Министерство образования, науки и молодежной политики Нижегородской области

Государственное бюджетное профессиональное образовательное учреждение "Нижегородский Губернский колледж"

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

**ПМ.02.** **ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ**

Специальность **09.02.07 Информационные системы и программирование**

|  |  |
| --- | --- |
| Выполнил |  |
| Студент | В.В. Батракова |
| подпись |  |
| Группа 33П курс 3 |  |
| Руководитель | Ю.С.Мамшева |
| подпись |  |
|  | Е.П.Голубева |
| подпись |  |
|  | Е.А.Романова |
| подпись |  |

Нижний Новгород

2024

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc170829521)

[**1.** **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 4](#_Toc170829522)

[1.1. Краткая характеристика предприятия 4](#_Toc170829523)

[1.2. Организационная структура предприятия 4](#_Toc170829524)

[1.3. Анализ программного обеспечения и технического оснащения предприятия 4](#_Toc170829525)

[**2.** **ПРАКТИЧЕСКАЯ ЧАСТЬ** 4](#_Toc170829526)

[2.1. Составление технического задания 4](#_Toc170829527)

[2.2. Проектирование информационной системы 9](#_Toc170829528)

[2.3. Макет приложения 17](#_Toc170829529)

[2.4. Разработка графического интерфейса информационной системы 21](#_Toc170829530)

[2.5. Разработка тестовой документации 23](#_Toc170829531)

[**Заключение** 35](#_Toc170829532)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 36](#_Toc170829533)

[**ПРИЛОЖЕНИЯ** 37](#_Toc170829534)

[ПРИЛОЖЕНИЕ А 37](#_Toc170829535)

[ПРИЛОЖЕНИЕ Б 48](#_Toc170829536)

# **ВВЕДЕНИЕ**

Производственная практика по ПМ.02 Осуществление программных модулей в «1С-Рарус» в период с 06.06.2023г. по 03.07.2024г.

Целью данного учебной практики является изучение и применение основных принципов осуществления и интеграции программных модулей.

В рамках производственной практики были поставлены следующие задачи:

1. Изучить основные принципы разработки требований к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
2. Освоить навыки выполнения интеграцию модулей в программное обеспечение.
3. Изучить методы выполнения отладки программного модуля с использованием специализированных программных средств.
4. Изучить основные принципы осуществления разработки тестовых наборов и тестовых сценариев для программного обеспечения
5. Освоить навыки инспектирования компонент программного обеспечения на предмет соответствия стандартам кодирования.

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## Краткая характеристика предприятия

Группа компаний «1С-Рарус» — разработчик, системный интегратор и дистрибьютор решений на платформе «1С:Предприятие». Совместное предприятие фирмы «1С» и компании «Рарус» с 1994 г.

3 000 сотрудников, 18 региональных офисов, 3 500 партнеров ИТ‑компаний в 230 городах.

170 000 клиентов коммерческого и бюджетного сектора. 21 000 пользователей облачных сервисов Аренды 1С. 1 000 корпоративных клиентов.

Более 100 проектов‑лауреатов в конкурсах 1С:Проект года, Global CIO, TAdviser. Программы разработки «1С‑Рарус» входят в топ лучших отраслевых продуктов 1С. Выпущено 90 решений для 17 отраслей экономики.

Организатор ежегодной технической конференции 1С‑RarusTechDay. Создает базовые кафедры 1С в университетах по всей стране.

## Организационная структура предприятия

Рисунок 1 – Организационная структура предприятия

* 1. Анализ программного обеспечения и технического оснащения предприятия

На рабочем месте сотрудника установлены персональные компьютеры Dell OptiPlex 7010 MT 210-39444 i5 3470.

Рассмотрим характеристики комплектующих рабочего ПК. Данные представлены на рисунке 2.

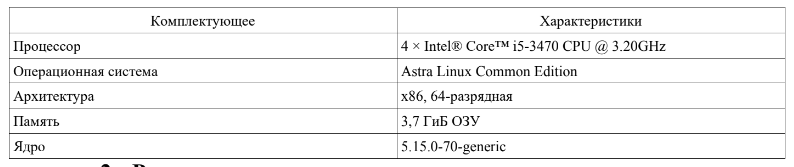


Рисунок 2 – Конфигурация аппаратного обеспечения

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

## Анализ предметной области

* + 1. Информация о продукте

Название: Учетная система для пункта выдачи заказов

Дата создания: 06.06.2024 г.

* + 1. Актуальность

Сейчас крайне высокую популярность приобрели различные торговые онлайн-площадки (маркетплейсы). Основа их бизнес-модели – широкая сеть пунктов выдачи заказов (ПВЗ), которые есть практически везде, даже в не-больших городах и поселках. При этом ПВЗ – это не подразделения маркет-плейса, а сторонние компании, которые работают по франчайзинговой моде-ли, под единым брендом и в единой инфраструктуре.

* + 1. Целевая аудитория

1. Покупатели:
   * Физические лица (розничные покупатели) - широкая аудитория потребителей, совершающих покупки для личных нужд.
   * Юридические лица (корпоративные покупатели) - представители бизнеса, делающие закупки для нужд компании.
   * Различные сегменты покупателей по возрасту, доходу, интересам и другим демографическим и поведенческим характеристикам.
2. Продавцы:
   * Индивидуальные предприниматели и малый бизнес, использующие маркетплейс как канал продаж.
   * Крупные бренды и ритейлеры, расширяющие свое присутствие в онлайн-среде.
   * Производители товаров, выходящие напрямую к конечным потребителям.
3. Партнеры экосистемы:
   * Логистические компании, обеспечивающие доставку товаров.
   * Платежные системы, предоставляющие инструменты для осуществления транзакций.
   * Сервисные провайдеры, оказывающие дополнительные услуги продавцам и покупателям.
     1. Цель

Целью проекта является повышен е удобства и доступности для покупателей

* + 1. Задачи
* Регистрация новых клиентов и сохранение их данных.
* Обработка заказов от интернет-магазинов или других поставщиков товаров.
* Отслеживание статуса каждого заказа (например, ожидает оплаты, в пути, готов к выдаче).
* Ведение учета товаров на складе пункта выдачи заказов.
* Подготовка документов для выдачи товара клиенту (счета-фактуры, акты приема-передачи и т. д.).
* Оформление возвратов товаров и проведение соответствующих операций в системе учета.
* Формирование отчетности о работе пункта выдачи заказов для руководства компании.
  + 1. Стадии и сроки

Таблица 1 – Этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Этап | Работы этапа | Ответственность | Сроки |
| Определение требований | Выявление и документирование потребностей | Батракова Валерия | 1 день |
| Анализ существующих систем и выявление проблем | Батракова Валерия  И Гаряев Семен |
| Определение функциональных и нефункциональных требований к продукту | Батракова Валерия и Гаряев Семен |
| Проектирование | Разработка архитектуры системы | Батракова Валерия и Гаряев Семен | 2 дня |
| Проектирование модулей, интерфейсов и взаимодействия между ними | Батракова Валерия и Гаряев Семен |
| Определение технологий, инструментов и методологий разработки | Батракова Валерия и Гаряев Семен |
| Разработка | Написание исходного кода программы | Батракова Валерия и Гаряев Семен | 4 дня |
| Модульное тестирование отдельных компонентов | Батракова Валерия |
| Интеграция и тестирование всей системы | Гаряев Семен |
| Тестирование | Проверка функциональности, производительности, безопасности и других характеристик | Батракова Валерия и Гаряев Семен | 2 дня |
| Выявление и устранение ошибок и дефектов | Гаряев Семен |
| Обеспечение качества и соответствие требованиям | Батракова Валерия |
| Развертывание | Подготовка программного продукта к эксплуатации | Батракова Валерия | 1 день |
| Установка, настройка и предоставление доступа пользователям | Гаряев Семен |
| Сопровождение и поддержка | Исправление ошибок и внесение изменений по запросам пользователей | Батракова Валерия и Гаряев Семен | 1 год |
| Модернизация, обновление и масштабирование системы | Гаряев Семен |
| Мониторинг работоспособности и решение возникающих проблем | Батракова Валерия |

* + 1. Ресурсы

1. Человеческие ресурсы: разработчики, пользователи.
2. Инфраструктурные ресурсы: база данных пользователей, база данных товаров, база данных сотрудников пункта выдачи.
3. Программные ресурсы: ОС Windows 10 и выше, 1С:Предприятие.
4. Информационные ресурсы: 1С ИТС.
5. Финансовые ресурсы: оплата труда разработчиков.
   * 1. Функциональные требования

Примерный набор модулей:

1. Базовая автоматизация бизнес-процессов
2. Интеграция системы взаимодействия
3. Интеграция системы распознавания на основе ИИ
4. Интеграция системы аналитики

## Проектирование программного продукта

В базе данных «FoodLog» хранятся сведения о пользователях, рецептах и необходимых для них ингредиентов.

У каждого пользователя известны его никнейм, рост, вес, пол, его предпочтения в еде, цель использования приложения (похудение, набор мышечной массы, поддержание здорового питание, диета, связанная с состоянием здоровья и т.д.), номер телефона и дата рождения.

В базе данных хранится информация о рецептах блюд, их ингредиентах, пищевой ценности и заведениях, в которых можно заказать их в готовом виде.

Каждое заведение, осуществляющее доставку продуктов и готовых блюд, вносится в таблицу учета поставщиков.

База данных координирует доставку блюд навынос и продуктов пользователям.

В результате проведенного анализа предметной области базы данных «FoodLog» легко перечислить основные сущности этой БД. Основные таблицы БД представлены на рисунке 19.

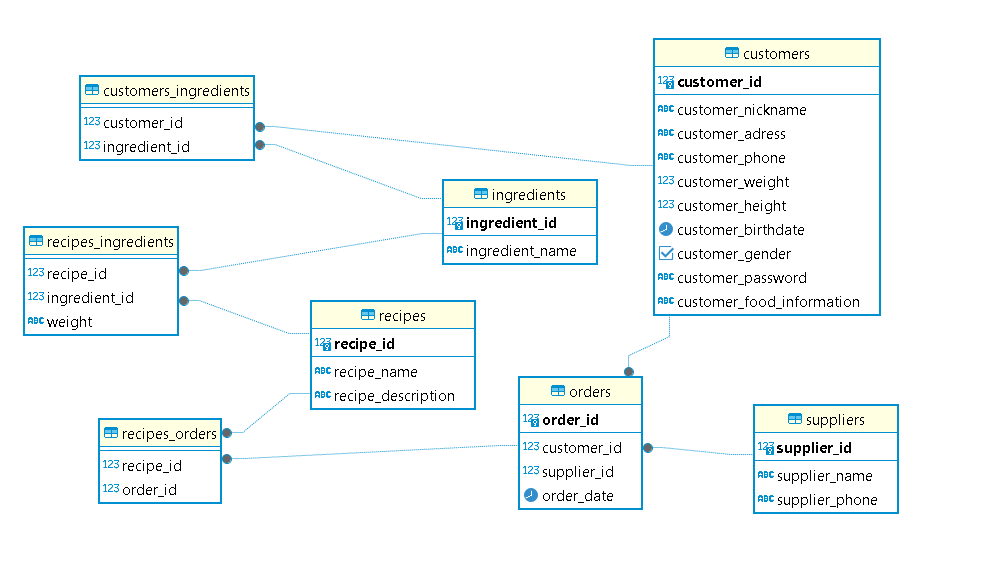


Рисунок 19 – ER-диаграмма

Диаграмма, представленная на рисунке 20 описывает работу приложения «FoodLog»: пользователь регистрируется, вводя свои предпочтения в еде и свои цели, ему предлагается индивидуальный план на день с возможностью заказать предложенные блюда.

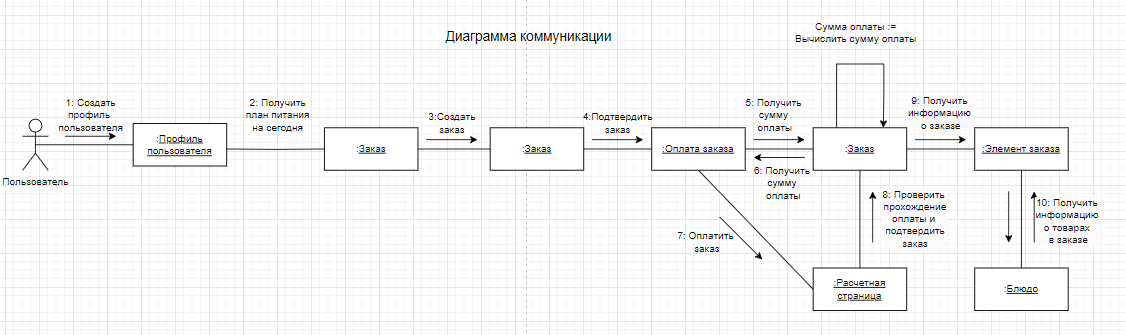


Рисунок 20 – Диаграмма коммуникации

Диаграмма приложения «FoodLog», представленная на рисунке 21, описывает классы (включая частные случаи классов: интерфейсы, примитивные типы, классы-ассоциации и многие другие) между которыми устанавливаются ассоциация, обобщение и зависимости. Из диаграммы следует, что каждый пользователь может зарегистрировать свой аккаунт, редактировать его и удалить его, каждый пользователь может осуществить заказ, добавляя и удаляя его элементы.

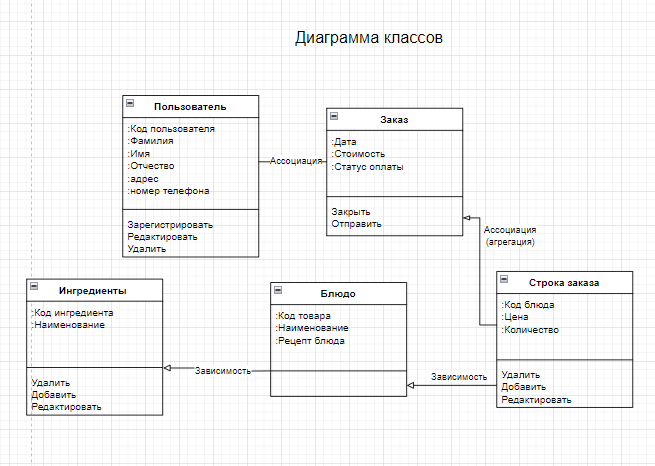


Рисунок 21 – Диаграмма классов

Диаграмма последовательности приложения «FoodLog», представленная на рисунке 22, описывает протокол конкретного сеанса работы системы – оформление пользователем заказа, где обращаясь к информационной системе пользователь запрашивает план питания и каталог блюд, далее выбирает блюдо и оформляет заказ на доставку, а в базу данных «заказы» вносится информация о данной доставке.

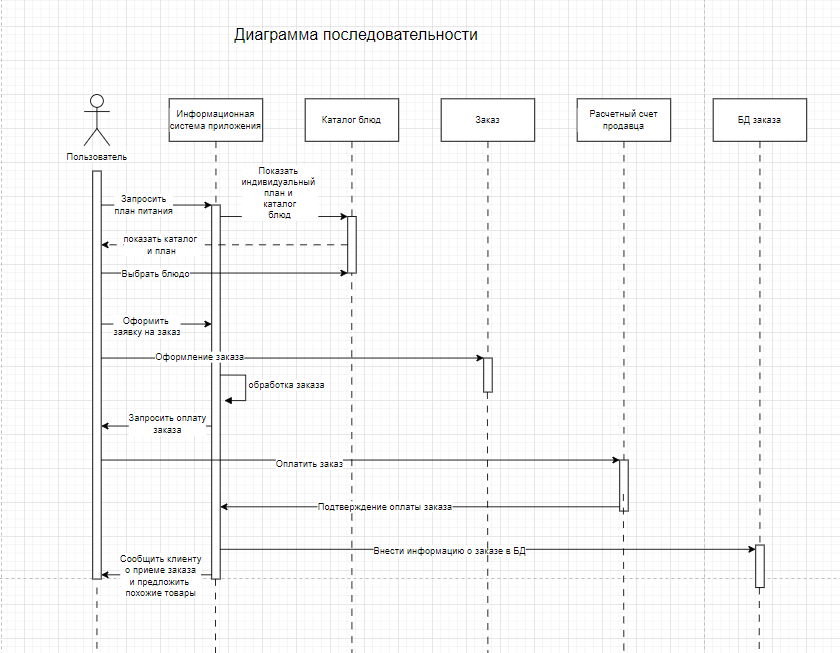


Рисунок 22 – Диаграмма последовательности

Диаграмма состояний приложения «FoodLog», представленная на рисунке 23, описывает ситуацию оформления пользователем заказа, где происходит проверка на наличие всех позиций заказа в определенном заведении, осуществляющем доставку

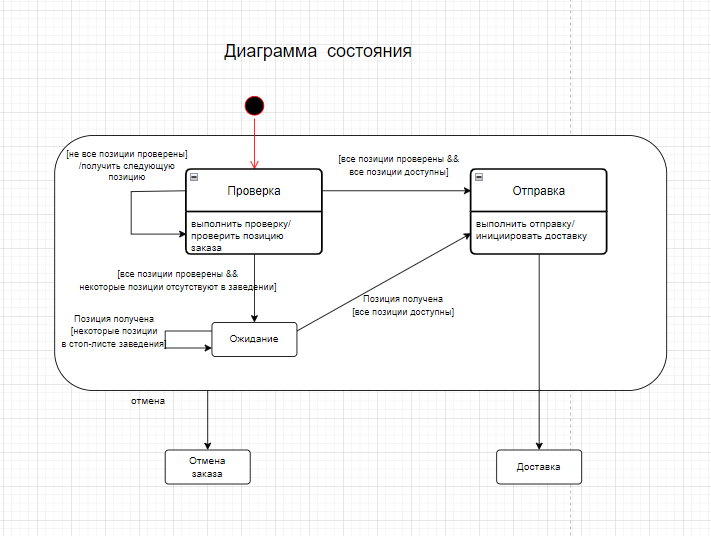


Рисунок 23 – Диаграмма состояния

Диаграмма деятельности системы «FoodLog», представленная на рисунке 24, описывает поведение системы при оформлении пользователем заказа и передачу информации о заказе заведению, которое в свою очередь занимается доставкой.

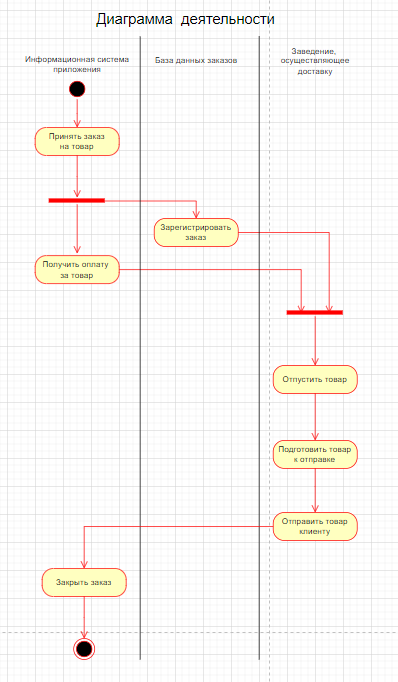


Рисунок 24 – Диаграмма деятельности

Диаграмма развертывания системы «FoodLog», представленная на рисунке 25, визуализирует элементы и компоненты программы, существующих лишь на этапе ее исполнения.

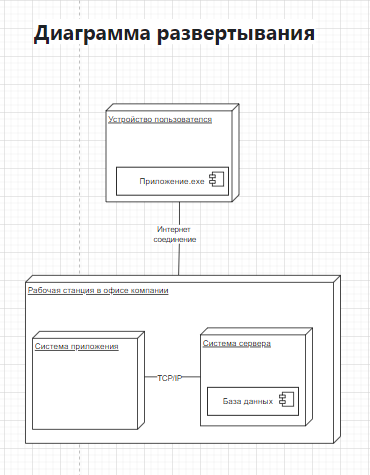


Рисунок 25 – Диаграмма развертывания

Диаграмма компонентов системы «FoodLog», представленная на рисунке 26, позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код

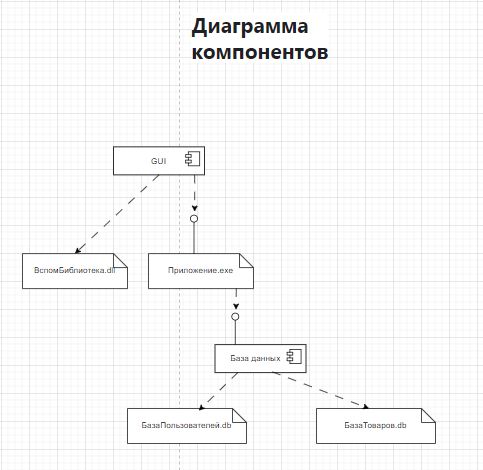


Рисунок 26 – Диаграмма компонентов

На рисунке 27 представлена диаграмма использования, которая описывает варианты использования системы составления индивидуального плана питания «FoodLog», где рассматриваются возможности ознакомления с полной базой рецептов системы, редактирования собственного плана питания, заказа продуктов из магазина и заказа готовых блюд из специализированных заведений.

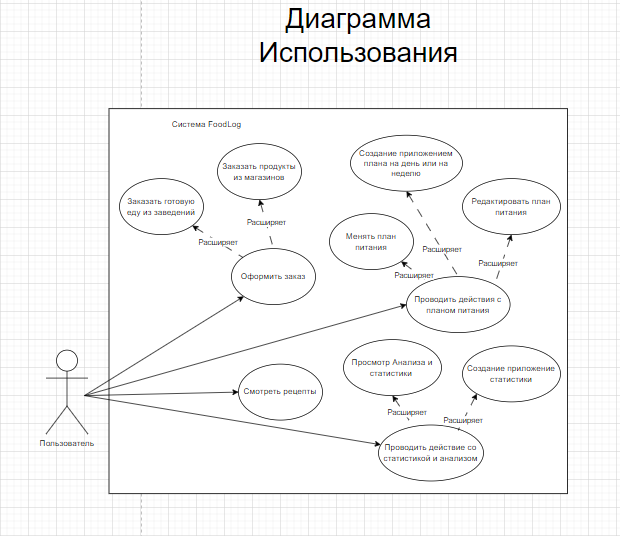


Рисунок 27 – Диаграмма использования

На рисунке 28 представлена диаграмма потоков данных приложения «FoodLog», которая описывает внешние для системы источники данных, потоки и хранилища данных, к которым имеется доступ для процесса оформления заказа пользователем.



Рисунок 28 – Диаграмма потоков данных

## Разработка программного продукта

Для разработки приложения «FoodLog» была выбрана Avalonia UI. Avalonia - это открытая, кроссплатформенная графическая система отображения. Она входит в .NET Foundation и является одним из самых активных проектов сообщества в области IT. Она используется для создания продвинутых и красивых интерфейсов. Avalonia поддерживает множество платформ, включая Windows, Linux, macOS, iOS, Android и WebAssembly. Созданная на современном стеке .NET, Avalonia позволяет использовать любой язык из семейства .NET (C#, F#, Visual Basic) и язык разметки XAML для определения интерфейса. Подобно WPF, Avalonia использует систему стилей на основе XAML, а ее система компоновки и привязки данных предоставляет знакомую среду для разработчиков, имеющих опыт работы с фреймворками, основанными на XAML. Ключевым отличием Avalonia от других графических систем является отсутствие привязки к графическим элементам операционной системы. Вместо этого используются собственные элементы, обеспечивающие гибкость и высокую степень единообразия поведения на разных платформах.

Для комфортной работы с Avalonia была выбрана интегрированная среда разработки Visual Studio. Расширение Avalonia для Visual Studio включает конструктор XAML, который можно использовать для предварительного просмотра XAML в реальном времени по мере его написания.

В соответствии с техническим заданием и утвержденным макетом был разработан графический интерфейс приложения «FoodLog», программный код разметки всех страниц приложения представлен в приложении Б на рисунках 1-7, а результат выполнения кода разметки представлен на рисунках 36 – 39.

С полным программным кодом можно ознакомиться по ссылке: <http://gogs.ngknn.local:3000/Praktica/FoodLog2.0.git>

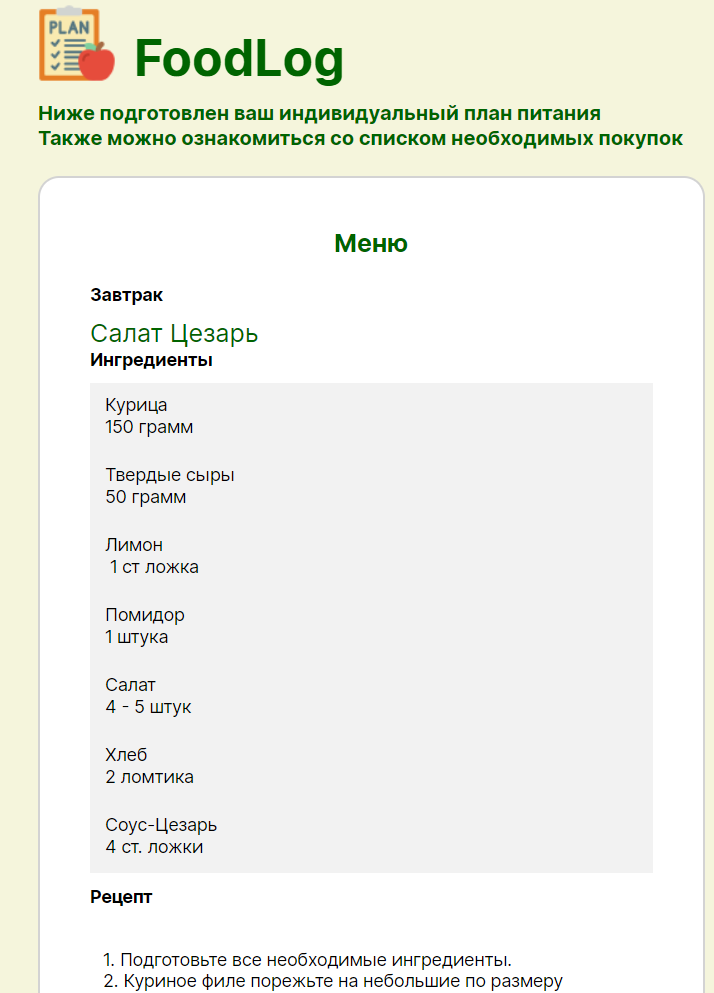


Рисунок 36 – Главная страница



Рисунок 37 – Страница каталога рецептов

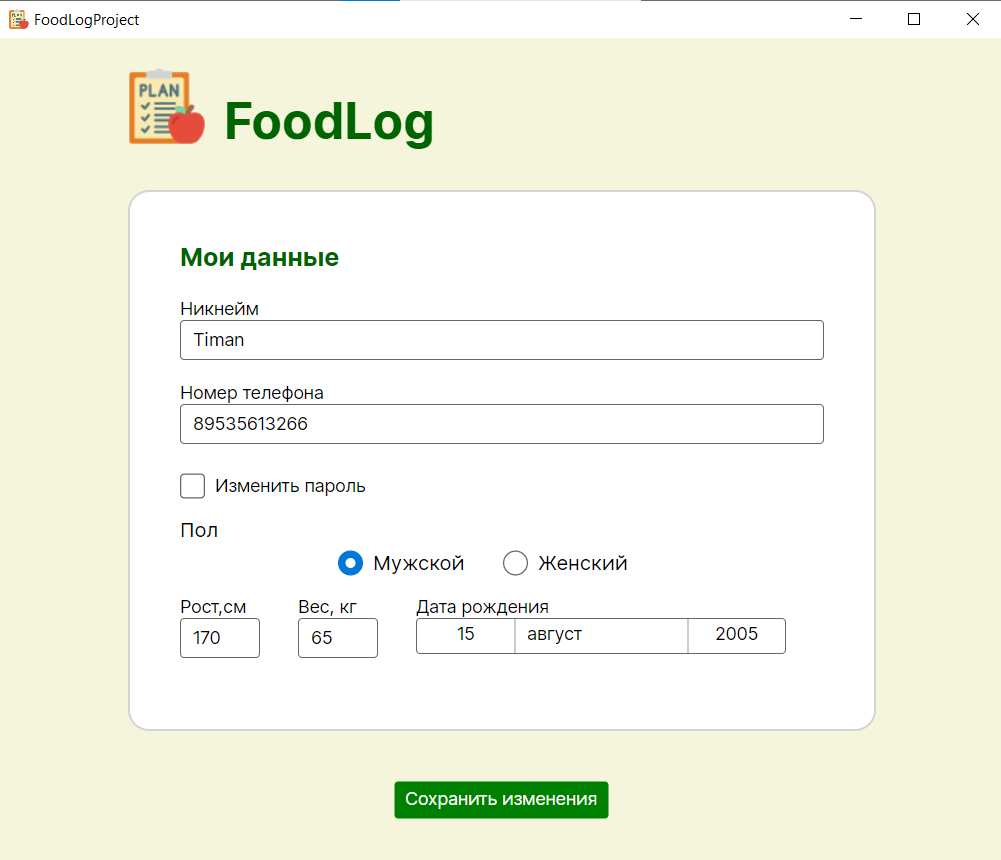


Рисунок 38 – Страница Профиля пользователя

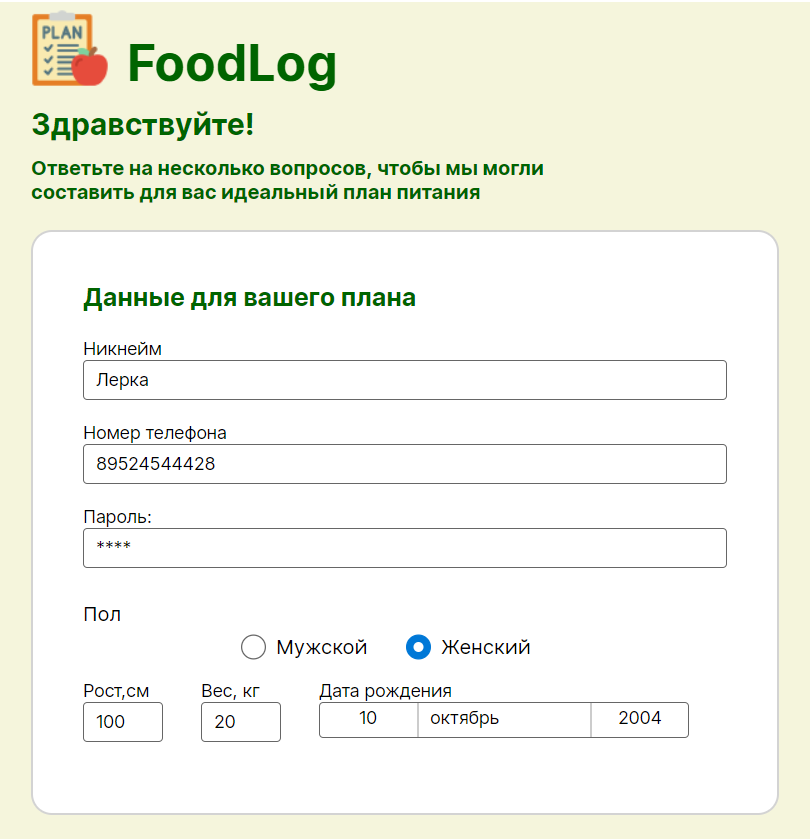


Рисунок 39 – Страница регистрации пользователя

## Тестирование программного продукта

Тест-план и стратегия тестирования для приложения «Пунк выдачи заказов»

* + 1. Название и версия тестируемой программы

Название тестируемой программы: «Пункт выдачи заказов», версия 1.0

* + 1. Техническое окружение

Техническое окружение при помощи, которого будет происходить тестирование приложения представлены на рисунках 40 и 41.

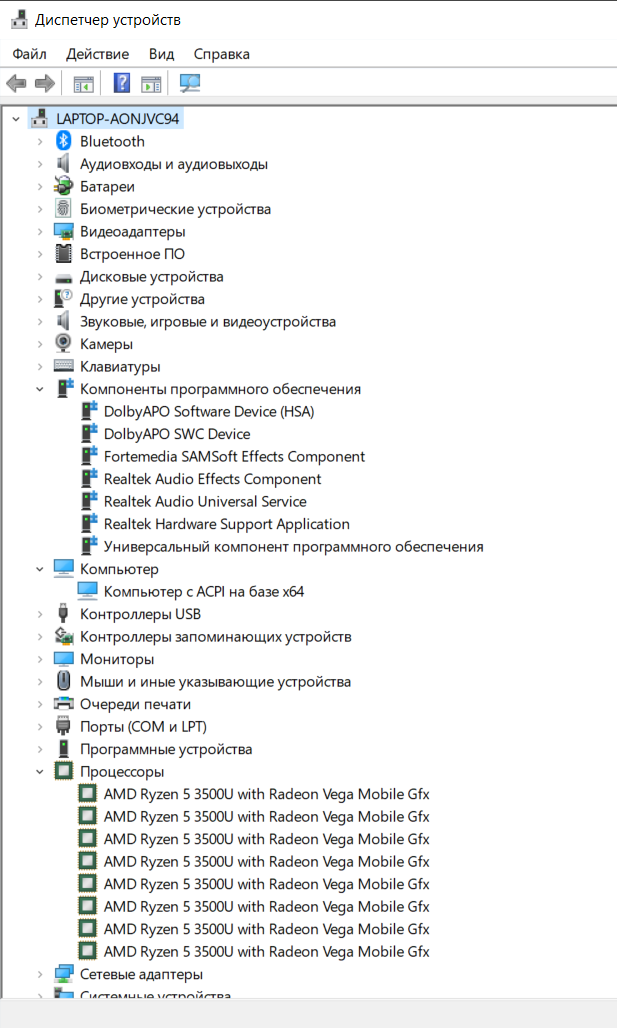


Рисунок 40 – Техническое окружение

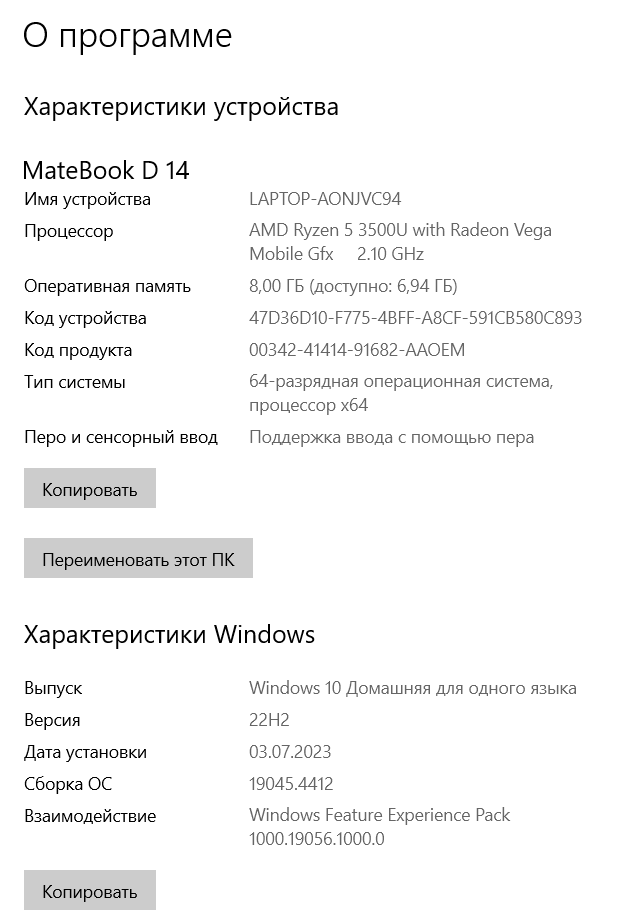


Рисунок 41 – Техническое окружение

* + 1. Перечень работ:
* логика поступления товара на пункт выдачи и логика выдачи заказов с пункта выдачи
* проверка на корректность входных данных (пароль длиной не меньше 8 символов, дата рождения пользователя не может быть больше текущего дня, номер телефона длиной не менее 11 символов);
* взаимодействие с пользователем (ввод данных, обработка ошибок ввода, рассылка уведомлений).
  + 1. Критерии качества:
* точность: продукт должен работать в соответствии с требованиями и техническим заданием.
* надежность: продукт не должен содержать критических и блокирующих дефектов в окончательной версии проекта;
* удобство использования: интерфейс должен быть интуитивно понятным и удобным для пользователя;
* производительность: программа должна обеспечивать быстрый ответ на запрос пользователя;
* поддерживаемость: продукт должен быть легко поддерживаем и модифицируем в случае необходимости.
  + 1. Оценка рисков

Таблица 2 – Анализ и оценка рисков

|  |  |  |
| --- | --- | --- |
| Риск | Вероятность появления | Варианты решения |
| Недостаточная защита от некорректного ввода данных пользователем, что может привести к сбою программы. | Часто | Разработка пользовательского интерфейса с уведомлениями об ошибках ввода данных и обработкой исключений для предотвращения сбоев программы. |
| Недостаточная обработка различных случаев пользовательских данных и действий, что может привести к некорректным результатам. | Редко | Создание обширного набора тестовых данных, включающих разнообразные случаи поведения пользователя, чтобы удостовериться в корректности работы программы. |
| Неправильное обращение к входным данным, например, если программа не учитывает, что пользователь может не выбрать ни одного заказа | Крайне мала | Добавление дополнительных проверок на входные данные, включая условия и ограничения на допустимые выборы заказов пользователем. |

* + 1. Документация
* тест-план;
* тест-кейсы;
* анализ дефектов:
* результаты тестирования.
  + 1. Ресурсы:
* человеческие: команда из тестировщика и разработчика;
* временные;
* аппаратные.
  + 1. Тестовый сценарий

Тест-кейсы смок-тестирования, UI и UX тестирования и функционального тестирования представлены, соответственно, в таблицах 3-5.

Таблица 3 – Тест-кейсы смок-тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест-кейс | Шаги тест-кейса | Ожидаемый результат | Фактический результат | Прохождение теста |
| 1 | Запуск программы | 1. Запустить программу с помощью отладки 1С:Предприятие | Успешный запуск программы |  | Пройден |
| 2 | Возможность авторизации в приложении | 1. Запустить программу с помощью отладки 1С:Предприятие 2. Пройти авторизацию | Вход в профиль пользователя |  | Пройден |
| 3 | Возможность регистрации нового пользователя в приложении | 1. Запустить программу с помощью отладки 1С:Предприятие  2. От роли администратора зарегистрировать нового сотрудника | Происходит регистрация нового пользователя |  | Пройден |

Таблица 3 – Тест-кейсы смок-тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест-кейс | Шаги тест-кейса | Ожидаемый результат | Фактический результат | Прохождение теста |
| 4 | Масштабирование окна программы | 1. Запустить программу с помощью отладки 1С:Предприятие  2. Изменить размер окна вручную и при помощи кнопки «развернуть» | Программа должна адаптироваться под изменение размера окна |  | Пройден |
| 5 | Выход из программы | 1. Запустить программу с помощью отладки 1С:Предприятие  2. Выйти из программы | Программа должна предусматривать возможность выхода пользователя из программы |  | Пройден |

Таблица 4 – Тест-кейсы UI и UX тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест-кейс | Шаги тест-кейса | Ожидаемый результат | Фактический результат | Прохождение теста |
| 1 | Интуитивная понятность интерфейса | 1. Запустить программу с помощью отладки 1С:Предприятие 2. Начать работу приложения 3. Оценить визуальный интерфейс приложения | Пользователь должен понимать, что от него требуется, что достигается либо путем инструктирования, либо путем интуитивной понятности интерфейса | В приложении все элементы имеют свою подпись-пояснение, интерфейс несложный и понятный | Пройден |

Таблица 4 – Тест-кейсы UI и UX тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест-кейс | Шаги тест-кейса | Ожидаемый результат | Фактический результат | Прохождение теста |
| 2 | Эстетика | 1.Запустить программу с помощью отладки 1С:Предприятие  2. Начать работу приложения  3.Оценить визуальный интерфейс приложения | Интерфейс должен быть визуально приятным и привлекательным | Интерфейс минималистичен, не имеет лишних отвлекающих факторов, отсутствует назойливая реклама | Пройден |
| 3 | Реактивность | 1. Запустить программу с помощью отладки 1С:Предприятие  2. Начать работу приложения  3. Оценить быстроту реакции программы на нажатие кнопок, на авторизацию, регистрацию, создание плана | Программа не должна заставлять пользователя долго ждать отклик, учитывая ее простоту. | Программа реагирует на действия пользователя быстро и без задержек | Пройден |
| 4 | Масштабируемость окна программы | 1. Запустить программу с помощью отладки 1С:Предприятие  2. Начать работу приложения  3. Попробовать изменить окно программы вручную или при помощи кнопки «развернуть» | Программа должна менять размер своего окна в зависимости от действий пользователя | Окно программы меняет свой размер, элементы интерфейса адаптируются под размер окна | Пройден |
| 5 | Удовлетворение | 1. Запустить программу с помощью отладки 1С:Предприятие  2. Начать работу приложения  3.Оценить насколько удобно, увлекательно и приятно пользоваться приложением | Программа должна приносить пользователю удовлетворение, увлекать его внимание и вызывать желание играть снова и снова | Общее впечатление положительное, приложение выглядит привлекательно и удобно в использовании, что вызывает желание пользоваться им в дальнейшем | Пройден |

Таблица 5 – Тест-кейсы функциональное тестирование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест-кейс | Шаги тест-кейса | Ожидаемый результат | Фактический результат | Прохождение теста |
| 1 | Запуск программы | 1.Запустить программу с помощью отладки 1С:Предприятие | Успешный запуск программы |  | Пройден |
| 2 | Обновление виджетов на начальной странице | 1. Запустить программу с помощью отладки 1С:Предприятие  2. Авторизоваться  3.Попробовать произвести прием заказа и выдачу заказа | Система обновляет показатели склада автоматически и передает информацию в виджеты |  | Пройден |
| 3 | Рассылка уведомлений о прибытии заказа на склад | 1.Запустить telegram  2.Начать диалог с ботом  3.Зарегистрировать нового пользователя  4.В системе провести документ «Прием заказа» | Система присылает пользователю сообщение о том, что заказ прибыл на склад |  | Пройден |

Таблица 5 – Тест-кейсы функциональное тестирование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тест-кейс | Шаги тест-кейса | Ожидаемый результат | Фактический результат | Прохождение теста |
| 4 | Прием заказа при заполненном складе | 1. Запустить программу с помощью отладки 1С:Предприятие  2. Авторизоваться  3.Попробовать произвести прием заказа, при условии, что все ячейке на складе заняты | Система не позволяет создать документ, выводя соответствующее предупреждение |  | Пройден |
| 5 | Регистрация нового пользователя | 1. Запустить программу с помощью отладки 1С:Предприятие  2.Под ролью администратора попробовать создать нового сотрудника | Происходит регистрация нового пользователя и в дальнейшем программа предлагает авторизоваться под новой учетной записью |  | Пройден |
| 6 | Реактивность интерфейса | 1. Запустить программу с помощью отладки 1С:Предприятие  2.Оценить реактивность интерфейса | Все кнопки в приложении выполняют свои функции, которые понятны интуитивно или указаны напрямую. | Все кнопки работают и их назначение понятно. | Пройден |
| 7 | Рассылка уведомлений об истечении срока хранения заказа | 1. Запустить telegram 2. Начать диалог с ботом 3. Зарегистрировать нового пользователя 4. Создать в системе заказ с истекающим сроком хранения и провести обработку «Уведомить об истечении срока хранения» | После выполнения обработки бот присылает в чат уведомление о том, что срок хранения заказа истекает пользователю |  | Пройден |
| 8 | Регистрация через telegram бота | 1.Запустить telegram  2.Начать диалог с ботом  3.Зарегистрировать нового пользователя | После успешной регистрации пользователя в telegram боте система автоматически вносит нового пользователя в соответствующий справочник |  | Пройден |
| 9 | Просмотр команд telegram бота | 1. Запустить telegram 2. Начать диалог с ботом | После ввода команды /help бот выводит список всех команд, которые он знает |  | Пройден |

Показатели нагрузки и затраты системных ресурсов представлены на рисунках 42-45. Из полученных показателей можно сделать вывод, что нагрузка на систему незначительная и затрачивает минимальные системные ресурсы, так как интерфейс приложения несложный, сама игра не требует большой вычислительной мощности.

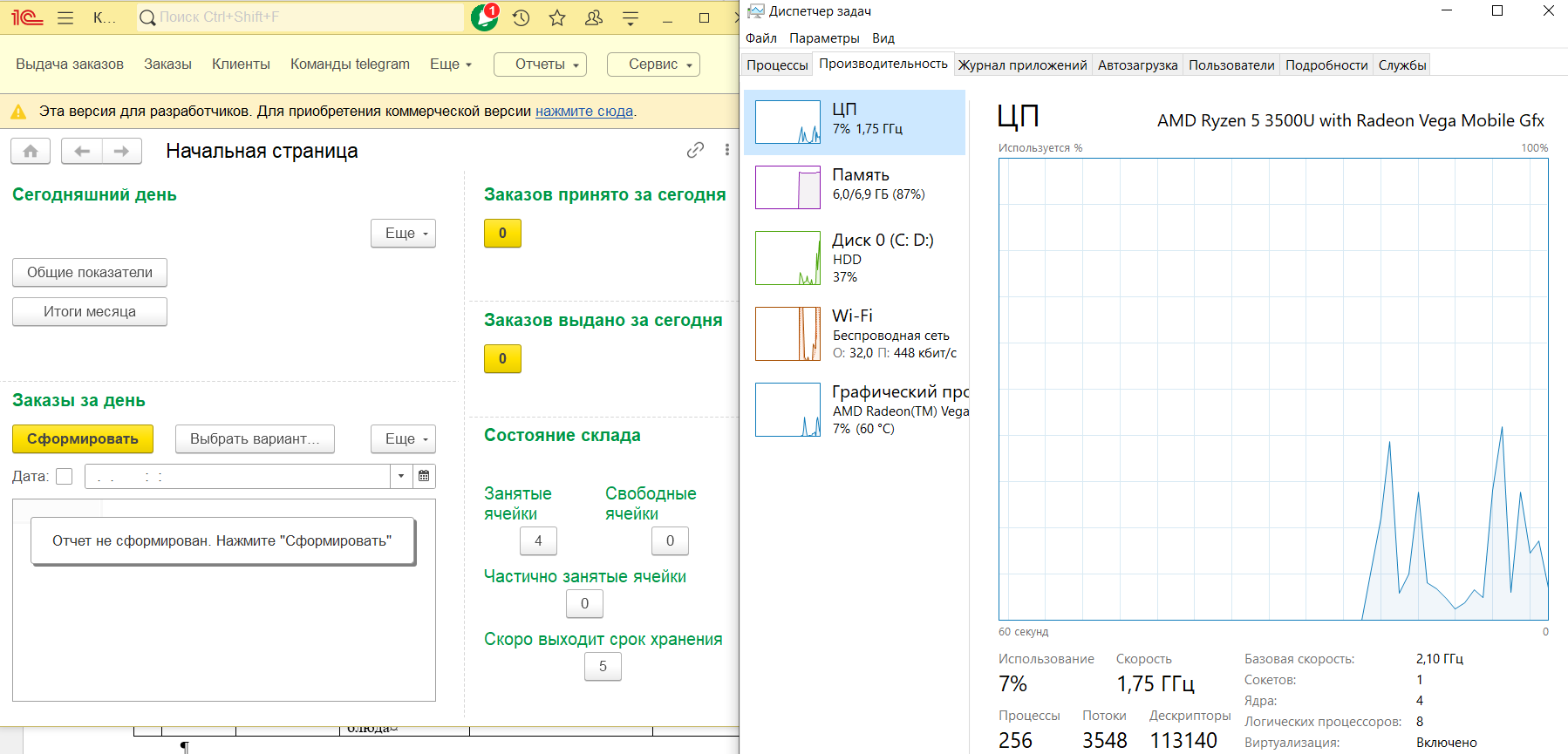


Рисунок 42 - Использование ЦП

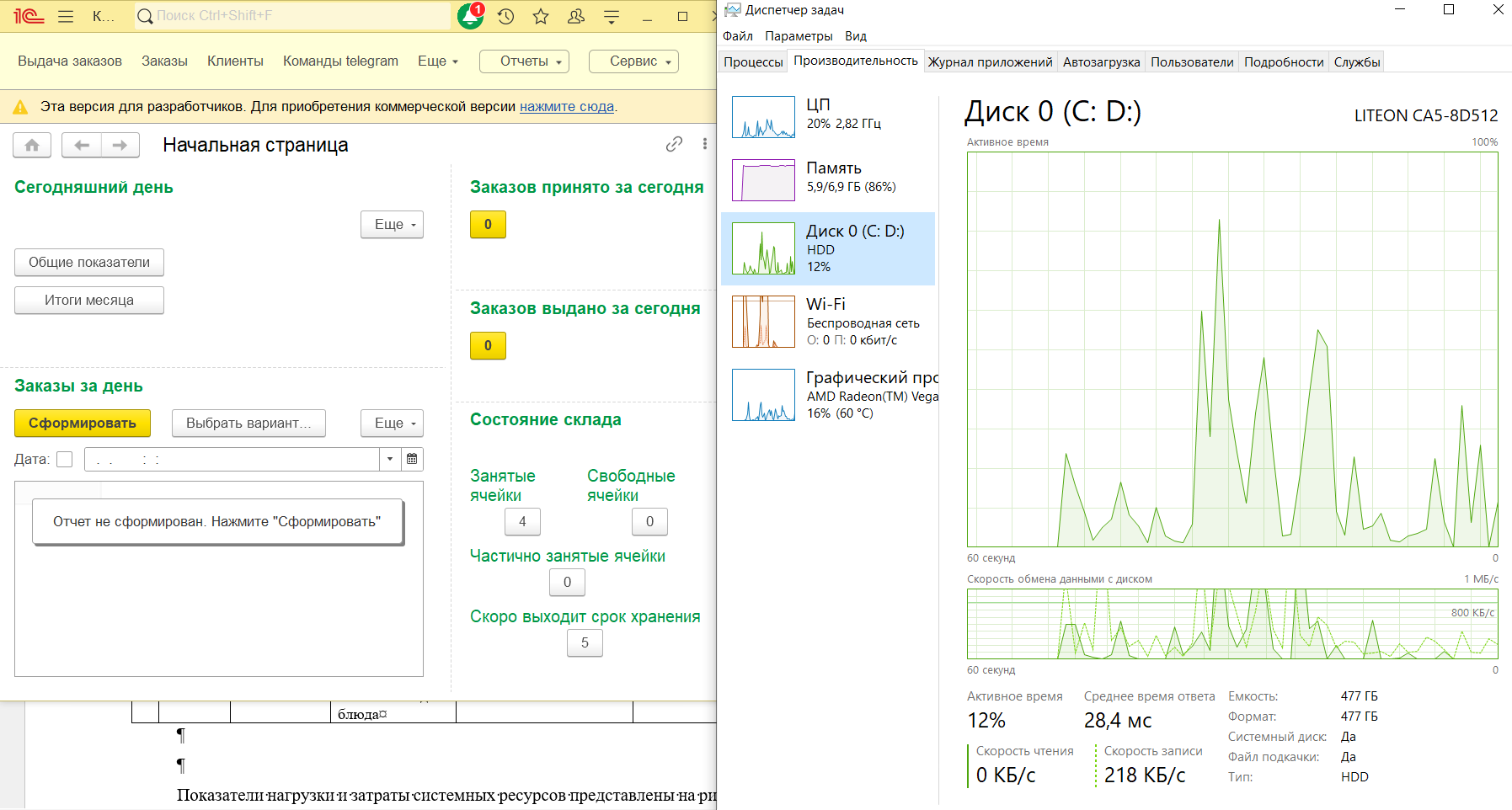


Рисунок 43 - Использование Диска

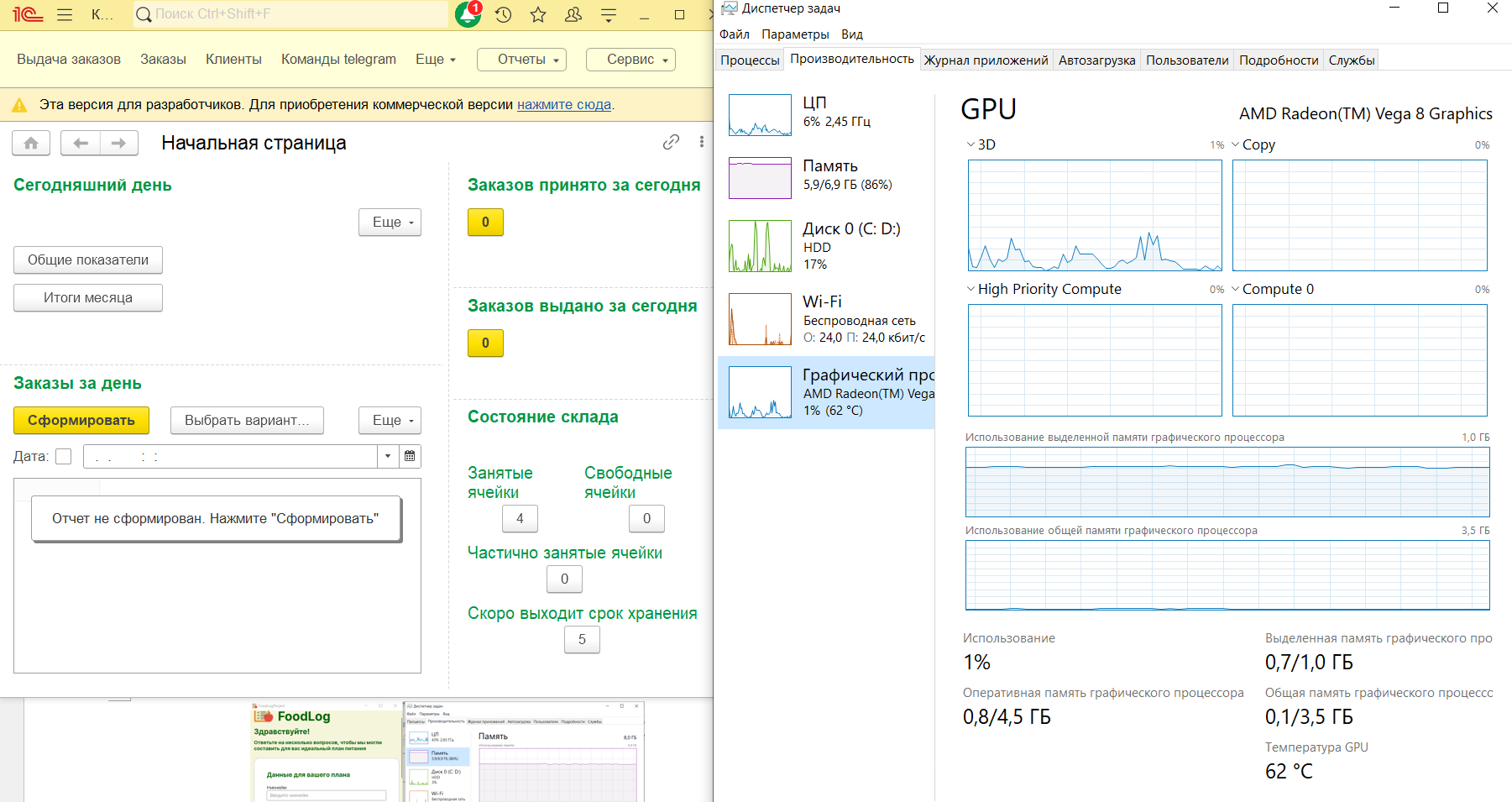


Рисунок 44 - Использование GPU

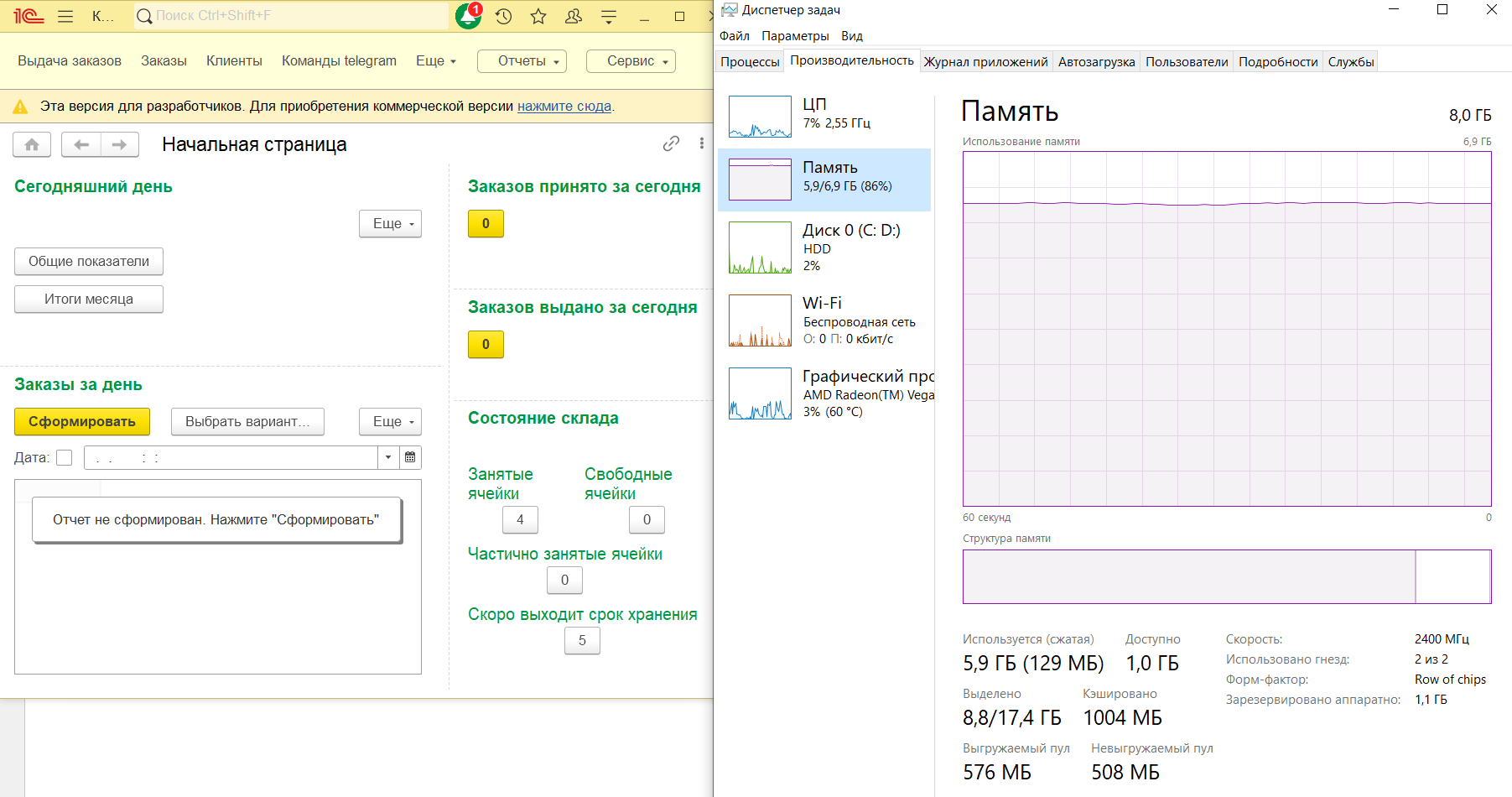


Рисунок 45 - Использование памяти

# **Заключение**

В результате производственной практики по профессиональному модулю Осуществление интеграции программных модулей были достигнуты поставленные цели и выполнены поставленные задачи.

Таким образом, выполнение производственной практики по профессиональному модулю Осуществление интеграции программных модулей позволило приобрести не только теоретические знания, но и практические навыки, необходимые для успешной работы с программными модулями и их интеграцией. Полученные результаты и опыт будут полезны в дальнейшей профессиональной деятельности и подготовке к будущим задачам в области осуществление интеграции программных модулей.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Гладкий А. 1С: Управление торговлей 8.2. Настройка, конфигурирование, администрирование. – Strelbytskyy Multimedia Publishing, 2018.
  2. Глотова М. А. 1С: Библиотека стандартных подсистем //НАУКА РОССИИ: ЦЕЛИ И ЗАДАЧИ. – 2020. – С. 17-20.
  3. Дадян Э. Г. Конфигурирование и моделирование в системе" 1С: Предприятие 8" //Новые информационные технологии в образовании. – 2018. – С. 200-203.
  4. Журба С. А. РАЗРАБОТКА И ИНТЕГРАЦИЯ TELEGRAM-БОТА С 1С //ББК 72.3 (2Рос-4Арх) я43 Л 75. – 2021. – С. 199.
  5. Зеленский О. С. Разработка системы управления проектом на основе платформы 1С: предприятие с интеграцией Telegram бота //Глав. ред. к. э. н., доц. Казаков ОД, отв. ред. к. э. н., доц. Азаренко НЮ, к. э. н., доц. Юркова ОН. – 2022. – С. 127.
  6. Кугаевских А. В. Проектирование информационных систем. Системная и бизнес-аналитика. – 2018.
  7. Кузнецова Р. В. 1С: библиотека стандартных подсистем как помощник разработчика //Российская наука: тенденции и возможности. – 2021. – С. 57-60.
  8. Куликова Т. А., Поддубная Н. А. Инструментальные средства разработки мультимедийных приложений. – 2019.
  9. Логачева Н. В., Сидорова Н. П. Программные средства разработки информационных технологий //СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. – 2017. – С. 122-128.
  10. Орешкова М. Н. Реализация в САФУ курса по выбору «Введение в конфигурирование на платформе «1С: Предприятие 8.3» //Новые информационные технологии в образовании. – 2023. – С. 68-71.
  11. Полевич К. Б. Интеграция информационных систем в единое информационное пространство на платформе" 1С: Предприятие" //Современные научные исследования и разработки. – 2018. – №. 9. – С. 304-309.
  12. Радченко М., Хрусталева Е. 1С: Предприятие 8.3. Практическое пособие разработчика. Используем 1C: EDT (+ 2epub). – Litres, 2023.
  13. Рочев К. В. Информационные технологии. Анализ и проектирование информационных систем. – 2018.
  14. Старолетов С. М. Основы тестирования и верификации программного обеспечения. – 2020.
  15. Хрусталева Е. Разработка сложных отчетов в «1С: Предприятии 8». Система компоновки данных (+ epub). – Litres, 2022.

# **ПРИЛОЖЕНИЯ**

## ПРИЛОЖЕНИЕ А