.cs (например, ContractsController.cs, AlarmsController.cs).
* Методы действий (Actions): Использовать стандартные имена HTTP-методов (Get, Post, Put, Delete) или уточняющие (например, GetContractById, CreateNewAlarmEvent).
* Сервисы (Services): I[ServiceName]Service.cs (интерфейс), [ServiceName]Service.cs (реализация) (например, IContractService, ContractService).
* Модели DTO (Data Transfer Objects): [EntityName][Action/Context]Dto.cs (например, ContractDetailsDto, CreateAlarmEventDto, UserProfileUpdateDto).
* Классы контекста БД: AppDbContext.cs (или более специфичное имя).

### **Библиотеки классов и архитектура:**

#### **Четкое разделение слоев (Layered Architecture):**

* **Клиент (WinUI):** Слой представления (Views/Pages). Минимальная бизнес-логика. Взаимодействие с бэкендом через четко определенные API вызовы (HTTP/REST).
* **Сервер (ASP.NET Core):** API (Контроллеры): Обработка HTTP-запросов, валидация входящих данных. Application/Services: содержит бизнес-логику, координирует работу домена и инфраструктуры. Domain (Опционально, но желательно для сложных систем): содержит основные бизнес-объекты (Entities), агрегаты, доменные сервисы, интерфейсы репозиториев. *Ядро бизнес-логики, не зависит от других слоев.* Infrastructure/Persistence: Реализация доступа к данным (Entity Framework Core DbContext, Репозитории), интеграции с внешними сервисами. Реализует интерфейсы, определенные в Domain или Application. Shared/Common: Общие DTO, утилиты, константы, валидаторы, используемые и клиентом, и сервером (если требуется).

#### **Использование NuGet пакетов:**

Стандартные пакеты Microsoft (WinUI, ASP.NET Core, EF Core) и общепризнанные сторонние пакеты (например, для логирования - Serilog, для маппинга - AutoMapper) должны использоваться вместо "велосипедов". Версии пакетов должны быть четко зафиксированы (использовать PackageReference с конкретными версиями в .csproj).

## Проектные ограничения

Настоящим устанавливаются ключевые проектные решения и технологические рамки, обязательные для соблюдения при разработке и развитии системы "Shield". Отклонение от этих ограничений допустимо только после формального пересмотра и утверждения данного документа.

### **Технологический стек клиентского приложения**

* Клиентское приложение для административного персонала ОВО должно быть реализовано исключительно с использованием WinUI 3.
* Язык программирования для клиентского приложения - C#. Использование других языков (C++, VB.NET, JavaScript и т. д.) для основной бизнес-логики клиента запрещено.
* Целевая платформа клиентского приложения - Windows 10 (версия 1809 или новее) и Windows 11. Поддержка других ОС (macOS, Linux, мобильные ОС) не требуется, если иное не оговорено отдельно для роли Патрульного.
* Архитектурный паттерн клиентского приложения - MVVM. Использование других паттернов (например, MVC, MVP) допустимо только для отдельных, изолированных компонентов при наличии веских оснований.

### **Технологический стек серверного приложения**

* Серверная часть системы должна быть реализована на ASP.NET Core (версии .NET 6 LTS или новее, например .NET 8).
* Основной язык программирования сервера - C#.
* Серверное API должно следовать принципам RESTful (или строго REST-подобного) дизайна. Использование других стилей (например, SOAP, gRPC в качестве основного) запрещено без отдельного обоснования.
* Сервер должен использовать реляционную базу данных. Первичная и обязательная СУБД - PostgreSQL. Поддержка MSSQL допустима как опция через конфигурацию строки подключения, но не является заменой требования к PostgreSQL. При локальной разработке имеется право использовать sqlite.
* Доступ к данным из серверного кода должен осуществляться через Entity Framework Core (EF Core) как основную ORM. Использование "чистого" ADO.NET или других ORM (Dapper, NHibernate) допустимо только для узкоспециализированных, высокопроизводительных операций и требует отдельного обоснования.
* Сервер должен быть контейнеризирован с использованием Docker. Поставка в виде "голого" исполняемого файла или установщика для ОС без Docker не соответствует требованиям.

### **Архитектурные и инфраструктурные ограничения**

* Система должна строго следовать слоистой архитектуре на стороне сервера, как определено в разделе 3.5 (Возможности поддержки), пункт 3 ("Библиотеки классов и архитектура"): четкое разделение на слой API (Controllers), слой приложения/бизнес-логики (Services), слой доступа к данным (Infrastructure/Persistence с EF Core). Смешивание ответственности слоев запрещено.
* Клиентское приложение должно взаимодействовать с сервером исключительно через четко определенное HTTP(S) API, предоставляемое ASP.NET Core сервером. Прямой доступ клиента к базе данных запрещен.
* Все компоненты системы (клиент, сервер, БД) должны быть развертываемы в изолированной, защищенной сети ОВО. Публичный доступ к компонентам системы из интернета не предусмотрен базовым проектом. Требования к интеграции с внешними системами (например, аппаратные сигнализации) должны быть определены отдельно и реализованы через защищенные шлюзы/API.

## Требования к интерактивной пользовательской документации и справочной системе.

Система "Shield" должна сопровождаться полным комплектом программной документации, обеспечивающей эффективное внедрение, эксплуатацию, техническое обслуживание и соответствие нормативным требованиям. Документация должна быть доступна как в электронном (предпочтительно PDF), так и, при необходимости, в печатном виде.

### **Состав обязательной программной документации:**

* **Техническое задание (ТЗ):**
* Должно включать подробное описание назначения системы, функциональных и нефункциональных требований, сценариев применения в контексте деятельности ОВО (обработка тревог, управление договорами, формирование актов, отчетность).
* Является основным исходным документом для разработки и приемки.
* **Программа и методика испытаний (ПМИ):**
* Должна содержать детальный план, сценарии (тест-кейсы) и процедуры проведения приемо-сдаточных испытаний системы "Shield".
* Должна охватывать все ключевые функции: регистрация договоров, обработка сигналов тревоги, формирование актов осмотра, генерация отчетов, безопасность, работа в многопользовательском режиме.
* Должна включать критерии успешного прохождения испытаний.
* **Руководство Системного программиста (Администратора):**
* Должно содержать детальные инструкции по установке, настройке и обновлению серверной (ASP.NET Core, PostgreSQL, Docker) и клиентской (WinUI) частей системы в среде ОВО.
* Должно содержать детальные инструкции по настройке и управлению базами данных (резервное копирование, восстановление, мониторинг).
* Должно содержать детальные инструкции по конфигурации сети, безопасности (SSL/TLS, брандмауэр), интеграции с системами сигнализации (если применимо).
* Должно содержать детальные инструкции по управлению пользователями, ролями и правами доступа на системном уровне.
* Должно содержать детальные инструкции по мониторингу работы системы и анализу логов (ELK/Grafana Stack).
* **Руководство Оператора (Дежурного персонала ОВО):**
* Должно быть практическим, пошаговым пособием для сотрудников ОВО (Дежурных операторов, Руководителей смен), непосредственно работающих с системой:
  + Регистрация и мониторинг сигналов тревоги.
  + Назначение патрулей на выезд.
  + Контроль статуса обработки инцидентов.
  + Оповещение ответственных лиц клиентов.
  + Просмотр и поиск информации по договорам, объектам, контактным лицам.
  + Формирование стандартных отчетов (для смены/дня).
  + Действия в нештатных ситуациях (сбой связи, отказ компонента).
* **Руководство Пользователя (для Роли "Патрульный"):**
* Должно описывать работу с мобильным интерфейсом патрульного (если применимо) или специализированным разделом десктопного клиента:
  + Прием и подтверждение заданий на выезд.
  + Просмотр деталей объекта и контактов ответственного лица.
  + Заполнение электронного акта осмотра объекта (внесение данных о сохранности, работе сигнализации, подписание).
  + Фиксация геолокации и времени прибытия/убытия.
  + Отправка завершенного акта в систему.
* **Руководство Программиста:**
* Должно содержать информацию, необходимую для поддержки и развития системы разработчиками:
  + Описание архитектуры системы (клиент-сервер, слои).
  + Стандарты и соглашения по кодированию (ссылка на раздел 3.5 SRS).
  + Описание API сервера (Swagger/OpenAPI) и принципов взаимодействия клиент-сервер.
  + Инструкции по настройке среды разработки (Visual Studio, .NET SDK, Docker).
  + Описание ключевых алгоритмов, структур данных, используемых библиотек (NuGet).
* **Ведомость эксплуатационных документов (ВЭД):**
* Должна представлять собой полный перечень *всей* поставляемой с системой "Shield" документации с указанием наименований, кодов, версий и краткого назначения каждого документа.
* Обеспечивает контроль комплектности документации при поставке и обновлении системы.
* **Формуляр:**
* Должен содержать основные сведения о системе "Shield", необходимые для ее эксплуатации и учета:
  + Полное наименование системы и компонентов.
  + Наименование разработчика и заказчика.
  + Перечень комплектующих (серверное ПО, клиентское ПО, СУБД).
  + Системные требования к аппаратному и программному обеспечению.
  + Сведения об установке (дата, место, ответственный).
  + Учет ввода в эксплуатацию.
  + Учет технического обслуживания, ремонтов и модификаций.
  + Гарантийные обязательства.

## Приобретаемые компоненты

Система "Shield" использует ряд сторонних (приобретаемых или открытых) компонентов. Настоящий раздел определяет эти компоненты, требования к их лицензированию, стандартам совместимости и взаимодействия.

### **Компоненты Серверной Части (ASP.NET Core)**

#### **Система управления базами данных (СУБД):**

* Наименование: PostgreSQL (версия 12 или выше).
* Лицензия: Open Source (PostgreSQL License) – разрешительная лицензия, схожая с MIT/BSD, позволяет свободное использование, модификацию и распространение, в том числе в коммерческих целях, без обязательств по раскрытию исходного кода "Shield".
* Требования к совместимости**:**
  + Серверное приложение должно использовать официальный .NET драйвер Npgsql (лицензия: PostgreSQL License) для подключения.
  + Поддержка версий PostgreSQL должна соответствовать политике поддержки выбранной LTS-версии .NET.

#### **Object-Relational Mapper (ORM)**

* Наименование: Entity Framework Core (EF Core) (версия, соответствующая версии .NET, например, EF Core 6.0/7.0/8.0).
* Лицензия: Apache License 2.0 (разрешительная, позволяет коммерческое использование).
* Требования к совместимости: должен использоваться поставщик данных Npgsql.EntityFrameworkCore.PostgreSQL для работы с PostgreSQL.

#### **Контейнеризация**

* Наименование: Docker Engine (для развертывания серверного приложения и PostgreSQL).
* Лицензия: Apache License 2.0 для Docker Community Edition (CE). *Требуется проверка актуальной лицензии для целевой среды развертывания (особенно для Docker Engine на серверах ОВО).*
* Требования к совместимости**:**
  + Серверное приложение должно поставляться в виде Docker-образа (.dockerfile).
  + Рекомендуется использование официальных образов .NET SDK/Runtime и PostgreSQL из Docker Hub.

### **Компоненты Клиентской Части (WinUI3)**

Библиотеки MVVM / Вспомогательные (Примеры, возможны другие):

* + Наименование: CommunityToolkit.Mvvm (Microsoft.Toolkit.Mvvm).
  + Лицензия: MIT License (разрешительная).
  + Требования к совместимости: Совместимость с WinUI 3 и целевой версией .NET для клиента.
  + Наименование: Newtonsoft.Json (или System.Text.Json).
  + Лицензия: MIT License (Newtonsoft.Json).
  + Требования к совместимости: используется для сериализации/десериализации данных при взаимодействии с API сервера.

### **Компоненты аппаратной части**

Приёмочные испытания проводятся на программно-аппаратном комплексе Заказчика в следующей минимальной конфигурации:

Серверная площадка: оборудование, выделенное Заказчиком на территории для проведения приемочных испытаний.

Рабочее место: ПК в составе АРМ пользователя, операционная система MS Windows 11 и выше, программы С#, PostgreSQL.

## Интерфейсы

В этом разделе определяются интерфейсы, которые должны поддерживаться данным приложением.

### **Пользовательские интерфейсы**

После запуска программы программисту или пользователю отображается форма ввода логина и пароля, показанная на рисунке 1. Здесь пользователи могут либо пройти регистрацию или аутентификацию. Если пользователь использует систему впервые, то он обязан пройти регистрацию, в ином случае аутентификация.

Окно для регистрации представлено на рисунке 1. На рисунке 2 представлено окно аутентификации.

Изображение выглядит как текст, снимок экрана, мультимедиа, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 – окно регистрации

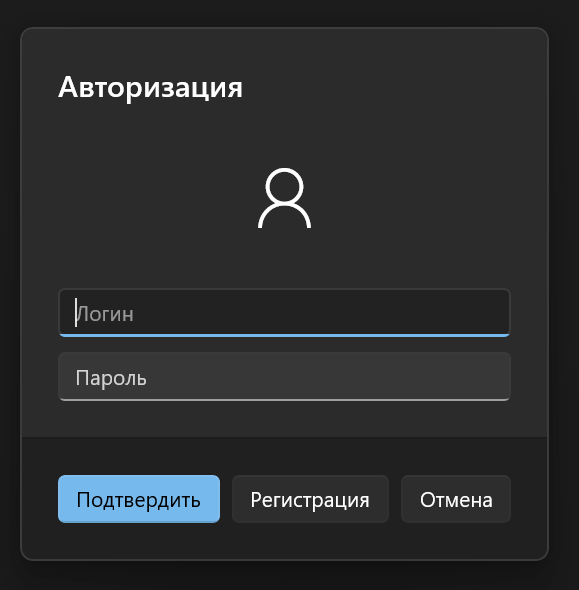


Рисунок 2 – окно авторизации

После удачной аутентифкации пользователи ИС оказываются в главном разделе приложения. На Рисунке 3 показан главный экран.

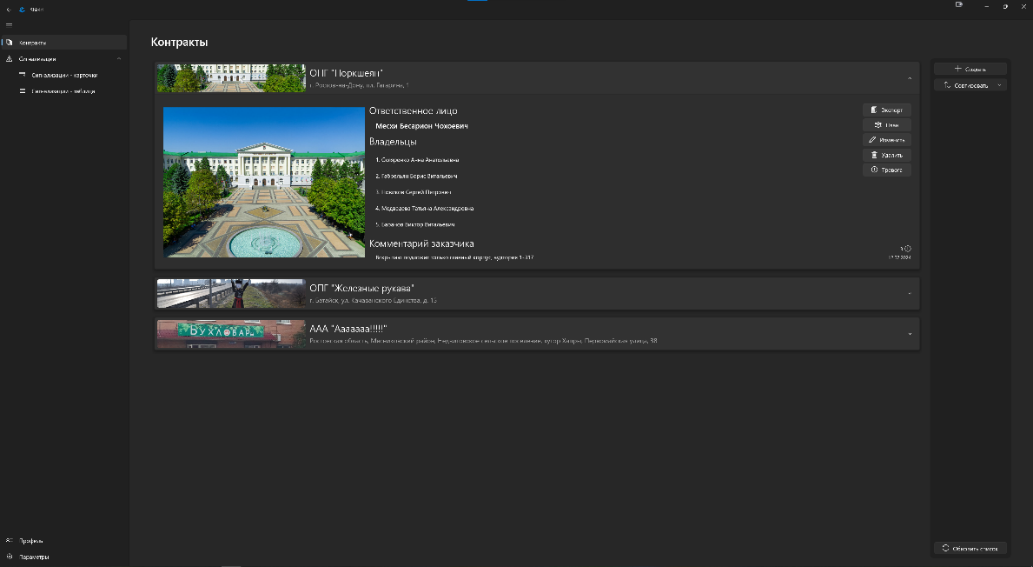


Рисунок 3 – главный экран приложения

После выбора программист или администратор могут просмотреть данные, содержащиеся в соответствующей выбранной вкладке таблице БД.

Рассмотрим пример того, как добавляется новый контракт. Здесь текущему пользователю необходимо ФИО лица, адрес, комментарий для охранников, название организации, которую надо защитить. Выбор места представлен через API карты.

Примерный интерфейс вкладок для работы выстроен по общему шаблону, представленному на Рисунке 4.

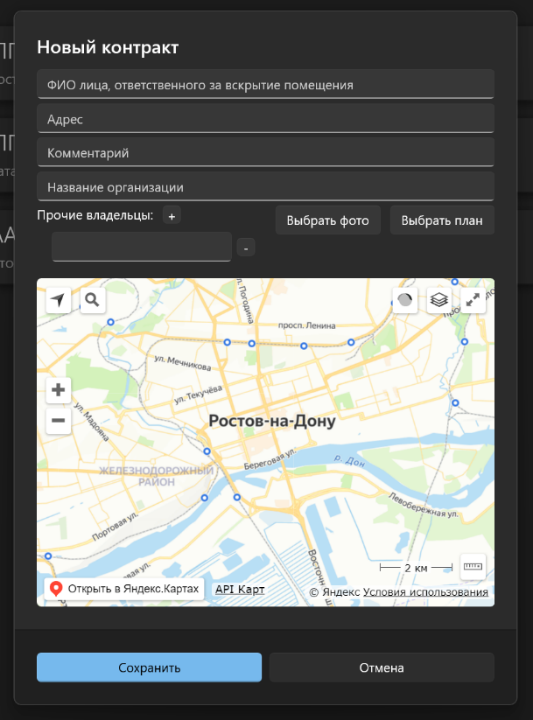


Рисунок 4 – Добавление нового контракта

Система позволяет авторизованным пользователям (Администраторам, Руководителям, Системным администраторам) просматривать журнал зарегистрированных системой срабатываний сигнализации для выбранного объекта охраны. Этот журнал содержит информацию о фактах тревог. Интерфейс системы предоставляет два режима просмотра журнала тревог: в виде карточек и в виде таблицы. Данные интерфейсы представлены на рисунках 5 и 6.

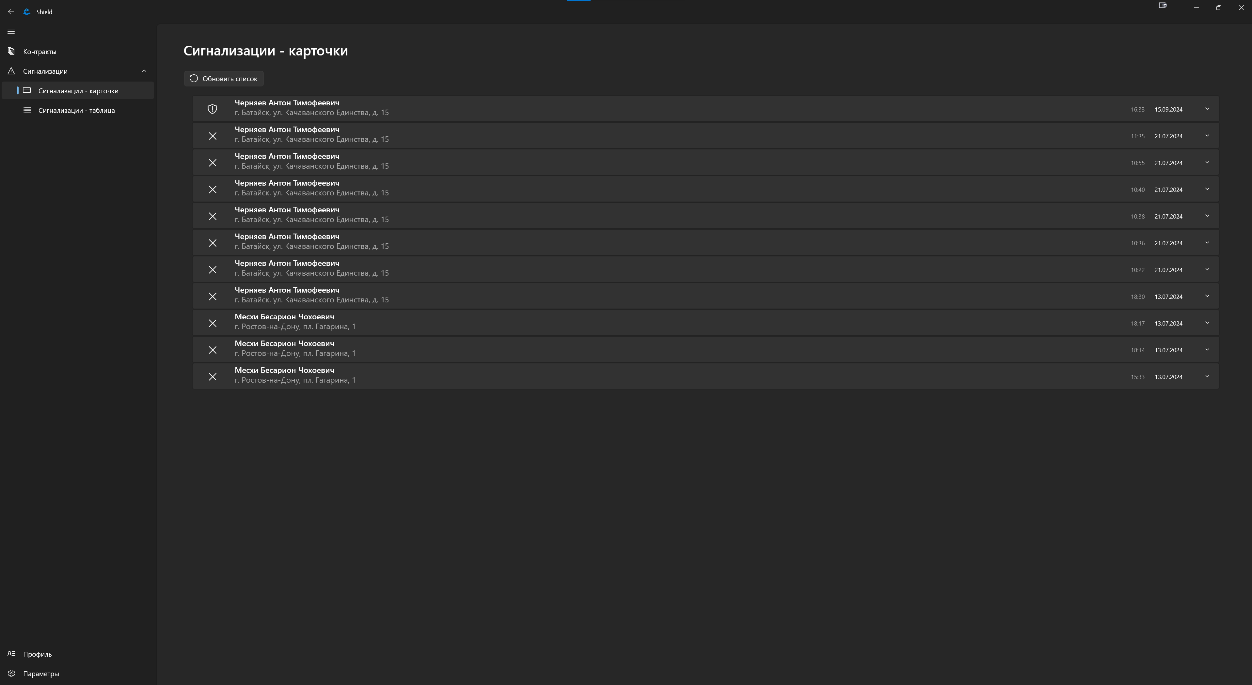


Рисунок 5 – окно “Cигнализации – карточки”

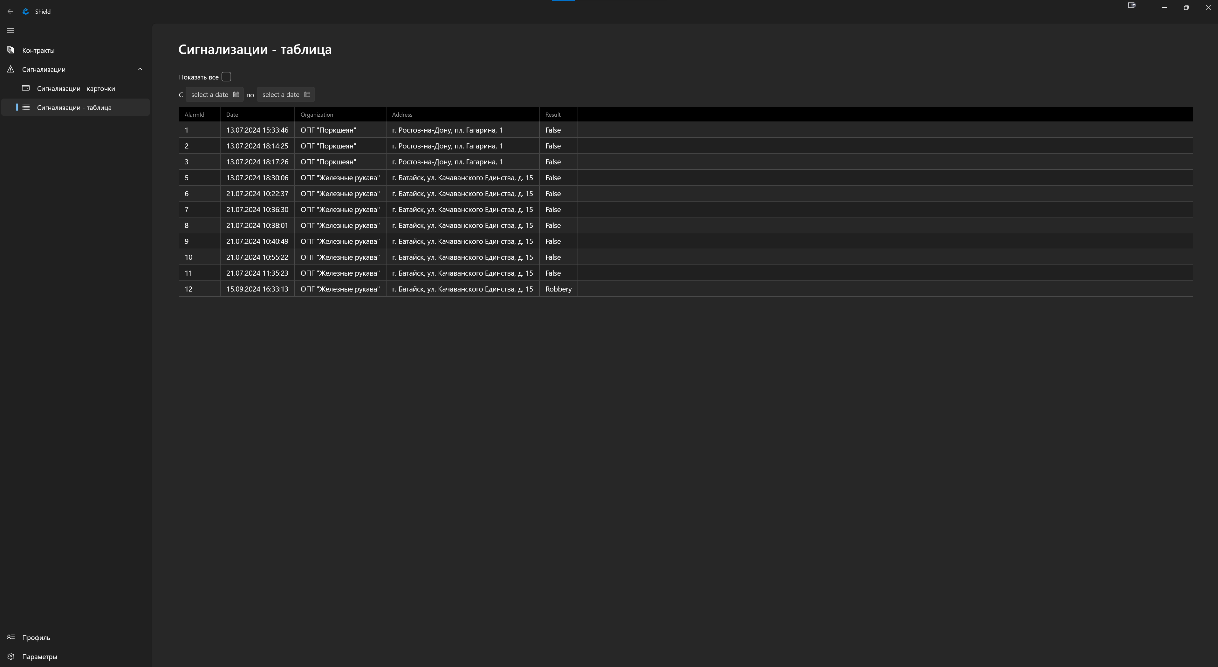


Рисунок 6 – окно “Сигнализации – таблица”

### **Аппаратные интерфейсы**

Система "Shield" взаимодействует с внешними аппаратными системами охранной сигнализации, установленными на охраняемых объектах. Это взаимодействие является односторонним (прием информации) и осуществляется через специализированное программное обеспечение производителей сигнализаций или аппаратные шлюзы. Прямое подключение "Shield" к "железу" сигнализации (датчикам, контрольным панелям) не предусмотрено базовой архитектурой.

## Лицензионные требования

Лицензионные ограничения накладываются с помощью лицензии MIT.

## Предупреждения, касающиеся законодательства, авторских прав и другие замечания

На данный момент на данное программное обеспечение не накладываются никакие предупреждения, касающиеся законодательства, авторских прав, так как программа не зарегистрирована в Едином реестре и не имеет зарегистрированного патента.

## Применяемые стандарты

[ГОСТ 28195-89](https://gost.ruscable.ru/Index/11/11212.htm) - Оценка качества программных средств. Общие положения

[ГОСТ 28806-90](https://gost.ruscable.ru/Index/10/10605.htm) - Качество программных средств. Термины и определения.

[ГОСТ Р 51188-98](https://gost.ruscable.ru/Index/18/18664.htm) - Защита информации. Испытания программных средств на наличие компьютерных вирусов. Типовое руководство.

ГОСТ 19 105–78 - Единая система программной документации. Общие требования к программным документам.

**4. Сопровождающая информация**

**Приложение А. Глоссарий**

* База данных (БД) – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.
* Информационная система (ИС) – система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.
* Программное обеспечение (ПО) — это совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники.
* Система управления базами данных (СУБД) - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.
* Пользователь – лицо или организация, которое использует действующую систему для выполнения конкретной функции.
* Логическое проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных.
* Даталогическая схема –  организация данных, выделенных на предыдущем этапе проектирования в форму, принятую в выбранной СУБД.

**Приложение Б. Модели процессов и предметной области (диаграммы)**

Изображение выглядит как текст, диаграмма, План, Параллельный

Автоматически созданное описание

Рисунок 7 – диаграмма верхнего уровня

Изображение выглядит как снимок экрана, диаграмма, Прямоугольник, прямоугольный

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 8 – DFD диаграмма нулевого уровня иерархии (приведенная)

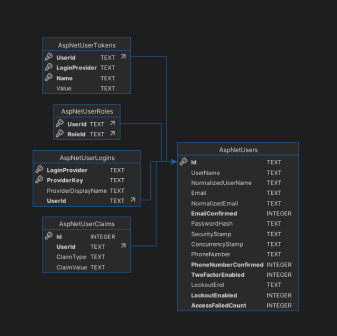


Рисунок 9 – даталогическая модель для пользователей

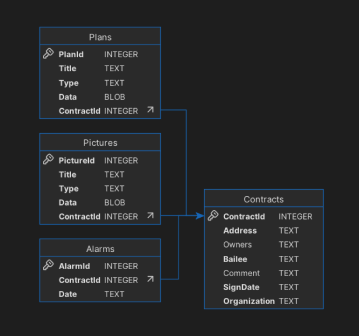


Рисунок 10 – даталогическая модель для ОВО

**Приложение В. Список ключевых задач, что необходимо сделать в будущем.**

* Разработать Мобильное приложение для патрульных;
* Добавить аналитику на основе AI;
* Добавить поддержку распределенных ОВО (облачная инфраструктура).