Министерство образования и науки Волгоградской области государственное бюджетное профессиональное  
образовательное учреждение   
«Волжский политехнический техникум»

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПРОЕКТА**

**«Создание программы «FoodOrder»»**

**(Приложения «FoodOrder»)**

Волжский 2025г.

СОДЕРЖАНИЕ

1. Общие сведения
   1. Полное наименование приложения и ее условное обозначение
   2. Плановые сроки начала и окончания работ

1.3.Порядок оформления и предъявления результатов работ

2. Назначение и цели создания приложения

2.1. Назначение приложения

2.2. Цели создания приложения

3. Характеристики объекта информатизации

4. Требования к системе

4.1. Требования к системе в целом

4.1.1. Описание цели и задач

4.2. Требования к структуре и функционированию системы

5. Стадии и этапы разработки

6. Программная реализация

6.1. Диаграмма прецедентов

6.2. Диаграмма классов

6.3. Диаграмма активностей

6.4. Диаграмма последовательностей

6.5. Диаграмма объектов

6.6. Диаграмма состояний

7. Листинг

7.1. Листинг 1. Program.cs

7.2. Листинг 2. ClientOrAdminForm.cs

7.3. Листинг 3. ClientForm.cs

7.4. Листинг 4. DishSelectionForm.cs

7.5. Листинг 5. AdminForm.cs

8. Инструкция пользователя

9. Отчет руководителя проекта о проделанной работе

9.1. Поставленная задача

9.2. Обязанности каждого участника команды

9.3. Выполнение работы команды (по дням)

**1. Общие сведения**   
Настоящее техническое задание по созданию мобильного приложения «Food Order» разработано ООО «ИСП - 9» в соответствии с государственным стандартом Российской Федерации «Информационная технология. Техническое задание на создание информационной системы».  
  
**1.1. Полное наименование приложения и ее условное обозначение**   
Полное наименование приложения: приложение для заказа еды «Food Order».   
Условное обозначение приложения: FoodOrder.   
Краткое наименование приложения, используемое в настоящем Техническом задании: Приложение «FoodOrder», Система, Приложение.  
  
**1.2. Плановые сроки начала и окончания работ**   
Начало работ: 28.01.2025   
Окончание работ: 11.02.2025   
  
**1.3. Порядок оформления и предъявления результатов работ**   
1. Оформление технического задания – подготавливается техническое задание для программистов, дизайнеров, аналитиков.   
Результат: готовое техническое задание, оформленный макет приложения, план продвижения приложения.   
2. ...   
10. Предоставление результатов работ заказчику – …   
Необходимо четко и структурировано описать, как и в каком виде будут оформляться результаты каждого этапа разработки, и как они будут предъявляться заказчику или заинтересованным сторонам. Этот пункт должен обеспечивать прозрачность и отслеживаемость процесса разработки.   
  
Коммерческое предложение Участника в обязательном порядке должно содержать план-график реализации проекта.   
В плане-графике реализации проекта должны быть указаны этапы реализации проекта, а также перечень работ по каждому этапу реализации проекта, предполагаемых Участником к выполнению. По всем работам необходимо указать длительность выполнения работ, количество нормо-дней, необходимых Участнику для выполнения работ, а также общую стоимость для каждой выполняемой работы.   
  
По завершению каждого из этапов работ Исполнитель сдает Заказчику соответствующие отчетные документы этапа, состав которых будет определен Договором между Заказчиком и Исполнителем по разработке и внедрению.

**2. Назначение и цели создания приложения**   
**2.1. Назначение приложения**   
Основным назначением Приложения «Food Order» является:   
- Обеспечение удобного способа заказа еды из ресторана.  
**2.2. Цели создания приложения**   
Главной целью создания Приложения «Food Order» является:   
- Популяризация онлайн-заказов (стимулирование пользователей заказывать еду через приложение, что может увеличить клиентскую базу ресторана)

**3. Характеристики объекта информатизации**

Краткие сведения об объекте автоматизации:

* Наименование объекта автоматизации: Программа «Food Order»
* Назначение: Программа предназначена для удобного заказа еды из ресторана. Она позволяет пользователю просматривать меню, выбирать блюда.
* Разработчик: Студенты
* Платформа: Персональный компьютер
* Пользователь: Домашние пользователи, любители еды, студенты, занятые профессионалы.
* Область применения: Ресторан, доставка еды, домашнее хозяйство.
* Основные функции:   
  - Просмотр меню ресторана в удобном формате   
  - Выбор и заказ блюд

- Просмотр данных о клиентах

**4. Требования к системе**   
**4.1. Требования к системе в целом**

**4.1.1. Описание цели и задач**

Цель: обеспечить удобный, быстрый и экономичный способ заказа еды из ресторанов на дом для пользователей, а также предоставить ресторанам платформу для расширения клиентской базы и увеличения продаж.

Задачи:

* **Расширение клиентской базы:**
  + Предоставить доступ к различным приложениям.
  + Увеличить узнаваемость бренда.
* **Увеличение продаж:**
  + Предоставить дополнительный канал продаж.
  + Увеличить средний чек за счет удобства оформления заказа.
* **Автоматизация процесса приема заказов:**
  + Оптимизировать процесс приема, обработки и подтверждения заказов.
  + Сократить время обработки заказов.
* **Управление меню:**
  + Предоставить возможность легкого добавления, изменения и удаления блюд из меню.
  + Управляйте ценами и наличием блюд.
* **Управление доставкой:**
  + Интегрироваться со службами доставки или предоставить возможность использовать собственные службы доставки.
  + Отслеживать курьеров и заказы в режиме реального времени.
* **Аналитика:**
  + Предоставить доступ к аналитической информации о заказах, популярных блюдах, клиентах и ​​т.д.
  + Помочь настроить меню и маркетинговые кампании.

Назначение системы:   
Обеспечить удобный и эффективный способ заказа еды из ресторана на дом

Надежность:   
Система должна быть стабильной, не допускать потерь данных, корректно выполнять все предусмотренные функции.   
  
Удобство использования:   
Система должна обладать интуитивно понятным интерфейсом, не требующим специальных навыков для работы.   
  
Производительность:   
Система должна быстро реагировать на действия пользователя, обеспечивая комфортную работу даже с большим количеством пользователей и заказов.   
  
Совместимость:

Система должна работать на большинстве компьютеров с операционной системой Windows.

**4.2. Требования к структуре и функционированию системы**

Программа содержит функционал регистрации пользователя. Пользователь должен ввести:

* номер телефона,
* адрес,
* номер дома.

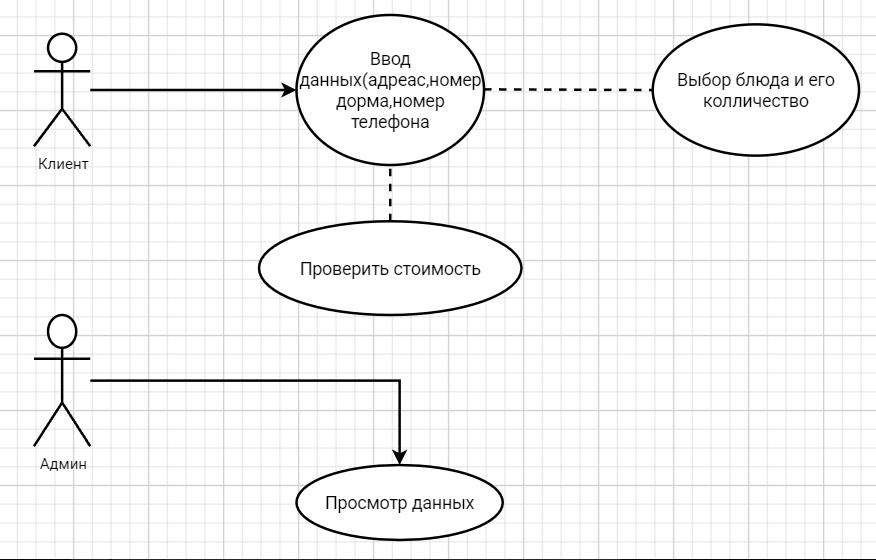
Далее клиент выбирает блюдо, подходящее по цене и сохраняет свой заказ.

**5. Стадии и этапы разработки**

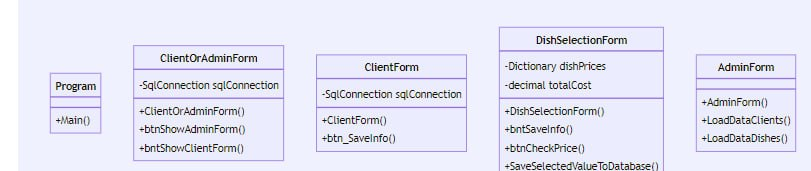
* Для начала мы изучили основы Windows Forms, чтобы понимать какие возможности у нас есть и как от них отталкиваться.
* Определились с основными функциями, которые будут у нас в приложении.
* Распределили работу на каждого человека.
* Начали работу над приложением.
* В конце протестировали каждую функцию приложения.

**6. Программная реализация**

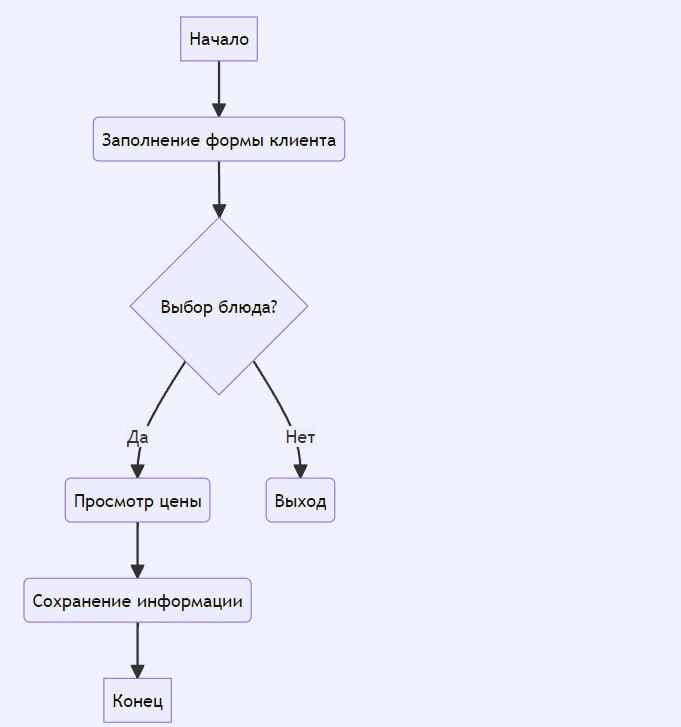
**6.1. Диаграмма прецедентов**

****

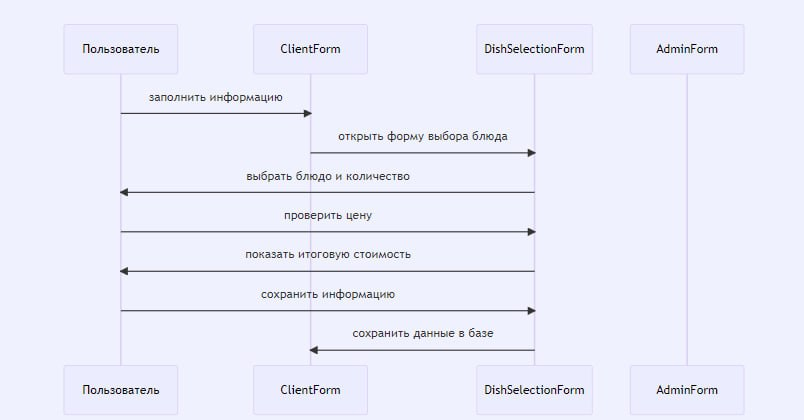
**6.2. Диаграмма классов**



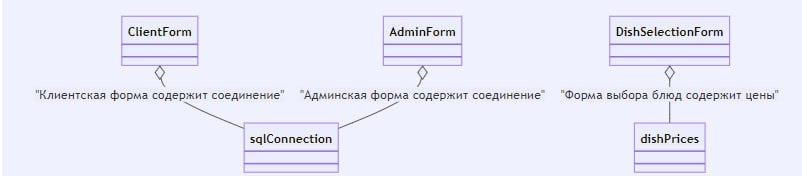
**6.3. Диаграмма активностей**



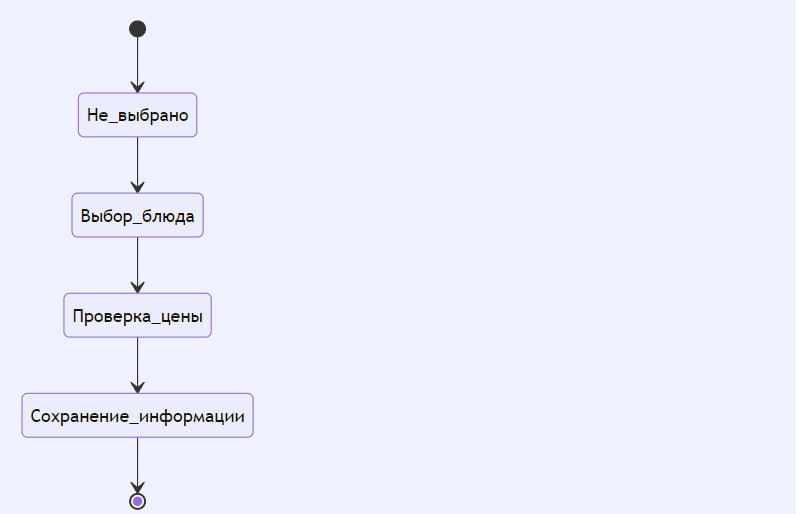
**6.4. Диаграмма последовательностей**



**6.5. Диаграмма объектов**



**6.6. Диаграмма состояний**



**7. Листинг**

**7.1. Листинг 1. Program.cs**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Threading.Tasks;  using System.Windows.Forms;    namespace WindowsFormsApp1  {      internal static class Program      {          /// <summary>          /// Главная точка входа для приложения.          /// </summary>          [STAThread]          static void Main()          {              Application.EnableVisualStyles();              Application.SetCompatibleTextRenderingDefault(false);              Application.Run(new ClientOrAdminForm());          }      }  } |

**7.2 Листинг 2. ClientOrAdminForm.cs**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.Common;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;    namespace WindowsFormsApp1  {        public partial class ClientOrAdminForm : Form      {          SqlConnection sqlConnection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\romas\\source\\repos\\ordering-food2\\WindowsFormsApp1\\ClientInto.mdf;Integrated Security=True");          SqlCommand command = new SqlCommand();          public ClientOrAdminForm()          {              InitializeComponent();              if (sqlConnection != null)              {                  sqlConnection.Open();                  MessageBox.Show("Connection true");              }              else              {                  MessageBox.Show("Connection false");              }            }            private void ClientOrAdminForm\_Load(object sender, EventArgs e)          {            }            private void btnShowAdminForm(object sender, EventArgs e)          {              AdminForm form = new AdminForm();              form.ShowDialog();          }            private void bntShowClientForm(object sender, EventArgs e)          {              ClientForm form = new ClientForm();              form.ShowDialog();          }      }  } |

**7.3. Листинг 3. ClientForm.cs**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;    namespace WindowsFormsApp1  {      public partial class ClientForm : Form      {            SqlConnection sqlConnection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\romas\\source\\repos\\ordering-food2\\WindowsFormsApp1\\ClientInto.mdf;Integrated Security=True");          SqlCommand command = new SqlCommand();          public ClientForm()          {              InitializeComponent();          }            private void ClientForm\_Load(object sender, EventArgs e)          {            }            private void btn\_SaveInfo(object sender, EventArgs e)          {              sqlConnection.Open();              string query = "INSERT INTO Clients (Address, House, PhoneNumber) VALUES (@address, @house, @phone)";                using (var cmd = new SqlCommand(query, sqlConnection))              {                  cmd.Parameters.AddWithValue("@address", addressTextBox.Text);                  cmd.Parameters.AddWithValue("@house", houseTextBox.Text);                  cmd.Parameters.AddWithValue("@phone", phoneTextBox.Text);                  cmd.ExecuteNonQuery();              }              DishSelectionForm dishSelectionForm = new DishSelectionForm();              dishSelectionForm.ShowDialog();          }      }  } |

**7.4. Листинг 4. DishSelectionForm.cs**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Data.SqlClient;  using System.Windows.Forms;    namespace WindowsFormsApp1  {      public partial class DishSelectionForm : Form      {          private string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\romas\\source\\repos\\ordering-food2\\WindowsFormsApp1\\ClientInto.mdf;Integrated Security=True";          private Dictionary<string, decimal> dishPrices = new Dictionary<string, decimal>();          private decimal totalCost;          public DishSelectionForm()          {              InitializeComponent();                dishPrices.Add("Карпача из лосося", 5.00m);              dishPrices.Add("Борщ", 7.50m);              dishPrices.Add("Салат", 6.00m);              dishPrices.Add("Чай", 2.00m);              dishPrices.Add("Кофе", 4.50m);              dishPrices.Add("Гренки", 5.30m);              dishPrices.Add("Паста", 12.00m);              dishPrices.Add("Пицца", 8.70m);              dishPrices.Add("Кока Кола", 3.45m);              dishPrices.Add("Пельмени", 8.25m);              dishPrices.Add("Бургер", 13.25m);              dishPrices.Add("Роллы", 10.10m);                foreach (string dish in dishPrices.Keys)              {                  comboBox1.Items.Add(dish);              }              comboBox2.Items.Add("1");              comboBox2.Items.Add("2");              comboBox2.Items.Add("3");              comboBox2.Items.Add("4");              comboBox2.Items.Add("5");              comboBox2.Items.Add("6");              comboBox2.Items.Add("7");              comboBox2.Items.Add("8");              comboBox2.Items.Add("9");              comboBox2.Items.Add("10");            }          private void bntSaveInfo(object sender, EventArgs e)          {              if (comboBox1.SelectedItem == null || comboBox2.SelectedItem == null)              {                  MessageBox.Show("Пожалуйста, выберите блюдо и количество.");                  return;              }                  string selectedName = comboBox1.SelectedItem.ToString();              string selectedQuantityString = comboBox2.SelectedItem.ToString();                if (!int.TryParse(selectedQuantityString, out int selectedQuantity))              {                  MessageBox.Show("Некорректное значение в ComboBox2. Пожалуйста, выберите число.");                  return;              }                SaveSelectedValueToDatabase(selectedName, selectedQuantity, (int)totalCost);          }          private void SaveSelectedValueToDatabase(string Name, int Quantity, int totalPrice)          {              string query = "INSERT INTO Dishes (Name, Quantity, TotalPrice) VALUES (@name, @quantity, @totalPrice)";              using (SqlConnection sqlConnection = new SqlConnection(connectionString))              {                  using (SqlCommand command = new SqlCommand(query, sqlConnection))                  {                      sqlConnection.Open();                      command.Parameters.AddWithValue("@name", Name);                      command.Parameters.AddWithValue("@quantity", Quantity);                      command.Parameters.AddWithValue("@TotalPrice", totalPrice);                        command.ExecuteNonQuery();                  }              }          }              private void btnCheckPrice(object sender, EventArgs e)          {              if (comboBox1.SelectedItem == null || comboBox2.SelectedItem == null)              {                  MessageBox.Show("Пожалуйста, выберите блюдо и количество.");                  return;              }              string selectedDish = comboBox1.SelectedItem.ToString();              string selectedQuantityString = comboBox2.SelectedItem.ToString();              decimal selectedPrice;              if (!int.TryParse(selectedQuantityString, out int selectedQuantity))              {                  MessageBox.Show("Некорректное значение в ComboBox2. Пожалуйста, выберите число.");                  return;              }              if (!dishPrices.ContainsKey(selectedDish))              {                  MessageBox.Show("Для выбранного блюда нет цены.");                  return;              }              else              {                  selectedPrice = dishPrices[selectedDish];              }              totalCost = selectedPrice \* selectedQuantity;              labelTotal.Text = $"Итоговая стоимость: {totalCost:C}";          }          }  } |

**7.5. Листинг 5. AdminForm.cs**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;    namespace WindowsFormsApp1  {      public partial class AdminForm : Form      {            SqlCommand command = new SqlCommand();          public AdminForm()          {              InitializeComponent();          }            private void AdminForm\_Load(object sender, EventArgs e)          {              LoadDataClients();              LoadDataDishes();          }          private void LoadDataClients()          {              try              {                    using (SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\romas\\source\\repos\\ordering-food2\\WindowsFormsApp1\\ClientInto.mdf;Integrated Security=True"))                  {                        string query = "SELECT \* FROM Clients";                        // Создаем объект адаптера данных                      using (SqlDataAdapter adapter = new SqlDataAdapter(query, connection))                      {                            DataTable dataTable = new DataTable();                            adapter.Fill(dataTable);                            dataGridView1.DataSource = dataTable;                      }                  }              }              catch (Exception ex)              {                  MessageBox.Show($"Ошибка при загрузке данных: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);              }          }          private void LoadDataDishes()          {              try              {                    using (SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\romas\\source\\repos\\ordering-food2\\WindowsFormsApp1\\ClientInto.mdf;Integrated Security=True"))                  {                        string query = "SELECT \* FROM Dishes";                        // Создаем объект адаптера данных                      using (SqlDataAdapter adapter = new SqlDataAdapter(query, connection))                      {                            DataTable dataTable = new DataTable();                            adapter.Fill(dataTable);                            dataGridView2.DataSource = dataTable;                      }                  }              }              catch (Exception ex)              {                  MessageBox.Show($"Ошибка при загрузке данных: {ex.Message}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);              }          }      }  } |

**8. Инструкция пользователя**

Выбрать должность:

* **Если клиент** 
  + Ввести данные (адрес, номер дома, номер телефона)
  + Выбрать блюдо из предлагаемого списка и количество порций
  + Узнать цену выбранных позиция
  + Сохранить информацию о выбранном блюде
* **Если администратор**
  + Просмотреть данные клиентов по заказу еды

**9. Отчет руководителя проекта о проделанной работе**

9.1. Поставленная задача

* Обеспечение удобного способа заказа еды из ресторана.

9.2. Обязанности каждого участника команды

* Афанасьев К.А. – Руководитель команды
  + Программирование
  + Назначение должностей
  + Визуализация проекта
* Метельков Д.Е.
  + Оформление технического задания
  + Оформление документации
* Семёнов Р.С.
  + Визуализация проекта
  + Оформление документации

9.3. Выполнение работы команды (по дням)

1 неделя (28.01.2025-04.02.2025) – Программирование проекта

2 неделя (05.02.2025 – 11.02.2025) – Составление документации

Афанасьев К.А.:

* Программирование: Задачи были выполнены в срок, решены основные технические проблемы, возникающие в процессе разработки.
* Назначение должностей: Провел встречи с командой, определил роли и распределил задачи согласно компетенциям участников.
* Визуализация проекта: Созданы и представлены макеты интерфейса для обсуждения с командой.

Метельков Д.Е.:

* Оформление технического задания: Составил и утвердил техническое задание, учитывающее все требования проекта.
* Оформление документации: Подготовлена вся необходимая документация, включая инструкции по использованию системы.

Семёнов Р.С.:

* Визуализация проекта: Разработал графические элементы и дизайн интерфейса, обеспечив удобство для пользователей.
* Оформление документации: Совместно с Метельковым создавал документацию, добавляя необходимые пояснения и описания.

Приложение разрабатывалось в течении 26 дней. За время работы конфликтных ситуаций не было.

Данный проект был выполнен в срок и приложение готово к использованию.

Руководитель проекта Афанасьев К.А. \_\_\_\_\_\_\_\_\_\_\_\_