

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ

**Национальный исследовательский технологический университет «МИСИС»**  
**Новотроицкий филиал**

ФАКУЛЬТЕТ Металлургических Технологий

КАФЕДРА Математики и Естествознания

НАПРАВЛЕНИЕ Прикладная информатика

ДИСЦИПЛИНА Программная инженерия

## КУРСОВОЙ ПРОЕКТ

на тему: Разработка мобильного приложения для заказов в столовой

Студент группы БПИ-21

Руководитель \_\_\_\_\_

Оценка работы \_\_\_\_\_

И. А. Слинько

Р. Р. Абдулвелеева

« \_\_\_\_ » \_\_\_\_\_ 2024 г.

Новотроицк, 2024 г.

**Национальный исследовательский технологический университет «МИСИС»  
Новотроицкий филиал**

Факультет Металлургических технологий

**УТВЕРЖДАЮ**

Кафедра Математики и естествознания

Зав. кафедрой Швалева А.В.

«\_\_\_» \_\_\_\_\_ 2024 г.

**ЗАДАНИЕ НА ВЫПОЛНЕНИЕ**

курсового проекта

студенту группы БПИ-21

Слинько Илье Андреевичу

(*Ф.И.О. полностью*)

1. Тема КП Разработка мобильного приложения для заказов в столовой

2. Исходные данные ГОСТ Р ИСО/МЭК 12207 Процессы жизненного цикла программных средств, ГОСТ Р ИСО/МЭК 14764 Сопровождение программных средств, ГОСТ 34.602-2020 Информационные технологии (ИТ). Комплекс стандартов на автоматизированные системы, ГОСТ 19.504-79. ЕСПД. Руководство программиста

3. Основная литература Учебник «Разработка мобильных приложений для iOS и Android» В.Н. Клементьев; Учебник «Программная Инженерия» Е.М. Лаврищев; Учебник «Методология разработки программного обеспечения» А.А. Веселов; Учебник «Разработка и оптимизация мобильных приложений» И.Ю. Азарова, Е.А. Гришко

4. Лабораторное оборудование и методики, которые должны быть использованы \_\_\_\_\_

---

5. Использование ЭВМ \_\_\_\_\_ Ноутбук Lenovo IdeaPad 3 Gaming 15ARH05

---

6. План выполнения КП

Название разделов работы	Сроки	Форма промежуточной отчетности	Примечания
Описание и анализ предметной области	06.02.2024–01.03.2024		
Постановка задачи и разработка требований	02.03.2024–16.03.2024		
Проектирование системы	17.03.2024–20.04.2024		
Конструирование интерфейса приложения	21.04.2024–30.04.2024		
Тестирование базы данных приложения	01.05.2024–04.05.2024		
Защита курсового проекта	07.05.2024		

7. Руководитель работы \_\_\_\_\_ доцент к.п.н., Абдулвелеева Р. Р.  
(подпись) (должность, звание, Ф.И.О.)

Дата выдачи задания « \_\_\_\_\_ » \_\_\_\_\_ 20 24 г.

Задание принял к исполнению студент \_\_\_\_\_ Слинко И. А.  
(подпись) (Ф.И.О.)

# Содержание

Введение.....	6
1 Описание и анализ предметной области.....	7
1.1 Описание предметной области .....	7
1.2 Описание систем-аналогов.....	8
1.2.1 «KFC», «Burger King» и «Додо-Пицца» .....	8
1.2.2 Уникальные особенности «Онлайн-Столовки».....	9
2 Постановка задачи и разработка требований .....	10
2.1 Постановка задачи .....	10
2.2 Формирование требований к программному обеспечению .....	10
2.2.1 Функциональные требования .....	10
2.2.2 Эргономические требования.....	11
2.2.3 Технические требования .....	11
2.2.4 Производительные требования .....	12
2.3 Ограничения и особенности .....	12
2.4 Методология разработки.....	12
3 Проектирование системы .....	14
3.1 Обоснование выбора ЯП и средств разработки.....	14
3.1.1 Flutter.....	14
3.1.2 Python .....	15
3.1.3 PostgreSQL .....	15
3.1.4 Figma .....	16
3.2 Язык UML .....	17
3.2.1 Use-Case Diagram .....	17

3.2.2	ER Diagram .....	20
4	Конструирование интерфейса приложения .....	24
4.1	Обзор рабочей области Figma.....	24
4.2	Макеты экранов приложения.....	25
5	Тестирование базы данных мобильного приложения .....	32
	Заключение .....	34
	Список использованных источников .....	36

## Введение

Современное общество стремительно движется в направлении цифровизации, что оказывает значительное влияние на различные сферы нашей жизни. Технологический прогресс позволяет не только упрощать рутинные задачи, но и значительно повышать эффективность процессов.

В данном контексте особенно актуальными становятся инновационные подходы в организации питания, где информационные технологии играют ключевую роль. Введение мобильных приложений для заказа еды является ярким примером такого подхода, обеспечивающего пользователям удобство и свободу выбора.

На сегодняшний день многие столовые и рестораны начинают внедрять системы онлайн-заказов для оптимизации процессов обслуживания и сокращения времени ожидания в очередях.

Однако, несмотря на активное развитие этой области, в столовых образовательных учреждений зачастую отсутствуют современные решения для автоматизации процессов заказа блюд.

Это приводит к возникновению очередей и снижению общей эффективности работы столовой, так как кассиры вынуждены выполнять сразу несколько функций: от накладывания пищи до оформления заказов.

Цель данной курсовой работы – разработка мобильного приложения "Онлайн-Столовка", которое позволит студентам и преподавателям НФ НИТУ «МИСИС» заказывать еду заранее через мобильные устройства.

Это приложение будет интегрировано с базой данных, обеспечивая автоматизацию процесса заказа и оплаты, что значительно снизит нагрузку на кассира и повысит общий уровень сервиса.

В рамках курсовой работы будут рассмотрены различные аспекты разработки мобильного приложения.

# **1 Описание и анализ предметной области**

## **1.1 Описание предметной области**

В современных образовательных учреждениях столовые играют важную роль, обеспечивая студентов и преподавателей качественным питанием. Однако, существующая система обслуживания в столовых часто сталкивается с проблемой долгих очередей и неэффективного управления заказами. Это вызвано тем, что кассиры вынуждены выполнять несколько функций одновременно: оформлять заказы, принимать оплату и накладывать пищу. В результате время ожидания клиентов увеличивается, что снижает общую удовлетворенность сервисом.

Для решения этой проблемы необходима разработка автоматизированной системы заказа и оплаты еды. Такая система позволит пользователям заказывать блюда заранее через мобильное приложение, что значительно сократит время ожидания в столовой и повысит эффективность работы кассиров, оставив за ними лишь функцию наложения готовой еды.

Основные участники процесса включают студентов и преподавателей (пользователей системы), кассиров и административный персонал столовой. Пользователи смогут через мобильное приложение просматривать меню, заказывать блюда и оплачивать их онлайн. Кассиры будут получать заказы в реальном времени и готовить их к выдаче. Административный персонал будет управлять меню и отслеживать статистику заказов.

Таким образом, предметная область мобильного приложения "Онлайн-Столовка" охватывает сферу общественного питания в НФ НИТУ «МИСИС».

Основные элементы предметной области включают в себя 4 элемента.

1-ым элементом является заказ и предзаказ еды. Пользователи могут выбирать блюда из предоставленного меню, добавлять их в корзину и заказывать заранее, чтобы избежать долгих очередей.

2-ым элементом являются процессы в столовой. Приложение оптимизирует работу персонала, предоставляя им информацию о заказах и помогая мониторить запасы продуктов, хотя оно не включает взаимодействие сотрудников столовой напрямую.

3-им элементом является информационное взаимодействие. Пользователи могут получать информацию об актуальных блюдах, калорийности, ценах, изменениях в меню и действующих акциях через приложение;

4-ым элементом является развитие функционала. В будущем в приложение планируется внедрить систему акций, подписки и добавление различных диет.

Главными целями приложения являются:

- оптимизация процесса заказа и получения питания для студентов и преподавателей;
- существенное сокращение временных затрат, связанных с ожиданием в очереди;
- повышение общего комфорта при питании в столовой ВУЗа.

## **1.2 Описание систем-аналогов**

### **1.2.1 «KFC», «Burger King» и «Додо-Пицца»**

«KFC» предлагает обширное меню и уведомления о текущих акциях.

Пользователи могут делать заказ и оплачивать его онлайн, получать бонусы, акции и скидки для постоянных клиентов. Приложение также предоставляет развлекательный контент для пользователей.

«Burger King» предоставляет функционал онлайн-меню, где пользователи могут просматривать разнообразное меню, включая акции и специальные предложения. Пользователи могут индивидуализировать заказ, выбирая ингредиенты, размер порции и добавляя опции. Программа лояльности



предлагает бонусы и скидки для постоянных клиентов. Также доступно отслеживание заказа в реальном времени и безналичная оплата через приложение.

«Додо Пицца» позволяет просматривать меню, настраивать и размещать заказы. Пользователи могут отслеживать приготовление своего заказа на кухне и перемещение курьера с заказом в реальном времени. Приложение также уведомляет о специальных предложениях и предоставляет бонусную программу для накопления бонусов за каждый заказ.

Все рассмотренные приложения предоставляют функционал для выбора блюд, размещения заказа и его электронной (безналичной) оплаты, имеют системы поощрений для постоянных клиентов, уведомляют о новых акциях и скидках, и позволяют отслеживать статус заказа в реальном времени.

### **1.2.2 Уникальные особенности «Онлайн-Столовки»**

Каждое из этих приложений имеет свои особенности и функциональные возможности, которые могут быть полезны при разработке "Онлайн-Столовки". Однако, ни одно из них не учитывает специфику работы столовых в образовательных учреждениях, что делает нашу задачу уникальной.

Рассмотрим уникальные особенности "Онлайн-Столовки".

1-ой особенностью является направленность на ВУЗ. Приложение ориентировано на студентов и преподавателей НФ НИТУ «МИСИС»;

2-ой особенностью является предварительный заказ. У пользователей всегда будет возможность заказа блюд заранее для минимизации времени в очереди;

3-ей особенностью являются акции для студентов. Перспектива внедрения акций и подписок для студентов и преподавателей в будущем.

## **2 Постановка задачи и разработка требований**

### **2.1 Постановка задачи**

Перед разработчиком стоит задача создать автоматизированную систему для университетской столовой под названием "Онлайн-Столовка".

Цель данного проекта является обеспечение удобного и эффективного взаимодействия студентов и преподавателей с процессом заказа и получения питания.

Это мобильное приложение должно обеспечить оптимизацию процесса питания, сократить время ожидания в очереди и повысить общий комфорт посетителей столовой института.

Подводя итоги, можно сказать, что для успешного создания мобильного приложения "Онлайн-Столовка" необходимо решить ряд ключевых задач:

- 1) Создать удобную систему, позволяющей студентам и преподавателям университета легко и быстро ознакомиться с меню столовой (включая информацию о блюдах, ценах и калорийности), сделать предварительный заказ (включая возможность указания количества порций) и оплатить обед онлайн;
- 2) Внедрить функции отслеживания статуса заказа и уведомления студентов и преподавателей о готовности их блюд;
- 3) Гарантировать стабильную работу системы на территории института.

### **2.2 Формирование требований к программному обеспечению**

#### **2.2.1 Функциональные требования**

- 1) Мобильное приложение должно предоставлять пользователям возможность просматривать меню столовой с актуальной информацией о блюдах, ценах и калорийности;

- 2) Необходимо обеспечить наличие функционала выбора блюд и указания количества порций;
- 3) В приложении должна быть реализована система онлайн-оплаты;
- 4) Необходимо обеспечить возможность предварительного заказа блюд с указанием времени получения;
- 5) Необходимо обеспечить возможность отслеживания статуса заказа в реальном времени и получения уведомлений о готовности заказа;
- 6) В приложении должна быть возможность управления меню и заказами со стороны административного персонала.

### **2.2.2 Эргономические требования**

- 1) Приложение должно иметь интуитивно понятный и удобный интерфейс, который легко освоить пользователям без специальной подготовки;
- 2) Дизайн интерфейса должен быть адаптивным, корректно отображаться на различных устройствах и размерах экранов.

### **2.2.3 Технические требования**

- 1) Приложение должно корректно функционировать на основных мобильных операционных системах (iOS, Android);
- 2) Система должна быть интегрирована с существующими информационными системами университета;
- 3) Данные пользователей должны быть надежно защищены в соответствии с современными стандартами безопасности, обеспечивая конфиденциальность и целостность данных.

## **2.2.4 Производительные требования**

- 1) Приложение должно обеспечивать высокую скорость работы и минимальное время отклика на действия пользователя;
- 2) Система должна быть масштабируемой и способной обрабатывать большое количество одновременных запросов без снижения производительности.

## **2.3 Ограничения и особенности**

- 1) Система ограничена территорией университета и не предоставляет услуги за его пределами.
- 2) Бронирование столов не предусмотрено; пользователи занимают свободные столики по мере их наличия.
- 3) Система функционирует по фиксированному расписанию, которое не может быть изменено пользователями.

## **2.4 Методология разработки**

Для реализации проекта "Онлайн-Столовка" будет использоваться методология итеративной и инкрементальной разработки. Эта модель позволяет гибко управлять разработкой, внедрять изменения на ранних этапах, а также реагировать на потребности пользователей. Учитывая, что проект представляет собой мобильное приложение для столовой, данная модель позволяет поэтапно создавать и улучшать функционал, что важно для адаптации к требованиям клиентов и эффективной разработки.

Методология итеративной и инкрементальной разработки состоит из 6 фаз.

1-я фаза – это анализ и планирование.

В этой фазе происходит определение требований к приложению, анализ рынка и аудитории, разработка плана разработки.

2-я фаза – это проектирование.

В этой фазе происходит создание дизайна пользовательского интерфейса (UI) и определение архитектуры приложения; а также создание прототипов.

3-я фаза – это реализация и тестирование.

В этой фазе начинается разработка приложения с внедрением основного функционала. После реализации каждой итерации проводится тестирование для выявления и устранения ошибок.

4-я фаза – это оценка итерации.

В этой фазе после каждой итерации проводится оценка результатов. На основе обратной связи от тестирования и пользователей корректируются планы на следующие итерации;

5-я фаза – это итерации и инкременты.

В этой фазе разработка происходит итеративно, каждая итерация добавляет новый функционал или улучшения к предыдущей версии приложения. Это позволяет быстро реагировать на изменения требований или обратную связь пользователей;

6-я фаза – это завершение проекта.

После ряда итераций приложение считается завершенным и готовым к выпуску.

Эта модель часто применяется в разработке мобильных приложений, так как она позволяет эффективно управлять проектом и быстро реагировать на изменения в требованиях или рыночных условиях.

По сравнению с Waterfall, модель "Итеративная и инкрементальная" позволяет более гибко реагировать на изменения и обратную связь, что особенно важно в сфере разработки мобильных приложений, где требования и условия рынка могут меняться быстро. Эта модель также способствует созданию более прозрачного и качественного продукта благодаря постоянному тестированию.

По сравнению с Agile, модель "Итеративная и инкрементальная" может быть менее радикальной и более управляемой, что может быть предпочтительным в случае жёстких сроков или ограниченных ресурсов.

## **3 Проектирование системы**

### **3.1 Обоснование выбора языка программирования и средств разработки**

#### **3.1.1 Комплект средств разработки Flutter**

Flutter – это открытый фреймворк для разработки мобильных приложений, созданный компанией Google. Он позволяет создавать нативные мобильные приложения для iOS и Android с использованием одного кода.

Основной язык программирования для Flutter – Dart, который также был разработан Google. Dart обеспечивает высокую производительность и быстрый отклик приложений. Он поддерживает асинхронное программирование, что делает его идеальным для работы с сетевыми запросами и большими объемами данных.

Рассмотрим преимущества Flutter перед другими фреймворками.

1-ым и одним из главных преимуществ является кроссплатформенность, то есть возможность создания приложений для iOS и Android с единой кодовой базой. Это значительно сокращает время и затраты на разработку.

2-ым преимуществом является поддержка горячей перезагрузки, позволяя разработчикам мгновенно видеть изменения в коде без необходимости перезапуска приложения.

3-им преимуществом является высокая производительность. Приложения на Flutter компилируются в нативный код, что обеспечивает высокую производительность и быстрый отклик.

4-ым преимуществом является богатый набор виджетов. Flutter предлагает множество предустановленных виджетов, которые легко кастомизировать под потребности приложения.

5-ым преимуществом является поддержка AI-инструментов, то есть инструментов с поддержкой искусственного интеллекта, таких как FlutterFlow, который упрощает процесс разработки интерфейсов и логики приложений.

### **3.1.2 Язык программирования Python**

Python – это высокоуровневый язык программирования общего назначения с открытым исходным кодом, известный своей простотой и читаемостью кода.

Рассмотрим преимущества Python перед другими языками программирования.

1-ым преимуществом является простота и читаемость синтаксиса Python, что делает его идеальным для быстрого прототипирования и разработки.

2-ым преимуществом является богатая экосистема библиотек Python. Python имеет огромную стандартную библиотеку и множество сторонних библиотек, которые упрощают разработку приложений.

3-им преимуществом является кроссплатформенность Python, то есть программы на Python могут выполняться на различных операционных системах без изменений в коде.

В рамках данной курсовой работы Python использовался для создания клиентского приложения для тестирования базы данных, что позволяет эффективно взаимодействовать с базой данных и обеспечивать корректность работы системы.

Об этом будет подробнее рассказано в последующих разделах курсовой работы.

### **3.1.3 Система управления базами данных PostgreSQL**

PostgreSQL – это мощная объектно-реляционная система управления базами данных с открытым исходным кодом, известная своей надежностью и богатым набором функций.

Рассмотрим преимущества PostgreSQL перед другими СУБД.

1-ым преимуществом является расширяемость. PostgreSQL позволяет добавлять новые функции, типы данных и индексные методы через плагины.

2-ым преимуществом является надежность. PostgreSQL поддерживает транзакции, соблюдение целостности данных и высокую доступность.

3-им преимуществом является поддержка сложных запросов. PostgreSQL имеет мощные средства для выполнения сложных SQL-запросов и аналитических задач.

Для проекта "Онлайн-Столовка" была разработана база данных на PostgreSQL, которая обеспечивает надежное хранение данных о заказах, меню и пользователях. PostgreSQL выбрана за ее надежность, расширяемость и способность обрабатывать большие объемы данных.

### **3.1.4 Облачный инструмент для прототипирования Figma**

Figma – это инструмент для создания дизайна интерфейсов и прототипирования, который работает в браузере и поддерживает совместную работу в реальном времени.

Рассмотрим преимущества Figma перед другими инструментами для создания дизайна.

1-ым преимуществом является совместная работа. Figma позволяет нескольким дизайнерам одновременно работать над одним проектом, что улучшает коммуникацию и ускоряет процесс разработки.

2-ым преимуществом является кроссплатформенность. Figma работает в любом браузере и не требует установки дополнительных программ.



3-им преимуществом является наличие плагинов, таких как "Figma to Code", которые могут конвертировать макеты в программный код, что упрощает процесс разработки и интеграции дизайна в приложение.

Figma использовалась для создания макетов пользовательского интерфейса приложения "Онлайн-Столовка". Инструмент позволил быстро и эффективно создать прототипы, которые затем были преобразованы в код с помощью плагинов, ускоряя процесс разработки.

## 3.2 Язык UML

UML (Unified Modeling Language) – это стандартный язык для визуализации, спецификации, конструирования и документирования программных систем. UML используется для создания диаграмм, которые помогают разработчикам и проектировщикам лучше понимать структуру и поведение системы.

### 3.2.1 Use-Case диаграмма

Диаграмма вариантов использования (англ. *use-case diagram*) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Основные элементы диаграммы – участник (*actor*) и прецедент (*use-case*).

Участник – это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Участником может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности. Графически участник изображается “человечком”.

Прецедент – описание множества последовательных событий (включая варианты), выполняемых системой, которые приводят к наблюдаемому участником результату. Прецедент представляет поведение сущности, описывая

взаимодействие между участниками и системой. Прецедент не показывает, “как” достигается некоторый результат, а только “что” именно выполняется. Прецеденты обозначаются очень простым образом – в виде эллипса, внутри которого указано его название.

Рассмотрим Use-Case диаграмму для «Онлайн-Столовки».

I. Слева от системы располагаются первичные актёры.

1) *Customer* (клиент) – это актёр, представляющий как новых, так и постоянных клиентов. Взаимодействует с системой для регистрации, входа в систему, поиска блюд, добавления их в корзину, оформления заказа, проведения оплаты, просмотра профиля, выхода из системы

2) *Admin* (администратор) – это актёр, с дополнительными возможностями, такими как обновление ассортимента блюд, изменение цен и другие административные функции.

New customer и Old customer, а также Admin унаследованы от Customer.

II. Справа от системы располагаются вторичные актёры.

1) *Bank* (банк) – это актёр, с которым взаимодействует система для проведения оплаты заказа.

2) *Server* (database) – это актёр, реагирующий на действия Customer, а также содержащий и обрабатывающий данные системы.

III. Посередине располагается бинарный актёр. Он может выполнять роль и вторичного и первичного актёра.

1C-Terminal/Кассирша – это актёр, выполняющий прецедент «*Set order status*» и реагирующий на «*Create order*».

Рассмотрим список прецедентов:

1) Register. Customer активирует данный прецедент для создания нового аккаунта и становления Old Customer;

2) Login (include VERIFY PASSWORD). Old Customer активирует данный прецедент для входа в систему, включая верификацию пароля. Реагирует Server (Database);

3) Search food. Old Customer активирует данный прецедент для поиска блюд в системе;

4) Add food item to basket. Old Customer активирует данный прецедент для добавления выбранных блюд в корзину;

5) Place order (include CHECK AVAILABILITY and CREATE ORDER) (extension: CANCEL ORDER or CHANGE ORDER). Old Customer активирует данный прецедент для оформления заказа, включая проверку доступности блюд, создание заказа. Реагирует Server (Database) и 1C Terminal;

6) Make payment (include CHECK SOLVENCY). Old Customer активирует данный прецедент для проведения оплаты заказа. Реагирует Bank;

7) Watch profile. Old Customer активирует данный прецедент для просмотра своего профиля;

8) Logout. Old Customer активирует данный прецедент для выхода из системы;

9) Set order status. 1C Terminal активирует данный прецедент для установки статуса заказа. Реагирует Server(Database).

Далее будет перечислено, какие актёры какие прецеденты выполняют, учитывая то, что первичные актёры активируют действие в системе, а вторичные реагируют на эти действия:

1) выполняет New Customer и становится Old Customer;

2) выполняет Old Customer, реагирует Server (Database);

3) выполняет Old Customer;

4) выполняет Old Customer;

5) выполняет Old Customer, реагирует Server (Database), реагирует 1C Terminal;

6) выполняет Old Customer, реагирует Bank;

7) выполняет Old Customer;

8) выполняет Old Customer;

9) выполняет 1C Terminal, реагирует Server (Database).

Саму Use-Case диаграмму можно рассмотреть на рисунке 1

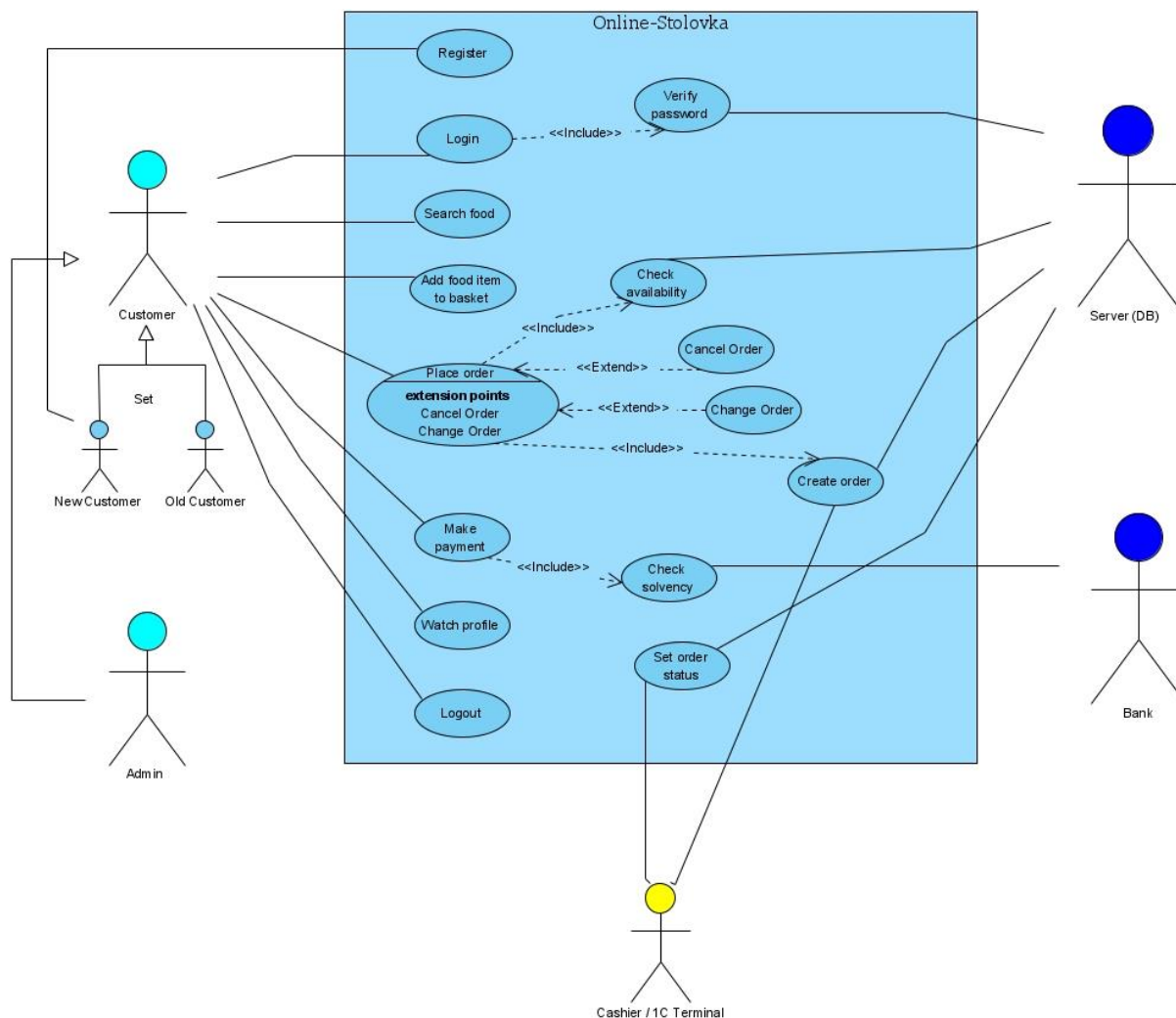


Рисунок 1 – Диаграмма Use-Case

### 3.2.2 ER-диаграмма

ER-диаграмма (Entity-Relationship diagram) является графическим инструментом моделирования данных в рамках ER-модели (Entity-Relationship model). ER-модель представляет собой абстрактную модель данных, используемую при концептуальном проектировании баз данных. Она позволяет описывать концептуальные схемы предметной области, выделяя ключевые сущности и связи между ними.

ER-диаграмма базы данных для «Онлайн-Столовки» была создана в CASE-системе (Computer-Aided Software Engineering) Visual Paradigm.

Выбор Visual Paradigm обусловлен рядом преимуществ. Этот инструмент предоставляет интуитивно понятный интерфейс, богатый функционал для создания различных видов диаграмм (UML, ERD, BPMN и др.), а также инструменты для коллаборации и командной работы. Visual Paradigm поддерживает методологии Agile и предоставляет средства для визуализации и анализа бизнес-процессов. Более того, данное приложение является платформонезависимым, что обеспечивает гибкость и удобство в работе независимо от операционной системы.

Рассмотрим, 8 таблиц, которые входят в структуру базы данных.

1-я таблица – это Customers (Заказчики).

Таблица Customers хранит информацию о зарегистрированных клиентах с уникальным идентификатором Customer\_ID.

Она связана с таблицей Orders по полю Customer\_ID, позволяя отслеживать заказы клиента.

2-я таблица – это Products (Блюда).

Таблица Products хранит информацию о всех блюдах, имеющих в меню столовой, предоставляя Product\_ID в качестве уникального идентификатора блюда.

Она связана с таблицей Order\_details по полю Product\_ID, определяя количество и цену каждого блюда в заказе.

3-я таблица – это Diet (Диеты).

Таблица Diet содержит информацию о различных диетах, предоставляя Diet\_ID в качестве уникального идентификатора диеты.

Она связана с таблицей Products по полю Diet\_ID, позволяя привязывать блюда к определённым диетам.

4-я таблица – это TimeVarying\_prices (Цены, изменяющиеся со временем).

Таблица TimeVarying\_prices содержит информацию о ценах на все блюда, а также информацию об изменении цен на все блюда с течением времени.

Она связана с таблицей Products по полю Product\_ID, позволяя обновлять в последней актуальные цены на продукты.

5-я таблица – это Schedule (Расписание).

Таблица Schedule представляет собой расписание на чётную и нечётную неделю.

Она связана с таблицей Available\_dishes по полю Schedule\_ID, указывая, какие блюда доступны в соответствие с расписанием.

6-я таблица – это Available\_dishes (Актуальные блюда).

Таблица Available\_dishes представляет собой список доступных блюд на основе действующего расписания.

Она связана с таблицей Products по полю Product\_ID, определяя доступные блюда в соответствии с расписанием.

7-я таблица – это Orders (Заказы).

Таблица Orders хранит информацию о заказах с уникальным идентификатором Order\_ID.

Она связана с таблицей Order\_details по полю Order\_ID, позволяя эффективно ассоциировать записи последней с соответствующими заказами и устанавливая, какие блюда были заказаны в каждом конкретном заказе.

8-я таблица – это Order\_details (Детали заказов).

Таблица Order\_details хранит информацию о деталях каждого заказа, включая количество и цену каждого заказанного блюда.

Вышеперечисленные таблицы окрашены на ER-диаграмме в абрикосовый цвет.

Помимо них существуют ещё 3 объекта, окрашенные в лимонный цвет – Titles (Звания), Categories (Категории), Days\_of\_week (Дни недели). Они являются перечислениями.

Изначально предполагалось включить их в виде таблиц, однако приняв во внимание то, что каждая из них содержит всего лишь одно поле, не считая поле-

идентификатор, и обладает ограниченным числом записей, было принято решение оптимизировать структуру, представив их в виде перечислений.

Рассмотреть ER-диаграмму в её графической реализации можно на рисунке 2.

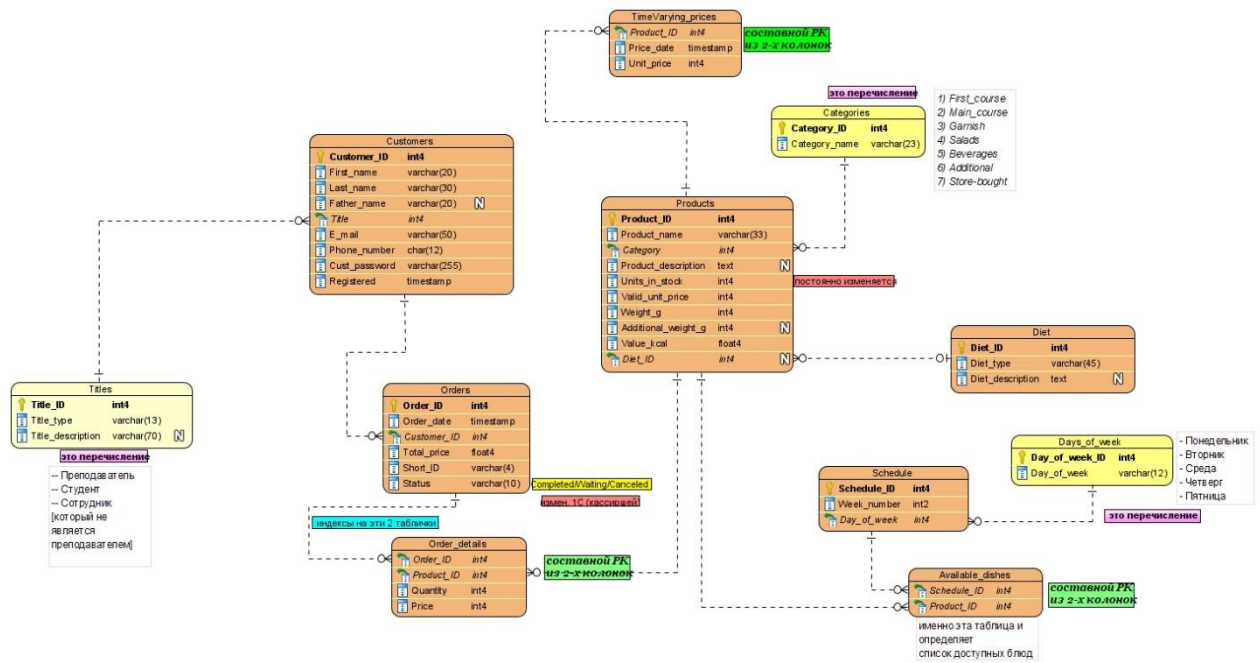


Рисунок 2 – ER-диаграмма базы данных для «Онлайн-Столовки»

## 4 Конструирование интерфейса приложения

### 4.1 Обзор рабочей области Figma

Разработка интерфейса мобильного приложения является одним из ключевых этапов проектирования. В данном разделе будут рассмотрены основные аспекты создания интерфейса для мобильного приложения "Онлайн-Столовка" с использованием Figma.

На рисунке 3 представлен главный рабочий экран проекта в Figma, где ведётся работа над ним.



Рисунок 3 – Главный рабочий экран проекта в Figma

Здесь отображаются все макеты, сгруппированные по функциональным элементам.

Главный экран разделен на логические блоки для облегчения навигации и редактирования:

1-ым логическим блоком является рабочая область. Центральная часть экрана представляет собой рабочую область, где располагаются макеты;



2-ым логическим блоком является меню слоев и активов. Левое меню включает в себя список всех макетов (*Layers*) и активов (*Assets*), что позволяет настраивать и управлять компонентами интерфейса.

В *Layers* находятся все макеты, представленные в виде списка. Этот список пользователь сам может настраивать, перетаскивать их, менять расположение. Также пользователь может "раскрыть" макет и посмотреть, из каких элементов он состоит;

3-им логическим блоком является меню настроек. Правое меню предназначено для изменения параметров выбранных элементов, таких как положение, эффекты, цвета, функции и связи;

Пользователь выбирает определенный макет или элемент и в правом меню может изменить различные настройки. положение, эффекты, цвета, функции, действия, связи.

## **4.2 Макеты экранов приложения**

1-ым макетом является загрузочный экран.

Загрузочный экран приветствует пользователей надписью "Столовка". Под надписью расположен логотип университета НИТУ "МИСиС" и текст "Загрузка...". Этот экран отображается при запуске приложения, создавая первое впечатление у пользователей.

Загрузочный экран изображён на рисунке 4.



Рисунок 4 – Загрузочный экран моб. приложения «Онлайн-Столовка»

2-ым макетом является экран регистрации.

На экране регистрации пользователи видят надпись "Регистрация". Здесь предусмотрены два поля ввода: для e-mail или номера телефона и пароля. Также есть две кнопки:

А) Кнопка подтверждения (черная галочка в зеленом кружочке)  
При нажатии перенаправляет пользователя на следующий макет "Код подтверждения".

Б) Кнопка "У меня уже есть аккаунт. Войти"  
При нажатии перенаправляет пользователя на экран авторизации.

Экран регистрации изображён на рисунке 5.

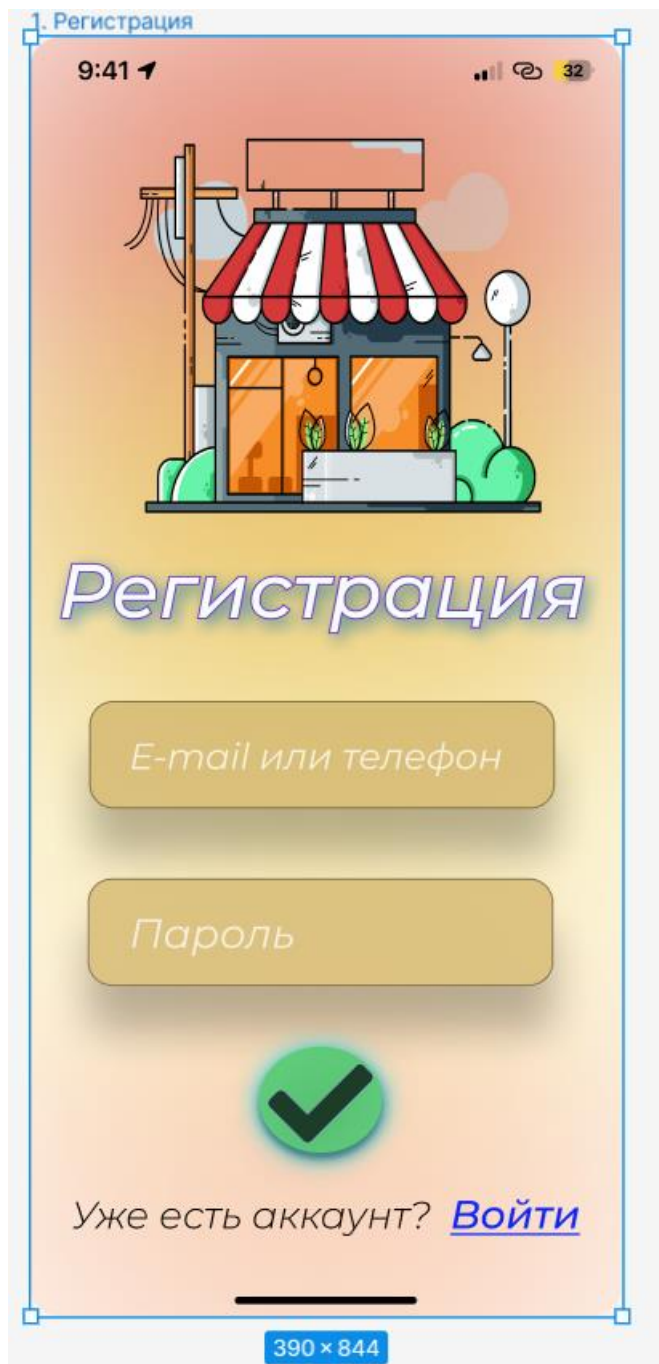


Рисунок 5 – Экран регистрации моб. приложения «Онлайн-Столовка»

3-им макетом является экран ввода кода подтверждения.

На экране ввода кода подтверждения отображается надпись "Введите код". Пользователи вводят код, полученный по e-mail или SMS, в специальное поле ввода. Этот код необходим для завершения процесса регистрации.

Экран ввода кода подтверждения изображён на рисунке 6.



Рисунок 6 – Экран ввода кода подтверждения моб. приложения «Онлайн-Столовка»

4-ым макетом является экран авторизации.

Если пользователь экране регистрации нажал на 2-ую кнопку "У меня уже есть аккаунт. Войти", то его перекидывает именно сюда.

Экран авторизации предназначен для пользователей, у которых уже есть аккаунт. На этом экране аналогично экрану регистрации имеются два поля ввода для e-mail или номера телефона и пароля, а также две кнопки:

А) Кнопка подтверждения, для входа в аккаунт.

Б) Кнопка отмены, которая возвращает пользователя на экран регистрации.

Экран авторизации изображён на рисунке 7.

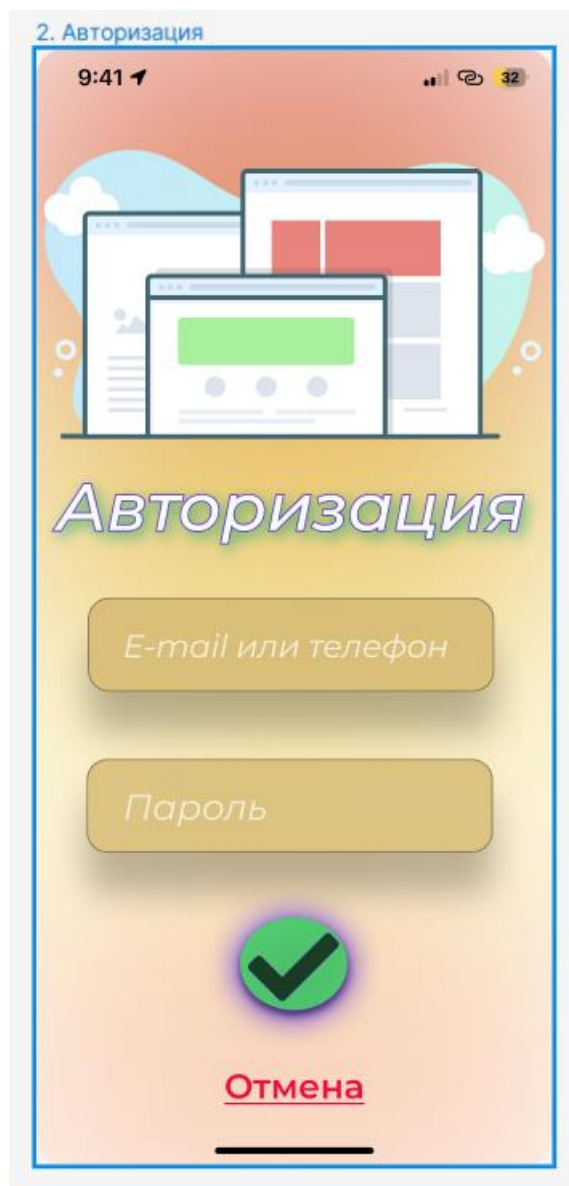


Рисунок 7 – Экран авторизации моб. приложения «Онлайн-Столовка»

5-ым макетом является главное меню.

Главное меню приложения содержит следующие элементы:

- А) Надпись «Столовка» с иконкой «гамбургер-меню» справа;
- Б) Поле поиска для удобного поиска блюд;
- В) Горизонтальный список категорий (например, "1 блюда", «2 блюда", "напитки", "дополнительное"), который можно прокручивать;
- Г) Список блюд с возможностью нажать на каждое для получения детальной информации.

Главное меню изображено на рисунке 8.



Рисунок 8 – Главное меню моб. приложения «Онлайн-Столовка»

6-ым макетом является так называемое «гамбургер-меню».

При нажатии на «гамбургер-меню» справа всплывает боковое меню, содержащее аватар пользователя и разделы: "Профиль", "Заказы", "О приложении", "Закреть".

«Гамбургер-меню» изображено на рисунке 9.



Рисунок 9 – «Гамбургер-меню» моб. приложения «Онлайн-Столовка»

## 5 Тестирование базы данных мобильного приложения

Как было сказано ранее, Python обладает богатой библиотекой, включая библиотеки для работы с сетями, веб-разработки и базами данных. В контексте создания клиентского приложения для работы с базой данных PostgreSQL, Python предоставляет удобные библиотеки для подключения к базе данных и выполнения SQL-запросов. Модуль `psycopg2`, например, позволяет эффективно взаимодействовать с PostgreSQL из кода на Python, делая его отличным выбором для разработки клиентской части приложения. Python также известен своей простотой использования, что упрощает процесс разработки и поддержки клиентского приложения.

Клиентское приложение “Online-Stolovka” разработано на языке программирования Python, используя библиотеку `tkinter` для создания графического пользовательского интерфейса.

Автоматическое подключение к базе данных, размещенной на виртуальной машине с ОС Ubuntu, позволяет пользователям взаимодействовать с приложением без необходимости ручного ввода данных подключения.

При успешном подключении отображается главное окно приложения, предоставляя пользователю 3 возможности.

1-ой возможностью является просмотр списка продуктов.

Пользователь может просматривать все блюда сразу или выбирать категории, например, 1-ое блюдо, 2-ое блюдо, гарнир и т.д.

2-ой возможностью является добавление блюд в корзину.

Пользователь может добавлять блюда в корзину с возможностью указания количества. Система также обрабатывает добавление уже существующих продуктов, позволяя пользователям легко управлять корзиной.

3-ей возможностью является оформление заказа.

После добавления желаемых блюд в корзину, пользователь может разместить заказ, нажав кнопку "Place Order". Информация о заказе



отображается во всплывающем окне, а также записывается в базу данных. Создается запись в таблице Orders, а также соответствующие записи в таблице Order\_details.

В дополнение к текущей функциональности, предоставленной клиентским приложением, в планах внедрение системы регистрации. Клиентское приложение в будущем использоваться не будет. Оно было разработано для тестирования функционала базы данных и обеспечения предварительного понимания структуры классов и методов для последующей разработки бэкенда под мобильное приложение.

Результаты тестирования приложения приведены на рисунке 10.

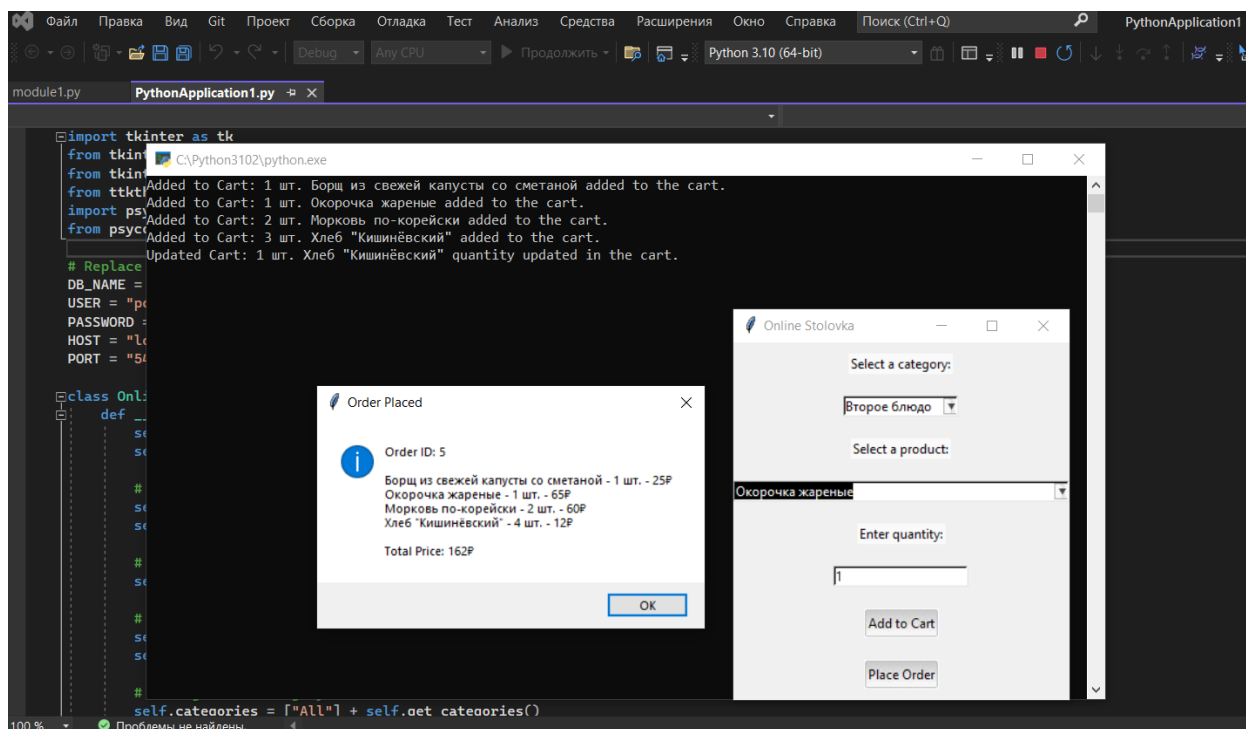


Рисунок 10 – Тестирование базы данных «Онлайн-Столовки» с помощью клиентского приложения

## Заключение

В данной курсовой работе была проведена разработка и анализ прототипа мобильного приложения "Онлайн-Столовка" для оптимизации процесса заказа и получения питания в столовой НФ НИТУ «МИСиС». Основные этапы работы включали описание предметной области, постановку задачи и разработку требований, проектирование системы и конструирование интерфейса приложения, а также тестирование базы данных.

Подведем итоги по каждому из разделов.

В 1-ом разделе была рассмотрена предметная область мобильного приложения "Онлайн-Столовка", охватывающая сферу общественного питания в университете. Главные цели приложения включают оптимизацию процесса заказа и получения питания, сокращение времени ожидания в очереди и повышение комфорта студентов и преподавателей. Также были проанализированы существующие системы-аналоги, такие как приложения "Burger King", "KFC" и "Додо Пицца", для выявления лучших практик и уникальных особенностей "Онлайн-Столовки".

Во 2-ом разделе были сформулированы основные задачи проекта: создание системы для удобного заказа и оплаты обедов, отслеживания статуса заказа и уведомления пользователей о готовности блюд. Были определены функциональные, эргономические и технические требования к программному обеспечению. Также была выбрана методология разработки – итеративная и инкрементальная модель, что позволило гибко управлять проектом и быстро реагировать на изменения.

В 3-ем разделе было обоснование выбора языка программирования и средств разработки. Был выбран фреймворк Flutter с языком программирования Dart за его кроссплатформенные возможности и преимущества перед другими фреймворками. Для серверной части и тестирования базы данных использовался Python, который обладает простотой и мощной экосистемой. База данных была

реализована на PostgreSQL, что обеспечило надежность и масштабируемость. Инструмент Figma использовался для создания дизайна интерфейса приложения, благодаря своим мощным возможностям и интеграции с другими инструментами разработки.

В 4-ом разделе были рассмотрены основные макеты, созданные в Figma, включая загрузочный экран, экраны регистрации и авторизации, главное меню и другие элементы интерфейса. Были описаны функциональные возможности и навигация между различными экранами приложения. Несмотря на то, что не все макеты были завершены, работа продемонстрировала ключевые элементы и пользовательский путь в приложении.

В заключение можно сделать следующий вывод.

В ходе выполнения курсовой работы была успешно разработана концепция мобильного приложения "Онлайн-Столовка", направленного на улучшение процесса заказа и получения питания в университетской столовой. Были рассмотрены предметная область, поставлены задачи, разработаны требования и выбраны соответствующие инструменты и методология разработки. Проектирование системы и создание интерфейса позволили продемонстрировать основные функциональные возможности и взаимодействие пользователей с приложением.

В результате проведенной работы можно сделать вывод, что предложенное мобильное приложение "Онлайн-Столовка" имеет потенциал значительно улучшить удобство и эффективность обслуживания в университетской столовой, снизив время ожидания в очередях и повысив удовлетворенность студентов и преподавателей. Внедрение данного решения будет способствовать улучшению общего опыта питания в университете и может быть успешно адаптировано для использования в других учебных заведениях.

## Список использованных источников

1. Азарова, И.Ю., Гришко, Е.А. Разработка и оптимизация мобильных приложений / И.Ю. Азарова, Е.А. Гришко. – Москва: Наука, 2019. – 456 с. – ISBN 978-5-02-041672-7.
2. Боярский, М.В. Программирование на Dart: от основ до профессионального уровня / М.В. Боярский. – Санкт-Петербург: Питер, 2021. – 384 с. – ISBN 978-5-4461-1735-3.
3. Веселов, А.А. Методология разработки программного обеспечения / А.А. Веселов. – Москва: ДМК Пресс, 2018. – 296 с. – ISBN 978-5-94074-856-5.
4. Вишняков, К.П. Базы данных: проектирование и использование / К.П. Вишняков, И.Н. Поляков. – Санкт-Петербург: Питер, 2020. – 512 с. – ISBN 978-5-4461-0931-0.
5. Джексон, М. Создание мобильных приложений с Flutter и Dart / М. Джексон. – Киев: Диалектика, 2022. – 544 с. – ISBN 978-966-945-121-9.
6. Захаров, А.В. Объектно-ориентированное программирование с Python / А.В. Захаров. – Москва: МЦНМО, 2020. – 432 с. – ISBN 978-5-4439-1198-2.
7. Карпов, А.Н. Базы данных: проектирование, реализация, сопровождение / А.Н. Карпов. – Москва: Издательский дом "Вильямс", 2009. – 384 с. – ISBN 978-5-8459-1337-3.
8. Клементьев, В.Н. Разработка мобильных приложений для iOS и Android / В.Н. Клементьев. – Санкт-Петербург: БХВ-Петербург, 2021. – 416 с. – ISBN 978-5-9775-5010-5.
9. Ложкин, И.В. Теория баз данных: учебное пособие / И.В. Ложкин. – Санкт-Петербург: Питер, 2016. – 368 с. – ISBN 978-5-496-01397-7.

10. Милов, С.Д. Разработка программного обеспечения: методы и подходы / С.Д. Милов. – Москва: Академия, 2019. – 312 с. – ISBN 978-5-4461-0931-1.
11. Официальная документация Flutter [Электронный ресурс]. – URL: <https://docs.flutter.dev/> (дата обращения: 15.04.2024).
12. Официальная документация PostgreSQL [Электронный ресурс]. – URL: <https://www.postgresql.org/docs/> (дата обращения: 15.04.2024).
13. Официальная документация Python [Электронный ресурс]. – URL: <https://docs.python.org/3/> (дата обращения: 15.04.2024).
14. Романенко, Е.А. UML: язык моделирования систем / Е.А. Романенко. – Москва: Инфра-М, 2018. – 240 с. – ISBN 978-5-16-005992-8.
15. Фомин, Д.С. Создание дизайна интерфейсов с Figma / Д.С. Фомин. – Москва: Альпина Пабlishер, 2021. – 368 с. – ISBN 978-5-9614-2187-3.