

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ЭВМ
_____ Д.В. Топольский
«___» _____ 2022 г.

РАЗРАБОТКА КЛИЕНТСКОЙ ЧАСТИ СИСТЕМЫ ПОЗИЦИОНИРОВАНИЯ
ВНУТРИ ПРОИЗВОДСТВЕННЫХ ПОМЕЩЕНИЙ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2022.406 ПЗ ВКР

Руководитель работы,
к.т.н., доцент каф. ЭВМ
_____ П.О. Шабуров
«___» _____ 2022 г.

Автор работы,
студент группы КЭ-406
_____ И.Р. Нестеров
«___» _____ 2022 г.

Нормоконтролёр,
к.п.н., доцент каф. ЭВМ
_____ М.А. Алтухова
«___» _____ 2022 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«__» _____ 2022 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ-406

Нестерову Илье Романовичу

обучающемуся по направлению

09.03.01 «Информатика и вычислительная техника»

1. Тема работы: «Разработка клиентской части системы позиционирования внутри производственных помещений» утверждена приказом по университету от 12 декабря 2021 г. № 308/141.

2. Срок сдачи студентом законченной работы: 1 июня 2022 г.

3. Исходные данные к работе:

Обеспечить основной функционал приложения:

- работа с объектами системы позиционирования в графическом интерфейсе посредством виртуальной карты;
- возможность загружать план помещения в качестве карты;
- обеспечение связи с серверной частью программного обеспечения;
- настройка аппаратной части системы позиционирования;
- система авторизации и настройка прав доступа пользователей.

4. Перечень подлежащих разработке вопросов:

- анализ существующих программных продуктов, используемых для позиционирования объектов внутри помещений;
- анализ технологических решений для разработки клиентского приложения, проектирование архитектуры программного обеспечения;
- разработка программного продукта, реализующего клиентскую часть системы позиционирования в производственных помещениях;
- отладка программного обеспечения, тестирование разработанного с использованием выбранных методологий.

5. Дата выдачи задания: 1 декабря 2021 г.

Руководитель работы _____ /П.О. Шабуров/

Студент _____ /И.Р. Нестеров/

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы (анализ существующих программных продуктов, используемых для позиционирования объектов внутри помещений)	07.02.2022	
Разработка модели, проектирование (анализ технологических решений для разработки клиентского приложения, проектирование архитектуры программного обеспечения)	07.03.2022	
Реализация системы (разработка программного продукта, реализующего клиентскую часть системы позиционирования в производственных помещениях)	04.04.2022	
Тестирование, отладка, эксперименты (отладка программного обеспечения, тестирование разработанного с использованием выбранных методологий)	10.05.2022	
Компоновка текста работы и сдача на нормоконтроль	16.05.2022	
Подготовка презентации и доклада	24.05.2022	

Руководитель работы _____ /П.О. Шабуров/

Студент _____ /И.Р. Нестеров/

АННОТАЦИЯ

И.Р. Нестеров. Разработка клиентской части системы позиционирования внутри производственных помещений. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2022, 36 с., 14 ил., библиогр. список – 15 наим.

В рамках выпускной квалификационной работы выполнены все этапы по проектированию и разработке клиентской части программного обеспечения системы позиционирования внутри производственных помещений. Проанализированы недостатки имеющегося на предприятии программного обеспечения, на основе сравнительного анализа выбраны технологические решения для реализации клиентского приложения. Выбран шаблон проектирования архитектуры приложения и определена структура клиентского приложения. На основе описанной архитектуры с использованием выбранных технологических решений была разработана клиентская часть программного обеспечения системы позиционирования. После разработки было произведено тестирование программного обеспечения по выбранной методологии.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 Обзор аналогов	9
1.2 Анализ основных технологических решений	10
1.3 Вывод.....	13
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	15
2.1 Функциональные требования	15
2.2 Нефункциональные требования	15
3 ПРОЕКТИРОВАНИЕ.....	17
3.1 Архитектура предлагаемого решения.....	17
3.2 Алгоритмы решения	19
4 РЕАЛИЗАЦИЯ	23
5 ТЕСТИРОВАНИЕ	29
5.1 Методология тестирования.....	29
5.2 Проведение процедуры тестирования	29
ЗАКЛЮЧЕНИЕ	33
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	34

ВВЕДЕНИЕ

С развитием информационных технологий в мире интерес к позиционированию внутри помещений в различных отраслях только растет, а на крупных предприятия данное нововведение может быть востребовано для оптимизации производственных процессов, связанных с логистикой и передвижением сотрудников или материальных объектов [1].

Для реализации системы позиционирования могут быть использованы различные технологии, такие как BLE (Bluetooth Low Energy), Wi-Fi, RFID, системы локального позиционирования, основанные на ультразвуковых волнах и инфракрасном излучении. Основные различия заключаются в дальности и точности, на которые рассчитана выбранная система. Ультразвуковые волны обеспечивают высочайшую среди аналогов точность вплоть до нескольких сантиметров и не распространяются через стены, что гарантирует однозначное определение нахождения человека или материального объекта в конкретном помещении [2].

Программное обеспечение, поставляемое с выбранной предприятием системой, не удовлетворяет всем требованиям, поэтому заказчику необходимо разработка собственного программного обеспечения, реализующего взаимодействие с аппаратной частью системы позиционирования.

Целью выпускной квалификационной работы является проектирование и разработка ПО, соответствующего функциональным требованиям, предоставленным предприятием ООО «ДСТ-УРАЛ». Создание клиентского приложения должно обеспечить оптимизацию производственного процесса на этапах транспортировки.

Задачи в ходе создания системы. В программе должен быть реализован понятный пользователю графический интерфейс с интерактивной координатной

плоскостью для отображения местоположения отслеживаемых объектов с использованием координат, полученных от серверной части программного обеспечения, а также возможность загружать изображения в качестве плана помещения и соотносить их на карте с объектами. Программа должна реализовывать следующие функции: получение координат в разных форматах, настройки главного контроллера системы навигации (далее Модем), настройки всех маяков как подвижных, так и стационарных, сообщать о наличии ошибок в позиционировании. Для корректной и полноценной работы всей системы в клиентском ПО требуется внедрить спроектированный программный интерфейс приложения (Application Programming Interface, далее API), описывающий все возможные методы и типы данных, которые может запросить и передать серверная часть системы.

В данной работе были поставлены следующие задачи:

- при наличии аналогов проанализировать их функционал;
- выбрать язык программирования, среду и методы разработки;
- на основе ТЗ и сбора данных по аналогам спроектировать ПО, описав модули;
- реализовать функциональную версию ПО;
- провести тестирование и отладку рабочей версии ПО.

В первой главе представлен обзор аналогов и анализ технологических решений для реализации программного обеспечения. Во второй главе описаны требования к реализуемому приложению. Третья глава содержит в себе проектирование архитектуры и описание используемых алгоритмов. В четвертой главе – реализация программного продукта. В пятой главе описано тестирование разработанного программного обеспечения.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Обзор аналогов

На предприятии выбрана система позиционирования «Marvelmind», основанная на использовании ультразвуковых волн. Данная система была выбрана из-за точности ультразвуковых технологий, невысокой стоимости и возможности связаться с разработчиками данной системы [3].

Основной задачей, поставленной предприятием, является создание программного обеспечения, которое заменит ПО, поставляемое с выбранной системой.

Поставляемое программное обеспечение не удовлетворяет предприятие по следующим критериям:

- визуально перегруженный интерфейс;
- нестабильная работа программного обеспечения;
- отсутствие модуля авторизации пользователя;
- невозможность интегрировать клиентскую часть приложения в имеющиеся производственные процессы.

Главным критерием, по которому не подходит данное ПО, является возможность интегрировать его в процессы на производстве. Открытый исходный код написанного взамен приложения позволит использовать его на предприятии в дальнейшем, реализовывать новые необходимые функции и связывать с уже имеющимися на предприятии приложениями и структурами данных.

На рисунке 1 представлен интерфейс поставляемого программного обеспечения системы «Marvelmind».

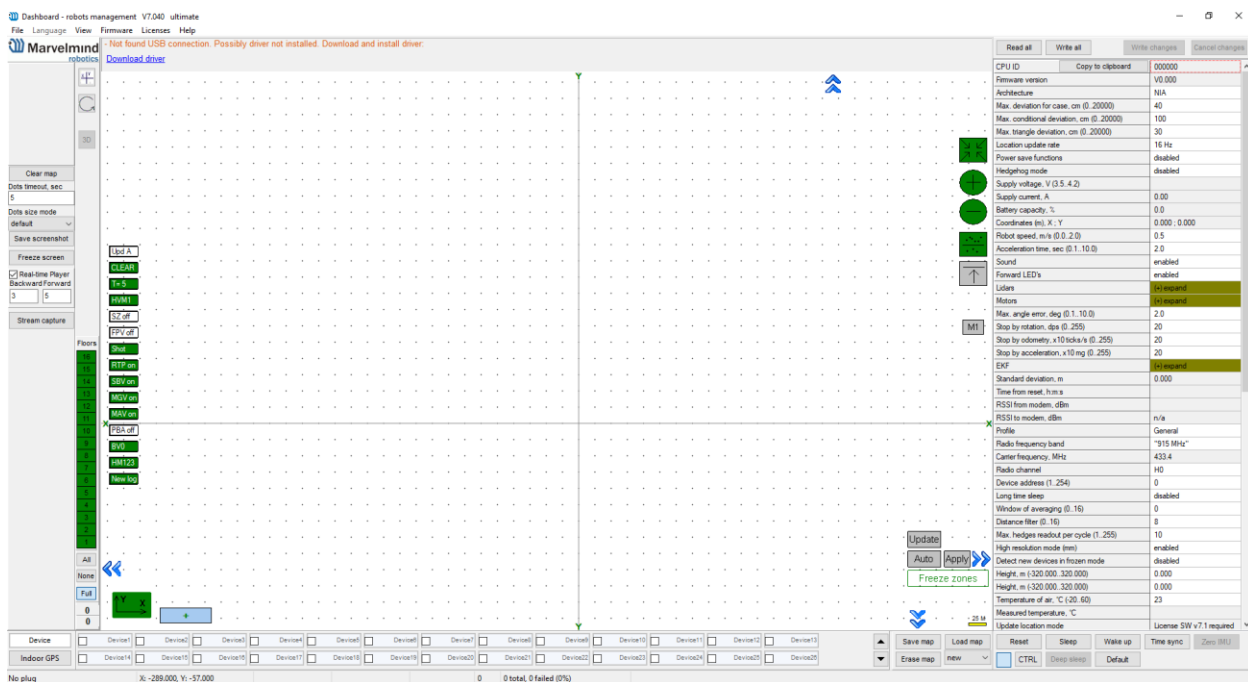


Рисунок 1 – Поставляемое ПО системы «Marvelmind»

1.2 Анализ основных технологических решений

Исходя из задания, необходимо выбрать язык программирования, который позволит разработать клиентскую часть программного обеспечения, приложение для персонального компьютера. Для создания приложения, помимо языка программирования, необходимо выбрать систему для разработки клиентских приложений. Одними из самых известных являются следующие платформы: Windows Forms, Qt, WPF. Проанализируем преимущества и недостатки данных платформ для принятия решения.

Windows Forms – платформа для создания пользовательского интерфейса приложений Windows. Пользуясь данной платформой, разработчик может обращаться к стандартным элементам интерфейса операционной системы Microsoft Windows и создавать с их помощью клиентские приложения. Windows Forms была представлена в 2002 году и является одной из самых простых

систем для создания приложений, но новые возможности перестали появляться в данной системе с 2014 года [4].

Таким образом, преимущества данной системы заключаются в следующем:

- простота в изучении и использовании;
- большое количество элементов управления;
- меньшие требования к производительности и памяти.

К недостаткам данной системы можно отнести следующие пункты:

- отсутствие языка разметки для создания графического интерфейса;
- ограниченный функционал системы по сравнению с современными альтернативами;
- отсутствие обновлений и поддержки данной системы.

Qt – платформа для разработки кроссплатформенного программного обеспечения на различных языках программирования, среди которых, например, C++, Python, Java и другие. Qt является популярным и мощным средством для разработки приложений, которое используется во многих крупных компаниях [5].

Преимущества Qt:

- широкий функционал для создания приложений;
- обширная, современная, обновляющаяся документация;
- возможность использовать различные языки программирования и операционные системы;

Недостатки Qt:

- высокий порог вхождения;
- большой вес приложений.

WPF (Windows Presentation Foundation) – графическая система для создания клиентских приложения, развитие известной графической системы

Windows Forms, включающая в себя более обширные возможности для создания приложений. WPF включает в себя средства для создания визуального интерфейса, включая язык разметки, элементы управления Windows, возможность изменять и создавать собственные элементы управления, привязку данных, макеты, стили, шаблоны, документы, текст, изображения и видео, двумерную и трёхмерную графику, анимацию. Для графических элементов используется DirectX, за счет которого производительность данной системы выше, чем у Windows Forms. При создании приложений, реализации логики их работы используется .NET-совместимые языки программирования [6].

Преимущества WPF:

- большой набор инструментов для создания графического интерфейса;
- поддержка Microsoft и актуальная, обновляющаяся документация;
- производительность системы при работе с графикой;
- широкие возможности для рисования графических объектов и анимации;
- возможность интеграции сетевой части программного обеспечения на языке C#.

Таким образом, при разработке приложения будет использоваться WPF, так как эта система сочетает в себе продвинутый функционал для работы с графикой, производительность и возможность использовать .NET-совместимый язык, например, VB.NET или C#, с помощью которого будет обеспечено взаимодействие с серверной частью программного обеспечения.

VB.NET (Visual Basic .NET) – язык объектно-ориентированный язык программирования, развитие языка Visual Basic, реализованное на платформе .NET Framework. Различия между VB.NET и C# в основном заключается лишь в синтаксисе данных языков, но, так как C# является C-подобным языком, большее количество программистов смогут работать с исходным кодом

программного обеспечения [7]. Несмотря на то, что VB.NET и C# являются первыми языками, разработанными компанией Microsoft для данной платформы, C# более популярен среди разработчиков, в следствие чего документации, книг, учебных пособий, статей, сообществ программистов гораздо больше, чем для VB.NET. Согласно исследованию, проведенному в 2021 году ресурсом для разработчиков Stack Overflow, 27,86% разработчиков используют C# и лишь 4.66% используют Visual Basic [8].

C# – компилируемый язык программирования, который относится к семейству C-подобных языков, имеет статическую типизацию, возможность оптимизировать память, посредством размещения в памяти данных в строгом порядке, является высокоуровневым и объектно-ориентированным. Также отметим, что платформа .NET Framework позволяет использовать разные языки программирования в рамках одного проекта, что может быть использовано заказчиком при дальнейшем развитии и масштабировании данного программного обеспечения. Принципы объектно-ориентированного программирования позволяют эффективно работать с модулями программного обеспечения, например, инкапсуляция данных и методов для их защиты от пользователей [9].

В качестве среды разработки используется Microsoft Visual Studio 2019, в которой возможна работа с WPF и которая предоставляет широкий набор библиотек и инструментов для создания клиентского программного обеспечения.

1.3 Вывод

Для решения поставленной задачи была выбрана система построения пользовательского интерфейса WPF, так как она включает в себя широкий

функционал по работе с графическим интерфейсом, производительность и доступность.

Из ограниченного набора доступных .NET-совместимых языков программирования был выбран язык C#. Данный язык был выбран из-за его распространенности и богатого функционала в сочетании со средой разработки Microsoft Visual Studio 2019.

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

2.1 Функциональные требования

Требования к разрабатываемому программному обеспечению заключаются в следующем:

- настройка аппаратной части системы позиционирования посредством графического интерфейса согласно документации системы позиционирования;
- настройка главного управляющего устройства (Модем);
- получение данных от Модема о характеристиках всей системы, ее состоянии и возможных ошибках в позиционировании маяков;
- настройка авторизации и прав доступа к данным всей системы для клиентских приложений;
- вывод данных о текущих координатах объектов в графическом интерфейсе в реальном времени;
- вывод объектов на графической карте с возможностью загрузить изображение в качестве плана помещения;
- вывод системной статистики: по ошибкам системы позиционирования, по ошибкам связи с аппаратной частью системы, по внутренним ошибкам клиентского приложения при их наличии.
- возможность сохранить конфигурационный файл с настройками маяков в текстовом формате.

2.2 Нефункциональные требования

Приложение должно соответствовать следующим системным требованиям:

- иметь совместимость с платформами семейств Windows.

- код программы должна быть написан на языке программирования C#;
- все вычисления должны производиться программой локально на самом устройстве;
- приложение должно быть разработано с использованием унифицированной платформы .NET5.0.

3 ПРОЕКТИРОВАНИЕ

3.1 Архитектура предлагаемого решения

В качестве шаблона проектирования архитектуры приложения используется паттерн MVVM (Model-View-ViewModel). Данный паттерн позволяет разделить внешнее представление приложения и его логику работы с данными. На рисунке 2 представлена структура приложения при использовании данного шаблона.

Согласно данному шаблону, приложение делится на три части:

- модель (Model);
- представление (View);
- модель представления (ViewModel).



Рисунок 2 – Шаблон MVVM

Модель включает в себя структуры данных, используемые в приложении и логику, связанную с этими данными. Данная часть приложения не должна содержать в себе методов и логики, связанной с визуальным представлением этих данных.

Представление определяет логику работы визуального интерфейса, взаимодействие с языком разметки и непосредственно графическим интерфейсом, с которым взаимодействует пользователь.

Модель представления содержит в себе логику по получению данных из модели, которые передаются в представление и реализацию привязки данных,

благодаря которой свойства в представлении обновляются без прямой связи с моделью [10].

Структурно программное обеспечение разделено на следующие модули:

- графический модуль;
- модуль авторизации пользователя;
- модуль настройки маяков и Модема;
- интерпретация входящих пакетов;
- API.

Принципы объектно-ориентированного программирования позволят эффективно работать с модулями программного обеспечения, например, инкапсуляция данных и методов для их защиты от пользователей.

Графический модуль должен включать в себя метод обработки изображения для их использования при наложении плана помещения/здания на передвижение маяков, метод графической отрисовки подвижных маяков и их пройденного пути, возможность изменения пропорций карты, отображение координат и расстояний между маяками.

Модуль авторизации пользователей является окном, вызываемым при запуске приложения, с формой для авторизации, т.е. полями ввода логина и пароля.

Модуль настройки маяков и Модема включает в себя все аппаратные и программные настройки доступные в системе «Marvelmind», возможность сохранения их в текстовом формате, их загрузка из файла, настройки отдельных маяков и всех сразу.

Программный интерфейс приложения необходимо интегрировать для взаимодействия клиентской и серверной частей программного обеспечения. Данный модуль включает в себя методы, которые описывают прием и передачу информации между серверной частью и клиентским приложением.

3.2 Алгоритмы решения

Основными структурами данных на уровне Модель являются классы Beacon.cs и Modem.cs. Класс Beacon.cs содержит в себе параметры маяка, например, идентификационный номер, координаты, статус активен ли маяк, является ли маяк подвижным. Класс Modem.cs содержит в себе перечень параметров, доступных для настройки Модема.

Базовым классом на уровне Модель представления является класс ViewModel.cs. Данный класс унаследован от интерфейса INotifyPropertyChanged, что позволяет производить автоматическое обновление данных в Модели и отображение данных в Представлении без прямой связи между этими уровнями. Данный класс представлен в листинге 1.

Листинг 1 – Класс ViewModel.cs

```
internal abstract class ViewModel : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    protected virtual void OnPropertyChanged([CallerMemberName]
string PropertyName = null)
    {
        PropertyChanged?.Invoke(this, new
PropertyChangedEventArgs (PropertyName));
    }

    protected virtual bool Set<T>(ref T field, T value,
[CallerMemberName] string PropertyName = null)
    {
        if (Equals(field, value)) return false;
        field = value;
        OnPropertyChanged (PropertyName);
        return true;
    }
}
```

Для реализации основной логики программы используется класс MainWindowViewModel.cs, наследуемый от класса ViewModel. В нем содержится логика программы по взаимодействию пользователя с элементами

управления приложения, логика работы элементов управления, привязка данных к Представлению. Пример создания полей, которые будут использованы для привязки данных в представлении, описан в листинге 2.

Листинг 2 – Пример создания поля на уровне модель представления

```
private string _UartBaud = "";
public string UartBaud
{
    get => SelectedBeacon.Settings.UartBaud.ToString();
    set => Set(ref _UartBaud, value);
}

private string _Title = "Marvelous GUI";
public string Title
{
    get => _Title;
    set => Set(ref _Title, value);
}
```

Для взаимодействия пользователя с элементами управления в архитектуре MVVM реализуются команды. Команды реализуются с помощью интерфейса ICommand. Базовый класс команд Command.cs представлен в листинге 3.

Листинг 3 – Класс Command.cs

```
internal abstract class Command : ICommand
{
    public event EventHandler CanExecuteChanged
    {
        add => CommandManager.RequerySuggested += value;
        remove => CommandManager.RequerySuggested -= value;
    }

    public abstract bool CanExecute(object parameter);
    public abstract void Execute(object parameter);
}
```

Пример создания команды на уровне модель представления описан в листинге 4.

Листинг 4 – Пример создания команды на уровне модель представления

```
public ICommand CloseAppCommand { get; }

private bool CanCloseAppCommandExecuted(object p) =>
true;
```

```

private void OnCloseAppCommandExecuted(object p)
{
    Application.Current.Shutdown();
}
public MainWindowViewModel()
{
    CloseAppCommand = new Cmd(OnCloseAppCommandExecuted,
CanCloseAppCommandExecuted);
}

```

На уровне Представление реализован графический интерфейс. Графический интерфейс представлен файлом MainWindow.xaml. XAML (eXtensible Application Markup Language) – основанный на XML, расширяемый язык разметки для приложений, используемый в системе WPF. В нем содержатся разметка окна приложения, осуществляется привязка данных к конкретным элементам управления [10]. Согласно модели MVVM класс MainWindow.xaml.cs, не должно содержать основной логики программы, поэтому класс имеет следующий вид, представленный в листинге 2 [11].

Для связи с серверной частью выбран сетевой протокол передачи данных UDP [12]. Данный протокол обеспечивает надежность, целостность и упорядоченность данных. Сетевое взаимодействие организует класс Network.cs, в котором содержатся методы, описывающие прием и отправку пакетов.

Для взаимодействия с серверной частью интегрирована система сериализации данных в формате JSON для удобного представления структур данных в процессе передачи данных между клиентской и серверной частями программного обеспечения. Текстовый формат обмена данными JSON легко читается людьми, что позволит быстро обрабатывать и анализировать данные. На листингах 5, 6, 7 представлены примеры JSON-сообщений, получаемых при взаимодействии с серверной частью программного обеспечения [13].

Листинг 5 – Пример передаваемого JSON-сообщения для смены адреса маяка

маяка

```
{
  "packet_type": 6001,
  "prev_address": 0x02,
  "new_address": 0x03
}
```

Листинг 6 – Пример передаваемого JSON-сообщения для запуска маяка

```
{
  "packet_type": 6003,
  "dst_address": 0x03,
  "access_mode": 0x02,
  "command": 0x02
}
```

Листинг 7 – Пример ответного JSON-сообщения на запрос версии прошивки

прошивки

```
{
  "packet_type": 4010,
  "device_address": 0x03,
  "version": "6.01",
  "device_type_id": 42
}
```

Реализация функции загрузки изображения в качестве плана помещения представлена в листинге 8.

Листинг 8 – Загрузка изображения в качестве плана помещения

```
OpenFileDialog op = new OpenFileDialog();
op.Title = "Select a picture";
op.Filter = "All supported graphics|*.jpg;*.jpeg;*.png|" +
  "JPEG (*.jpg;*.jpeg)|*.jpg;*.jpeg|" +
  "Portable Network Graphic (*.png)|*.png";
if (op.ShowDialog() == true)
{
  MapImg.Source = new BitmapImage(new Uri(op.FileName));
}
```

4 РЕАЛИЗАЦИЯ

Двумя основными окнами в приложении являются окна авторизации пользователя и главное окно. Главное окно содержит в себе поля для ввода имени пользователя и пароля, кнопку для подтверждения ввода и перехода к главному окну при правильности ввода. Окно авторизации пользователя представлено на рисунке 3.

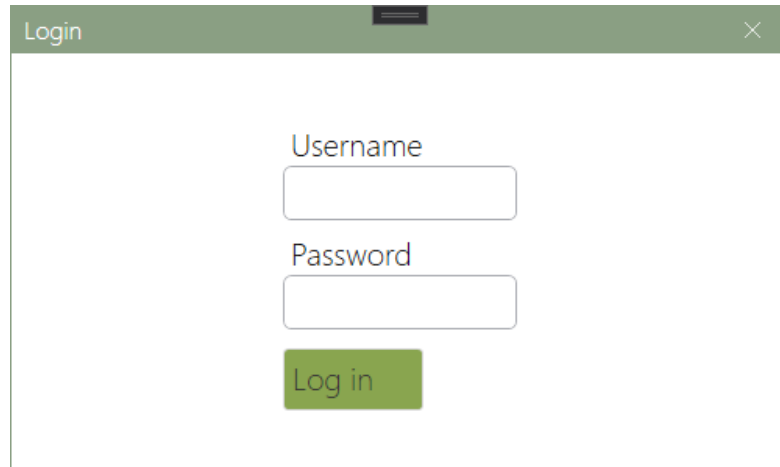


Рисунок 3 – Окно авторизации пользователя

Главное окно изображено приложения на рисунке 4.

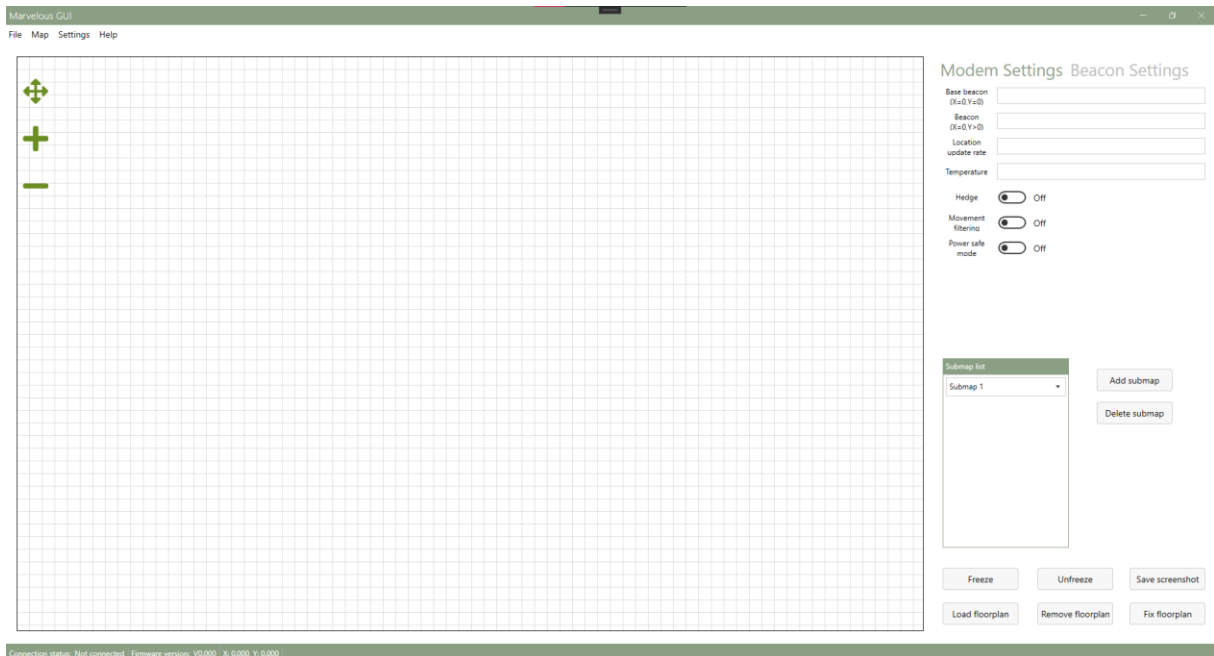
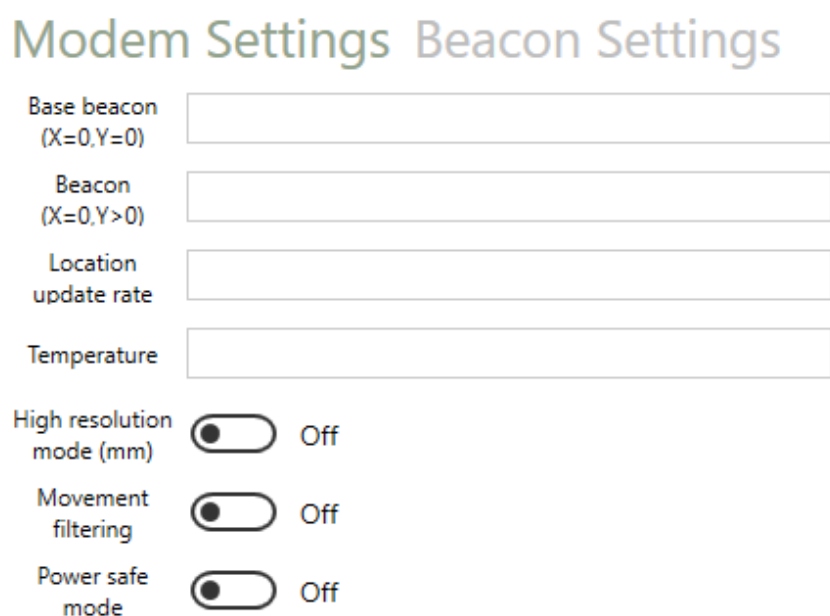


Рисунок 4 – Главное окно приложения

В правой части окна расположен модуль настроек Модема и маяков. Оставшуюся часть занимает графический модуль, представленный в виде виртуальной карты с возможностью загрузки плана помещения и масштабирования. Модуль настроек включает в себя четыре раздела: Modem Settings, Beacon Settings, Submap List и элементы управления, взаимодействующие с графическим модулем.

Модуль настройки Модема (Modem Settings) представлен на рисунке 5.



Modem Settings Beacon Settings

Base beacon
(X=0,Y=0)

Beacon
(X=0,Y>0)

Location
update rate

Temperature

High resolution
mode (mm) Off

Movement
filtering Off

Power safe
mode Off

Рисунок 5 – Модуль настройки Модема

Данный модуль включает в себя доступные пользователю настройки главного контроллера системы навигации, например, включение режимов высокой точности (измерение дистанций, координат в миллиметрах), экономии заряда Модема, изменение параметров температуры окружающей среды и частоты обновления местоположения.

На рисунке 6 изображен модуль настройки маяков (Beacon Settings).

Modem Settings Beacon Settings

The image shows a web interface for configuring beacon settings. At the top, there is a dropdown menu. Below it are several input fields: 'ID', 'UART baudrate', 'Radio profile', 'Radio band', and 'Coord, X;Y'. Below these is another dropdown menu labeled 'Submap'. Further down is a 'Hedge' toggle switch, which is currently turned off. At the bottom of the interface are three buttons: 'Wake up', 'Sleep', and 'Reset'.

Рисунок 6 – Модуль настройки маяков

В данном модуле пользователь с помощью выпадающего списка в верхней части может выбрать маяк, который необходимо настроить. Для просмотра и изменения пользователю доступны следующие настройки:

- идентификационный номер маяка;
- скорость передачи данных по интерфейсу UART;
- битрейт;
- частота, используемая в маяке;
- координаты маяка;
- принадлежность к подмножеству маяков на карте;
- статус, является ли данный маяк подвижным (hedge) или статичным.

Также в данном модуле пользователя может активировать или отправить маяк в спящий режим (wake up, sleep), восстановить исходные настройки маяка.

В разделе Submap list пользователь может настраивать подмножества маяков. Этот раздел необходим для распределения маяков с одинаковой частотой в разные подмножества. Данный раздел представлен на рисунке 7.

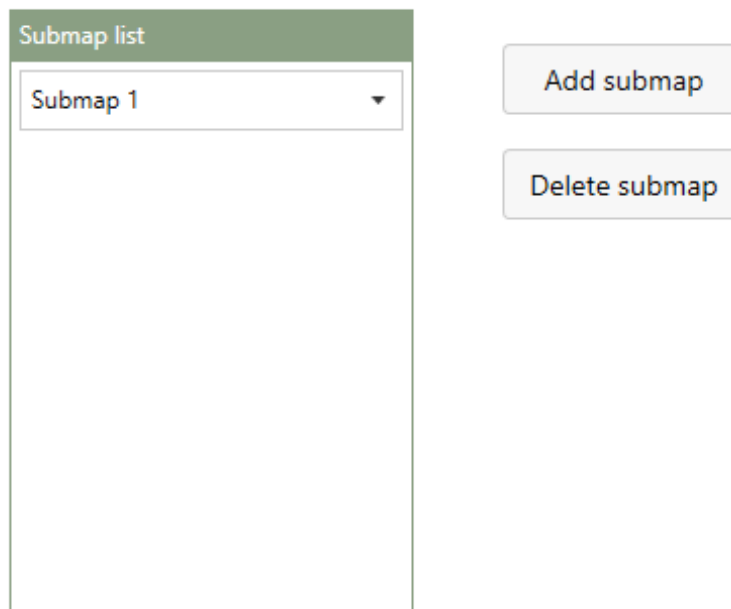


Рисунок 7 – Модуль настройки подмножеств маяков

Взаимодействие с графическим модулем реализовано в виде элементов управления, представленных на рисунке 8.

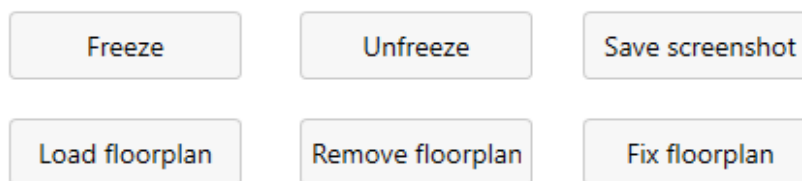


Рисунок 8 – Элементы управления, взаимодействующие с графическим модулем.

В данном блоке пользовательского интерфейса расположены следующие элементы управления:

– кнопки «Freeze» и «Unfreeze» позволяют закрепить и открепить соответственно карту маяков, т.е. закрепить местоположение стационарных маяков;

– кнопка «Save screenshot» позволяет сделать снимок карты с расположенными на ней маяками;

– кнопка «Load floorplan» и «Remove floorplan» позволяет загрузить в качестве фона план помещения в формате изображения и удалить его соответственно;

– кнопка «Fix floorplan» позволяет закрепить план помещения по отношению к расположенным на ней маякам после их соотнесения относительно друг друга.

На рисунках 9 и 10 показана возможность выбора изображения в качестве плана помещения.

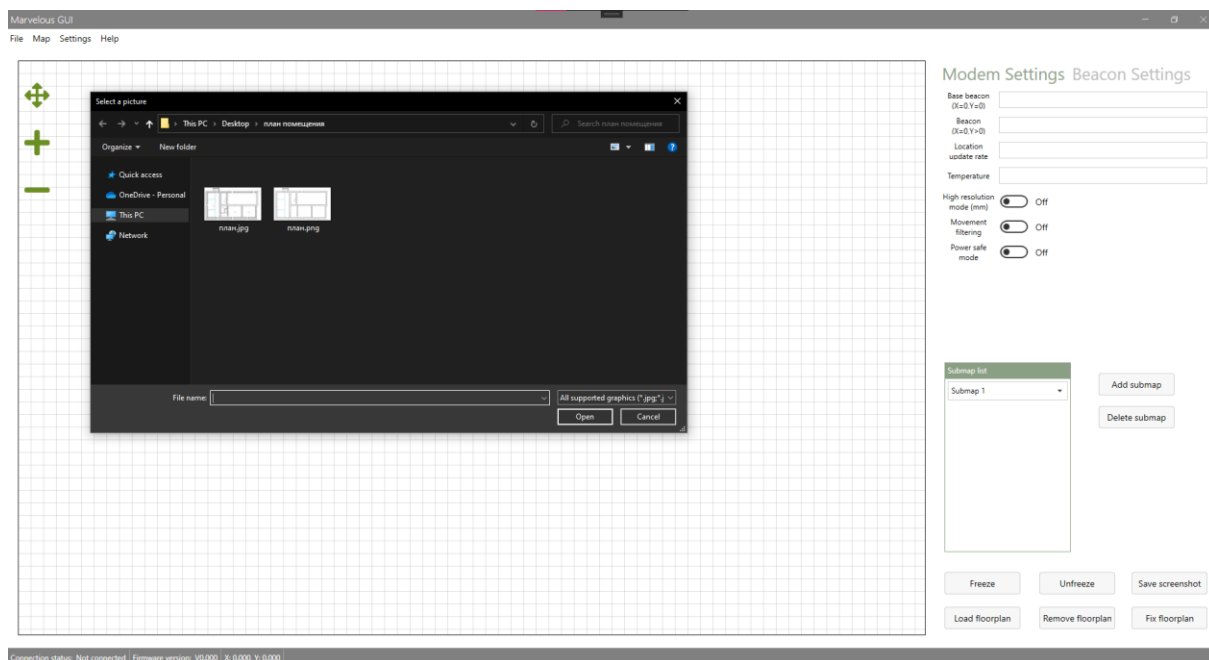


Рисунок 9 – Выбор плана помещения.

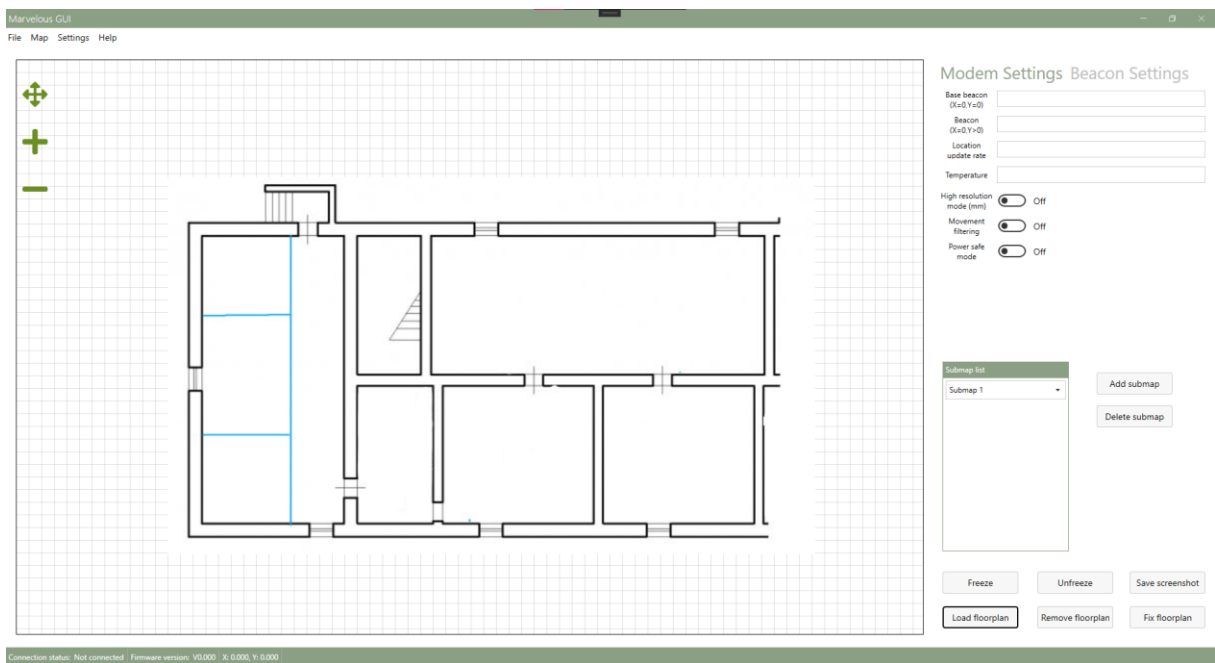


Рисунок 10 – Использование изображения в качестве плана помещения

План помещения можно изменять с помощью мыши, реализованы методы нажатия левой и правой клавиши мыши, метод масштабирования с помощью колеса мыши.

5 ТЕСТИРОВАНИЕ

5.1 Методология тестирования

Для тестирования программного обеспечения выбраны следующие виды тестирования:

- тестирование удобства использования;
- интеграционное тестирование.

Тестирование удобства использования – метод тестирования интерфейса со стороны степени удобства использования, понятности и привлекательности для пользователей [14].

Интеграционное тестирование – метод тестирования программного обеспечения, направленный на проверку связи и правильного взаимодействия между компонентами системы [15].

5.2 Проведение процедуры тестирования

Тестирование удобства использования было проведено методом предоставления программного обеспечения пользователям разного уровня владения персональным компьютером. В ходе тестирования проблем с использованием пользовательского интерфейса не возникало, интерфейс не перегружен и доступен.

Интеграционное тестирование заключалось в проверки взаимодействия разработанной клиентской части программного обеспечения с имеющейся на предприятии серверной частью.

В рамках данного тестирования были проверены следующие пункты:

- отображение доступных для настройки маяков в модуле настройки маяков;

- отображение настроек Модема в модуле настройки модема;
- отображение маяков в графическом модуле, т.е. на виртуальной карте;
- проверка состояния подвижного маяка.

Результаты данного тестирования представлены на рисунках 11-14.

Modem Settings Beacon Settings

Beacon 1

Beacon 2

Beacon 3

Beacon 4

Radio profile

Radio band

Coord, X;Y

Submap

Hedge Off

Wake up Sleep Reset

Рисунок 11 – Отображение доступных для настройки маяков в модуле настройки маяков

Modem Settings Beacon Settings

Base beacon (X=0,Y=0)	<input type="text" value="0x01"/>
Beacon (X=0,Y>0)	<input type="text" value="0x02"/>
Location update rate	<input type="text" value="1"/>
Temperature	<input type="text" value="22"/>
High resolution mode (mm)	<input checked="" type="checkbox"/> On
Movement filtering	<input type="checkbox"/> Off
Power safe mode	<input type="checkbox"/> Off

Рисунок 12 – отображение настроек Модема в модуле настройки модема

На рисунке 13 можно увидеть, что маяки отображаются на карте, маяк с идентификационным номером является подвижным и выделен другим цветом. На данном рисунке в нижней части окна также можно увидеть, что статус подключения к Модему Connected, версия прошивки Модема получена.

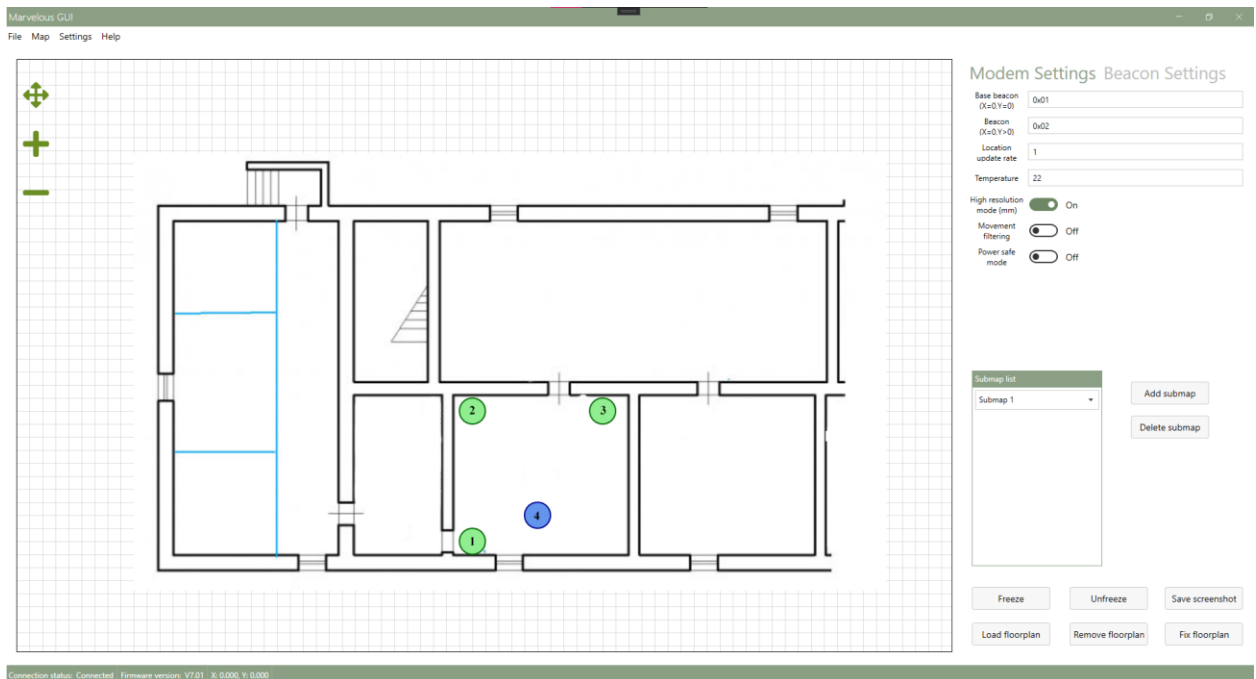


Рисунок 13 – Отображение маяков в графическом модуле

На рисунке 14 представлен модуль настройки подвижного маяка, на котором мы видим, что у 4 маяка переключатель Hedge установлен в положение On.

Modem Settings Beacon Settings

Beacon 4

ID

UART baudrate

Radio profile

Radio band

Coord, X;Y

Submap

Hedge On

Рисунок 14 – Модуль настройки 4 маяка

По итогам тестирования можно сделать вывод, что разработанная клиентская часть программного обеспечения успешно взаимодействует с серверной частью.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была разработана клиентская часть программного обеспечения системы позиционирования в производственных помещениях.

Были решены следующие задачи:

- проведен анализ существующего аналога программного обеспечения;
- выбраны технологические решения для реализации;
- выполнено проектирование системы;
- выполнена реализация системы;
- проведено тестирование рабочей версии программного обеспечения.

Разработанное программное обеспечение соответствует требованиям предприятия заказчика. Клиентское приложение, установив связь с серверным приложением, позволяет пользователю отслеживать местоположение маяков, настраивать их и взаимодействовать с аппаратной частью системы позиционирования. Реализована интерактивная виртуальная карта с возможностью загружать план помещения в качестве фона, соотносить маяки с планом и отслеживать их в реальном времени.

В дальнейшем данное программное обеспечение возможно интегрировать в имеющуюся на предприятии систему, связать с сервисами, которые используются в процессе производства. Структура приложения позволяет добавлять новые модули без значительного переписывания имеющегося исходного кода, что будет использовано при желании заказчика расширить функционал.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вахрушева, А. А. Технологии позиционирования в режиме реального времени / А. А. Вахрушева. – Текст : электронный // Вестник СГУГиТ (Сибирского государственного университета геосистем и технологий). – 2017. – №1. – С. 170–177. – URL: http://vestnik.ssga.ru/wp-content/uploads/2017/04/Вестник-Том-22-№-1_rus.pdf (дата обращения: 11.04.2022).

2. Монгуш, А. В. Обзор технологий indoor-навигации / А. В. Монгуш, П. М. Кикин. – Текст : электронный // Интерэкспо Гео-Сибирь. – 2017. – №1. – С. 119–123. – URL: <https://geosib.sgugit.ru/wp-content/uploads/kongress/Sborniki/2017/Магистерская1.pdf> (дата обращения: 10.04.2022).

3. Marvelmind Indoor Navigation System Operating manual. – Текст. Изображение : электронные. // Marvelmind Robotics : [официальный сайт]. – URL: https://marvelmind.com/pics/marvelmind_navigation_system_manual.pdf (дата обращения: 22.02.2022).

4. Документация по Windows Forms. – Текст. Изображение : электронные // Microsoft : [официальный сайт]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/?view=netframeworkdesktop-4.8> (дата обращения: 25.04.2022).

5. Qt : [официальный сайт] / Qt. – URL: <https://www.qt.io> (дата обращения: 25.04.2022). – Текст. Изображение : электронные.

6. Документация по Windows Presentation Foundation. – Текст. Изображение : электронные // Microsoft : [официальный сайт]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/?view=netdesktop-6.0> (дата обращения: 25.02.2022)..

7. Документация по VB.NET. – Текст. Изображение : электронные // Microsoft : [официальный сайт]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/visual-basic/> (дата обращения: 26.04.2022).

8. Developer Survey Results 2018. – Текст. Изображение : электронные // Stack Overflow : [сайт]. – URL: <https://insights.stackoverflow.com/survey/2021/> (дата обращения: 26.04.2022).

9. Документация по C#. – Текст. Изображение : электронные // Microsoft : [официальный сайт]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 24.02.2022).

10. Обзор XAML (WPF .NET). – Текст. Изображение : электронные // Microsoft : [официальный сайт]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/xaml/?view=netdesktop-6.0> (дата обращения: 14.04.2022).

11. Приложения WPF с шаблоном проектирования модель-представление-модель представления. – Текст. Изображение : электронные // Microsoft : [официальный сайт]. – URL: <https://docs.microsoft.com/ru-ru/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern> (дата обращения: 10.03.2022).

12. User Datagram Protocol. – Текст : электронные // IETF : [сайт]. – URL: <https://datatracker.ietf.org/doc/html/rfc768> (дата обращения: 05.04.2022).

13. Как сериализовать и десериализовать (маршалирование и демаршалирование) JSON в .NET. – Текст. Изображение : электронные // Microsoft : [сайт]. – URL: <https://docs.microsoft.com/ru-ru/dotnet/standard/serialization/system-text-json-how-to?pivot=dotnet-6-0> (дата обращения: 12.04.2022).

14. Тестирование удобства пользования или Usability Testing. – Текст. Изображение : электронные // ПРОтестинг.ru : [сайт]. – <http://www.protesting.ru/testing/types/usability.html> (дата обращения: 15.05.2022).

15. Интеграционное тестирование (Integration Testing). – Текст.
Изображение : электронные // ПРОтестинг.ru : [сайт]. –
<http://www.protesting.ru/testing/levels/integration.html> (дата обращения:
15.05.2022).