

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский экономический университет имени Г.В. Плеханова»

Московский приборостроительный техникум

АТТЕСТАЦИОННЫЙ ЛИСТ
(характеристика профессиональной деятельности студента
во время производственной практики)

Суслин Александр Михайлович

Фамилия, Имя, Отчество

обучающийся на 4 курсе в группе П50-2-18 по специальности СПО 09.02.07 «Информационные системы и программирование» успешно прошел(ла) производственную практику ПП.02.01 «Осуществление интеграции программных модулей» по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей» в объеме 108 часов в период с 17.11.2021 по 21.12.2021; с 12.01.2022 по 08.02.2022.

1. Виды, объем и качество выполненных работ обучающимся во время практики

Виды работ	Объем выполненных работ (часов)
ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент – анализ требований к программному обеспечению; – определение характера взаимодействия компонентов программного обеспечения; – анализ проектной и технической документации на уровне взаимодействия компонент программного обеспечения; – точность и грамотность оформления технологической документации.	20
ПК 2.2 Выполнять интеграцию модулей в программное обеспечение – определение этапов разработки программного обеспечения; – демонстрация построения концептуальной, логической и физической моделей программного обеспечения и отдельных модулей; – выбор технологии разработки исходного модуля исходя из его назначения; – выбор методов разработки программных модулей; – выбор средств разработки программных модулей; – демонстрация навыков модификации программных модулей.	22
ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств – выявление ошибок в программных модулях; – определение возможности увеличения быстродействия программного продукта; – определение способов и принципов оптимизации; – выбор методов отладки программных модулей и программного продукта; – выбор специализированных средств для отладки программного продукта; – демонстрация навыков использования программных средств для отладки программного продукта.	22

<p>ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения</p> <ul style="list-style-type: none"> – разработка тестовых наборов и тестовых сценариев; – демонстрация устранения ошибок в программных модулях; – демонстрация использования методов тестирования программного обеспечения; – демонстрация навыков внесения изменения в программные модули для обеспечения качества программного обеспечения; – демонстрация навыков правильного использования инструментальных средств тестирования программных модулей. 	22
<p>ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования</p> <ul style="list-style-type: none"> – выбор методов обеспечения качества и надежности в процессе разработки сложных программных средств; – изложение основных принципов тестирования; – произведение инспектирования компонент программного продукта на предмет соответствия стандартам кодирования. 	22
Итого часов	108

2. Качество выполнения работ в соответствии с требованиями программы практики: _____ (_____)
оценка цифрой (оценка прописью)

3. База прохождения производственной практики

Предприятие (организация): ООО «План Б»

Руководитель по практической
подготовке от профильной
организации
МП

ФИО

Подпись

Руководитель по практической
подготовке от Московского
приборостроительного техникума
МП

А.А. Шимбирёв
ФИО

Подпись

Итоговая оценка по практике _____ (_____)
оценка цифрой (оценка прописью)

Дата: «08» февраля 2022 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ

Дневник производственной практики

ПП.02.01 Осуществление интеграции программных модулей

название практики по УП

ПМ.02 Осуществление интеграции программных модулей

наименование профессионального модуля, в рамках которого проводится практика

Студент 4 курса

**Специальности 09.02.07 «Информационные системы и
программирование»**

Квалификация: «Программист»

код, наименование специальности

Группа П50-2-18

Фамилия: Суслин

Имя: Александр

Отчество: Михайлович

Периоды проведения практики

- с «17» ноября 2021 года по «21» декабря 2021 года;
- с «12» января 2022 года по «08» февраля 2022 года.

СВЕДЕНИЯ О БАЗЕ ПРАКТИКИ

***Руководитель по практической подготовке
от техникума:***

Ф.И.О.: Шимбирёв Андрей Андреевич _____

Фамилия, Имя и Отчество (полностью)

***Руководитель по практической подготовке
от профильной организации:***

Ф.И.О.:

Фамилия, Имя и Отчество (полностью) руководителя по месту прохождения практики

Профильная организация-база практики:

ООО «План Б»

название организации

Адрес профильной организации с почтовым индексом:

115114, Российская Федерация, г.Москва, Дербеневская наб., 11

Период прохождения практики:

- с «17» ноября 2021 года по «21» декабря 2021 года;

- с «12» января 2022 года по «08» февраля 2022 года.

ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ

Практика имеет целью комплексное освоение студентами всех видов профессиональной деятельности по специальности 09.02.07 «**Информационные системы и программирование**» **Квалификация «Программист»**, формирование общих и профессиональных компетенций, а также приобретение необходимых умений и опыта практической работы по специальности.

Производственная практика включает в себя следующие этапы: практика по профилю специальности и преддипломная практика.

Практика по профилю специальности направлена на формирование у студентов общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках профессиональных модулей программы подготовки специалистов среднего звена (ППССЗ) по каждому из видов профессиональной деятельности, предусмотренных ФГОС СПО по специальности.

ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В результате освоения образовательной программы у выпускника должны быть сформированы общие и профессиональные компетенции.

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере.

В результате изучения профессионального модуля ПМ.02 **Осуществление интеграции программных модулей** студент должен обладать профессиональными компетенциями, соответствующими основным видам профессиональной деятельности:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

ИНСТРУКТАЖ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ

Провел инструктаж по охране труда, технике безопасности и промышленной санитарии

фамилия, имя, отчество проводившего инструктаж

Вид инструктажа: (заполнять только тот вид инструктажа, который проводится)

ИНСТРУКТАЖ ПО ПОЖАРНОЙ БЕЗОПАСНОСТИ

Подпись _____ Дата _____

ВВОДНЫЙ ИНСТРУКТАЖ

Подпись _____ Дата _____

ПЕРВИЧНЫЙ ИНСТРУКТАЖ НА РАБОЧЕМ МЕСТЕ

Подпись _____ Дата _____

Инструктаж получил(а)

Суслин Александр Михайлович

фамилия, имя, отчество получившего инструктаж

Подпись _____ Дата _____

Подпись _____ Дата _____

Подпись _____ Дата _____

СОДЕРЖАНИЕ ЗАДАНИЯ НА ПРАКТИКУ

Необходимо разработать базу данных с дальнейшим её администрированием для мобильного приложения по доставке газа, которая должна содержать процедуры добавления, редактирования и удаления для всех необходимых данных, а также создать веб-Арі для работы с созданной базой данных.

М.П.
(печать профильной
организации)

«17» ноября 2021 год

Руководитель по практической подготовке от
профильной организации: _____

фио

подпись

Ознакомлен студент: Суслин А.М.

подпись

ПАМЯТКА

для студента о практике в профильной организации

1. Прибыв на место практики, следует явиться в отдел кадров и оформить соответствующие документы.
2. Пройти инструктаж по технике безопасности, по пожарной безопасности, иные инструктажи, обязательные для прохождения в профильной организации, ознакомиться с правилами внутреннего трудового распорядка.
3. С момента зачисления студента в период практик в качестве практикантов на рабочие места, на них распространяются правила охраны труда и правила внутреннего распорядка, действующие в профильной организации-базе практики. Кроме того, на студента, зачисленных на рабочие должности, распространяется трудовое законодательство Российской Федерации, и он подлежит государственному социальному страхованию наравне со всеми работниками.
4. Студент при прохождении практики в профильной организации обязан:
 - полностью выполнять задания, предусмотренные программой данной практики;
 - соблюдать действующие в организациях правила внутреннего трудового распорядка;
 - изучать и строго соблюдать нормы охраны труда и правила пожарной безопасности;
 - бережно относиться к расходованию материалов, энергоресурсов.
5. В период прохождения практики студентом ведется **дневник практики**. Дневник заполняется ежедневно путем внесения записей краткого описания проделанной работы.
6. По результатам практики студентом составляется **отчет**, который утверждается в профильной организации, где проводилась практика. Отчет должен оформляться в последние дни пребывания студента на месте практики. Рекомендуемый объем отчета – от 5 до 10 страниц текста (с использованием рисунков, фотографий, схем). Основу содержания отчета должны составлять: самостоятельные личные наблюдения, критический анализ, оценка действующих технических средств, процессов и организации работ, а также личные рационализаторские предложения, выводы и заключения. Отчет составляется в соответствии с планом и программой практики.
7. По результатам практики руководителями по практической подготовке от профильной организации и от техникума формируется **аттестационный лист**, содержащий сведения об уровне освоения студентом профессиональных и общих компетенций, а также характеристика на студента.
8. Студент представляет визированные и заверенные печатью профильной организации в установленном порядке документы (отчет, дневник по практике и аттестационный лист) руководителю по практической подготовке от техникума в последний день практики.

Дата или период выполнения работ	Краткое содержание выполняемых работ	Подпись руководителя по практической подготовке от профильной организации
18.11.2021	Проведён инструктаж по охране труда при работе на персональном компьютере.	
25.11.2021	Получение задания на производственную практику. Ознакомление с предметной областью.	
29.11.2021	Написание технического задания.	
02.12.2021	Написание модели базы данных.	
09.12.2021	Создание таблиц для базы данных.	
13.12.2021	Создание процедур для базы данных.	
16.12.2021	Обновление структуры базы данных.	
20.12.2021	Разработка api. (Модели и контекст данных)	
13.01.2022	Разработка api. (Контроллеры и методы get)	
17.01.2022	Разработка api (методы post)	
20.01.2022	Разработка api (доработка и тестирование методов)	
31.01.2022	Тестирование и публикация api.	
03.02.2022	Написание отчетной документации.	
08.02.2022	Сдача и проверка документации в организации.	

Характеристика студента (с указанием степени его профессиональной подготовки, качества выполняемой работы, трудовой дисциплины и недостатков, если они имели место).

За время практики студентом была выполнена следующая работа (перечислить):

При прохождении практики студент

Суслин А.М.
фамилия, инициалы

зарекомендовал себя

Примечание: дать характеристику профессиональному опыту, умениям, навыкам и знаниям студента, степени его ответственности, дисциплинированности, исполнительности, адаптивности в коллективе, стремлению к саморазвитию, совершенствованию в профессиональной деятельности, уровню сформированности компетенций.

Оценка работы студента за практику _____ (_____)
оценка (прописью)

**Руководитель по практической подготовке
от профильной организации**

_____/_____
подпись фамилия, инициалы

МП
печать профильной организации

Заключение руководителя по практической подготовке от техникума и оценка
результатов практики

Итоговая оценка по практике

Руководитель по практической подготовке
от техникума

_____/_____
подпись фамилия, имя, отчество

Дневник практики проверен

_____/_____
подпись фамилия, имя, отчество

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский экономический университет имени Г.В. Плеханова»
МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ

ОТЧЕТ

по производственной практике

ПП.02.01 «Осуществление интеграции программных модулей»

Профессионального модуля ПМ.02 «Осуществление интеграции программных модулей»

Специальность 09.02.07 Информационные системы и программирование

Квалификация: Программист

Студент _____
подпись

Суслин Александр Михайлович
фамилия, имя, отчество

Группа П50-2-18

Руководитель по практической подготовке от профильной организации
ООО «План Б»
название профильной организации

подпись

фамилия, имя, отчество

М.П.

«08» февраля 2022 года

Руководитель по практической подготовке от
Московского приборостроительного техникума

подпись

Шимбирёв Андрей Андреевич
фамилия, имя, отчество

«08» февраля 2022 года

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	2
1. ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ	3
2. БАЗА ПРАКТИКИ.....	5
3. СОДЕРЖАНИЕ ПРАКТИКИ	7
ЗАКЛЮЧЕНИЕ	8
ИСПОЛЬЗУЕМЫЕ МАТЕРИАЛЫ.....	9
ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ	
ПРИЛОЖЕНИЕ Б. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	
ПРИЛОЖЕНИЕ Б1. ТЕХНИЧЕСКИЙ ПРОЕКТ	
ПРИЛОЖЕНИЕ В. СЦЕНАРИЙ ТЕСТОВЫХ ИСПЫТАНИЙ	
ПРИЛОЖЕНИЕ Г. РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ	
ПРИЛОЖЕНИЕ Д. ТЕКСТ ПРОГРАММЫ	
ПРИЛОЖЕНИЕ Е. СКРИПТ БАЗЫ ДАННЫХ	

ВВЕДЕНИЕ

Производственная практика по теме: ПП.02.01 «Осуществление интеграции программных модулей» проходила в организации ООО «План Б» в периоды с «17» ноября 2021 года по «21» декабря 2021 года; с «12» января 2022 года по «08» февраля 2022 года.

В ходе выполнения данной производственной практики необходимо было разработать Арі с дальнейшим её администрированием и интеграцией для мобильного приложения по доставке газа.

Основными должностными обязанностями во время практики были:

- Выбор программных решений для реализации задачи;
- Проектирование, разработка, администрирование и защита базы данных;
- Создание, тестирование и публикация арі;

1. ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ

Практика имеет целью комплексное освоение студентами всех видов профессиональной деятельности по специальности 09.02.07 «Информационные системы и программирование» Квалификация «Программист», формирование общих и профессиональных компетенций, а также приобретение необходимых умений и опыта практической работы по специальности.

Производственная практика включает в себя следующие этапы: практика по профилю специальности и преддипломная практика.

Практика по профилю специальности направлена на формирование у студентов общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках профессиональных модулей программы подготовки специалистов среднего звена (ППССЗ) по каждому из видов профессиональной деятельности, предусмотренных ФГОС СПО по специальности.

В результате освоения образовательной программы у выпускника должны быть сформированы общие и профессиональные компетенции.

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке.

ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере.

В результате изучения профессионального модуля ПМ.02 Осуществление интеграции

программных модулей студент должен обладать профессиональными компетенциями,

соответствующими основным видам профессиональной деятельности:

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и

технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных

программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного

обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет

соответствия стандартам кодирования.

2. БАЗА ПРАКТИКИ

Полное название организации: ООО «План Б».

Краткое название: План Б.

Руководитель от организации: _____

Адрес организации: 115114, г.Москва, Дербеневская наб., 11

В качестве средств вычислительной техники использовался персональный компьютер. Его характеристики представлены в таблице 1.

Таблица 1 – Программные средства

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Система управления базами данных	DataGrip 2021.2.4	Создание и управление базой данных
2	Текстовый редактор	Microsoft Word 2016	Разработка документации, формирование отчётных документов по шаблонам
3	Средство проектирования	ERwin Data Modeler 7.2.0.1644	Разработка схем для проектирования приложения
4	Среда разработки	Visual Studio 2019	Разработка web-api приложения

Таблица 2 – Технические средства

№	Тип оборудования	Наименование оборудования
1	2	3
Персональный компьютер7		
1	Размер экрана:	23.5"
2	Разрешение экрана:	1920x1080
3	Линейка процессора:	Intel Core i5 3450
4	Количество ядер процессора:	4
5	Оперативная память:	8 ГБ
6	Тип видеокарты:	дискретная
7	Видеокарта:	Nvidia Geforce GTX 1060 3Gb
8	Конфигурация накопителей:	HDD 1TB, SSD 256
9	Общий объем всех накопителей:	1280 ГБ
10	Операционная система	Windows 10 Ultimate

В качестве средств периферийной техники использовались устройства, приведённые в таблице 3.

Таблица 3 – Периферийные устройства

Наименование	Описание
1	2
Мышь	Оклик 715G
Клавиатура	MSI VIGOR GK50 Elite

3. СОДЕРЖАНИЕ ПРАКТИКИ

Тема: «Осуществление интеграции программных модулей».

Практика длилась в период с «17» ноября 2021 года по «21» декабря 2021 года; с «12» января 2022 года по «08» февраля 2022 года (понедельник и четверг) с 9:00 до 17:00.

В ходе выполнения данной производственной практики необходимо было разработать Api с дальнейшим её администрированием и интеграцией для мобильного приложения по доставке газа.

Перед началом разработки я ознакомился с предметной областью. После чего я приступил к поиску и анализу информации для выполнения поставленной задачи.

Была использована база данных MySQL.

Виды повседневных работ, выполнимых на практике представлены в таблице 4.

Таблица 4 – Виды работ

Дата	Виды работ выполнимые на практике.
1	2
18.11.2021	Проведён инструктаж по охране труда при работе на персональном компьютере.
25.11.2021	Получение задания на производственную практику. Ознакомление с предметной областью.
29.11.2021	Написание технического задания.
02.12.2021	Написание модели базы данных.
09.12.2021	Создание таблиц для базы данных.
13.12.2021	Создание процедур для базы данных.
16.12.2021	Обновление структуры базы данных.
20.12.2021	Разработка api. (Модели и контекст данных)
13.01.2022	Разработка api. (Контроллеры и методы get)
17.01.2022	Разработка api (методы post)
20.01.2022	Разработка api (доработка и тестирование методов)
31.01.2022	Тестирование и публикация api.
03.02.2022	Написание отчетной документации.
08.02.2022	Сдача и проверка документации в организации.

ЗАКЛЮЧЕНИЕ

В ходе производственной практики была разработана и интегрирована база данных и веб-Арі для мобильного приложения по доставке газа работы с созданной базой данных.

Благодаря производственной практике по теме ПП.02.01 «Осуществление интеграции программных модулей» улучшил свои знания в работе с программными обеспечениями DataGrip, Erwin Data Modeler и Visual Studio, повысил знания в написании, проектировании, разработки и защиты базы данных, разработки, тестировании и публикации веб-арі. По завершению практики были произведены работы по формированию и ведению проектной документации.

ИСПОЛЬЗУЕМЫЕ МАТЕРИАЛЫ

1. ГОСТ 7.80-2000 СИБИД. Библиографическая запись. Заголовок. Общие требования и правила составления.
2. ГОСТ Р 7.0.5-2008 БИБЛИОГРАФИЧЕСКАЯ ССЫЛКА. Общие требования и правила составления.
3. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
4. ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов.
5. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам.
6. Бэрон Шварц, Вадим Ткаченко, Петр Зайцев. MySQL по максимуму - Питер СПб, 2018 - 864 с.
7. Мартишин С., Симонов В., Храпченко М. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench. Методы и средства проектирования информационных систем и технологий. Инструментальные средства информационных систем. Учебное пособие: Форум, 2018 - 160 с.
8. Белл Ч., Киндал М., Талманн Л.. Обеспечение высокой доступности систем на основе MySQL: Русская редакция, 2012 - 624 с.
9. Васвани В. MySQL Использование и администрирование: ЛОРИ, 2011. - 368 с.

ПРИЛОЖЕНИЕ А. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

УТВЕРЖДЕН

РАЗРАБОТКА БАЗЫ ДАННЫХ И ВЕБ-АРІ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО ДОСТАВКЕ ГАЗА

Техническое задание

Руководитель разработки

_____ / _____

«08» февраля 2022 г.

Исполнитель

Студент _____ А.М. Суслин

«08» февраля 2022 г.

АННОТАЦИЯ

В данном программном документе приведено техническое задание на разработку базы данных и веб-арі для мобильного приложения по доставке газа.

В данном программном документе, в разделе «Введение» указано наименование, краткая характеристика области применения программы (программного изделия).

В разделе «Основания для разработки» указаны документы, на основании которых ведется разработка, наименование и условное обозначение темы разработки.

В данном программном документе, в разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программы (программного изделия).

Раздел «Требования к программе» содержит следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- специальные требования.

В данном программном документе, в разделе «Требования к программной документации» указаны предварительный состав программной документации и специальные требования к ней.

В разделе «Технико-экономические показатели» указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки.

В данном программном документе, в разделе «Стадии и этапы разработки» установлены необходимые стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ.....	5
1.1. Наименование программы	5
1.2. Краткая характеристика области применения программы.....	5
2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ	6
2.1. Основание для проведения разработки	6
2.2. Наименование и условное обозначение темы разработки.....	6
3. НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	7
3.1. Функциональное назначение программы.....	7
3.2. Эксплуатационное назначение программы.....	7
4. ТРЕБОВАНИЯ К ПРОГРАММЕ.....	8
4.1. Требования к функциональным характеристикам	8
4.1.1. Требования к составу выполняемых функций.....	8
4.1.2. Требования к организации входных данных.....	8
4.1.3. Требования к организации выходных данных	8
4.1.4. Требования к временным характеристикам	8
4.2. Требования к надежности	8
4.2.2. Время восстановления после отказа	9
4.2.3. Отказы из-за некорректных действий оператора	9
4.3. Условия эксплуатации	9
4.3.1. Климатические условия эксплуатации	9
4.3.2. Требования к видам обслуживания.....	9
4.3.3. Требования к численности и квалификации персонала.....	9
4.4. Требования к составу и параметрам технических средств.....	10
4.5. Требования к информационной и программной совместимости.....	10
4.5.1. Требования к информационным структурам и методам решения.....	10
4.5.2. Требования к исходным кодам и языкам программирования.....	10
4.5.3. Требования к программным средствам, используемым программой	10
4.5.4. Требования к защите информации и программ	10
4.6. Специальные требования	10
5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ	11
5.1. Предварительный состав программной документации	11

5.2. Специальные требования к программной документации	11
6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	12
6.1. Ориентировочная экономическая эффективность.....	12
6.2. Предполагаемая годовая потребность	12
6.3. Экономические преимущества разработки	12
7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ.....	13
7.1. Стадии разработки	13
7.2. Этапы разработки.....	14
7.3. Исполнители	14
8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ.....	15
8.1. Виды испытаний.....	15
8.2. Общие требования к приемке работы	15

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование - «Разработка базы данных и веб-арі для мобильного приложения по доставке газа».

1.2. Краткая характеристика области применения программы

Программа предназначена для клиентов, желающих заказать баллоны газа через мобильное приложение и водителей, принимающих и выполняющих заказы клиентов.

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

2.1. Основание для проведения разработки

Программа практики ПП 02.01 разработка модулей программного обеспечения для компьютерных систем.

2.2. Наименование и условное обозначение темы разработки

Наименование темы разработки - «Осуществление интеграции программных модулей».

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение программы

Функциональным назначением арі является работа с базой данных для мобильного приложения по доставке газа.

3.2. Эксплуатационное назначение программы

Конечными пользователями программы должны являться клиенты, желающие заказать баллоны газа или водителя, выполняющие заказы клиентов.

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональным характеристикам

4.1.1. Требования к составу выполняемых функций

Арі должна обеспечивать возможность выполнения перечисленных ниже функций:

- Регистрация и авторизация клиента;
- Регистрация и авторизация водителя;
- Генерация кода авторизации;
- Добавление новых товаров;
- Регистрация новых заказов;
- Получение списка свободных заказов;
- Получение списка заказов конкретного пользователя и их статусы;
- Изменение статуса заказа и точного времени доставки, по мере его

выполнения водителем.

4.1.2. Требования к организации входных данных

Требования к организации входных данных не предъявляются.

4.1.3. Требования к организации выходных данных

Требования к организации выходных данных не предъявляются.

4.1.4. Требования к временным характеристикам

Требования к временным характеристикам программы не предъявляются.

4.2. Требования к надежности

4.2.1. Требования к обеспечению надежного (устойчивого) функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- а) организацией бесперебойного питания технических средств;
- б) необходимым уровнем квалификации сотрудников профильных подразделений.

4.2.2. Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

4.2.3. Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий пользователя при взаимодействии с приложением

Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

4.3. Условия эксплуатации

4.3.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

4.3.2. Требования к видам обслуживания

См. Требования к обеспечению надежного (устойчивого) функционирования программы.

4.3.3. Требования к численности и квалификации персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 1 штатной единицы - конечный пользователь программы - оператор.

Конечный пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

4.4. Требования к составу и параметрам технических средств

В состав технических средств должно входить любое электронное устройство с доступом в интернет.

4.5. Требования к информационной и программной совместимости

4.5.1. Требования к информационным структурам и методам решения

Требования к информационным структурам на входе и выходе, а также к методам решения не предъявляются.

4.5.2. Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке с#. В качестве интегрированной среды разработки программы должна быть использована среда Visual Studio 2019 и Data Grip.

4.5.3. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены локализованной версией операционной системы Windows 10 и выше.

4.5.4. Требования к защите информации и программ

Требования к защите информации и программ не предъявляются.

4.6. Специальные требования

Специальные требования к программе не предъявляются.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1. Предварительный состав программной документации

Состав программной документации должен включать в себя:

1. Техническое задание;
2. Пояснительная записка;
3. Текст программы;
4. Технический проект;
5. Сценарий тестовых испытаний;
6. Результат тестовых испытаний.

5.2. Специальные требования к программной документации

Специальные требования к программной документации не предъявляются.

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

6.1. Ориентировочная экономическая эффективность

Ориентировочная экономическая эффективность не рассчитывается.

6.2. Предполагаемая годовая потребность

Предполагаемое число использования программы в год – круглосуточная работа программы на одном рабочем месте.

6.3. Экономические преимущества разработки

Экономические преимущества разработки не рассчитываются.

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

7.1. Стадии разработки

Таблица 1 - Стадии разработки

Стадии разработки	Этапы работ	Содержание работ
1. Техническое задание	Обоснование необходимости разработки программы	Постановка задачи Сбор исходных материалов Выбор и обоснование критериев эффективности и качества разрабатываемой программы. Обоснование необходимости проведения научно-исследовательских работ.
	Научно-исследовательские работы	Определение структуры входных и выходных данных. Предварительный выбор методов решения задач. Обоснование целесообразности применения ранее разработанных программ. Определение требований к техническим средствам. Обоснование принципиальной возможности решения поставленной задачи
	Разработка и утверждение технического задания	Определение требований к программе. Разработка технико-экономического обоснования разработки программы. Определение стадий, этапов и сроков разработки программы и документации на неё. Выбор языков программирования. Определение необходимости проведения научно-исследовательских работ на последующих стадиях. Согласование и утверждение технического задания.
2. Технический проект	Разработка технического проекта	Уточнение структуры входных и выходных данных. Разработка алгоритма решения задачи. Определение формы представления входных и выходных данных. Определение семантики и синтаксиса языка. Разработка структуры программы. Окончательное определение конфигурации технических средств.

Стадии разработки	Этапы работ	Содержание работ
	Утверждение технического проекта	Разработка плана мероприятий по разработке и внедрению программ. Разработка пояснительной записки. Согласование и утверждение технического проекта.
3. Рабочий проект	Разработка программы	Программирование и отладка программы.
	Разработка программной документации	Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.
	Испытания программы	Разработка, согласование и утверждение порядка и методики испытаний. Проведение предварительных государственных, межведомственных, приёмо-сдаточных и других видов испытаний. Корректировка программы и программной документации по результатам испытаний.
4. Внедрение	Подготовка и передача программы.	Подготовка и передача программы и программной документации для сопровождения и (или) изготовления. Оформление и утверждение акта о передаче программы на сопровождение и (или) изготовление. Передача программы в фонд алгоритмов и программ.

7.2. Этапы разработки

См. пункт 7.1.

7.3. Исполнители

Исполнитель: Суслин А.М.

Срок сдачи: 08.02.2022

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1. Виды испытаний

Испытания будут проводиться руководителем и исполнителем на основе программы и наличии документов: «Техническое задание», «Текст программы», «Пояснительная записка», «Руководство пользователя», «Технический проект», «Сценарий тестовых испытаний», «Результат тестовых испытаний», «Скрипт базы данных»

8.2. Общие требования к приемке работы

После проведения испытаний в полном объеме, на основании документов и программы, заполняется «Аттестационный лист».

ПРИЛОЖЕНИЕ Б. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

УТВЕРЖДЕН

РАЗРАБОТКА БАЗЫ ДАННЫХ И ВЕБ-АРІ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО
ДОСТАВКЕ ГАЗА

Пояснительная записка

АННОТАЦИЯ

В данном программном документе приведена пояснительная записка на разработку базы данных и веб-арі для мобильного приложения по доставке газа.

В данном программном документе, в разделе «Введение» указано наименование программы, условное обозначение темы разработки и основание проведения разработки.

В разделе «Назначение и область применения» указано назначение программы, краткая характеристика области применения программы.

Раздел «Технические характеристики» содержит следующие подразделы:

- постановка задачи;
- алгоритм и функционирование программы;
- входные и выходные данные;
- состав технических и программных средств.

СОДЕРЖАНИЕ

АННОТАЦИЯ	2
1. ВВЕДЕНИЕ.....	4
1.1. Наименование программы	4
1.2. Условное обозначение темы разработки	4
1.3. Основание проведения разработки	4
2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ	5
2.1. Функциональное назначение программы.....	5
2.2. Эксплуатационное назначение программы.....	5
3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	6
3.1. Постановка задачи	6
3.2. Алгоритм и функционирование программы	6
3.4. Состав технических и программных средств	6
4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	8
4.1. Ориентировочная экономическая эффективность	8
4.2. Предполагаемая годовая потребность	8
4.3. Экономические преимущества разработки	8

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование - «Разработка базы данных и веб-арі для мобильного приложения по доставке газа».

1.2. Условное обозначение темы разработки

Условное обозначение темы разработки - «Осуществление интеграции программных модулей»

1.3. Основание проведения разработки

Программа практики ПП 02.01 Осуществление интеграции программных модулей.

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Функциональное назначение программы

Функциональным назначением арі является работа с базой данных для мобильного приложения по доставке газа.

2.2. Эксплуатационное назначение программы

Конечными пользователями программы должны являться клиенты, желающие заказать баллоны газа или водителя, выполняющие заказы клиентов.

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи

Необходимо разработать базу данных с дальнейшим её администрированием для мобильного приложения по доставке газа, которая должна содержать процедуры добавления, редактирования и удаления для всех необходимых данных, а также создать веб-Арі для работы с созданной базой данных.

3.2. Алгоритм и функционирование программы

Арі должна обеспечивать возможность выполнения перечисленных ниже функций:

- Регистрация и авторизация клиента;
- Регистрация и авторизация водителя;
- Генерация кода авторизации;
- Добавление новых товаров;
- Регистрация новых заказов;
- Получение списка свободных заказов;
- Получение списка заказов конкретного пользователя и их статусы;
- Изменение статуса заказа и точного времени доставки, по мере его

выполнения водителем.

3.4. Состав технических и программных средств

Для разработки веб приложения использовались технические средства, которые представлены в Таблице 3.

Таблица 3- Технические средства

№	Тип оборудования	Наименование оборудования
1	2	3
Персональный компьютер		
1	Центральный процессор:	Intel Core i5 3450
2	Количество ядер	4
3	Видеоадаптер:	nVidia GeForce GTX 1060 (3 ГБ)
4	Системная память	8 ГБ (DDR3)
5	Твердотельный накопитель:	1 TB HDD 256 SSD
6	Разрешение экрана	1920x1080
7	Операционная система	Windows 10 Ultimate

Для разработки веб приложения использовались программные средства, представленные в Таблице 4.

Таблица 4 - Программные средства

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Инструментальное средство разработки программных решений	Visual Studio	Разработка веб приложения
2	Операционная система	Microsoft Windows 10	Организация взаимодействия программ и пользователя
3	Текстовый редактор	Microsoft Word 2016	Разработка документации, формирование отчётных документов по шаблонам
4	Инструментальное средство разработки базы данных	DataGrip 2022.2.4	Разработка базы данных

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Ориентировочная экономическая эффективность

Ориентировочная экономическая эффективность не рассчитывается.

4.2. Предполагаемая годовая потребность

Предполагаемая годовая потребность не рассчитывается.

4.3. Экономические преимущества разработки

Экономические преимущества разработки не рассчитываются.

ПРИЛОЖЕНИЕ Б1. ТЕХНИЧЕСКИЙ ПРОЕКТ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

УТВЕРЖДЕН

РАЗРАБОТКА БАЗЫ ДАННЫХ И ВЕБ-АРІ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО
ДОСТАВКЕ ГАЗА

Технический проект

АННОТАЦИЯ

В данном программном документе приведен технический проект на разработку базы данных и веб-арі для мобильного приложения по доставке газа.

В разделе «Входные данные» представлена таблица с входными данными и их ограничениями, типами и описанием.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
СОДЕРЖАНИЕ	3
1. ВХОДНЫЕ ДАННЫЕ.....	4

1. ВХОДНЫЕ ДАННЫЕ

Входными данными являются данные, в виде json-объекта, отправляемые на сервер к api.

```
1 {
2   ...."LastName": "Алексей",
3   ...."FirstName": "Иванов",
4   ...."Patronymic": "Михайлович",
5   ...."Phone": "88005553536"
6 }
```

Регистрация клиента или водителя.

```
1 {
2   ...."Address":{
3     ...."Address": "Ул. Петровская, д5",
4     ...."ApartmentNum":23,
5     ...."FrontDoorNum":1,
6     ...."FloorNum":3,
7     ...."Intercom": "78к9102"
8   },
9   ...."ClientPhone": "88005553535",
10  ...."DateDelivery": "02.11.2021",
11  ...."DesiredTimeFrom": "11:00",
12  ...."DesiredTimeTo": "19:00",
13  ...."Products":[
14    ....{
15      ...."Id":1,
16      ...."Quantity":1
17    }
18  ]
19 }
```

Регистрация нового заказа

```
1 {
2   ...."Phone": "88005553535"
3 }
```

Получение кода авторизации

ПРИЛОЖЕНИЕ В. СЦЕНАРИЙ ТЕСТОВЫХ ИСПЫТАНИЙ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

УТВЕРЖДЕН

РАЗРАБОТКА БАЗЫ ДАННЫХ И ВЕБ-АРІ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО
ДОСТАВКЕ ГАЗА

Сценарий тестовых испытаний

АННОТАЦИЯ

В разделе «Цель испытаний» указана цель проведения испытаний. В разделе «Требования к программе» указаны требования, подлежащие проверке во время испытаний и заданные в техническом задании на программу.

В разделе «Средства и порядок испытаний» указаны технические и программные средства, используемые во время испытаний, а также порядок проведения испытаний.

В разделе «Методы испытаний» приведено описание используемых методов испытаний

В разделе «Тестовые примеры» приведены таблицы с результатами тестовых испытаний.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
1. ОБЪЕКТ ИСПЫТАНИЙ	4
1.1. Наименование объекта	4
1.2. Область применения объекта.....	4
1.3. Обозначение испытываемой программы	4
2. ЦЕЛЬ ИСПЫТАНИЙ.....	5
3. ТРЕБОВАНИЕ К ПРОГРАММЕ	6
3.1. Схема тестирования	6
3.2. Требования, подлежащие проверке	6
4. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ	7
5. МЕТОДЫ ИСПЫТАНИЙ	9
6. ТЕСТОВЫЕ ПРИМЕРЫ.....	10
6.1. Генерация кода авторизации.....	10
6.2. Корректная регистрация клиента/водителя.....	10
6.3. Корректная авторизация клиента/водителя	11
6.4. Регистрация нового заказа.	11

1. ОБЪЕКТ ИСПЫТАНИЙ

1.1. Наименование объекта

Наименование – «Разработка базы данных и веб-арі для мобильного приложения по доставке газа».

1.2. Область применения объекта

Программа предназначена для клиентов, желающих заказать баллоны газа через мобильное приложение и водителей, принимающих и выполняющих заказы клиентов.

1.3. Обозначение испытываемой программы

Обозначение – «Разработка базы данных и веб-арі для мобильного приложения по доставке газа».

2. ЦЕЛЬ ИСПЫТАНИЙ

Целью проведения испытаний является проверка соответствия разработанной программы требованиям, изложенным в документах «Техническое задание».

3. ТРЕБОВАНИЕ К ПРОГРАММЕ

3.1. Схема тестирования

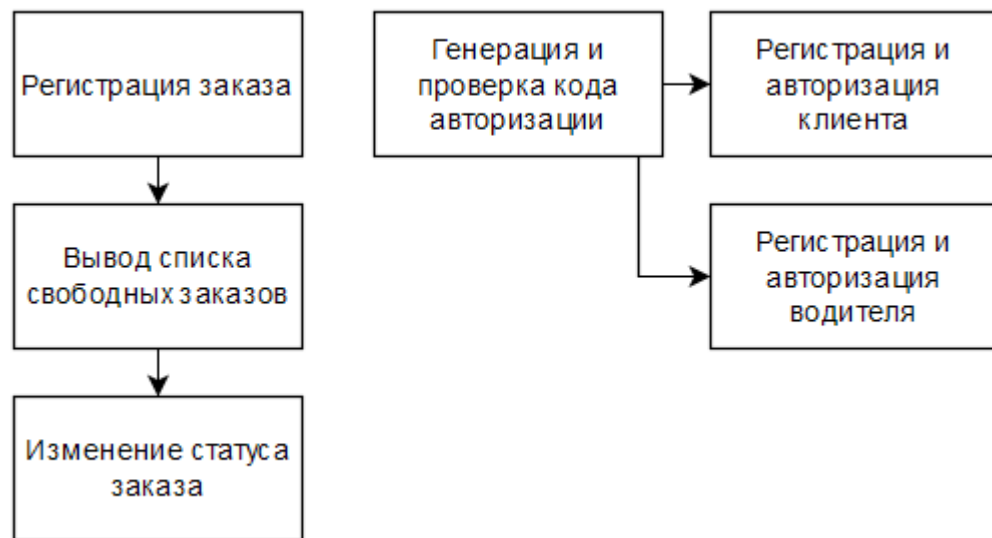


Рисунок 1 – Схема тестирования

3.2. Требования, подлежащие проверке

- генерация кода авторизации;
- корректная регистрация и авторизация клиента;
- корректная регистрация и авторизация водителя;
- корректная регистрация нового заказа;
- корректный вывод списка свободных заказов;
- корректное изменение статуса заказа.

4. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

Состав технических средств, используемых для проведения тестирования разработка веб приложения для взаимодействия с формами заполнения и базой данных представлен в Таблице №1.

Таблица 1 – Технические средства

№	Тип оборудования	Наименование оборудования
1	2	3
Стационарный компьютер		
1	Центральный процессор:	Intel Core i5 3450
2	Количество ядер	4
3	Видеоадаптер:	nVidia GeForce GTX 1060 (3 ГБ)
4	Системная память	8 ГБ (DDR3)
5	Твердотельный накопитель:	1ТБ HDD 256 SSD
6	Разрешение экрана	1920x1080
7	Операционная система	Windows 10 Home

Состав программных средств, используемых для проведения тестирования разработка веб приложения для взаимодействия с формами заполнения и базой данных представлен в Таблице 2.

Таблица 2 – Программные средства

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Инструментальное средство разработки программных решений	Visual Studio 2019	Разработка веб приложения
2	Операционная система	Microsoft Windows 10	Организация взаимодействия программ и пользователя
3	Текстовый редактор	Microsoft Word 2016	Разработка документации, формирование отчётных документов по шаблонам

4	Инструментальное средство разработки базы данных	DataGrip 2021.2.4	Разработка базы данных
---	--	-------------------	------------------------

Испытания должны проводиться поэтапно согласно разделу «Требования к программе» настоящего документа.

5. МЕТОДЫ ИСПЫТАНИЙ

1. По формальности тестирования.

Тестирование по тестам – тестирование по предварительно написанным тест кейсам.

6. ТЕСТОВЫЕ ПРИМЕРЫ

Все тестовые данные для проведения тестирования api для мобильного приложения по доставке газа.

6.1. Генерация кода авторизации.

Таблица 3 – Тест генерации кода авторизации.

№	Действие	Ожидаемый результат	Предпосылка
1	2	3	4
1	Отправить post-запрос с json-объектом с полем «phone» на адрес gopher-server.xyz:49153/api/client/getcode	Вернется 5-ти значный код авторизации	Предварительно установить и запустить postman
2	Отправить post-запрос с json-объектом с полем «phone» на адрес gopher-server.xyz:49153/api/client/getcode с телефоном, существующим в базе данных	Вернется статус код 404	Предварительно установить и запустить postman

6.2. Корректная регистрация клиента/водителя.

Таблица 4 – Тест регистрации клиента/водителя.

№	Действие	Ожидаемый результат	Предпосылка
1	2	3	4
1	Перейти по ссылке server.xyz:49153/api/	Откроется главная страница с отображением списка, состоящем из двух форм	Запустить браузер, внести в пустую базу данных 2 формы
2	Нажать на кнопку «Ответы»	Откроется страница «Список форм для получения ответов» с отображением списка, состоящем из двух форм	Зайти на главную страницу, внести в пустую базу данных 2 формы
3	Нажать на кнопку «Пройти форму»	Откроется страница «Прохождения формы» на которой корректно отобразятся 3 вопроса и поля	Зайти на главную страницу, внести в пустую базу данных 2 формы и составить к ней 3 вопроса

№	Действие	Ожидаемый результат	Предпосылка
1	2	3	4
		ввода к каждому вопросу	

6.3. Корректная авторизация клиента/водителя

Таблица 5 – Тест авторизации клиента/водителя.

№	Действие	Ожидаемый результат	Предпосылка
1	2	3	4
1	Нажать на кнопку «Создать»	Отобразиться страница «Создания формы» с полем ввода количества вопросов и кнопкой «Сгенерировать»	Зайти на главную страницу
2	Ввести цифру «2» и нажать кнопку «Сгенерировать»	Страница обновится и появится поле ввода для наименования формы, и два поля ввода для введения вопросов	Зайти на главную страницу и нажать кнопку «Создать»

6.4. Регистрация нового заказа.

Таблица 6 – Тест регистрации нового заказа.

№	Действие	Ожидаемый результат	Предпосылка
1	2	3	4
1	Нажать кнопку «Пройти форму»	Осуществится переход на главную страницу и в отображаемом списке форм будет отображаться форма с наименованием «Кто ты?», нажимая на кнопку «Пройти форму» осуществится переход на страницу заполнения формы, с вопросами «Как тебя зовут?» и «Сколько тебе лет?»	Зайти на главную страницу и нажать кнопку «Создать», ввести цифру «2» и нажать кнопку «Сгенерировать», далее заполнить поле наименования формы значением «Кто ты?», поле «Содержание вопроса №1» значением «Как тебя зовут?», поле «Содержание вопроса №2» значением «Сколько тебе лет?» и нажать на кнопку

№	Действие	Ожидаемый результат	Предпосылка
1	2	3	4
			сгенерировать
2	Ввести в первое поле значение «Иван», во второе поле значение «25» и нажать кнопку Сгенерировать	Осуществится переход на главную страницу, после по нажатию кнопки «Ответы» и дальнейшего нажатия на кнопку «Показать ответы» напротив наименования «Кто ты?», в странице отображения ответов мы увидим ответ, заполненный от анонимного пользователя, значениями «Иван» и «25»	Зайти на главную страницу и нажать кнопку «Создать», ввести цифру «2» и нажать кнопку «Сгенерировать», далее заполнить поле наименования формы значением «Кто ты?», поле «Содержание вопроса №1» значением «Как тебя зовут?», поле «Содержание вопроса №2» значением «Сколько тебе лет?» и нажать на кнопку сгенерировать

ПРИЛОЖЕНИЕ Г. РЕЗУЛЬТАТ ТЕСТОВЫХ ИСПЫТАНИЙ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

УТВЕРЖДЕН

РАЗРАБОТКА БАЗЫ ДАННЫХ И ВЕБ-АРІ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО
ДОСТАВКЕ ГАЗА

Результат тестовых испытаний

АННОТАЦИЯ

В данном программном документе приведены результаты тестовых испытаний на разработку базы данных и веб-арі для мобильного приложения по доставке газа.

В разделе «Объект испытаний» указаны наименования объекта, область применения объекта и обозначение испытываемой программы.

В разделе «Требования к программе» указаны требования, подлежащие проверке во время испытаний и заданные в техническом задании на программу.

В разделе «Результаты тестирования» приведены таблицы с результатами тестовых испытаний.

СОДЕРЖАНИЕ

АННОТАЦИЯ	2
1. ОБЪЕКТ ИСПЫТАНИЙ	4
1.1. Наименование объекта	4
1.2. Область применения объекта.....	4
2. ТРЕБОВАНИЯ К ПРОГРАММЕ.....	5
2.1. Требования, подлежащие проверке	5
3. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ	6
3.1. Генерация кода авторизации.....	6
3.2. Корректная регистрация клиента/водителя.....	7
3.3. Корректная авторизация клиента/водителя.	8
3.4. Регистрация нового заказа.	9

1. ОБЪЕКТ ИСПЫТАНИЙ

1.1. Наименование объекта

Наименование – «Разработка базы данных и веб-арі для мобильного приложения по доставке газа».

1.2. Область применения объекта

Программа предназначена для клиентов, желающих заказать баллоны газа через мобильное приложение и водителей, принимающих и выполняющих заказы клиентов.

2. ТРЕБОВАНИЯ К ПРОГРАММЕ

2.1. Требования, подлежащие проверке

- генерация кода авторизации;
- корректная регистрация и авторизация клиента;
- корректная регистрация и авторизация водителя;
- корректная регистрация нового наказа;
- корректный вывод списка свободных заказов;
- корректное изменение статуса заказа.

3. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Все тестовые данные для проведения тестирования api для мобильного приложения по доставке газа представлены в пунктах 3.1-3.4.

Тестирование производилось по тестовым данным, представленным в Приложении В. Сценарий тестовых испытаний.

3.1. Генерация кода авторизации.

Таблица 1 – Тест генерации кода авторизации.

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
1	Отправить post-запрос с json-объектом с полем «phone» на адрес gopher-server.xyz:49153/api/client/getcode	Вернется 5-ти значный код авторизации	Вернется 5-ти значный код авторизации	Пройден
2	Отправить post-запрос с json-объектом с полем «phone» на адрес gopher-server.xyz:49153/api/client/getcode с телефоном, существующим в базе данных	Вернется статус код 404	Вернется статус код 404	Пройден
3	Нажать на главной странице кнопку «Ответы»	Переход на страницу «Список форм для получения ответов».	Осуществился переход на страницу «Список форм для получения ответов».	Пройден
4	Нажать на главной странице кнопку «Создать»	Переход на страницу «Создания формы».	Осуществился переход на страницу «Создания формы».	Пройден
5	Нажать на кнопку «Пройти форму» у первого элемента в списке форм	Переход в окно «Прохождение формы».	Осуществился переход в окно «Прохождение формы».	Пройден

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
6	Нажать на элемент «Показать ответы»	Переход в окно «Просмотр ответов у выбранной формы»	Осуществился переход в окно «Просмотр ответов у выбранной формы»	Пройден

3.2. Корректная регистрация клиента/водителя.

Таблица 2 – Тест регистрации клиента/водителя.

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
1	Перейти по ссылке server.xyz:49153/api/	Откроется главная страница с отображением списка, состоящем из двух форм	Открывается главная страница с отображением списка, состоящем из двух форм	Пройден
2	Нажать на кнопку «Ответы»	Откроется страница «Список форм для получения ответов» с отображением списка, состоящем из двух форм	Открывается страница «Список форм для получения ответов» с отображением списка, состоящем из двух форм	Пройден
3	Нажать на кнопку «Пройти форму»	Откроется страница «Прохождения формы» на которой корректно отобразятся 3 вопроса и поля ввода к каждому вопросу	Открывается страница «Прохождения формы» на которой корректно отобразились 3 вопроса и поля ввода к каждому вопросу	Пройден
4	Нажать на кнопку «Показать ответы»	Откроется страница «Отображение ответов для	Откроется страница «Отображение ответов для	Пройден

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
		выбранной формы» на которой корректно отобразятся 3 вопроса и ответы к каждому из них	выбранной формы» на которой корректно отобразились 3 вопроса и ответы к каждому из них	

3.3. Корректная авторизация клиента/водителя.

Таблица 3 – Тест авторизации клиента/водителя.

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
1	Нажать на кнопку «Создать»	Отобразиться страница «Создания формы» с полем ввода количества вопросов и кнопкой «Сгенерировать »	Отобразилась страница «Создания формы» с полем ввода количества вопросов и кнопкой «Сгенерировать »	Пройден
2	Ввести цифру «2» и нажать кнопку «Сгенерировать»	Страница обновится и появится поле ввода для наименования формы, и два поля ввода для введения вопросов	Страница обновилась и появилось поле ввода для наименования формы, и два поля ввода для введения вопросов	Пройден
3	Заполнить поле наименования формы значением «Кто ты?», поле «Содержание вопроса №1» значением «Как тебя зовут?», поле «Содержание вопроса №2»	Осуществится переход на главную страницу и в отображаемом списке форм будет отображаться форма с наименованием	Осуществился переход на главную страницу и в отображаемом списке форм отображается форма с наименованием «Кто ты?»,	Пройден

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
	значением «Сколько тебе лет?» и нажать на кнопку сгенерировать	«Кто ты?», нажимая на кнопку «Пройти форму» осуществится переход на страницу заполнения формы, с вопросами «Как тебя зовут?» и «Сколько тебе лет?»	нажимая на кнопку «Пройти форму» осуществился переход на страницу заполнения формы, с вопросами «Как тебя зовут?» и «Сколько тебе лет?»	

3.4. Регистрация нового заказа.

Таблица 4 – Тест регистрации нового заказа.

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
1	Нажать кнопку «Пройти форму»	Осуществится переход на главную страницу и в отображаемом списке форм будет отображаться форма с наименование м «Кто ты?», нажимая на кнопку «Пройти форму» осуществится переход на страницу заполнения формы, с вопросами «Как тебя зовут?» и «Сколько тебе лет?»	Осуществился переход на главную страницу и в отображаемом списке форм отображается форма с наименованием «Кто ты?», нажимая на кнопку «Пройти форму» осуществился переход на страницу заполнения формы, с вопросами «Как тебя зовут?» и «Сколько тебе лет?»	Пройден
2	Ввести в первое	Осуществится	Осуществился	Пройден

№ п/п	Действие	Ожидаемый результат	Фактический результат	Статус теста (пройден / не пройден)
	поле значение «Иван», во второе поле значение «25» и нажать кнопку Сгенерировать	переход на главную страницу, после по нажатию кнопки «Ответы» и дальнейшего нажатия на кнопку «Показать ответы» напротив наименования «Кто ты?», в странице отображения ответов мы увидим ответ, заполненный от анонимного пользователя, значениями «Иван» и «25»	переход на главную страницу, после по нажатию кнопки «Ответы» и дальнейшего нажатия на кнопку «Показать ответы» напротив наименования «Кто ты?», в странице отображения ответов виден ответ, заполненный от анонимного пользователя, значениями «Иван» и «25»	

ПРИЛОЖЕНИЕ Д. ТЕКСТ ПРОГРАММЫ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

УТВЕРЖДЕН

РАЗРАБОТКА БАЗЫ ДАННЫХ И ВЕБ-АРІ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО
ДОСТАВКЕ ГАЗА

Текст программы

АННОТАЦИЯ

В данном программном документе приведен код программы веб-арі для мобильного приложения по доставке газа.

В разделе «Текст программы» указаны следующие пункты:

- Наименование программы;
- Область применения программы;
- Модули;
- Код программы.

СОДЕРЖАНИЕ

АННОТАЦИЯ	2
1. ТЕКСТ ПРОГРАММЫ	4
1.1. Наименование программы	4
1.2. Область применения программы	4
1.3. Модули	4
1.4. Код программы	4

1. ТЕКСТ ПРОГРАММЫ

1.1. Наименование программы

Наименование – «Разработка базы данных и веб-арі для мобильного приложения по доставке газа»

1.2. Область применения программы

Программа предназначена для клиентов, желающих заказать баллоны газа через мобильное приложение и водителей, принимающих и выполняющих заказы клиентов.

1.3. Модули

Таблица 1- Модули

Модуль	Описание	Количество строк кода	Размер (в Кбайтах)
1	2	3	4
ClientController.cs	Контроллер клиента	144	5.24
DriverController.cs	Контроллер водителя	144	5.24
OrderController.cs	Контроллер заказов	148	6.26
ProductController.cs	Контроллер товаров	52	1.4

1.4. Код программы

ClientController.cs

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using GasDeliveryApi.Models;
using GasDeliveryApi.Models.Views;

namespace GasDeliveryApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
```

```

public class ClientController : ControllerBase
{
    /// <summary>
    /// Список пар: номер телефона + код авторизации, для верификации телефона
    /// </summary>
    private static List<PhoneValid> phones = new List<PhoneValid>();

    /// <summary>
    /// Контекст базы данных
    /// </summary>
    private GasDeliveryDBContext _context;

    /// <summary>
    /// Инициализация контекста данных
    /// </summary>
    /// <param name="context">контекст данных</param>
    public ClientController(GasDeliveryDBContext context)
    {
        _context = context;
    }

    /// <summary>
    /// Метод, генерирующий код авторизации (в дальнейшем нужно код отправлять на номер телефона)
    /// </summary>
    /// <param name="phoneValid">Объект с номером телефона</param>
    /// <returns>Возвращает сгенерированный код, или http-код 404, если аккаунт с таким номером уже
    есть</returns>
    [HttpPost]
    [Route("[action]")]
    public ActionResult<int> GetCode(PhoneValid phoneValid)
    {
        if (_context.Users.Where(c => c.Phone == phoneValid.Phone).FirstOrDefault() != null)
        {
            return NotFound();
        }

        var result = phones.Where(p => p.Phone == phoneValid.Phone).FirstOrDefault();
        if (result != null)
        {
            phones.Remove(result);
        }
    }
}

```

```

    }

    int code = new Random().Next(100000, 999999);

    phones.Add(new PhoneValid()
    {
        Phone = phoneValid.Phone,
        Code = code
    });

    return new ObjectResult(code);
}

/// <summary>
/// Метод, проверяющий корректность ранее отправленного кода
/// </summary>
/// <param name="phoneValid">Объект с номером телефона и кодом авторизации</param>
/// <returns>Возвращает http-код 200 или 404, в зависимости от успешности проверки кода</returns>
[HttpPost]
[Route("[action]")]
public ActionResult CheckCode(PHONEValid phoneValid)
{
    var result = phones.Where(p => p.Phone == phoneValid.Phone && p.Code ==
phoneValid.Code).FirstOrDefault();
    if (result == null)
    {
        return NotFound();
    }

    phones.Remove(result);
    return Ok();
}

/// <summary>
/// Метод регистрации нового клиента
/// </summary>
/// <param name="clientView">Объект, содержащий все данные, необходимые для
регистрации</param>
/// <returns>Возвращает http-код 200, 404 или 500, в зависимости от выполнения
регистрации</returns>

```

```

[HttpPost]
[Route("[action]")]
public ActionResult RegClient(ClientView clientView)
{
    if (_context.Users.Where(c => c.Phone == clientView.Phone).FirstOrDefault() != null)
    {
        return NotFound();
    }

    try
    {
        PersonalInfo personalInfo = new PersonalInfo()
        {
            LastName = clientView.LastName,
            FirstName = clientView.FirstName,
            Patronymic = clientView.Patronymic
        };
        _context.PersonalInfos.Add(personalInfo);

        User user = new User()
        {
            Phone = clientView.Phone,
            RoleId = _context.UserRoles.Where(u => u.Name == "Клиент").Select(u => u.Id).FirstOrDefault()
        };
        _context.Users.Add(user);

        Client client = new Client()
        {
            PersonalInfo = personalInfo,
            User = user
        };
        _context.Clients.Add(client);

        _context.SaveChanges();

        return Ok();
    }
    catch
    {
        return StatusCode(500);
    }
}

```

```

    }
}

/// <summary>
/// Тестовый метод, выводящий список пар: номер телефона + код авторизации
/// </summary>
/// <returns>Список пар: номер телефона + код авторизации</returns>
[HttpGet]
[Route("[action]")]
public IEnumerable<PhoneValid> GetPhones()
{
    return phones;
}
}
}

```

DriverController.cs

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using GasDeliveryApi.Models;
using GasDeliveryApi.Models.Views;

namespace GasDeliveryApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class DriverController : ControllerBase
    {
        /// <summary>
        /// Список пар: номер телефона + код авторизации, для верификации телефона
        /// </summary>
        private static List<PhoneValid> phones = new List<PhoneValid>();

        /// <summary>
        /// Контекст базы данных
        /// </summary>

```

```

private GasDeliveryDBContext _context;

/// <summary>
/// Инициализация контекста данных
/// </summary>
/// <param name="context">контекст данных</param>
public DriverController(GasDeliveryDBContext context)
{
    _context = context;
}

/// <summary>
/// Метод, генерирующий код авторизации (в дальнейшем нужно код отправлять на номер телефона)
/// </summary>
/// <param name="phoneValid">Объект с номером телефона</param>
/// <returns>Возвращает сгенерированный код, или http-код 404, если аккаунт с таким номером уже
есть</returns>
[HttpPost]
[Route("[action]")]
public ActionResult<int> GetCode(PhoneValid phoneValid)
{
    if (_context.Users.Where(u => u.Phone == phoneValid.Phone).FirstOrDefault() != null)
    {
        return NotFound();
    }

    var result = phones.Where(p => p.Phone == phoneValid.Phone).FirstOrDefault();
    if (result != null)
    {
        phones.Remove(result);
    }

    int code = new Random().Next(100000, 999999);

    phones.Add(new PhoneValid()
    {
        Phone = phoneValid.Phone,
        Code = code
    });
}

```

```

        return new ObjectResult(code);
    }

    /// <summary>
    /// Метод, проверяющий корректность ранее отправленного кода
    /// </summary>
    /// <param name="phoneValid">Объект с номером телефона и кодом авторизации</param>
    /// <returns>Возвращает http-код 200 или 404, в зависимости от успешности проверки кода</returns>
    [HttpPost]
    [Route("[action]")]
    public ActionResult CheckCode(PhoneValid phoneValid)
    {
        var result = phones.Where(p => p.Phone == phoneValid.Phone && p.Code ==
phoneValid.Code).FirstOrDefault();
        if (result == null)
        {
            return NotFound();
        }

        phones.Remove(result);
        return Ok();
    }

    /// <summary>
    /// Метод регистрации нового водителя
    /// </summary>
    /// <param name="clientView">Объект, содержащий все данные, необходимые для
регистрации</param>
    /// <returns>Возвращает http-код 200, 404 или 500, в зависимости от выполнения
регистрации</returns>
    [HttpPost]
    [Route("[action]")]
    public ActionResult RegDriver(DriverView driverView)
    {
        if (_context.Users.Where(c => c.Phone == driverView.Phone).FirstOrDefault() != null)
        {
            return NotFound();
        }

        try

```



```

    {
        PersonalInfo personalInfo = new PersonalInfo()
        {
            LastName = driverView.LastName,
            FirstName = driverView.FirstName,
            Patronymic = driverView.Patronymic
        };
        _context.PersonalInfos.Add(personalInfo);

        User user = new User()
        {
            Phone = driverView.Phone,
            RoleId = _context.UserRoles.Where(u => u.Name == "Водитель").Select(u =>
u.Id).FirstOrDefault()
        };
        _context.Users.Add(user);

        Driver driver = new Driver()
        {
            PersonalInfo = personalInfo,
            User = user
        };
        _context.Drivers.Add(driver);

        _context.SaveChanges();

        return Ok();
    }
    catch
    {
        return StatusCode(500);
    }
}

/// <summary>
/// Тестовый метод, выводящий список пар: номер телефона + код авторизации
/// </summary>
/// <returns>Список пар: номер телефона + код авторизации</returns>
[HttpGet]
[Route("[action]")]

```

```

        public IEnumerable<PhoneValid> GetPhones()
        {
            return phones;
        }
    }
}

```

OrderController.cs

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using GasDeliveryApi.Models;
using GasDeliveryApi.Models.Views;

namespace GasDeliveryApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class OrderController : ControllerBase
    {
        /// <summary>
        /// Контекст базы данных
        /// </summary>
        private GasDeliveryDBContext _context;

        /// <summary>
        /// Инициализация контекста данных
        /// </summary>
        /// <param name="context">контекст данных</param>
        public OrderController(GasDeliveryDBContext context)
        {
            _context = context;
        }

        /// <summary>
        /// Регистрация нового заказа
        /// </summary>

```

```

    /// <param name="orderView">Объект, содержащий все данные, необходимые для
регистрации</param>
    /// <returns></returns>
    [HttpPost]
    [Route("[action]")]
    public ActionResult RegOrder(OrderView orderView)
    {
        try
        {
            var address = _context.AddressDeliveries.Where(a => a.Address == orderView.Address.Address &&
                a.ApartmentNum == orderView.Address.ApartmentNum &&
                a.FloorNum == orderView.Address.FloorNum &&
                a.FrontDoorNum == orderView.Address.FrontDoorNum).FirstOrDefault();

            if (address == null)
            {
                _context.AddressDeliveries.Add(orderView.Address);
            }

            var client = _context.Clients.Where(c => c.User.Phone == orderView.ClientPhone).FirstOrDefault();
            if (client == null)
            {
                return StatusCode(500);
            }

            Ordered order = new Ordered()
            {
                Address = orderView.Address,
                DateDelivery = DateTime.Parse(orderView.DateDelivery),
                DesiredTimeFrom = TimeSpan.Parse(orderView.DesiredTimeFrom),
                DesiredTimeTo = TimeSpan.Parse(orderView.DesiredTimeTo),
                Sum = 0.0,
                Client = client,
                Status = _context.OrderStatuses.Where(s => s.Name == "Обрабатывается").FirstOrDefault()
            };
            _context.Ordereds.Add(order);

            double allSumm = 0.0;
            foreach (var i in orderView.Products)
            {
                OrderCompo orderCompos = new OrderCompo()

```

```

        {
            Order = order,
            ProductId = i.Id,
            Quantity = i.Quantity,
            Sum = _context.Products.Where(p => p.Id == i.Id).Select(p => p.Price).FirstOrDefault() *
i.Quantity
        };
        _context.OrderCompos.Add(orderCompos);
        allSumm += orderCompos.Sum;
    }
    order.Sum = allSumm;

    _context.SaveChanges();
    return Ok();
}
catch
{
    return StatusCode(500);
}
}

[HttpGet]
[Route("[action]")]
public ActionResult<OrderView> GetFreeOrders()
{
    var result = (from order in _context.Ordereds
        where order.Status == _context.OrderStatuses.Where(s => s.Name ==
"Обрабатывается").FirstOrDefault()
        select new OrderView()
        {
            Id = order.Id,
            Address = order.Address,
            ClientLastName = order.Client.PersonalInfo.LastName,
            ClientFirstName = order.Client.PersonalInfo.FirstName,
            ClientPhone = order.Client.User.Phone,
            DateDelivery = order.DateDelivery.ToString("dd.MM.yyyy"),
            DesiredTimeFrom = order.DesiredTimeFrom.ToString(),
            DesiredTimeTo = order.DesiredTimeTo.ToString(),
            Summ = order.Sum,
            OrderStatus = order.Status.Name,

```

```

        Products = (from orderCompos in _context.OrderCompos
                     where orderCompos.OrderId == order.Id
                     select new OrderProductView()
                     {
                         Id = orderCompos.Product.Id,
                         Name = orderCompos.Product.Name,
                         Quantity = orderCompos.Quantity,
                         Summ = orderCompos.Sum
                     }).ToList()
    }).ToList();
    return new ObjectResult(result);
}

[HttpPost]
[Route("[action]")]
public ActionResult<Ordered> SetDriver(DriverOrderView driverOrder)
{
    var orderDb = (from order in _context.Ordereds
                   where order.Id == driverOrder.OrderId
                   select order).FirstOrDefault();

    var driverDb = (from driver in _context.Drivers
                   where driver.User.Phone == driverOrder.DriverPhone
                   select driver).FirstOrDefault();

    orderDb.DriverId = driverDb.Id;
    orderDb.ExactTime = TimeSpan.Parse(driverOrder.ExactTime);
    orderDb.StatusId = _context.OrderStatuses.Where(s => s.Name == "Принят в работу").Select(s =>
s.Id).FirstOrDefault();
    _context.Ordereds.Update(orderDb);
    _context.SaveChanges();

    var result = (from order in _context.Ordereds
                  where order.Id == driverOrder.OrderId
                  select order).FirstOrDefault();

    return new ObjectResult(result);
}
}
}

```

ProductController.cs

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using GasDeliveryApi.Models;
using GasDeliveryApi.Models.Views;

namespace GasDeliveryApi.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class ProductController : ControllerBase
    {
        /// <summary>
        /// Контекст базы данных
        /// </summary>
        private GasDeliveryDbContext _context;

        /// <summary>
        /// Инициализация контекста данных
        /// </summary>
        /// <param name="context">контекст данных</param>
        public ProductController(GasDeliveryDbContext context)
        {
            _context = context;
        }

        [HttpPost]
        [Route("[action]")]
        public ActionResult AddProduct(Product product)
        {
            try
            {
                Product newProduct = new Product()
                {
                    Name = product.Name,
                    Price = product.Price,
```

```
        Description = product.Description
    };
    _context.Products.Add(newProduct);
    _context.SaveChanges();
    return Ok();
}
catch
{
    return StatusCode(500);
}
}
}
```

ПРИЛОЖЕНИЕ Е. СКРИПТ БАЗЫ ДАННЫХ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»
Московский приборостроительный техникум

УТВЕРЖДЕН

РАЗРАБОТКА БАЗЫ ДАННЫХ И ВЕБ-АРІ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО
ДОСТАВКЕ ГАЗА
Скрипт базы данных

АННОТАЦИЯ

В данном программном документе приведен скрипт базы данных для веб-арі для мобильного приложения по доставке газа.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
1. СКРИПТ БАЗЫ ДАННЫХ.....	4
1.1.1. Наименование скрипта	4
1.2. Область применения скрипта.....	4
1.3. Скрипт	4

1. СКРИПТ БАЗЫ ДАННЫХ

1.1. Наименование скрипта

Наименование – «GasDelivery.sql».

1.2. Область применения скрипта

Скрипт предназначен для базы данных веб-аpи для мобильного приложения по доставке газа.

1.3. Скрипт

```
create table AddressDelivery
(
    id          int auto_increment
        primary key,
    address     varchar(100) charset utf8 not null,
    apartmentNum int          null,
    frontDoorNum int          null,
    floorNum    int           null,
    intercom     varchar(20) charset utf8 null
);
```

```
create table OrderStatus
(
    id int auto_increment
        primary key,
    name varchar(40) charset utf8 not null,
    constraint name
        unique (name)
);
```

```
create table PersonalInfo
(
    id          int auto_increment
        primary key,
    lastName    varchar(40) charset utf8 not null,
    firstName   varchar(40) charset utf8 not null,
    patronymic  varchar(40) charset utf8 null
);
```

```
create table Product
(
    id          int auto_increment
        primary key,
    name        varchar(100) charset utf8 not null,
    description  varchar(500) charset utf8 null,
    price       double          not null
);
```

```

create table UserRole
(
    id int auto_increment
        primary key,
    name varchar(40) charset utf8 not null,
    constraint name
        unique (name)
);

create table User
(
    id int auto_increment
        primary key,
    phone varchar(20) not null,
    roleId int not null,
    constraint phone
        unique (phone),
    constraint User_ibfk_1
        foreign key (roleId) references UserRole (id)
);

create table Client
(
    id int auto_increment
        primary key,
    personalInfoId int not null,
    userId int not null,
    constraint Client_ibfk_1
        foreign key (personalInfoId) references PersonalInfo (id),
    constraint Client_ibfk_2
        foreign key (userId) references User (id)
);

create index personalInfoId
    on Client (personalInfoId);

create index userId
    on Client (userId);

create table Driver
(
    id int auto_increment
        primary key,
    personalInfoId int not null,
    userId int not null,
    constraint Driver_ibfk_1
        foreign key (personalInfoId) references PersonalInfo (id),
    constraint Driver_ibfk_2
        foreign key (userId) references User (id)
);

create index personalInfoId

```

```

on Driver (personalInfoId);

create index userId
on Driver (userId);

create table Ordered
(
    id          int auto_increment
        primary key,
    dateDelivery date not null,
    desiredTimeFrom time null,
    desiredTimeTo time null,
    exactTime    time null,
    sum          double not null,
    addressId    int not null,
    driverId     int null,
    clientId     int not null,
    statusId     int not null,
    constraint Ordered_ibfk_1
        foreign key (addressId) references AddressDelivery (id),
    constraint Ordered_ibfk_2
        foreign key (driverId) references Driver (id),
    constraint Ordered_ibfk_3
        foreign key (clientId) references Client (id),
    constraint Ordered_ibfk_4
        foreign key (statusId) references OrderStatus (id)
);

create table OrderCompos
(
    id      int auto_increment
        primary key,
    quantity int not null,
    sum     double not null,
    orderId int not null,
    productId int not null,
    constraint OrderCompos_ibfk_1
        foreign key (orderId) references Ordered (id),
    constraint OrderCompos_ibfk_2
        foreign key (productId) references Product (id)
);

```