Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Работа защищена

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель

Прозоров А.Н.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2025

**РАЗРАБОТКА системЫ контроля времени пребывания детей на детской площадке НА ПЛАТФОРМЕ WPF**

Отчет о лабораторной работе по курсу

«Технология программирования»

ЯГТУ 09.03.02 - 004 ЛР

|  |  |
| --- | --- |
|  | Отчет выполнил  студент группы ЗЦИС-26  К.В. Фатеева  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2025 |

2025

**Цель работы**

Разработать приложение для системы контроля времени пребывания детей на детской площадке на платформе WPF.

**Задание**

Разработать программу, предоставляющую пользователю систему контроля времени пребывания детей на детской площадке. Система должна обеспечивать следующие требования:

* формирование заявки "посещение", с указанием данных "заказчика" (родитель, ребенок), время начала посещения, время пребывания
* контроль выполнения сформированных заявок в реальном времени
* ведение истории посещения с возможностью формирования отчёта посещений за указанный период времени

**Ход выполнения задания**

При запуске приложения открывается рабочее окно.

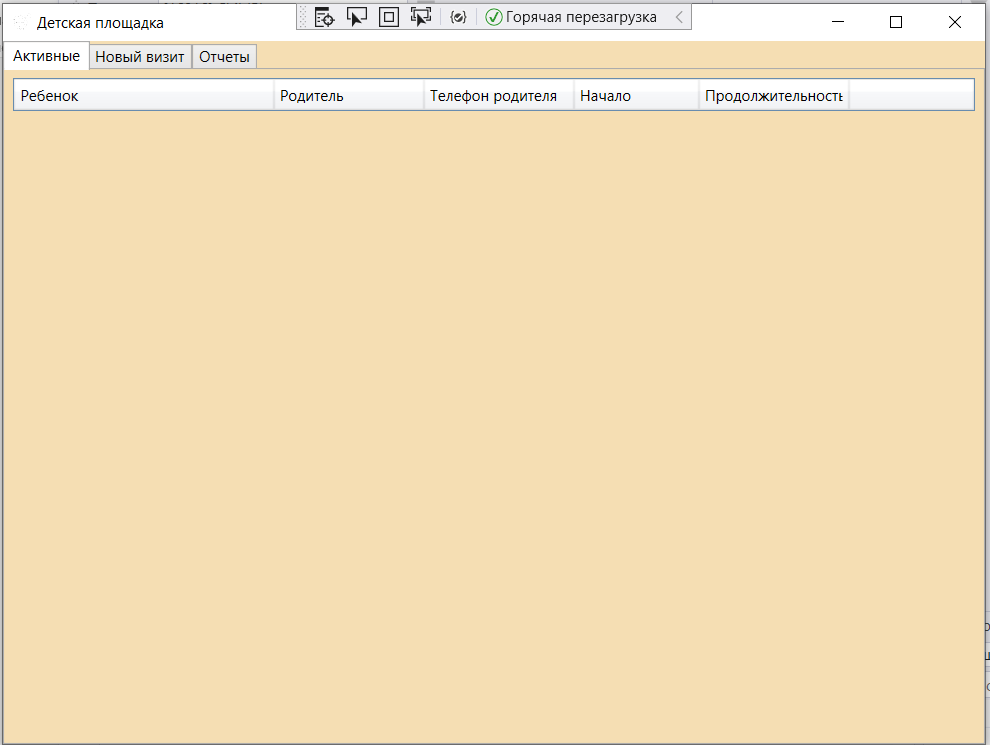


Рисунок 1 – Запуск приложения

Добавляем ребенка.

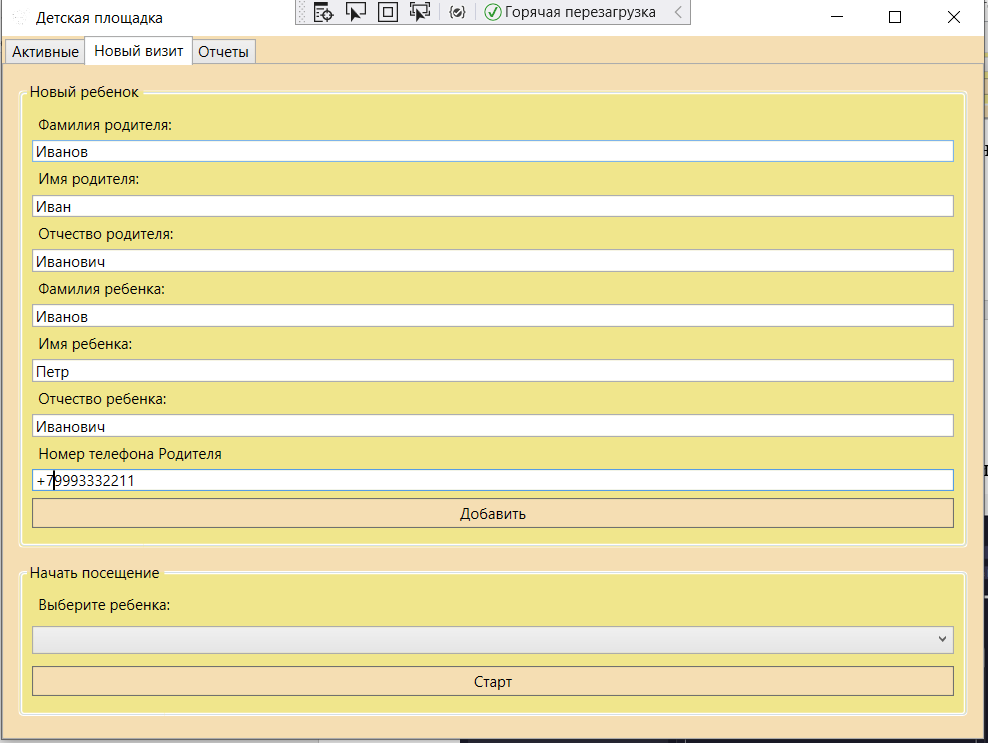


Рисунок 2 – Добавление ребенка

При добавлении ребенка, дает выбор ребенка из списка.

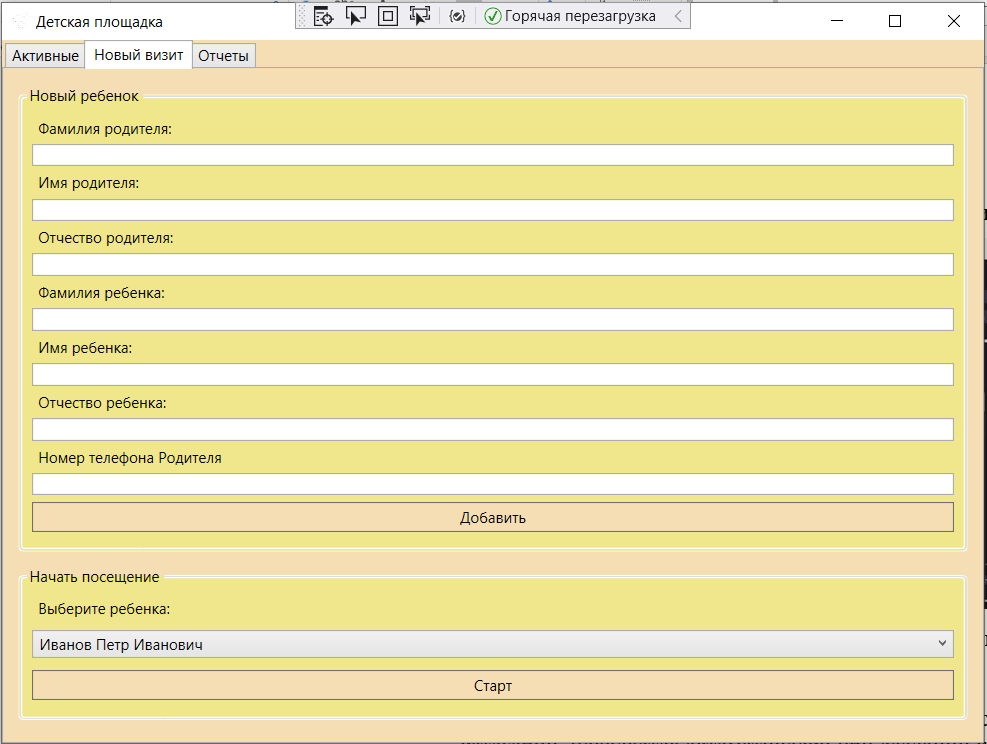


Рисунок 3 – Выбор ребенка из списка

Запуск таймер.

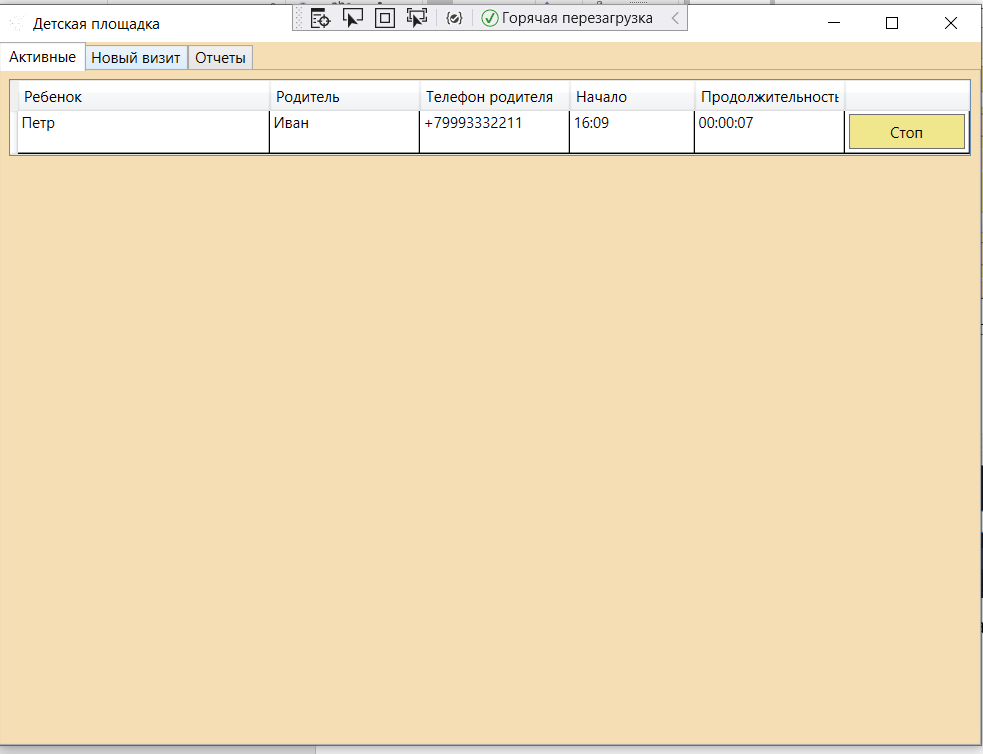


Рисунок 4 – Запуск таймера

Формирование отчета посещений.

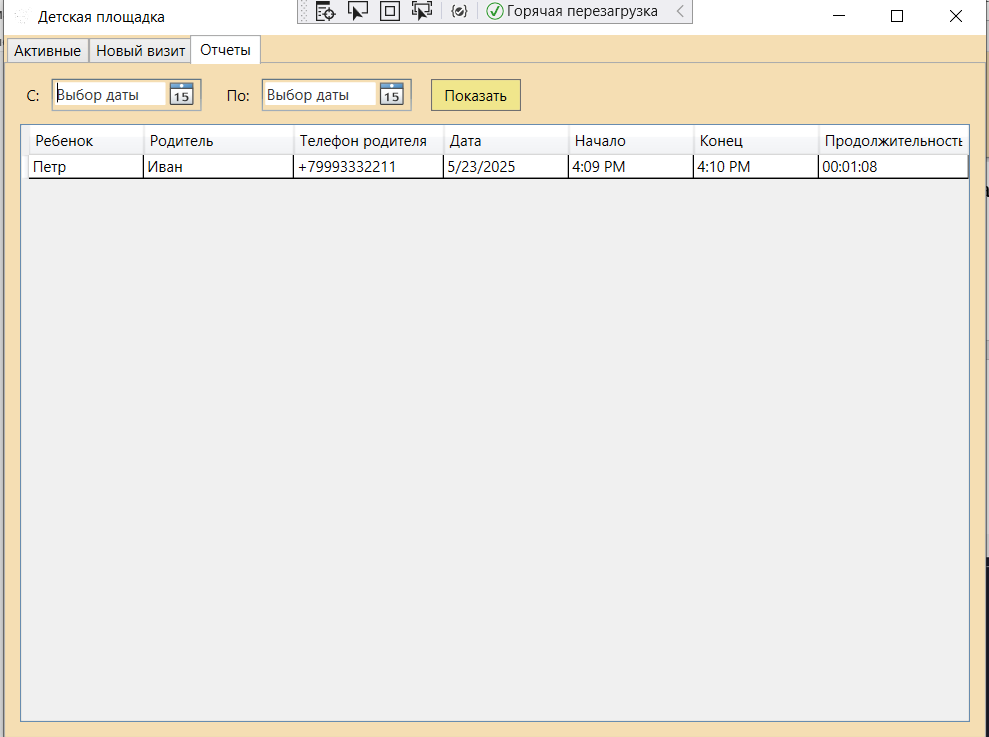


Рисунок 5- Отчет посещений

**Вывод**

Была разработана программа, предоставляющая пользователю систему контроля времени пребывания детей на детской площадке. Система соответствует следующим требованиям:

* формирование заявки "посещение", с указанием данных "заказчика" (родитель, ребенок), время начала посещения, время пребывания
* контроль выполнения сформированных заявок в реальном времени

ведение истории посещения с возможностью формирования отчёта посещений за указанный период времени

**Приложение А**

**Листинг приложения**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Data.Entity;

using System.Windows.Threading;

using System.Text.RegularExpressions;

namespace ChildPlaygroundTracker

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private Database db = new Database();

private DispatcherTimer timer = new DispatcherTimer();

public MainWindow()

{

InitializeComponent();

LoadData();

timer.Interval = TimeSpan.FromSeconds(1);

timer.Tick += UpdateDurations;

timer.Start();

}

private void LoadData()

{

db.Children

.Include(c => c.Parent)

.Load();

db.Visits

.Include(v => v.Child.Parent)

.Load();

ChildrenCombo.ItemsSource = db.Children.Local;

ActiveVisitsGrid.ItemsSource = db.Visits.Local.Where(v => v.EndTime == null);

ReportGrid.ItemsSource = db.Visits.Local.Where(v => v.EndTime != null);

}

private void UpdateDurations(object sender, EventArgs e)

{

ActiveVisitsGrid.Items.Refresh();

}

private void AddChild\_Click(object sender, RoutedEventArgs e)

{

// Проверка заполненности полей родителя

if (string.IsNullOrWhiteSpace(ParentLastName.Text))

{

MessageBox.Show("Введите фамилию родителя!");

return;

}

if (string.IsNullOrWhiteSpace(ParentFirstName.Text))

{

MessageBox.Show("Введите имя родителя!");

return;

}

if (string.IsNullOrWhiteSpace(ParentPatronymic.Text))

{

MessageBox.Show("Введите отчество родителя!");

return;

}

// Проверка заполненности полей ребенка

if (string.IsNullOrWhiteSpace(ChildLastName.Text))

{

MessageBox.Show("Введите фамилию ребенка!");

return;

}

if (string.IsNullOrWhiteSpace(ChildFirstName.Text))

{

MessageBox.Show("Введите имя ребенка!");

return;

}

if (string.IsNullOrWhiteSpace(ChildPatronymic.Text))

{

MessageBox.Show("Введите отчество ребенка!");

return;

}

// Проверка телефона

var phoneRegex = new Regex(@"^\+7\d{10}$");

if (!phoneRegex.IsMatch(ParentPhone.Text.Trim()))

{

MessageBox.Show("Некорректный формат телефона!\nПример: +71234567890");

return;

}

// Проверка на существующий номер

if (db.Parents.Any(p => p.Phone == ParentPhone.Text.Trim()))

{

MessageBox.Show("Этот номер уже зарегистрирован!");

return;

}

// Создание родителя и ребенка

var parent = new Parent

{

LastName = ParentLastName.Text.Trim(),

FirstName = ParentFirstName.Text.Trim(),

Patronymic = ParentPatronymic.Text.Trim(),

Phone = ParentPhone.Text.Trim()

};

db.Parents.Add(parent);

db.SaveChanges(); // Сохраняем родителя, чтобы получить его Id

var child = new Child

{

LastName = ChildLastName.Text.Trim(),

FirstName = ChildFirstName.Text.Trim(),

Patronymic = ChildPatronymic.Text.Trim(),

ParentId = parent.Id

};

db.Children.Add(child);

db.SaveChanges();

// Очистка полей

ParentLastName.Clear();

ParentFirstName.Clear();

ParentPatronymic.Clear();

ChildLastName.Clear();

ChildFirstName.Clear();

ChildPatronymic.Clear();

ParentPhone.Clear();

// Обновление данных и сообщение

LoadData();

MessageBox.Show("Ребенок и родитель успешно добавлены!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

private void StartVisit\_Click(object sender, RoutedEventArgs e)

{

if (ChildrenCombo.SelectedItem == null)

{

MessageBox.Show("Выберите ребенка!");

return;

}

var childId = ((Child)ChildrenCombo.SelectedItem).Id;

if (db.Visits.Any(v => v.ChildId == childId && v.EndTime == null))

{

MessageBox.Show("Этот ребенок уже на площадке!");

return;

}

var visit = new Visit

{

ChildId = childId,

StartTime = DateTime.Now

};

db.Visits.Add(visit);

db.SaveChanges();

// Сброс выбора и сообщение

ChildrenCombo.SelectedItem = null;

LoadData();

MessageBox.Show("Посещение начато!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

}

private void StopVisit\_Click(object sender, RoutedEventArgs e)

{

var visit = (Visit)((FrameworkElement)sender).DataContext;

visit.EndTime = DateTime.Now;

db.SaveChanges();

LoadData();

}

private void GenerateReport\_Click(object sender, RoutedEventArgs e)

{

var start = StartDate.SelectedDate ?? DateTime.MinValue;

var end = EndDate.SelectedDate ?? DateTime.MaxValue;

ReportGrid.ItemsSource = db.Visits.Local

.Where(v => v.StartTime >= start && v.EndTime <= end)

.ToList();

}

}

}