

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки – 09.03.02 Информационные системы и технологии
Направленность (профиль) образовательной программы Информационные системы и технологии

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«_____» _____ 2025г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Меню PRO (программное обеспечение для заведений общественного питания). В форме «Стартап как диплом»

Исполнитель студент группы 1104–об	_____	К.Р. Жбанников
	(подпись, дата)	
Руководитель доцент, канд. физ–мат. наук	_____	В.В. Ерёмина
	(подпись, дата)	
Консультант по безопасности и экологичности доцент, канд. техн. наук	_____	А.Б. Булгаков
	(подпись, дата)	
Нормоконтроль инженер кафедры	_____	В.Н. Адаменко
	(подпись, дата)	

Благовещенск 2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
И.о. зав. кафедрой
_____ А.В. Бушманов
«_____» _____ 2025 г.

З А Д А Н И Е

К выпускной квалификационной работе студента: Жбанников Кирилл Романович

1. Тема выпускной квалификационной работы: Меню PRO (программное обеспечение для заведений общественного питания). В форме «Стартап как диплом»

(утверждена приказом от 14 апреля 2025 № 980-уч)

2. Срок сдачи студентом законченной работы: 10.06.2025

3. Исходные данные к бакалаврской работе: формулировка темы ВКР, литературные источники.

4. Содержание бакалаврской работы (перечень подлежащих разработке вопросов): анализ предметной области; анализ рынка; расчёт финансовых показателей; прототипирование системы; разработка системы; тестирование системы.

5. Консультанты по выпускной квалификационной работе: по безопасности и экологичности – Булгаков А. Б., доцент, канд. техн. наук

6. Дата выдачи задания: 03.09.2024

Руководитель бакалаврской работы: Ерёмина Виктория Владимировна, доцент, канд. физ-мат. наук

Задание принял к исполнению (03.09.2024): _____ Жбанников К.Р.

РЕФЕРАТ

Бакалаврская работа содержит 91 страница, 10 рисунков, 32 таблицы, 6 приложений, 20 источников

NFC–ТЕХНОЛОГИИ, ИНТЕРАКТИВНОЕ МЕНЮ, ПРЕДПРИЯТИЯ ОБЩЕСТВЕННОГО ПИТАНИЯ, ЦИФРОВИЗАЦИЯ РЕСТОРАННОГО БИЗНЕСА.

Объектом исследования данной работы является процесс автоматизации взаимодействия клиентов с меню в предприятиях общественного питания с использованием NFC–технологий.

Цель работы – разработка мобильного приложения для системы интерактивного меню, обеспечивающей бесконтактное взаимодействие через NFC–метки и интеграцию с платёжными системами.

Задачами работы являются:

- анализ предметной области и существующих решений в сфере цифровизации ресторанного бизнеса;
- обоснование выбора технологий к разработке мобильного приложения;
- разработка полноценного функционала мобильного приложения.

В работе рассмотрена проблема автоматизации процессов взаимодействия клиентов с меню в ресторанах и кафе. Предложено решение на основе разработки мобильного приложения, которое использует NFC–технологии для доступа к интерактивному меню и оформления заказов. Разработана архитектура системы, реализовано взаимодействие с NFC–устройствами. Результаты работы могут быть применены в практике предприятий общественного питания.

НОРМАТИВНЫЕ ССЫЛКИ

Приказ Министерства труда и социальной защиты Российской Федерации от 29 октября 2021 г. N 774н «Об утверждении общих требований к организации безопасного рабочего места»

СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»

СанПиН 2.1.3684-21 «Санитарно-эпидемиологические требования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению населения, атмосферному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противоэпидемических (профилактических) мероприятий»

СТО СМК 4.2.3.21-2018 Оформление выпускных квалификационных и курсовых работ (проектов)

ГОСТ Р 50948-2001 Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности

ГОСТ Р 52872-2019 Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложение для стационарных и мобильных устройств, иные пользовательские интерфейсы

ГОСТ Р 58751-2019 Слаботочные системы. Кабельные системы. Телекоммуникационные пространства и помещения. Рабочее место

ГОСТ Р ИСО 9241-161-2016 Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

NFC (Near Field Communication) – технология ближней бесконтактной связи, позволяющая обмениваться данными между устройствами на расстоянии до 10 см. В данном проекте используется для мгновенного доступа к цифровому меню через сканирование меток, установленных на столах.

API (Application Programming Interface) – программный интерфейс, обеспечивающий взаимодействие между компонентами системы. В проекте используется для связи мобильного приложения с облачной базой данных (Firebase), а также интеграции платежных систем (Сбер Pay, Apple Pay).

БД (база данных) – структурированное хранилище информации. В системе используются облачная (Firebase Cloud Firestore) БД для хранения данных о меню, заказах, пользователях и статусах оплаты.

ПО (программное обеспечение) – совокупность программ, обеспечивающих работу системы. В проекте включает клиентскую часть (Flutter–приложение), серверную логику (Firebase Functions), а также инструменты администрирования (админ–панель).

RESTful API – тип веб–API, основанный на протоколе HTTP и архитектурном стиле REST. Используется при интеграции внешних сервисов, таких как Telegram API для передачи заказов в чат заведения или сторонние платёжные шлюзы.

Web NFC API – интерфейс веб–приложений для работы с NFC–устройствами. Рассматривался как альтернатива, но не был выбран из–за ограниченной поддержки на iOS и более высоких ограничений по безопасности и скорости реакции по сравнению с нативными решениями.

СОДЕРЖАНИЕ

Введение	8
1 Анализ предметной области	11
1.1 Общая характеристика предметной области	11
1.2 Анализ существующих решений	13
1.3 Формулировка проблемы и цели проекта	16
1.4 Бизнес–модель проекта	18
1.5 Финансовый план	22
1.6 Потенциал масштабирования проекта	25
2 Проектирование программного обеспечения	31
2.1 Цели и задачи проектирования	31
2.2 Состав функциональных и обеспечивающих систем	34
2.3 Выбор инструментов реализации проекта	39
2.4 Проектирование архитектуры системы	40
2.5 Проектирование базы данных	46
2.6 Макетирование пользовательского интерфейса	53
2.7 Оценка качества проектных решений	60
3 Разработка программного обеспечения	62
3.1 Общие сведения	62
3.2 Функциональное назначение	63
3.3 Тестирование и отладка программного обеспечения	65
4 Безопасность и экологичность	68
4.1 Безопасность	68
4.2 Экологичность	70
4.3 Чрезвычайные ситуации	72
Заключение	78
Библиографические ссылки	79
Библиографический список	80
Приложение А	82

Приложение Б	83
Приложение В	84
Приложение Г	85
Приложение Д	86
Приложение Е	90

ВВЕДЕНИЕ

Современные информационные технологии играют ключевую роль в цифровой трансформации различных отраслей экономики, включая сферу общественного питания. Растущие требования к скорости и качеству обслуживания клиентов, а также стремление предприятий к оптимизации операционных процессов стимулируют внедрение инновационных решений. Одним из таких направлений является разработка систем интерактивного меню, позволяющих автоматизировать взаимодействие между клиентом и заведением.

Особый интерес представляет использование NFC-технологий (Near Field Communication), которые обеспечивают бесконтактное взаимодействие через простое прикосновение смартфона к NFC-метке. Это позволяет клиентам быстро получать доступ к цифровому меню, просматривать блюда, оформлять заказы и даже производить оплату без необходимости использования дополнительных приложений или навигации по сложным интерфейсам. В мобильных приложениях NFC-функционал реализуется через специализированные библиотеки (например, `nfc_manager` или `flutter_nfc_kit`), которые обеспечивают интеграцию с аппаратными модулями Android и iOS.

Актуальность темы обусловлена необходимостью повышения эффективности работы ресторанов и кафе за счет снижения времени на обслуживание, минимизации человеческих ошибок, улучшения пользовательского опыта и обеспечения гибкой аналитики продаж. Особенно актуально внедрение подобных решений в условиях пандемии и после нее, когда важное значение имеет минимизация личного контакта между персоналом и клиентами. Мобильные приложения на базе Flutter обеспечивают более высокую производительность, автономную работу в офлайн-режиме и глубокую интеграцию с аппаратными возможностями устройств.

Цель выпускной квалификационной работы – разработка мобильного приложения для системы интерактивного меню с использованием NFC–технологий для предприятий общественного питания на платформе Flutter, которая позволит автоматизировать процессы взаимодействия клиентов с меню, повысить скорость и качество обслуживания, а также предоставить возможность сбора аналитических данных.

Для достижения указанной цели были поставлены следующие задачи:

- провести анализ предметной области и существующих решений в сфере цифровизации ресторанного бизнеса;
- обосновать выбор Flutter как основного фреймворка для разработки, а также библиотек для работы с NFC;
- спроектировать архитектуру мобильного приложения с использованием паттернов (например, BLoC, Provider) и диаграмм UML;
- разработать прототип и реализовать функционал мобильного приложения с интеграцией NFC;
- реализовать модуль взаимодействия с NFC–устройствами, включая чтение/запись меток и обработку NDEF–данных;
- описать пользовательский интерфейс, адаптированный под мобильные экраны, и алгоритмы работы системы;
- проанализировать экономическую целесообразность внедрения мобильного приложения;
- рассмотреть вопросы информационной безопасности (шифрование данных, защита от несанкционированного доступа к NFC–меткам) и удобства использования.

Предметом исследования являются методы и технологии проектирования и разработки мобильных приложений, основанных на применении NFC–технологий для автоматизации взаимодействия клиентов с меню ресторанов и кафе. Объектом исследования выступает процесс обслуживания клиентов в

предприятиях общественного питания с учётом особенностей мобильных решений.

Теоретическая значимость заключается в систематизации современных подходов к созданию интерактивных сервисов в сфере общепита на основе мобильных технологий и обосновании применения Flutter и NFC–технологий как одного из наиболее перспективных способов организации клиентской части сервиса.

Практическая ценность работы состоит в том, что созданное мобильное приложение может быть внедрено в реальных условиях работы ресторанов и кафе, что позволит повысить уровень автоматизации, снизить нагрузку на персонал, улучшить пользовательский опыт и обеспечить гибкую аналитику через интеграцию с облачными сервисами.

Структура работы включает введение, три основные главы, заключение, список литературы и приложения. Каждая глава посвящена отдельному этапу выполнения проекта: анализ предметной области, проектирование и разработка программного обеспечения.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Общая характеристика предметной области

В условиях стремительного развития цифровых технологий и увеличения потребности бизнеса в автоматизации процессов, сфера общественного питания также подвергается значительным изменениям. Современные клиенты ожидают высокого уровня сервиса, удобства и скорости обслуживания. Это стимулирует владельцев ресторанов, кафе и фуд-кортв внедрять инновационные решения, направленные на оптимизацию взаимодействия с посетителями.

Одним из ключевых направлений цифровой трансформации в ресторанном бизнесе является автоматизация процесса заказа блюд и напитков. Традиционные бумажные меню уступают место электронным решениям, позволяющим не только ознакомиться с ассортиментом, но и сделать заказ, оплатить его, получить информацию о составе блюд, акциях и бонусах – всё это без участия персонала.

На сегодняшний день существует множество решений, реализующих функционал интерактивного меню: от простых веб-страниц до полноценных мобильных приложений с расширенными возможностями. Однако большинство таких продуктов требуют предварительной установки приложения или активного взаимодействия с QR-кодами, что может быть не всегда удобно для пользователя. Также зачастую отсутствует единая система, объединяющая административную часть и клиентскую, что усложняет управление и синхронизацию данных между сторонами.

Актуальным становится использование технологии NFC (Near Field Communication) – беспроводной связи короткого радиуса действия, которая позволяет быстро и безопасно передавать данные между устройствами. С помощью NFC-меток можно организовать моментальный доступ к интерактив-

ному меню без необходимости сканирования кодов или поиска сайта в интернете. Это делает процесс взаимодействия с системой максимально простым и интуитивно понятным для конечного пользователя.

Кроме того, переход от бумажных меню к электронным решениям имеет важный экологический аспект. Снижение использования бумаги и печатной продукции способствует уменьшению антропогенного воздействия на окружающую среду. Это особенно важно в условиях роста экологической ответственности бизнеса и увеличения спроса на «зелёные» решения среди потребителей.

Важно отметить и социальный аспект внедрения таких систем. Использование современных технологий обеспечивает более широкую доступность услуг для людей с ограниченными возможностями. Например, поддержка масштабирования текста, совместимость с экранами-ридерами и возможность клавиатурного управления позволяют использовать систему людям с нарушением зрения или моторных функций.

Также нельзя игнорировать вопросы безопасности и надежности внедряемых решений. Внедрение протоколов шифрования данных, механизмов аутентификации, резервного копирования и отказоустойчивой архитектуры позволяет минимизировать риски потери информации и несанкционированного доступа. Такие меры обеспечивают защиту как личных данных пользователей, так и конфиденциальной информации заведения.

На фоне роста интереса к автоматизации процессов обслуживания, такие решения становятся не просто технической новинкой, но и элементом стратегии развития заведений. Они позволяют:

- повысить скорость и точность оформления заказов;
- снизить нагрузку на персонал;
- упростить администрирование и аналитику;
- укрепить имидж заведения как технологически продвинутого и клиентоориентированного.

1.2 Анализ существующих решений

В последние годы сфера общественного питания активно внедряет цифровые технологии для повышения эффективности работы заведений и улучшения клиентского опыта. Одним из ключевых направлений является автоматизация взаимодействия клиента с меню. Современные системы интерактивного меню могут быть реализованы в различных форматах: QR-коды, мобильные приложения, веб-интерфейсы, а также решения на основе NFC. Наиболее популярными являются:

- QR-коды: простое решение, позволяющее открыть меню через камеру смартфона. Широко используется благодаря минимальным затратам на внедрение;
- мобильные приложения: предоставляют расширенный функционал – просмотр меню, заказ, оплата, история покупок. Требуют установки, но обеспечивают высокий уровень персонализации.
- веб-приложения: доступны через браузер, не требуют установки. Обеспечивают кросс-платформенность и удобство обновления;
- NFC-решения: позволяют получить доступ к меню через простое прикосновение смартфона к метке. Это наиболее быстрый и интуитивно понятный способ взаимодействия, однако его внедрение пока ограничено из-за зависимости от поддержки NFC в устройствах.

Каждая из указанных технологий имеет свои преимущества и недостатки, что делает выбор оптимального решения зависимым от специфики заведения и целевой аудитории.

Для более объективной оценки были выделены ключевые критерии сравнения: функциональность, используемая технология, пользовательский опыт, безопасность данных и простота внедрения. Результаты анализа приведены в таблице 1.

Из таблицы видно, что NFC-решения демонстрируют высокий уровень

пользовательского опыта и безопасности, однако их внедрение связано с необходимостью установки меток и обеспечения совместимости с устройствами. В то же время веб и мобильные приложения позволяют реализовать полный цикл взаимодействия – от просмотра меню до оформления заказа и оплаты, но требуют отдельной разработки и синхронизации.

Таблица 1 – Сравнительный анализ технологий

Критерий	QR-коды	Мобильное приложение	Веб-приложение	NFC-решение
Функциональность	Ограниченная	Высокая	Средняя	Средняя
Технология	Универсальная	Зависит от платформы	Универсальная	Ограниченная
Пользовательский опыт	Средний	Высокий	Средний	Очень высокий
Безопасность	Средняя	Высокая	Средняя	Высокая
Простота внедрения	Высокая	Низкая	Средняя	Средняя

Несмотря на наличие множества решений, рынок всё ещё сталкивается с рядом проблем:

- фрагментация между клиентской и административной частями: большинство продуктов предлагают либо удобный интерфейс для пользователя, либо мощную панель управления, но редко сочетают оба аспекта в единой системе;
- ограниченная аналитика: не все решения предоставляют данные о поведении клиентов, популярных блюдах или времени заказа;
- отсутствие масштабируемости: многие продукты не предусматривают возможность адаптации под различные форматы заведений;
- проблемы с доступностью: редкие решения учитывают потребности

людей с ограниченными возможностями, такие как поддержка экранного доступа или клавиатурная навигация.

Эти пробелы открывают возможности для создания комплексного решения, которое будет:

- объединять веб и мобильную части через единое API;
- использовать NFC–технологию для мгновенного доступа к меню;
- обеспечивать гибкий интерфейс администрирования;
- учитывать принципы доступности и юзабилити;
- предоставлять аналитические данные для владельцев заведений.

Для обеспечения эффективной работы системы важно правильно организовать интеграцию веб и мобильной частей. Основные принципы:

- единое API: использование RESTful API позволяет обеспечить единый канал обмена данными между компонентами;
- общая база данных: MySQL в сочетании с ORM RedBeanPHP обеспечивает согласованность данных между веб и мобильной частями;
- механизмы синхронизации: данные о заказах, статусах и изменениях должны корректно отражаться в обеих частях системы;
- кроссплатформенная совместимость: интерфейс должен быть адаптирован как под мобильные устройства, так и под десктоп.

Анализ современного рынка показал, что цифровизация ресторанной индустрии активно развивается, однако большинство доступных решений имеют ряд ограничений, связанных с фрагментацией между клиентской и административной частями, отсутствием поддержки NFC и ограниченной масштабируемостью. Это создаёт возможность для внедрения нового подхода, сочетающего преимущества нескольких технологий и обеспечивающего комплексное решение задач заведений общественного питания.

В рамках данного исследования были рассмотрены как коммерческие, так и продукты с открытым исходным кодом, применяемые в сфере интерак-

тивного меню. Оценка проводилась по следующим критериям: функциональность, тип используемой технологии, удобство взаимодействия с пользователем, уровень безопасности и простота внедрения.

В таблице 2 приведены конкретные примеры решений, которые были проанализированы в ходе подготовки проекта. Эти системы представляют разные подходы к организации цифрового меню и имеют свои сильные и слабые стороны.

Таблица 2 – Существующие решения

Название	Тип решения	Особенности
TouchBistro	Мобильное приложение	Платформа для ресторанов, предоставляющая полный цикл заказа и управления через мобильное приложение. Интеграция с POS–системами.
Toast POS	Веб + мобильная система	Гибридная система с мощной административной панелью и мобильным клиентом. Поддерживает онлайн–заказы и доставку.
Zebra Ziosk	Клиентский планшет	Система, установленная на столах в виде стационарных планшетов. Обеспечивает моментальный доступ к меню и заказу.
TableCheck	NFC/QR–меню	Японская платформа, поддерживающая NFC и QR–коды. Используется в сетевых кафе и ресторанах.

1.3 Формулировка проблемы и цели проекта

Целью данного проекта является разработка клиентской части комплексного решения для организации интерактивного меню на основе технологии NFC, включающей как веб–приложение, так и мобильное приложение. Основной задачей стало обеспечение бесконтактного, быстрого и интуитивно по-

нятного способа взаимодействия клиента с меню заведения, а также реализация возможности оформления заказа без участия официанта.

Для достижения указанной цели были определены следующие задачи:

- разработать клиентское веб–приложение, доступное через браузер смартфона и активируемое с помощью NFC–метки. При этом оно должно быть полностью адаптировано под мобильные устройства, обеспечивать высокую скорость загрузки и простой интерфейс навигации;
- создать мобильное приложение, предоставляющее аналогичный функционал, но с возможностью хранения истории заказов, персонализированных рекомендаций и других функций, доступных после регистрации пользователя;
- обеспечить единое информационное пространство для обоих клиентских приложений. Для этого была реализована система API, позволяющая обмениваться данными между веб– и мобильной частями, сохраняя актуальность информации в реальном времени;
- протестировать оба клиента на совместимость с различными устройствами и операционными системами, проверить работу NFC–модуля, а также протестировать ограничения по геолокации, которые позволяют использовать систему только внутри заведения;
- уделить внимание вопросам безопасности передачи данных. Все данные между клиентскими приложениями и сервером передаются по защищённому каналу HTTPS, обеспечивается защита от несанкционированного доступа и проверяется целостность информации;
- оценить удобство использования интерфейса и провести юзабилити–тестирование, чтобы удостовериться, что приложения соответствуют принципам доступности и могут использоваться широкой аудиторией, включая людей с ограниченными возможностями.

Внедрение такого подхода позволяет достичь ряда положительных эффектов. Во–первых, клиент получает возможность быстро и без лишних усилий ознакомиться с меню и сделать заказ, минимизируя время ожидания. Во–

вторых, заведение получает возможность снизить нагрузку на персонал, уменьшить вероятность ошибок при оформлении заказов и повысить общую прозрачность процесса обслуживания.

1.4 Бизнес–модель проекта

Разрабатываемый продукт представляет собой интерактивное цифровое меню, доступ к которому осуществляется через NFC–метку. Это позволяет посетителю заведения мгновенно получить доступ к меню, выбрать блюда и оформить заказ без участия официанта. При этом система состоит из двух клиентских частей: веб–приложения, которое активируется при сканировании метки, и мобильного приложения, предоставляющего дополнительные возможности, такие как история заказов и персонализированные рекомендации.

Для описания бизнес–модели проекта используется универсальный инструмент Business Model Canvas(Бизнес–модель Остервальдера), который позволяет структурировать ключевые аспекты стартапа в виде девяти блоков , расписанных в приложении Г.

С точки зрения бизнеса, продукт можно рассматривать как цифровое решение, ориентированное на автоматизацию процесса взаимодействия клиента с рестораном или кафе. Основная ценность заключается в том, чтобы минимизировать время между ознакомлением с меню и оформлением заказа, повысить скорость обслуживания и уменьшить нагрузку на персонал. Это достигается благодаря отсутствию необходимости установки программного обеспечения для большинства пользователей и возможности мгновенного доступа к информации.

Целевой аудиторией являются как небольшие заведения, стремящиеся оптимизировать процесс обслуживания, так и крупные сети общепита, которые могут интегрировать данное решение в уже существующую цифровую экосистему. При этом система может быть адаптирована под специфику конкретного заведения, что делает её гибким решением с потенциалом масштабирования.

Каналы распространения будут включать прямые продажи, партнёрские программы и возможность использования решения по модели SaaS(Программное обеспечение как услуга), при которой заведение платит за использование сервиса ежемесячно или ежегодно.

Взаимодействие с клиентами будет строиться на принципах долгосрочного сотрудничества. Будет внедрена система технической поддержки, регулярного обновления функционала и сбора обратной связи от пользователей. Особое внимание будет уделено юзабилити и доступности, чтобы продукт мог использоваться максимально широкой аудиторией, включая людей с ограниченными возможностями.

Ключевыми партнерами выступят поставщики NFC–меток, облачные провайдеры, а также IT–интеграторы, способные внедрять систему в уже действующую инфраструктуру заведений. В дальнейшем возможно сотрудничество с CRM–системами, такими как iiko или R–Keeper, что позволит использовать данные для аналитики и персонализации.

Доходы от внедрения системы формируются за счёт нескольких источников, каждый из которых имеет чётко определённую структуру и механизм реализации. Все предложения предоставляются на условиях подписки или разовой оплаты, в зависимости от типа клиента и объёма необходимых функций. Подписочная модель особенно актуальна для небольших кафе и ресторанов, которые стремятся оптимизировать затраты и платить только за те функции, которые действительно используются. Также возможно внедрение партнёрских программ, позволяющих интегрировать систему в уже существующие решения для ресторанного бизнеса, основные доходы указаны в таблице 3.

Основные расходы на начальном этапе связаны с разработкой, тестированием и запуском MVP (минимально жизнеспособного продукта). По мере роста числа клиентов значительная часть затрат будет покрываться за счёт подписки или единовременной оплаты за подключение. Источниками дохода

станут не только платные тарифные планы, но и предоставление интеграционных услуг, а также реализация дополнительных модулей, таких как рекомендательные системы и аналитика поведения клиентов, в таблице 3 представлены основные источники дохода.

Таблица 3 – Основные источники дохода

Источник дохода	Описание	Стоимость
Ежемесячная подписка	Доступ ко всем функциям: аналитика, история заказов, рекомендации	990–2500 рублей в месяц
Годовая подписка	Скидка за годовое использование	8000–20 000 рублей в месяц
Интеграционные услуги	Настройка под конкретное заведение, доработка API	3000–10 000 рублей
Партнёрские программы	Совместные продажи с поставщиками NFC–меток, POS–систем	10–20% от стоимости
Рекламные площадки внутри приложения	Размещение рекламы брендов алкоголя, сопутствующих товаров	500–3000 рублей

Такая структура позволяет создать устойчивый поток доходов, который будет расти по мере увеличения числа клиентов и расширения функциональности системы. Важно отметить, что система построена таким образом, чтобы масштабирование не требовало значительных дополнительных инвестиций – разработанная серверная архитектура и облачная инфраструктура позволяют добавлять новых пользователей с минимальными затратами.

При этом плата за использование сервиса взимается независимо от объёма заказов или количества клиентов заведения, что делает её особенно выгодной для точек общепита с высокой проходимостью. Это снижает риски сезонных колебаний и позволяет прогнозировать денежные потоки с большей точностью. Подробнее информация представлена в таблице 4.

Таблица 4 – Основные расходы

Статья расходов	Описание	Сумма
Разработка программного обеспечения	Создание веб и мобильного приложений, серверной логики	100 000–300 000 рублей
Тестирование	Проверка работоспособности, юзабилити–тестирование	20 000–50 000 рублей
Серверное оборудование и хостинг	Облачный хостинг, домен, SSL–сертификат	3000–10 000 рублей в месяц
Маркетинг и продвижение	Контекстная реклама, SEO	20 000–100 000 рублей в месяц
Юридические и регистрационные расходы	Регистрация ООО или ИП, патентование, защита авторских прав	15 000–30 000 рублей
Поддержка и развитие продукта	Обновления, техническая поддержка, доработки	15 000–40 000 рублей в месяц
Командные расходы	Зарплата разработчиков, дизайнеров, менеджеров	зависит от штата

Ценовая политика строится на основе принципа «плати за то, что используешь». Для каждого заведения выбирается тарифный план, соответствующий его потребностям: от базового, включающего лишь возможность сканирования NFC–меток и просмотра меню, до расширенного, предусматривающего интеграцию с внешними системами, аналитику и персонализацию.

При этом цена формируется с учётом следующих факторов:

- количество столиков и установленных NFC–меток;
- объём функционала, необходимого заведению;
- уровень технической поддержки;

– срок использования сервиса (месячная или годовая подписка).

Такой подход позволяет сделать систему доступной даже для малых кафе, одновременно предлагая расширенные возможности сетевым операторам и крупным ресторанам.

На начальном этапе проекта основной акцент сделан на B2B–сегмент (Бизнес для бизнеса), где система продается непосредственно владельцам заведений. Однако в дальнейшем возможен выход на рынок B2C (Бизнес для потребителя), например, через партнерские программы с поставщиками алкоголя, сопутствующих товаров или служб доставки. Это позволит использовать данные о предпочтениях клиентов) для показа рекламы или персональных предложений.

1.5 Финансовый план

Финансовая модель проекта разрабатывалась с учётом его масштабируемости, коммерческой ориентированности и возможности дальнейшего развития как самостоятельного стартапа. Основной задачей на данном этапе стало формирование устойчивого экономического обоснования, которое позволит не только покрыть начальные затраты, но и обеспечить рентабельность при масштабировании продукта на новые рынки. Анализ проводился с учетом реальных условий внедрения и текущих рыночных цен на технические и программные решения.

На предварительном этапе реализации были рассчитаны инвестиции, необходимые для создания минимально жизнеспособного продукта (MVP). К ним относятся расходы на разработку клиентской части – веб и мобильного приложений, проектирование серверной логики, интеграцию с NFC–устройствами, тестирование и подготовку к запуску. В стоимость также вошли временные и трудовые ресурсы, связанные с анализом требований, проектированием архитектуры системы и созданием пользовательского интерфейса. Эти затраты составили значительную часть первоначального бюджета, так как именно на этом этапе закладывается основа всей системы.

После завершения разработки MVP необходимо учитывать ежемесячные расходы на эксплуатацию продукта. К ним относится стоимость аренды серверного оборудования или использование облачного хостинга, поддержка программного обеспечения, регулярное обновление функционала, а также техническая поддержка клиентов. Особое внимание уделено надёжности инфраструктуры: выбранная платформа хостинга обеспечивает высокую доступность, резервное копирование данных и защиту от DDoS-атак. Эти меры повышают уровень доверия со стороны клиентов, однако требуют дополнительных финансовых вложений.

Для оценки доходности проекта был выполнен прогноз поступлений от подписки. Учитывая текущую рыночную ситуацию и средние цены на аналогичные услуги, была определена стоимость месячного тарифного плана, который позволяет заведению использовать систему в полном объёме. Прогноз количества клиентов строился на основе анализа спроса в сегменте малых и средних кафе, заинтересованных в автоматизации процессов обслуживания. Также был учтён потенциал привлечения сетевых операторов, что может значительно увеличить количество подписчиков в перспективе ближайших месяцев. Конверсия из числа заинтересованных лиц в фактических клиентов оценивалась с учётом эффективности маркетинговой стратегии и качества предоставляемого сервиса.

Особое внимание было уделено расчёту точки безубыточности, которая показывает момент, когда суммарные доходы покроют все понесённые расходы, представлено в таблице 5. С учётом прогнозируемого темпа прироста клиентов и среднемесячного дохода, ожидается, что этот порог будет достигнут в течение первых 8–12 месяцев после запуска продукта. Такой срок считается приемлемым для стартапа такого уровня и открывает возможность дальнейшего привлечения инвестиций или выхода на международный рынок. Динамика роста клиентов и ежемесячного дохода представлена в таблице 6.

В качестве возможных источников финансирования рассматривается несколько направлений. На раннем этапе наиболее вероятным вариантом является использование собственных средств или привлечение частных инвесторов, готовых поддержать инновационный проект. Также планируется участие в конкурсах и грантовых программах, направленных на развитие цифровых решений и технологического предпринимательства. Расчёты приведены в таблице 5 и 6.

Таблица 5 – Расчет точки безубыточности

Показатель	Значение
Постоянные расходы (ежемесячно)	25 000 Р
Цена подписки на одно заведение	2000 Р
Маржинальная прибыль на одного клиента	1800 Р
Точка безубыточности (клиентов)	≈ 14 клиентов
Месяц достижения точки безубыточности	9-й месяц

Таблица 6 – Динамика роста клиентской базы и ежемесячных доходов

1	2	3	4	5
Месяц	Новые клиенты	Общее число клиентов	Доход за месяц, рублей	Накопленный доход, рублей
1	2	2	4 000	4 000
2	2	4	8 000	12 000
3	2	6	12 000	24 000
4	2	8	16 000	40 000
5	2	10	20 000	60 000
6	2	12	24 000	84 000
7	2	14	28 000	112 000
8	2	16	32 000	144 000

1	2	3	4	5
9	2	18	36 000	180 000
10	2	20	40 000	220 000
11	2	22	44 000	264 000
12	2	24	48 000	312 000

1.6 Потенциал масштабирования проекта

Одним из ключевых преимуществ системы является её универсальность – она может быть внедрена в любом заведении, независимо от его формата: кафе, ресторан, фуд–корт или точка быстрого питания. Система не требует значительной доработки под конкретное место, так как вся информация о меню, акциях и доступных столах хранится на сервере и легко обновляется через интерфейс администратора. Это позволяет быстро запускать продукт в новых точках без повторной разработки.

Благодаря облачной архитектуре и REST API, к системе можно подключать новые заведения в разных городах и регионах. Клиентам доступно единое пространство взаимодействия, а владельцам заведений – возможность управлять своим контентом удалённо. Такая универсальность делает продукт привлекательным для широкого круга потенциальных пользователей и открывает возможности для масштабирования в разных сегментах рынка.

Для малого бизнеса особенно важна простота внедрения, экономичность и минимальная зависимость от IT–специалистов. В этом случае система может быть предложена в виде минималистичного тарифа, включающего:

- базовое меню с возможностью редактирования через административную панель MODX;
- интеграцию с Telegram для получения заказов;
- ограничение по геолокации для предотвращения удалённого использования;

- минимальный набор меток (например, 5–10 штук на одно заведение).

Все функции реализуются без необходимости установки дополнительного ПО со стороны клиента – достаточно NFC-совместимого смартфона. Это снижает порог входа и позволяет быстро начать использование сервиса даже при ограниченном техническом опыте у владельца заведения.

Также предусмотрена возможность автоматического обновления информации о меню, что особенно важно для заведений, часто меняющих ассортимент или проводящих акции. Администратор может добавлять новые блюда, изменять цены и загружать изображения через удобный интерфейс MODX без участия программиста.

Для сетевых заведений и крупных ресторанов доступны расширенные функции, которые обеспечивают централизованное управление несколькими точками, аналитику поведения клиентов и персонализацию предложений.

Система поддерживает:

- многопользовательский режим, позволяющий управлять несколькими точками с одного аккаунта;
- интеграцию с CRM-системами (например, iiko, R-Keeper), что даёт возможность собирать данные о предпочтениях клиентов и использовать их для рекомендаций;
- расширенную аналитику: популярные блюда, время наибольшей активности, частота повторных заказов и другие метрики;
- группировку меток по столам и зонам, что особенно важно для больших ресторанов и банкетных залов.

Благодаря REST API и модульной архитектуре, к таким заведениям могут быть подключены дополнительные функции:

- программа лояльности с начислением бонусов за заказы;
- пуш-уведомления о новых акциях и сезонных предложениях.

Эти элементы значительно увеличивают ценность системы и делают её полноценным инструментом управления клиентским опытом.

Фуд–корты представляют собой уникальный сегмент общепита, где одновременно работают несколько брендов в одной территории. Для таких условий была предусмотрена возможность параллельной работы нескольких меню в рамках одного пространства.

Клиент, сканируя NFC–метку, получает доступ к меню конкретного заведения, которое соответствует расположению столика. При этом:

- каждое заведение в фуд–корте имеет отдельный интерфейс администрирования;
- заказы направляются только в выбранную точку;
- возможно применение единой системы оплаты и учета.

Такой подход позволяет не только обеспечить корректную работу системы, но и создать единое цифровое пространство, повышающее удобство для посетителей и эффективность управления для организаторов пространства.

Для точек быстрого питания и служб доставки предусмотрена возможность ускоренного оформления заказа. Благодаря сохранению истории покупок, клиент может всего за один клик повторить свой любимый заказ или получить рекомендации на основе предыдущих выборов.

Ключевые особенности:

- быстрая отправка заказа в Telegram или любую систему учёта;
- возможность комбинировать заказы с нескольких устройств (например, если группа людей делает общий заказ);
- поддержка QR/NFC–билетов для самовывоза и доставки, упрощающая идентификацию заказа.

Эти функции делают систему удобной как для посетителя, так и для сотрудников точки, ускоряя процессы и минимизируя ошибки при оформлении заказа.

Одной из ключевых возможностей дальнейшего развития проекта является интеграция с внешними системами, что позволит значительно расширить

функционал, повысить уровень автоматизации и обеспечить более тесное взаимодействие между различными компонентами цифровой экосистемы заведения общественного питания. В текущей системе основное внимание было уделено клиентской части – интерфейсу пользователя, NFC–взаимодействию и отправке заказов через Telegram. Однако архитектура системы спроектирована таким образом, чтобы в будущем можно было легко подключать сторонние сервисы, такие как CRM, ERP, учётные системы.

На сегодняшний день большинство заведений общественного питания используют профессиональные системы управления рестораном, такие как iiko, R-Keeper, Omicron, и другие. Эти платформы позволяют:

- учитывать остатки товаров на складе;
- формировать отчёты по продажам;
- управлять персоналом;
- интегрироваться с кассовым ПО и онлайн–кассами;
- автоматизировать финансовые операции.

MenuPRO может быть интегрирована с такими системами через REST API, предоставляя данные о сделанных заказах, времени их оформления, составе блюд и статусах выполнения. Это позволит:

- синхронизировать данные о продажах в реальном времени;
- автоматически обновлять остатки товаров после каждого заказа;
- генерировать аналитические отчёты, которые могут использоваться для планирования закупок и прогнозирования спроса;
- сократить ручной ввод данных, минимизируя ошибки и экономя время сотрудников.

Для повышения уровня персонализации и улучшения клиентского опыта в будущем планируется интеграция с CRM–системами, такими как Bitrix24, AmoCRM, Salesforce и другими. Это даст возможность:

- собирать данные о предпочтениях клиентов;
- вести историю заказов;

- создавать программы лояльности;
- отправлять персонализированные предложения;
- анализировать поведение пользователей и выявлять наиболее активных клиентов.

Для реализации этой задачи будет создано отдельное хранилище пользовательских данных (по желанию клиента), которое может быть связано с аккаунтом Telegram или мобильным приложением. Также будет разработан механизм авторизации, позволяющий сохранять предпочтения пользователя даже при переходе между устройствами.

Сбор и анализ данных – важная часть цифровизации любого бизнеса. MenuPRO уже сейчас собирает информацию о каждом заказе: время, состав, количество просмотров, популярность блюд. В дальнейшем эти данные могут быть переданы в BI-платформы (например, Power BI, Tableau) или встроенные модули аналитики, что откроет возможность:

- визуализировать продажи по дням и часам;
- выявлять самые популярные и непопулярные блюда;
- прогнозировать спрос на продукты;
- оптимизировать меню на основе поведения клиентов.

Кроме того, может быть реализована система рекомендаций, которая будет предлагать клиентам:

- блюда, аналогичные ранее заказанным;
- популярные позиции;
- сезонные или ограниченные предложения.

Рекомендательный алгоритм может быть простым (на основе истории заказов) или сложным (с использованием машинного обучения и кластеризации пользователей). Такая интеграция превращает систему из инструмента оформления заказов в полноценный маркетинговый и аналитический инструмент.

В перспективе система может быть развита в сторону:

- углубления аналитики: добавление рекомендательных алгоритмов, анализа популярности блюд, времени заказа и частоты посещений;
- программ лояльности: начисление бонусов, участие в акциях, история заказов и индивидуальные предложения;
- автоматизации маркетинга: рассылка push-уведомлений, SMS или email-оповещений о новых блюдах, скидках и событиях;
- расширения пользовательской базы: развитие B2C-направления, например, через партнёрские программы с поставщиками алкоголя или сопутствующих товаров.

Все эти элементы могут быть реализованы в виде отдельных модулей, которые будут доступны за дополнительную плату, что обеспечит рост доходов после достижения точки безубыточности.

Проект имеет хорошие перспективы для международного распространения, особенно в странах с высоким уровнем автоматизации ресторанного бизнеса и развитой инфраструктурой NFC-технологий. Технологическая простота и высокая скорость взаимодействия делают решение конкурентоспособным на глобальном уровне.

2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Цели и задачи проектирования

Этап проектирования программного обеспечения является фундаментальным в процессе создания мобильного приложения, так как именно на этом этапе определяются архитектурные принципы, состав функциональных модулей и технические решения, обеспечивающие соответствие системы бизнес-требованиям. Проектирование должно не только заложить основу для реализации, но и учесть специфику мобильной разработки: кроссплатформенность, автономность работы, безопасность данных и интеграцию с нативными API.

Целью проектирования стало создание устойчивой, масштабируемой и интуитивно понятной платформы, которая бы автоматизировала взаимодействие клиентов с меню через NFC-технологии и одновременно обеспечивала высокую степень гибкости для владельцев заведений. Эта цель включала в себя разработку архитектуры, которая соответствует современным стандартам безопасности, производительности и пользовательского опыта, а также обеспечивает возможность дальнейшего развития продукта в условиях изменяющихся технологических требований.

Для достижения указанной цели были определены задачи, которые требовали глубокого анализа специфики мобильной разработки и особенностей внедрения NFC-технологий. Эти задачи охватывали широкий спектр аспектов, от сбора требований до подготовки технической документации, и были направлены на создание системы, которая бы эффективно решала, как пользовательские, так и бизнес-задачи.

Ключевые аспекты проектирования

В процессе проектирования мобильного приложения были выделены следующие аспекты, которые способствовали успешной реализации. Представлены в таблице 7.

Таблица 7 – Аспекты проектирования

Аспект	Описание	Значение для проекта
Анализ требова-ний	Сбор и формализация функциональных и нефункциональных требований к системе.	Позволяет выявить ожидания пользователей и бизнеса, а также определить технические ограничения.
Выбор технологического стека	Определение оптимальных инструментов разработки, хранения данных и интеграции.	Обеспечивает кроссплатформенность, производительность и совместимость с мобильными API.
Проектирование архитектуры	Создание структуры приложения, включая слои представления, логики и данных.	Гарантирует устойчивость, тестируемость и простоту сопровождения, а также позволяет масштабировать систему.
Разработка модели данных	Формирование логической и физической схемы хранения информации.	Обеспечивает согласованность данных, их целостность и возможность работы в офлайн-режиме.
Проектирование пользовательского интерфейса	Создание макетов и прототипов, соответствующих стандартам Material Design и Cupertino.	Упрощает взаимодействие с приложением, повышает удобство использования и снижает порог входа для пользователей.
Обеспечение безопасности	Проработка механизмов защиты данных, аутентификации и интеграции платежных систем.	Исключает утечки конфиденциальной информации, обеспечивает доверие пользователей и соответствие нормативам.
Тестирование архитектуры	Проверка устойчивости и масштабируемости выбранной архитектуры.	Минимизирует риски ошибок на этапе реализации и обеспечивает стабильность приложения в различных условиях.
Подготовка спецификаций	Создание технической документации для команды разработки и тестирования.	Упрощает координацию работы, ускоряет реализацию и облегчает сопровождение продукта.

Цели проектирования

Цель проектирования заключалась в создании мобильного приложения, которое бы обеспечивало бесконтактное и мгновенное взаимодействие клиентов с меню через NFC-метки, при этом сохраняя высокую степень гибкости

для адаптации под нужды различных категорий заведений. Особое внимание было уделено архитектурным решениям, которые должны были:

- обеспечить кроссплатформенную совместимость (Android и iOS);
- поддерживать автономную работу в условиях ограниченного интернета;
- интегрировать NFC–технологии для упрощения доступа к меню;
- реализовать безопасность данных и платежных операций;
- обеспечить возможность расширения функционала в будущем.

Эти цели были сформулированы на основе анализа потребностей рынка и особенностей мобильных устройств, таких как ограниченная площадь экрана, зависимость от батареи, необходимость оптимизации производительности и интеграции с нативными функциями.

Задачи проектирования

Для достижения поставленной цели были определены задачи, которые охватывали все этапы проектирования:

- провести анализ требований к функциональности и качеству системы, включая удобство использования, скорость работы и совместимость с NFC–метками;
- выбрать технологический стек, который обеспечит кроссплатформенность, производительность и простоту сопровождения;
- спроектировать многослойную архитектуру приложения, разделённую на слои представления, бизнес–логики и данных;
- определить состав функциональных модулей: NFC–интеграция, оформление заказа, оплата, аналитика;
- проработать меры по защите информации, включая шифрование и интеграцию с Apple Pay/Сбер Pay;
- создать макеты пользовательского интерфейса, учитывающие особенности мобильных устройств и стандарты дизайна Material Design/Cupertino;

– подготовить техническую документацию, включающую описание модулей, алгоритмов работы и правил интеграции.

2.2 Состав функциональных и обеспечивающих систем

Процесс разработки мобильного приложения предполагает чёткое разделение системы на отдельные модули, каждый из которых выполняет определённые функции и взаимодействует с другими компонентами. Такой подход позволяет упростить разработку, тестирование и дальнейшее сопровождение приложения, а также обеспечивает гибкость при добавлении новых возможностей. Архитектура приложения спроектирована с учётом специфики мобильных устройств, включая кроссплатформенность, автономную работу и интеграцию с нативными API, что делает её устойчивой к изменениям в технологическом ландшафте и требованиях пользователей.

Функциональные подсистемы

Функциональные подсистемы приложения ориентированы на выполнение ключевых бизнес-задач, связанных с взаимодействием клиентов с меню, оформлением заказов, оплатой и сбором аналитики. Каждая из них реализована с учётом мобильной специфики, таких как ограниченная площадь экрана, необходимость автономной работы и глубокая интеграция с аппаратными возможностями смартфонов.

NFC-модуль является центральным элементом системы, обеспечивающим бесконтактное взаимодействие между клиентом и заведением. Он реализован через библиотеки `nfc_manager` и `flutter_nfc_kit`, которые позволяют считывать данные с NFC-меток, распознавать их уникальные идентификаторы и инициировать действия в приложении. Например, при прикосновении смартфона к метке на столе приложение автоматически открывает меню заведения, исключая необходимость ручного поиска URL. Модуль также обеспечивает шифрование данных на метке для предотвращения несанкционированного доступа, а также совместимость с различными типами NFC-чипов (например, NTAG213/215/216).

Этот модуль реализует логику формирования заказов, начиная с выбора блюд и заканчивая отправкой данных на кухню или в службу доставки. Он включает возможность изменения состава заказа, отслеживания его статуса в реальном времени. Для повышения удобства использования реализованы алгоритмы рекомендаций, которые анализируют историю заказов клиента и предлагают популярные блюда.

Интеграция с Apple Pay и Сбер Pay обеспечивает безопасную обработку транзакций, исключая необходимость ввода реквизитов карты вручную. Платёжный модуль реализует проверку данных клиента, обработку транзакции и выдачу электронного чека после завершения операции.

Этот компонент собирает данные о посещениях, популярных блюдах, среднем чеке и других метриках, что позволяет владельцам заведений принимать обоснованные управленческие решения. Аналитика реализована через Firebase Analytics. В мобильной версии добавлены функции персонализации – система рекомендует блюда на основе истории заказов и времени суток.

Обеспечивающие подсистемы приложения обеспечивают техническую реализацию, интеграцию и поддержку работы всей платформы. Их задача – создать устойчивую инфраструктуру, которая минимизирует риски сбоев, обеспечивает защиту данных и упрощает сопровождение продукта.

Техническая поддержка включает аппаратное обеспечение, необходимое для работы:

- смартфоны с NFC: Android 5.0 и выше, iPhone 7 и выше;
- NFC–метки: долговечные и недорогие чипы (стоимость от 10 рублей до 200 рублей RUB за штуку), интегрируемые в интерьер заведения;
- серверы резервного копирования: для хранения архивных данных и восстановления системы при сбое.

Особое внимание уделено совместимости с различными устройствами: приложение протестировано на бюджетных и премиум–смартфонах, а также

на устройствах с ограниченными ресурсами (например, с низкой производительностью процессора или слабым интернет-соединением).

Информационная подсистема отвечает за корректное хранение, обработку и защиту данных. Для этого используются:

- Firebase Cloud Firestore – для облачной синхронизации данных между устройствами и сбора аналитики;

- шифрование AES-256 – для защиты конфиденциальной информации, такой как данные о платежах и пользователях.

Программная часть охватывает как клиентскую, так и серверную логику:

- клиентская часть – разработана на Flutter с использованием архитектурного паттерна BLoC/Provider, что обеспечивает разделение слоёв представления, логики и данных;

- серверная часть – реализована через Firebase, где хранятся данные о заведениях, пользователях и заказах;

- API-интерфейсы – REST API для интеграции с внешними сервисами, такими как CRM-системы или сторонние платёжные шлюзы;

- административная панель – позволяет владельцам заведений управлять меню, просматривать аналитику и настраивать параметры NFC-меток.

Этот подход упрощает сопровождение и масштабирование, так как изменения в одном модуле не требуют переработки всей системы.

Математическая подсистема включает алгоритмы обработки данных, такие как:

- подсчёт суммы заказа с учётом скидок, налогов и промокодов;
- рекомендации блюд на основе истории покупок и предпочтений пользователя;

- аналитика продаж – расчёт популярности блюд, времени обслуживания и эффективности маркетинговых кампаний.

Алгоритмы рекомендаций реализованы через простые, но эффективные методы, такие как коллаборативная фильтрация на основе популярности блюд

среди пользователей. Это позволило избежать сложных вычислений, сохраняя производительность на бюджетных устройствах.

Эти меры обеспечивают стабильность приложения и упрощают его обслуживание, особенно при масштабировании на крупные сети ресторанов. Представлено в таблице 8.

Таблица 8 – Сравнение функциональных и обеспечивающих подсистем

1	2	3	4
Подсистема	Цель	Техническая реализация	Преимущества
NFC–модуль	Бесконтактный доступ к меню через NFC–метки.	Библиотеки <code>nfc_manager</code> и <code>flutter_nfc_kit</code> .	Исключает физический контакт, минимизирует время открытия меню.
Управление заказами	Формирование и обработка заказов.	Облачная синхронизация через Firebase.	Поддержка офлайн–режима, отслеживание статусов заказов.
Платёжный модуль	Безопасная обработка транзакций.	Интеграция с Apple Pay/Сбер Pay, шифрование данных.	Высокая степень защиты, биометрическая аутентификация.
Аналитика	Сбор и анализ данных о предпочтениях клиентов.	Firebase Analytics + алгоритмы рекомендаций.	Персонализация меню, оптимизация ассортимента.
Техническая поддержка	Обеспечение стабильной работы системы.	Совместимость с Android и iOS, использование недорогих NFC–меток.	Минимальные затраты на внедрение, простота масштабирования.
Информационное обеспечение	Хранение и защита данных.	SQLite для офлайн–режима, Firebase для синхронизации.	Целостность данных, шифрование AES–256.
Программное обеспечение	Обеспечение гибкости и поддержки системы.	Архитектура BLoC/Provider, REST API, административная панель.	Простота доработки, кроссплатформенность.

1	2	3	4
Математическое обеспечение	Алгоритмы подсчёта заказов и рекомендаций.	Коллаборативная фильтрация, расчёт времени обслуживания.	Повышение среднего чека, улучшение пользовательского опыта.
Организационное обеспечение	Администрирование и поддержка приложения.	Firebase Authentication, Git, интеграция с чат-ботами.	Упрощение управления правами, снижение нагрузки на службу поддержки.

Все перечисленные подсистемы тесно взаимосвязаны и работают совместно, обеспечивая бесперебойную работу приложения. Например, NFC-модуль инициирует открытие меню, которое загружается из базы данных. Платёжный модуль взаимодействует с NFC-модулем для ускорения транзакций, а аналитическая подсистема использует данные из модуля управления заказами для формирования отчётов.

Особенно важна роль обеспечивающих подсистем в поддержании стабильности. Например, Firebase Authentication контролирует доступ к данным, а Git обеспечивает управление версиями кода, что критично для кроссплатформенной разработки.

Разделение на функциональные и обеспечивающие подсистемы позволило:

- упростить разработку – отдельные модули можно тестировать и дорабатывать независимо, что ускорило вывод продукта на рынок;
- обеспечить масштабируемость – приложение может быть адаптировано под нужды как малых кафе, так и крупных сетей с минимальными изменениями в коде;
- повысить надёжность – чёткое разделение ответственности между модулями снизило количество ошибок при обновлениях и упростило отладку.

- улучшить пользовательский опыт: интеграция с нативными API (NFC, камеры) обеспечила высокую скорость и интуитивность интерфейса;
- снизить затраты – использование open-source технологий (Flutter, Firebase) сделало решение доступным даже для заведений с ограниченным бюджетом.

Разделение на функциональные и обеспечивающие подсистемы стало ключевым фактором успешной реализации. Функциональные модули обеспечивают выполнение бизнес-задач (работа с NFC, заказы, оплата, аналитика), тогда как обеспечивающие подсистемы (техническая, информационная, программная, организационная) гарантируют стабильность, безопасность и простоту сопровождения. Такой подход не только упорядочил процесс разработки, но и повысил надёжность, гибкость и масштабируемость приложения, что делает его перспективным решением для цифровизации ресторанного бизнеса. Исходя из этого было сформулированное техническое задание в приложении Д.

2.3 Выбор инструментов реализации проекта

При разработке цифровых решений в сфере общественного питания, таких как мобильные приложения для заказа еды, онлайн-меню, системы управления столиками и другие автоматизированные сервисы, особое значение имеет выбор правильных инструментов разработки. От качества подобранных технологий зависит не только скорость и удобство реализации проекта, но и его надёжность, масштабируемость, а также дальнейшее сопровождение [1].

Для эффективной разработки были отобраны современные и проверенные инструменты, соответствующие целям и задачам проекта. Предпочтение было отдано решениям, поддерживающим кроссплатформенную разработку, обеспечивающим высокую производительность и совместимость с различными устройствами, а также обладающим развитым сообществом и документацией. Представлено в таблице 9.

Таблица 9 – Основные инструменты

Категория инструмента	Инструменты	Назначение
Средства разработки (IDE)	Android Studio, Xcode, Visual Studio Code	Используются для написания, тестирования и отладки кода на платформах Android и iOS, а также для разработки веб-компонентов
Языки программирования	Dart	Применяются для создания приложений с высокой производительностью и качественным пользовательским интерфейсом
Управление проектом	Miro	Обеспечивают эффективную организацию рабочих процессов
Системы контроля версий	Git, GitHub	Позволяют управлять изменениями в коде, обеспечивая безопасное хранение, совместную работу и возможность отката к предыдущим версиям
Проектирование интерфейсов	Figma	Инструмент для создания дизайна, прототипирования и согласования пользовательского интерфейса до начала этапа разработки
Бэкенд / Серверная часть	Firebase	Фреймворки и среды для реализации серверной логики, обработки запросов и взаимодействия с базами данных
Базы данных	PostgreSQL, MongoDB	Используются для хранения информации о пользователях, заказах, меню и других данных проекта

Выбор указанных инструментов обусловлен их широким применением в профессиональной среде, наличием поддержки и регулярного обновления, а также возможностью интеграции между собой. Это позволило создать устойчивую и гибкую техническую основу для реализации проекта в сфере цифровизации общественного питания.

2.4 Проектирование архитектуры системы

Проектирование архитектуры мобильного приложения является одним из наиболее важных и ответственных этапов на пути к созданию качественного программного обеспечения. Именно на этом этапе закладываются основы всей

будущей системы — определяется структура приложения, логика взаимодействия между его компонентами, а также принципы обмена данными как внутри самого приложения, так и с внешними системами, например, серверами или сторонними сервисами.

Архитектурное решение играет роль фундамента, на котором строится вся дальнейшая разработка. Оно влияет не только на то, как будет работать приложение в текущий момент, но и на то, насколько легко его можно будет модифицировать, расширять и поддерживать в будущем. Правильно выбранная архитектура позволяет команде разработчиков избежать множества потенциальных проблем, таких как низкая производительность, сложности с тестированием, трудности в добавлении новых функций или сопровождении кода.

Кроме того, грамотное проектирование архитектуры способствует достижению ключевых целей проекта: обеспечение высокой надёжности работы приложения, возможность масштабирования под растущее количество пользователей, соблюдение требований к безопасности данных и удобству использования. Это особенно важно, когда приложение предназначено для взаимодействия с бизнесом — например, когда оно используется владельцами заведений для управления заказами, бронированием, аналитикой и прочими аспектами своей деятельности.

В рамках проекта была выбрана гибридная архитектура, сочетающая многослойный подход внутри клиента (на уровне Flutter–приложения) и облачную интеграцию через Firebase. Такое решение позволило создать автономное мобильное приложение, Общая схема архитектуры представлена на рисунке 1 и отражает распределение функционала между клиентской и серверной частями, а также механизм хранения и обмена данными. Обозначение архитектуры диаграмм системы представлено в таблице 10. Общая архитектура представлена на рисунке 1.

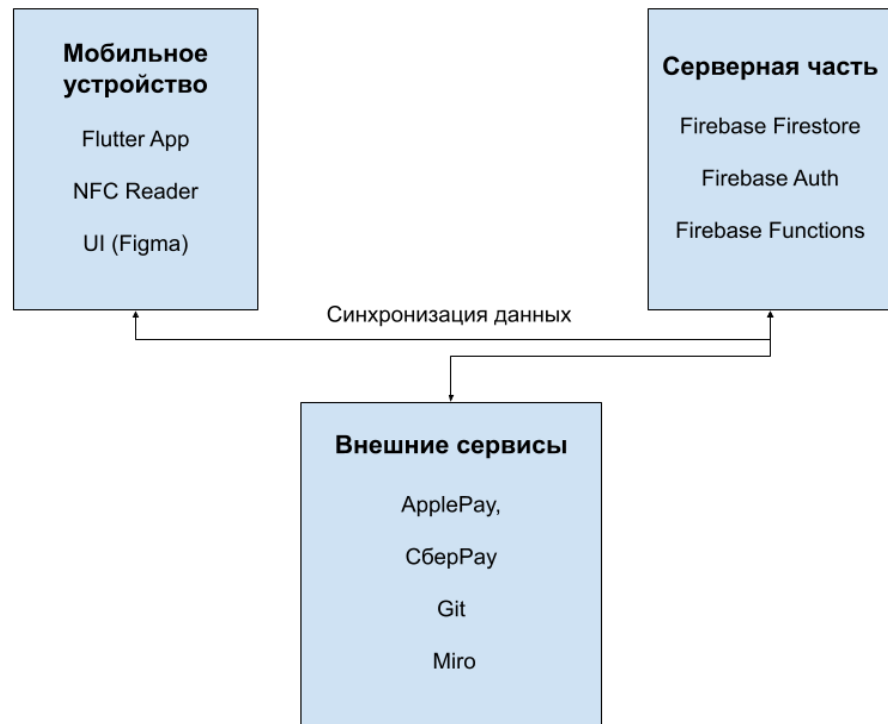


Рисунок 1 – Общая архитектура

Для наглядного представления логики системы были разработаны диаграммы UML, которые стали основой для реализации всех модулей. Представлено на рисунке 2, 3 и 4.

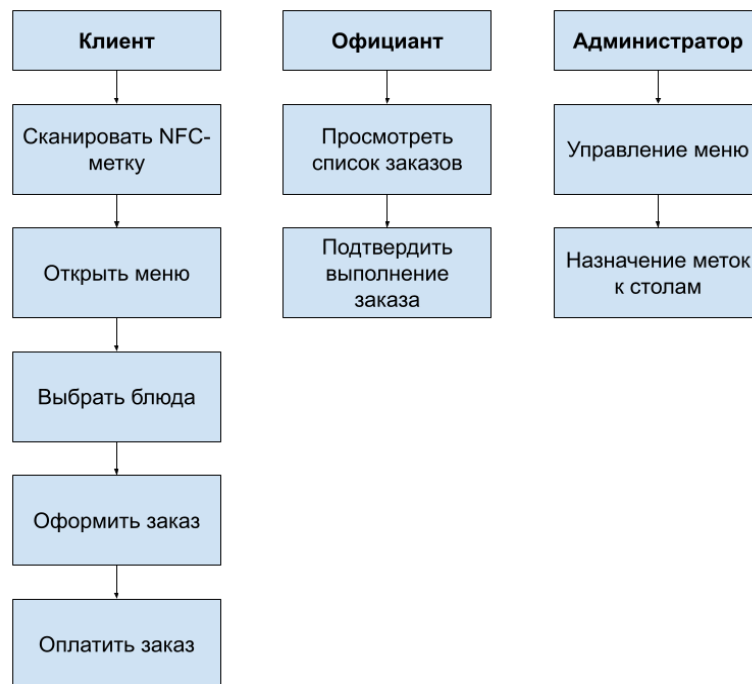


Рисунок 2 – Диаграмма сценариев

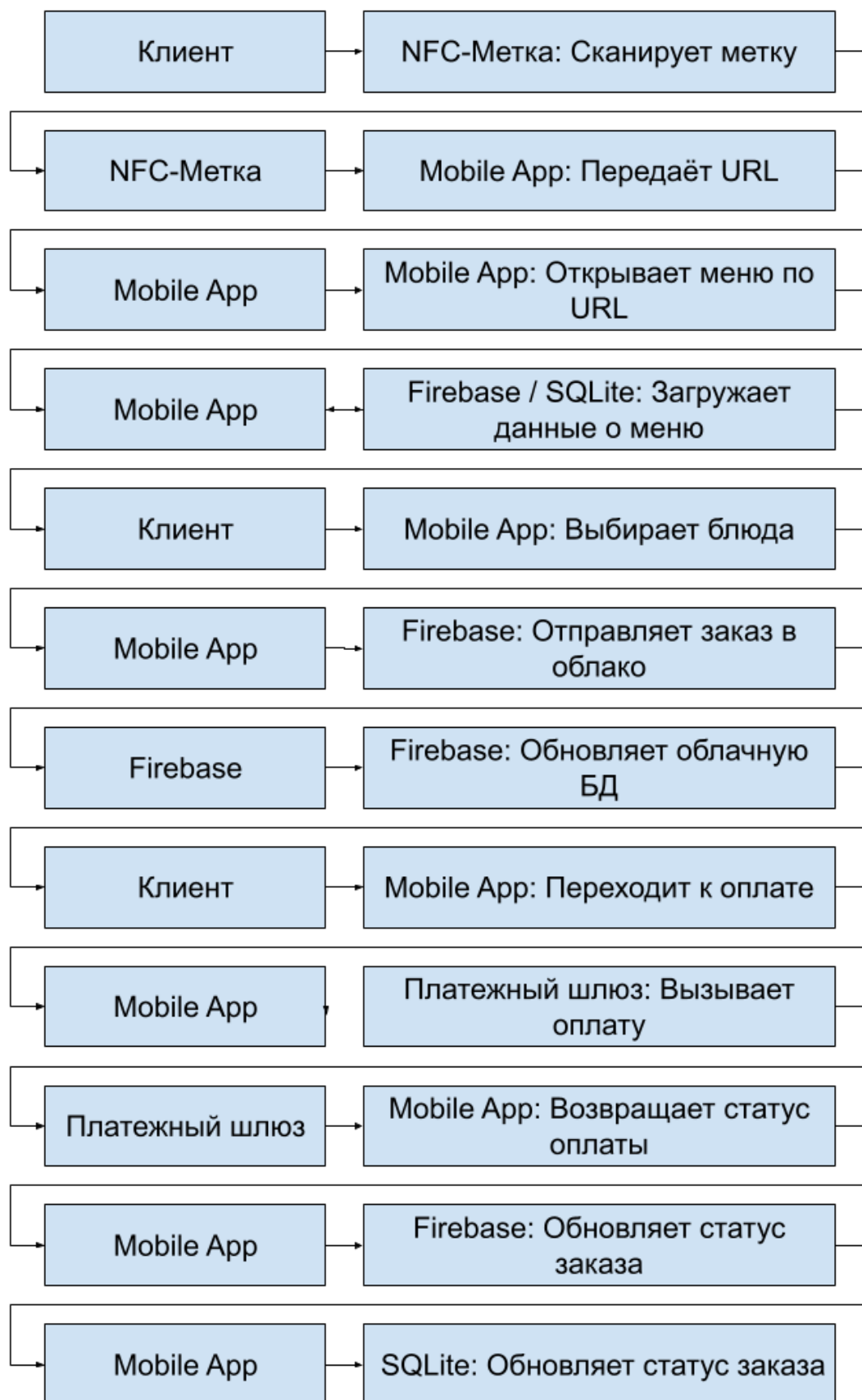


Рисунок 3 – Диаграмма последовательности

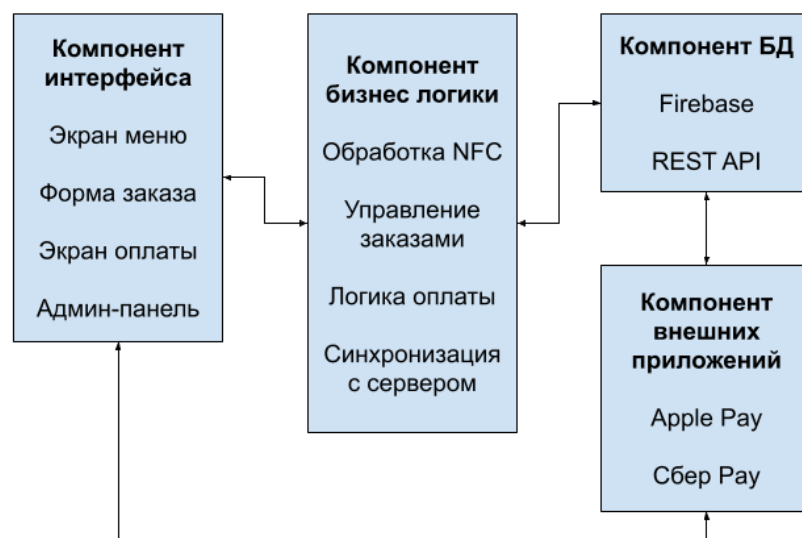


Рисунок 4 – Диаграмма компонентов

Таблица 10 – Диаграммы архитектуры системы

Диаграмма	Описание	Цель использования
Use Case (сценарии)	Отражает роли пользователей (клиент, официант, администратор) и их действия	Формализация требований и проверка полноты функционала
Sequence (последовательности)	Иллюстрирует поток вызовов методов при оформлении заказа через NFC и оплате	Тестирование логики и выявление точек возможных ошибок
Component (компонентов)	Показывает взаимодействие между модулями: UI, бизнес-логика, облачная синхронизация	Обеспечение модульности и устойчивости архитектуры

Эти диаграммы помогли чётко понять, как должны взаимодействовать модули, какие связи существуют между ними и как будут обрабатываться запросы в разных сценариях. Диаграммы классов и последовательностей были особенно важны при работе с NFC-модулем, где требуется точная обработка NDEF-сообщений и синхронизация с облачной БД [2].

На этапе проектирования были предусмотрены меры обеспечения без-

опасности и устойчивости системы. Представлены в таблице 11. Преимущества описаны в таблице 12.

Таблица 11 – Меры обеспечения безопасности

Аспект	Мера защиты
Шифрование данных	Использование AES–256 для локального хранения и TLS для облачных данных
Платежи	Интеграция с Apple Pay и Сбер Pay с биометрической аутентификацией
NFC–технологии	Шифрование данных на метке и использование одноразовых токенов
Целостность данных	Валидация на уровне приложения и контроль версий в Firebase
Резервное копирование	Автоматическое резервное копирование через Firebase Backup и Google Cloud Storage

Таблица 12 – Преимущества выбранной архитектуры

Параметр	Клиент–серверная архитектура (веб–решение)	Гибридная архитектура (мобильное приложение)
Кроссплатформенность	Требует браузерной совместимости	Поддержка Android и iOS через единую кодовую базу
Интеграция с NFC	Через Web NFC API (ограниченная поддержка)	Глубокая интеграция через нативные плагины
Производительность	Зависит от скорости загрузки страниц	Высокая, благодаря движку Skia и локальному запуску
Безопасность	Базовая (HTTPS, шифрование)	Расширенная (биометрия, шифрование меток, NFC–токены)
Скорость внедрения	Требуется настройка сервера	Ускоренная разработка через Firebase и Flutter

Многослойный подход позволил разделить ответственность между слоями:

– UI–слой – отвечает за отображение и взаимодействие;

- слой логики – управляет состоянием, обработкой NFC, формированием заказов;
- слой данных – работает с и облачной БД, обеспечивая согласованность и целостность.

Таким образом, архитектура мобильного приложения была спроектирована таким образом, чтобы обеспечить высокую степень гибкости, производительности и безопасности, при этом оставаясь экономически эффективной и простой в сопровождении.

2.5 Проектирование базы данных

Проектирование базы данных является важным этапом разработки информационной системы, поскольку именно от качества спроектированной модели зависит надёжность хранения данных, скорость обработки запросов и возможность дальнейшего масштабирования платформы. В рамках проекта была применена классическая схема проектирования, состоящая из трёх последовательных этапов: инфологическое, логическое и физическое проектирование. Каждый из них позволил постепенно формировать структуру данных, начиная с абстрактного представления предметной области и заканчивая конкретной реализацией в мобильном приложении на основе Firebase [3].

Инфологическое проектирование

На данном этапе определяется предметная область и выделяются ключевые сущности и их связи, без привязки к конкретной технологии хранения. Основная цель – понять, какие данные будут использоваться в системе и как они связаны между собой. Сущности представлены в таблице 13. Связи между ними в таблице 14.

Таблица 13 – Сущности базы данных

1	2	3
Сущность	Атрибуты	Описание
Пользователь	id, имя, email, история заказов	Клиент заведения, который взаимодействует с меню и делает заказы

1	2	3
Стол	id, номер стола, метка (NFC), URL	Физический объект, через который открывается меню
Заведение	id, название, адрес, меню	Ресторан или кафе, предоставляющее интерактивное меню
Меню	id, содержание, версия	Данные о блюдах, доступные клиенту
Категория	id, название, подкатегории	Группировка блюд (например, напитки, десерты)
Продукт	id, название, цена, описание, изображение	Конкретное блюдо или напиток
Заказ	id, стол, пользователь, список блюд, статус	Совокупность выбранных продуктов, оформленных клиентом
NFC–метка	id, url, заведение	Объект, связывающий стол с меню

Таблица 14 – Связи между сущностями

Связь	Тип связи	Описание
Пользователь → Заказ	Один ко многим	Один пользователь может сделать несколько заказов
Заказ ↔ Продукт	Многие ко многим	Один заказ может содержать несколько блюд, одно блюдо может быть в нескольких заказах
Категория → Продукт	Один ко многим	Одна категория может включать множество блюд
Заведение → Меню	Один ко многим	Заведение может иметь несколько версий меню
Стол → Заказ	Один ко многим	Один стол может быть источником нескольких заказов
Стол → NFC–метка	Один к одному	Каждому столу соответствует одна NFC–метка

На этом этапе инфологическая модель переводится в структурированный вид, учитывающий типы данных, ключи и нормализацию. Здесь формируются таблицы и устанавливаются правила целостности и взаимодействия между ними. Данный процесс представлен в таблицах 15, 16, 17, 18, 19, 20, 21.

Таблица 15 – «Стол»

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор
number	INTEGER	Номер стола
establishment_id	INTEGER FOREIGN KEY	Идентификатор заведения
nfc_url	VARCHAR	URL, записанный в NFC-метку

Таблица 16 – «Заказ»

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор
table_id	INTEGER FOREIGN KEY	Идентификатор стола
user_id (опц.)	INTEGER FOREIGN KEY	Идентификатор пользователя
created_at	TIMESTAMP	Время создания
status	VARCHAR	Статус: создан, готовится, доставлен

Таблица – 17 «Категория»

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор
name	VARCHAR	Название категории
parent_category	INTEGER FOREIGN KEY	Ссылка на родительскую категорию

Таблица 18 – «Продукт»

1	2	3
Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор
name	VARCHAR	Название блюда
price	FLOAT NOT NULL	Цена
description	TEXT	Описание
image_url	VARCHAR	Ссылка на изображение блюда

1	2	3
category_id	INTEGER FOREIGN KEY	Идентификатор категории
establishment_id	INTEGER FOREIGN KEY	Идентификатор заведения

Таблица 19 – «Продукты в заказе»

Поле	Тип данных	Описание
order_id	SERIAL PRIMARY KEY, FOREIGN KEY	Идентификатор заказа
product_id	SERIAL PRIMARY KEY, FOREIGN KEY	Идентификатор продукта
quantity	INTEGER	Количество
note	TEXT	Комментарии

Таблица 20 – Нормализация данных

Нормальная форма	Что было сделано
1НФ	Все поля атомарны, нет повторяющихся групп данных
2НФ	Все не–ключевые поля зависят только от первичного ключа; связи вынесены в отдельные таблицы
3НФ	Исключены транзитивные зависимости; добавлены внешние ключи и смежные таблицы

Таблица 21 – Достижения логического проектирования

Этап нормализации	До нормализации	После нормализации
1НФ	Дублирование информации о блюдах в заказе	Блюда вынесены в отдельную таблицу, ссылки через foreign key
2НФ	Зависимость полей от составных ключей	Создана таблица «Продукты в заказе» для разрешения связи
3НФ	Характеристики блюда хранились в одной таблице	Описание и параметры вынесены в JSON–поле data

Физическое проектирование

На физическом этапе реализуется конкретная схема базы данных, адаптированная под используемые технологии хранения: Firebase Cloud Firestore

(облачная синхронизация). Также здесь уточняются типы данных, индексы и механизмы шифрования. Данный процесс представлен в таблицах 22, 23, 24, 25, 26.

Таблица 22 – Стол

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор стола
number	INTEGER	Номер стола
nfc_url	VARCHAR	URL, хранящийся в NFC–метке
establishment_id	INTEGER FOREIGN KEY	Идентификатор заведения

Таблица 23 – Заказ

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор заказа
table_id	INTEGER FOREIGN KEY	Идентификатор стола
user_id (опц.)	INTEGER FOREIGN KEY	Идентификатор пользователя
created_at	TIMESTAMP	Время создания
status	VARCHAR	Статус заказа (ожидает, готовится, доставлен)

Таблица 24 – Категория

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор
name	VARCHAR	Название категории
parent_category	INTEGER FOREIGN KEY	Ссылка на родительскую категорию

Таблица 25 – Продукт

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор
name	VARCHAR	Название блюда
price	FLOAT NOT NULL	Цена за одну позицию
description	TEXT	Описание блюда
image_url	VARCHAR	Ссылка на изображение блюда
category_id	INTEGER FOREIGN KEY	Идентификатор категории
establishment_id	INTEGER FOREIGN KEY	Идентификатор заведения

Таблица 26 – Продукты в заказе

Поле	Тип данных	Описание
order_id	SERIAL PRIMARY KEY, FOREIGN KEY	Уникальный идентификатор заказа
product_id	SERIAL PRIMARY KEY, FOREIGN KEY	Уникальный идентификатор продукта
quantity	INTEGER	Количество порций
note	TEXT	Комментарии к блюду

Прогресс проектирования по этапам представлен в таблице 27.

Таблица 27 – Прогресс проектирования по этапам

1	2	3
Этап проектирования	Что было известно / определено	Что стало ясно / изменено
Инфологическое	Перечень сущностей: пользователи, столы, заказы, продукты, категории, NFC–метки	Определены связи между сущностями, выделены основные атрибуты
Логическое	Сущности, связи, ограничения, атрибуты	Приведены к нормализованному виду до 3НФ, создана структура таблиц и отношений между ними
Физическое	Таблицы, типы данных, внешние ключи	Реализована двухуровневая модель хранения (SQLite + Firebase), добавлены механизмы безопасности, шифрования и синхронизации

Сравнение таблиц по этапам проектирования представлено в таблице 28.

Таблица 28 – Сравнение таблиц по этапам проектирования

1	2	3	4
Сущность	Инфологическая модель	Логическая модель	Физическая модель
Стол	id, номер, метка, URL	id, number, nfc_url, establishment_id	id, number, nfc_url, establishment_id

1	2	3	3
Заказ	id, стол, пользователь, блюда, статус	id, table_id, user_id, created_at, status	id, table_id, user_id, created_at, status
Категория	id, name, parent_category	id, name, parent_category	id, name, parent_category
Продукт	id, name, price, data, category_id	id, name, price, description, image_url, category_id	id, name, price, description, image_url, category_id, establishment_id
Продукты в заказе	Отсутствует (связь "многие-ко-многим" в одной таблице)	order_id, product_id	order_id, product_id, quantity, note

Проектирование базы данных в рамках проекта было выполнено в строгом соответствии с методологиями нормализации и современными практиками хранения информации.

- инфологическая модель позволила понять, какие сущности и связи необходимы в системе;

- логическое проектирование обеспечило переход к структурированным данным и нормализации до 3НФ, что повысило целостность и снизило избыточность;

- физическое проектирование определило конкретные таблицы, типы данных и механизмы хранения.

Выбранная гибридная модель БД обеспечивает:

- высокую производительность за счёт оптимизации структур данных;
- гибкость благодаря JSON-полям и поддержке нескольких типов СУБД;
- безопасность через шифрование и контроль доступа;
- масштабируемость за счёт облачной синхронизации.

Этап проектирования стал основой для корректной реализации всех функциональных модулей и обеспечил высокую степень согласованности

между компонентами приложения. Подробная работа компонентов расписана в приложении Е.

2.6 Макетирование пользовательского интерфейса

На этапе проектирования была выполнена работа по макетированию пользовательского интерфейса мобильного приложения. Целью данного этапа стало создание удобного и интуитивно понятного интерфейса, который обеспечивает простоту взаимодействия для клиентов и соответствует современным требованиям к мобильным сервисам в сфере общественного питания.

Все элементы интерфейса были разработаны с учётом принципов юзабилити – дизайн выполнен минималистичным, структура – логичной и последовательной, текст – читаемым, а навигация – оптимизированной под экраны смартфонов. Это позволило сделать приложение доступным даже для пользователей, не имеющих опыта работы с цифровыми решениями ресторанного типа.

Экран 1: Приветственный экран

После сканирования NFC–метки пользователь попадает на приветственный экран – начальную точку взаимодействия с приложением. На этом экране отображается название заведения, его логотип и краткое приглашение перейти к просмотру меню. Такое оформление создаёт положительное первое впечатление и плавно переводит пользователя к основному функционалу. Представлено на рисунке 5.

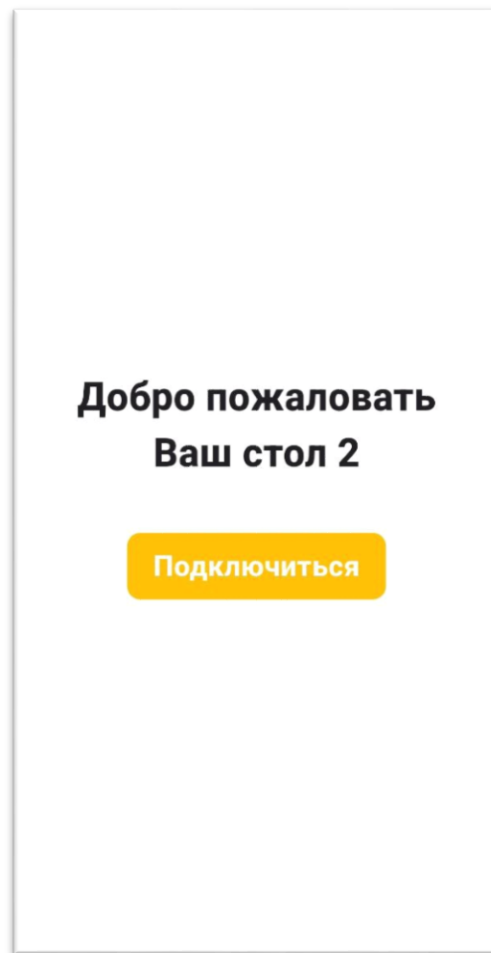


Рисунок 5 – Макет приветственного экрана

Экран 2: Главное меню категорий

Следующий экран представляет собой список основных категорий блюд или напитков. Каждая из категорий отображается с использованием графических элементов и кратких описаний, что позволяет быстро находить нужную информацию. Навигация организована таким образом, чтобы пользователь мог легко перейти к более детализированному просмотру содержимого выбранной категории. Представлено на рисунке 6.

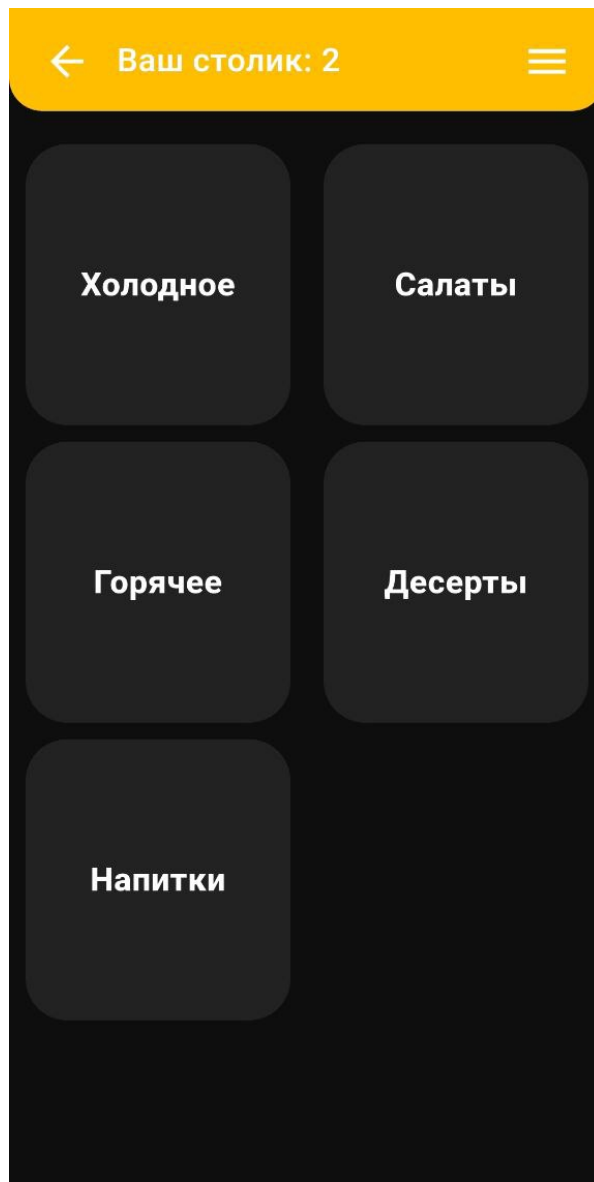


Рисунок 6 – Макет экрана "Меню категорий"

Экран 3: Просмотр списка блюд

При переходе к выбранной подкатегории открывается экран меню товаров, на котором представлены блюда с подробными характеристиками. Для каждого продукта указываются:

- название;
- фотография;
- цена.

Информация отображается в виде карточек, которые легко сканируются взглядом и обеспечивают быстрый выбор. Представлено на рисунке 7.

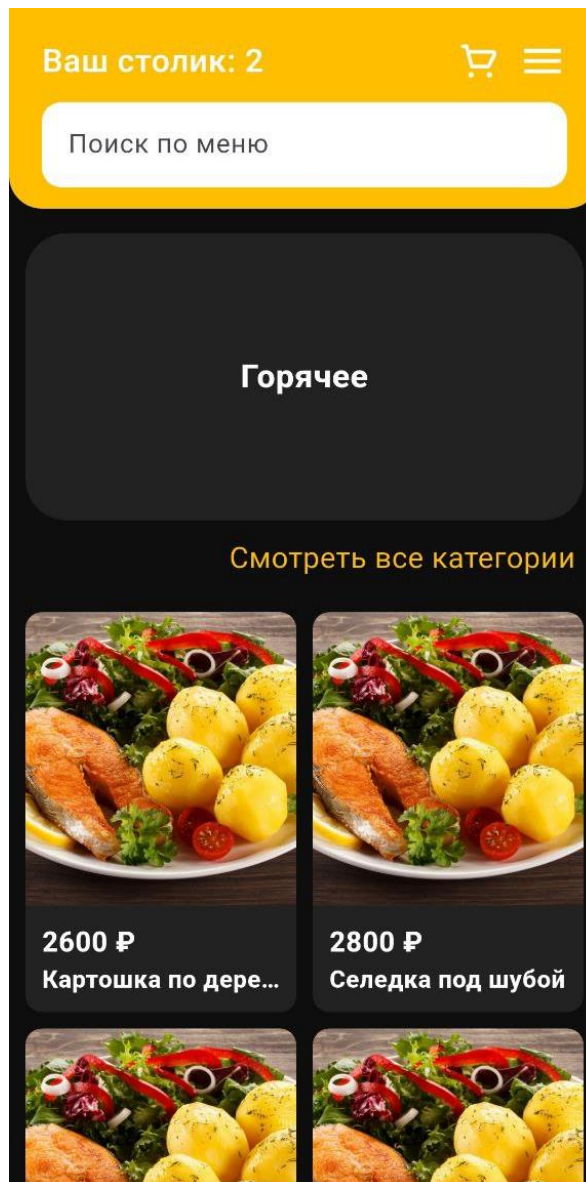


Рисунок 7 – Макет экрана "Меню напитков" / "Меню блюд"

Экран 4: Детальный просмотр блюда

При нажатии на конкретный товар пользователь переходит к детальному описанию, где собрана наиболее полная информация о данном продукте.

Карточка также содержит возможность добавления товара в корзину или возврата к предыдущему шагу. Представлено на рисунке 8.

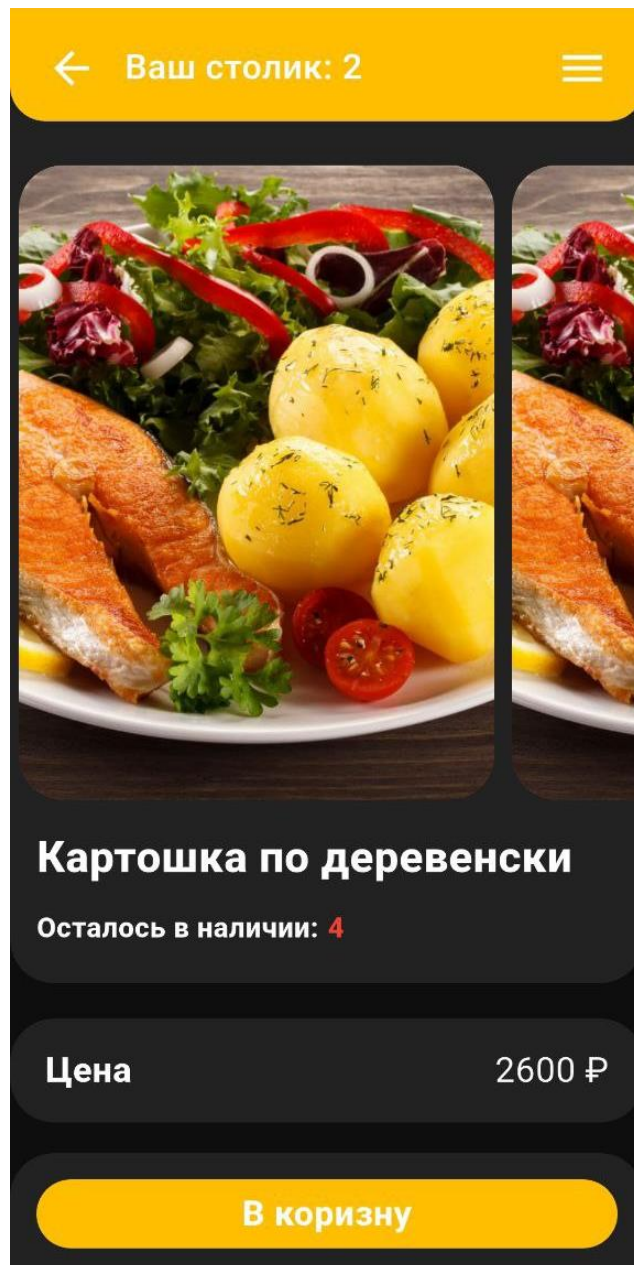


Рисунок 8 – Макет экрана "Карточка блюда"

Экран 5: Корзина заказа

Корзина показывает все выбранные блюда, их количество и цену. Пользователь может изменить состав заказа, удалить отдельные позиции или продолжить оформление. Все изменения вносятся динамически, что делает работу с заказом удобной и наглядной. Представлено на рисунке 9.



Рисунок 9 – Макет экрана «Корзина»

Экран 6: Боковое меню навигации

Для упрощения перемещения между разделами системы предусмотрен элемент бокового меню, доступный по свайпу или нажатию кнопки в верхнем углу. Представлено на рисунке 10.

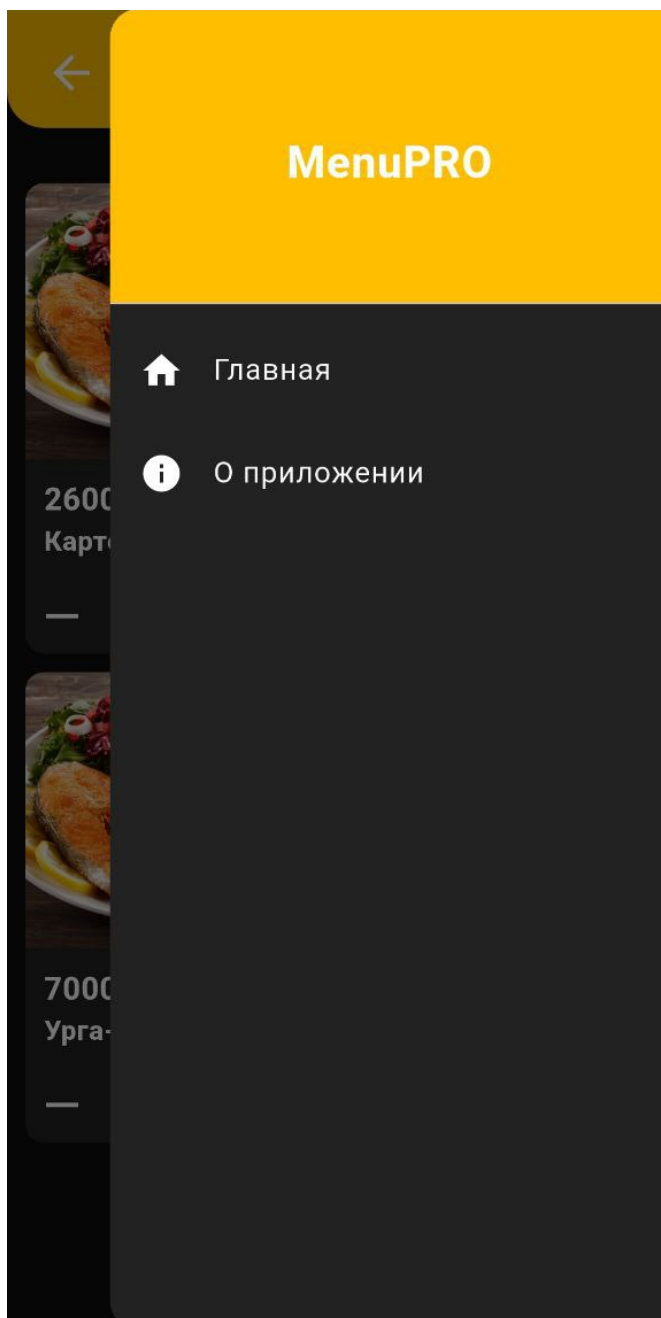


Рисунок 10 – Макет бокового меню

Технологическая реализация интерфейса

В мобильном приложении, разработанном на Flutter, интерфейс строится с помощью Material Design и Cupertino виджетов, что обеспечивает соответствие стандартам Android и iOS соответственно [4]. Flutter предоставляет широкие возможности для создания адаптивных и отзывчивых интерфейсов, а также поддерживает горячую перезагрузку, что ускоряет итерации дизайна.

Для реализации макетов были применены следующие компоненты:

- ListView / GridView: для отображения списков блюд;
- Card / ListTile: для отображения отдельных элементов меню;
- BottomNavigationBar / Drawer: для организации навигации между экранами;
- StatefulWidget / BLoC / Provider: для управления состоянием корзины и динамического обновления данных;
- NFC–интеграция: через библиотеки nfc_manager и flutter_nfc_kit , которые запускают интерфейс после считывания метки.

Принципы проектирования UI представлены в таблице 29.

Таблица 29 – Принципы проектирования UI

Принцип	Описание	Реализация
Простота использования	Минимальное количество действий для получения меню и оформления заказа	Считывание NFC–метки, одно прикосновение для выбора блюда
Читаемость	Четкий шрифт, контрастные цвета, минимальная загруженность экрана	Использование Material Theme, адаптация под темный режим
Навигация	Логичная структура экранов, наличие бокового меню и обратных ссылок	Drawer, Bottom Navigation, кнопки "Назад", хлебные крошки
Отзывчивость	Мгновенная реакция на действия пользователя	Горячая перезагрузка, использование BLoC для управления состоянием
Адаптация под экраны	Различные устройства требуют гибкости в отображении	Responsive layout, MediaQuery, Flex и Column–виджеты

2.7 Оценка качества проектных решений

После завершения этапа проектирования была проведена всесторонняя оценка качества разработанных решений с точки зрения соответствия надёжности, производительности, масштабируемости и уровня безопасности.

Одним из ключевых показателей стало соответствие основным требованиям – все основные модули – взаимодействие с NFC–метками, оформление

заказов, интеграция и платёжной системы— были реализованы в полном объеме. Помимо обязательных функций добавлены дополнительные улучшения, направленные на повышение удобства использования и общей эффективности.

Облачная синхронизация через Firebase обеспечивают целостность информации даже при временных перебоях интернета. В системе также реализованы транзакции и проверка целостности данных, что снижает риск потери информации.

Информационная безопасность была учтена на всех уровнях:

- передача данных осуществляется по защищённому протоколу HTTPS;
- доступ к платёжным операциям реализован через Apple Pay и Сбер Pay;
- NFC–метки содержат одноразовые токены для предотвращения несанкционированного чтения.

Анализ удобства сопровождения показал, что применение гибкой многослойной архитектуры и облачной синхронизации позволяет легко расширять функционал без глубокой переработки существующих модулей. Это даёт возможность добавлять новые возможности, такие как рекомендательный алгоритм, поддержка нескольких языков или интеграция с CRM–системами.

Выбор open–source технологий (Flutter, Firebase) сделал решение экономически выгодным и простым в сопровождении. Отсутствие лицензионных ограничений и наличие активного сообщества обеспечивают долгосрочную поддержку и развитие продукта.

Таким образом, проектные решения соответствуют современным стандартам разработки программного обеспечения, обладают высокой степенью надёжности, производительности и безопасности, а также возможностью дальнейшей адаптации под различные бизнес–сценарии [4].

3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Общие сведения

Разработка мобильного приложения велась поэтапно в рамках выбранной гибридной модели жизненного цикла программного обеспечения, которая позволила объединить преимущества строгого проектирования и гибкой итеративной реализации.

В основе разработки лежали материалы, подготовленные на предыдущих этапах: UML–диаграммы, модель данных, макеты интерфейсов и архитектурное описание системы. Эти документы помогли сформировать чёткое понимание структуры будущего приложения, его функциональных блоков и способов взаимодействия между ними, что позволило избежать дублирования усилий и упростило координацию команды разработчиков.

Процесс создания программного обеспечения включал несколько ключевых фаз:

- реализация клиентской части – разработка интерфейса, навигации и обработка действий пользователя;
- разработка серверной логики – интеграция с облачными сервисами (Firebase), организация хранения и синхронизации данных;
- внедрение NFC–модуля – обеспечение бесконтактного доступа к меню через метку и обработка NDEF–сообщений;
- тестирование и оптимизация – проверка корректности работы всех компонентов, выявление багов и повышение производительности.

Особое внимание уделялось соблюдению требований, сформулированных на этапе проектирования. Были реализованы механизмы, обеспечивающие:

- безопасность передачи и хранения данных;
- высокую скорость реакции интерфейса;
- интуитивно понятную навигацию.

Также были учтены рекомендации по защите информации, интеграции с нативными API, чтобы создать решение, которое будет надёжным, удобным и простым в масштабировании.

Таким образом, этап разработки стал практической реализацией спроектированного решения. Основные усилия были сосредоточены на создании клиентского приложения, его связывании с облачной частью и обеспечении бесперебойной работы всех модулей, начиная от считывания NFC–меток и заканчивая обработкой заказов и аналитикой.

3.2 Функциональное назначение

В рамках реализации мобильного приложения была спроектирована и наполнена функционалом клиентская часть, а также разработаны ключевые модули, отвечающие за взаимодействие пользователя с заведением. Все компоненты были ориентированы на обеспечение бесконтактного доступа к меню, удобного выбора блюд, формирования заказа и его доставки в кухню или службу обслуживания.

Модуль NFC–взаимодействия. Данный модуль обеспечивает мгновенный доступ к интерфейсу приложения через NFC–метку, установленную на столе или другой поверхности. При прикосновении смартфона к метке происходит автоматическое открытие меню конкретного заведения. Для реализации использовались библиотеки `nfc_manager` и `flutter_nfc_kit`, которые позволяют не только считывать данные с чипа, но и обрабатывать NDEF–сообщения – стандартный формат данных для NFC–обмена. Пример кода представлен в приложении А.

Модуль пользовательского интерфейса. Интерфейс приложения был разработан с учетом особенностей мобильных устройств и требований юзабилити. Он состоит из последовательной цепочки экранов:

- приветственный экран – отображает логотип и название заведения, активируется после сканирования метки;

- категории – список групп блюд (например, «Салаты», «Горячие блюда», «Напитки»);
- подкатегории – уточнение выбора внутри категории (например, «Пресное», «Острое»);
- просмотр блюд – отображение карточек продуктов с изображением, ценой, описанием и возможностью добавления в заказ;
- карточка товара – детальная информация о блюде, включая состав, рекомендации по подаче и сочетанию;
- корзина – просмотр текущего заказа, редактирование количества позиций.

Все экраны разрабатывались с учётом отзывчивости и адаптации под разные устройства, включая смартфоны среднего и низкого уровня, что важно для широкой целевой аудитории.

Этот компонент системы отвечает за формирование, обработку и отправку заказа. Его основные функции:

- добавление блюд в корзину с возможностью изменения количества и удаления;
- привязка к столу через идентификатор метки;
- отправка заказа в Firebase Cloud Messagin.

Для предотвращения несанкционированного использования была внедрена проверка геолокации. Система определяет координаты клиента и сверяет их с данными заведения, хранящимися в Firebase. Если пользователь находится вне заданного радиуса (обычно 50–100 метров), ему блокируется дальнейшее взаимодействие до тех пор, пока он не вернётся в зону обслуживания.

Геолокация определяется через API устройства и работает при наличии GPS, Wi-Fi или мобильного интернета. Для улучшения точности можно интегрировать внутренние точки доступа заведения. Этот механизм снижает риск массовых тестовых заказов и повышает надёжность всей платформы. Пример кода представлен в приложении Б

Структура данных была спроектирована заранее и включает следующие сущности:

- product – информация о блюдах и напитках;
- category – категории и подкатегории;
- order – данные о текущих и прошлых заказах;
- table – привязка к столам и NFC–меткам;
- product_in_order – связь между блюдами и заказами.

Текущая версия не предусматривает глубокой интеграции с ERP–системами, но такая возможность заложена в архитектуре. В будущем возможно подключение к решениям, таким как iiko, R–Keeper , 1С и другим. Предполагается, что это позволит:

- автоматически обновлять остатки товаров;
- синхронизировать данные продаж;
- формировать аналитические отчёты;
- интегрировать CRM и программы лояльности.

Для этих задач могут быть добавлены новые поля в базе данных, такие как external_id, sync_status, а также реализованы внешние методы REST API для обмена с внешними системами. Пример кода в приложении В.

Разработанные модули мобильного приложения полностью соответствуют заявленным целям – обеспечение бесконтактного доступа, простота выбора блюд, надёжность передачи заказов и экономическая эффективность.

Выбранные технологии позволили создать современное, безопасное и удобное решение, которое может применяться как в малых кафе, так и в крупных сетях ресторанов. Интеграция с мобильными возможностями (NFC, камера, геолокация, биометрия) сделала приложение актуальным и востребованным в условиях цифровой трансформации общественного питания.

3.3 Тестирование и отладка программного обеспечения

Этап тестирования мобильного приложения был направлен на проверку корректности работы всех модулей. Были проведены функциональное, нагрузочное тестирование, а также проверка интеграции с NFC и геолокационных ограничений.

Функциональное тестирование охватывало основные сценарии использования: считывание NFC–метки, выбор блюд, формирование заказа, отправка данных через Firebase. Проверялись как клиентская логика, так и серверная синхронизация.

Геолокационный контроль предотвращает оформление заказов вне заведения. Приложение сравнивает текущее местоположение пользователя с координатами заведения и блокирует действия при удалении за установленный радиус.

Нагрузочное тестирование показало стабильную работу системы при одновременном использовании до 200 клиентских сессий, что достаточно для заведений среднего размера. Результаты тестирования представлены в таблице 30.

Таблица 30 – Результаты тестирования

Вид тестирования	Цель	Результат
Функциональное	Проверка работы всех модулей	Все функции реализованы согласно ТЗ
Интерфейс	Адаптация под мобильные устройства	Успешно пройдена, интерфейс отзывчив
NFC	Скорость и точность считывания	Метки распознаются быстро и надёжно
Геолокация	Защита от удалённого использования	Эффективно ограничивает доступ вне заведения
Нагрузка	Устойчивость к высокой посещаемости	Система стабильна при нагрузке до 200 пользователей

В процессе тестирования были выявлены и исправлены:

- задержки при загрузке большого количества позиций;
- проблемы с повторным сканированием NFC–меток.

Для диагностики применялись Flutter DevTools, Firebase Crashlytics и системное логирование.

Тестирование показало готовность приложения к внедрению. Все ключевые функции работают корректно, интерфейс адаптирован под мобильные устройства, система устойчива к типовой нагрузке и защищена от несанкционированного использования. Полученные результаты легли в основу финальной доработки и подготовки приложения к запуску.

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

4.1 Безопасность

При использовании мобильного приложения для NFC–меню, разработанного в рамках данной работы, необходимо учитывать вопросы безопасности пользователей, персонала и данных. В условиях цифровизации ресторанного бизнеса безопасность становится одним из ключевых аспектов, влияющих как на эффективность использования системы, так и на комфорт, здоровье сотрудников и посетителей заведений. Учитывая, что система предполагает взаимодействие через NFC–метки и мобильный интерфейс, особое внимание уделяется вопросам организации безопасного рабочего места администратора, информационной защищённости и доступности продукта для всех категорий пользователей.

Рабочее место администратора должно отвечать современным эргономическим и санитарно–гигиеническим требованиям, направленным на снижение рисков профессиональных заболеваний и повышение общей производительности труда. При работе с административной панелью, важно обеспечить правильное расположение монитора на уровне глаз, использовать регулируемую мебель и удобные подставки для клавиатуры и мыши. Стул должен иметь поддержку поясничного отдела позвоночника, а стол – достаточную площадь для размещения оборудования и документов. Провода рекомендуется прятать в специальные каналы во избежание травмоопасности. Также важно соблюдать правило «20–20–20» для снижения зрительного напряжения, особенно при длительной работе с экраном.

Освещение рабочего места играет важную роль в создании комфортных условий труда. Оно должно быть равномерным, без бликов на экране монитора, и соответствовать нормативным требованиям – не менее 300 лк. Предпочтительно использование рассеянного света от люминесцентных или светоди-

одных ламп, а также дополнительного местного освещения для чтения печатных материалов. Температурный режим помещения должен находиться в пределах $+(20-25)^\circ\text{C}$ при относительной влажности 40–60 %, что способствует нормальному самочувствию работающего человека.

Электробезопасность обеспечивается за счёт заземления всего используемого оборудования и подключения его через устройства защитного отключения. Рекомендуется регулярно проверять состояние кабелей и разъёмов, избегать перегрузки розеток, а также использовать сетевые фильтры и источники бесперебойного питания для защиты техники от скачков напряжения. Графический интерфейс административной панели спроектирован с учётом требований доступности и эргономики: текст легко читается благодаря крупному размеру шрифта, элементы управления интуитивно понятны, возможна навигация с помощью клавиатуры, а также поддерживается масштабирование страниц без потери функциональности.

Информационная безопасность системы обеспечена даже несмотря на то, что приложение не работает с конфиденциальной информацией клиентов. Все данные передаются по защищённому протоколу HTTPS с использованием TLS, что гарантирует их целостность и конфиденциальность. Защита от SQL-инъекций реализована на уровне фреймворка RedBeanPHP, который автоматически обрабатывает входящие данные. NFC-метки содержат только идентификатор стола, работают в режиме только чтения и не хранят персональных данных, что делает их перезапись невозможной без физического доступа. Также внедрены геолокационные ограничения, которые блокируют оформление заказов вне радиуса заведения, обычно составляющего 50–100 метров. Это позволяет предотвратить массовые тестовые отправки и злоупотребления со стороны пользователей. Telegram-бот защищён от DDoS-атак и компрометации токена: он расположен на серверной стороне, запросы проходят валидацию и ограничиваются по частоте.

Доступность приложения для людей с ограниченными возможностями

здоровья реализована согласно требованиям ГОСТ Р 52872–2019. Интерфейс имеет высокую контрастность и увеличенный размер текста, поддерживает управление с клавиатуры и совместим с экранными ридерами. Возможность масштабирования страницы без нарушения её функциональности также была предусмотрена. Таким образом, мобильное приложение для NFC–меню общественного питания разработано с учётом требований безопасности, удобства использования и доступности для всех категорий пользователей.

4.2 Экологичность

Внедрение веб–приложения MenuPRO в ресторанный сферу не только повышает эффективность работы заведения, но и оказывает положительное влияние на окружающую среду. В условиях роста внимания к устойчивому развитию и экологической ответственности бизнеса, внедрение цифровых решений, снижающих негативное воздействие на природу, становится важным фактором конкурентоспособности и социальной ответственности. Применение электронного меню вместо традиционного бумажного позволяет значительно снизить нагрузку на экосистемы, оптимизировать потребление ресурсов и соответствовать современным требованиям устойчивого развития.

Одним из ключевых экологических преимуществ использования MenuPRO является значительное снижение потребления бумаги. Традиционные бумажные меню требуют регулярной замены при изменении цен, состава блюд или сезонных предложений, что приводит к постоянному расходу древесины, увеличению объёмов отходов и необходимости полиграфической печати. Цифровое меню полностью исключает необходимость бумажных носителей, что позволяет сократить вырубку лесов, уменьшить выбросы углекислого газа при производстве и доставке бумаги, а также отказаться от использования химических веществ, применяемых при её переработке. По оценкам Минприроды РФ, одно кафе, обслуживающее около 100 клиентов ежедневно, может экономить до 100 кг бумаги в год. При масштабировании такого подхода на всю индустрию общественного питания экологический эффект будет

значительным.

При использовании цифровой системы возникает вопрос обращения с электронными отходами, так как система работает на мобильных устройствах и серверах. После выхода техники из строя необходимо соблюдать установленные правила утилизации согласно Федеральному закону № 89–ФЗ "Об отходах производства и потребления". Все вышедшие из строя устройства должны храниться в безопасных условиях, защищённых от влаги и механических повреждений, после чего передаваться в специализированные пункты приёма, имеющие лицензии на переработку электроники. Отдельной утилизации подлежат аккумуляторы и литиевые элементы питания согласно Постановлению Правительства РФ № 861. Также рекомендуется заключать договоры с компаниями, занимающимися профессиональной переработкой электронных компонентов, включая содержащие драгоценные металлы. Это обеспечивает не только экологическую безопасность, но и возможность вторичного использования ценных материалов.

Еще одним важным аспектом является энергоэффективность используемой технологии. MenuPRO разработано на базе технологий с открытым исходным кодом: HTML5, CSS3, JavaScript, PHP, MySQL, которые отличаются низким энергопотреблением. Благодаря оптимизации кода и использованию легких фреймворков удалось минимизировать нагрузку как на серверы, так и на клиентские устройства. Отказ от установки стационарных терминалов для заказа позволил дополнительно снизить общее потребление электроэнергии в заведении. Например, смартфон клиента, используемый для взаимодействия с системой, потребляет всего около 2 Вт/час, тогда как планшеты для заказов могли бы использовать до 15 Вт/час. Таким образом, внедрение веб-приложения способствует снижению энергоёмкости операций и уменьшает общий углеродный след заведения.

Все мероприятия по утилизации оборудования, энергоэффективности и

соблюдению экологических стандартов полностью соответствуют действующему законодательству Российской Федерации. Федеральный закон № 89–ФЗ регулирует вопросы обращения с отходами, СанПиН 1.2.3685–21 устанавливает гигиенические нормативы безопасности, а ГОСТ Р 58751–2019 содержит требования к энергоэффективности и организации рабочих мест. Кроме того, использование открытого программного обеспечения снижает зависимость от коммерческих поставщиков и обеспечивает возможность модернизации без дополнительных лицензионных затрат.

Таким образом, внедрение веб–приложения MenuPRO в ресторанную индустрию не только повышает эффективность и удобство обслуживания клиентов, но и вносит реальный вклад в защиту окружающей среды. Основные экологические выгоды представлены в таблице 31.

Таблица 31 – Итоговые экологические показатели внедрения MenuPRO

Показатель	Значение/эффект
Снижение потребления бумаги	До 100 кг в год на одно заведение
Уменьшение выбросов CO ₂	За счёт отказа от печати и перевода на цифровую платформу
Обращение с электронными отходами	Организовано в соответствии с законодательством
Энергопотребление	Снижено за счёт оптимизации и отказа от лишних терминалов
Углеродный след	Существенно снижен
Соответствие экологическим стандартам	Полное соответствие требованиям законодательства

Эти данные подтверждают, что использование цифрового меню MenuPRO – это не только технологическое решение, но и шаг в сторону более устойчивого и экологически ответственного бизнеса.

4.3 Чрезвычайные ситуации

В процессе эксплуатации системы, как и любой другой информационной технологии, могут возникнуть чрезвычайные ситуации (ЧС), которые способны повлиять на её работоспособность, доступность данных и безопасность пользователей. Учитывая специфику использования веб–приложения MenuPRO в условиях ресторанного бизнеса, где важна непрерывная работа и оперативное выполнение заказов, необходимо предусмотреть комплекс мер по предупреждению, реагированию и восстановлению после аварийных ситуаций.

Виды возможных чрезвычайных ситуаций

- пожар одним из наиболее опасных рисков является возникновение пожара в помещении, связанного с неисправностью электропроводки или перегревом оборудования. Такие ситуации могут привести к выходу из строя серверов, уничтожению данных и создать угрозу жизни и здоровью сотрудников и клиентов заведения;

- отключение электроэнергии: неожиданное отключение питания может привести к временной недоступности сервиса, потере данных и нарушению работы заказной системы;

- выход из строя оборудования: поломка мобильных устройств, сервера или NFC–меток может привести к временной недоступности сервиса или невозможности взаимодействия с системой;

- киберугрозы: несмотря на то, что система не предусматривает работу с конфиденциальной информацией клиентов, она остаётся объектом возможных киберугроз, таких как DDoS–атаки, попытки несанкционированного доступа, вредоносные программы и другие виды атак;

- сбои программного обеспечения Программные ошибки, некорректные обновления, сбои в базе данных или конфликты между модулями могут вызывать непредвиденные остановки работы системы;

Профилактические меры и действия при возникновении ЧС

Пожар – Профилактические меры:

- установка автоматических систем пожаротушения и дымовых датчиков;
- обеспечение наличия огнетушителей вблизи серверов и рабочих мест;
- регулярное проведение противопожарных инструктажей персонала;
- соблюдение правил эксплуатации электропроводки согласно ПУЭ и ГОСТ 12.1.009–76;
- разработка и утверждение плана эвакуации.
- меры реагирования:
- немедленное отключение электроэнергии при обнаружении признаков возгорания;
- эвакуация персонала и клиентов в соответствии с разработанным планом;
- вызов пожарной службы;
- использование первичных средств тушения для локализации очага возгорания до прибытия спасателей.

Отключение электроэнергии – Профилактические меры:

- установка источников бесперебойного питания для серверов;
- организация двойного энергопитания (резервный источник);
- резервное копирование данных на внешние носители или в облачное хранилище;
- регулярное тестирование резервного оборудования;

Меры реагирования:

- переключение на резервное питание при внезапном отключении;
- сохранение текущего состояния системы перед прекращением работы;
- при длительном отключении – переход на ручной способ оформления заказов;
- восстановление работы после подачи электроэнергии с проверкой целостности данных.

Выход из строя оборудования – Профилактические меры:

- наличие запасных устройств и комплектующих;
- дублирование серверного оборудования или использование облачных решений;

- регулярное техническое обслуживание и диагностика оборудования;
- сервисное обслуживание по договору с поставщиками техники;

Меры реагирования:

- замена вышедшего из строя оборудования на резервное;
- восстановление системы из последней резервной копии;
- диагностика и ремонт оборудования в специализированных центрах;
- оповещение пользователей о временных ограничениях в работе.

Киберугрозы – Профилактические меры:

- регулярное обновление программного обеспечения и библиотек;
- использование антивирусного программного обеспечения на сервере;
- настройка фаервола и системы обнаружения вторжений;
- шифрование данных при передаче и хранении;
- контроль доступа к административной части системы (логин/пароль, двухфакторная аутентификация);

Мониторинг сетевых атак и подозрительной активности.

Меры реагирования:

- блокировка подозрительных IP-адресов и пользовательских аккаунтов;
- отключение системы от сети при подтверждённой атаке;
- восстановление системы из защищённой резервной копии;
- анализ инцидента и устранение уязвимостей;
- оповещение администраторов и, при необходимости, служб безопасности.

Сбои программного обеспечения – Профилактические меры:

- тестирование обновлений перед их применением;
- поддержание актуального состояния резервных копий;

- ведение журналов событий (логирование);
- разработка политики управления изменениями в системе;
- применение принципов отказоустойчивости при проектировании ПО.

Меры реагирования:

- откат системы к предыдущей стабильной версии;
- восстановление из резервной копии;
- анализ логов и устранение причины сбоя;
- оповещение пользователей о временной недоступности функционала;
- проведение внутреннего расследования и улучшение процедур тестирования.

Единый план действий при возникновении чрезвычайной ситуации

Для минимизации последствий ЧС рекомендуется разработать и внедрить единый план действий, который должен включать следующие этапы:

- обнаружение инцидента – фиксируются признаки ЧС: отсутствие связи с сервером, признаки возгорания, сообщения об ошибках;
- анализ ситуации – определяется тип ЧС, степень влияния на систему, необходимость включения резервных механизмов;
- принятие мер – выполняются действия по локализации проблемы: отключение питания, блокировка доступа, запуск резервного оборудования;
- информирование – уведомление ответственных лиц, сотрудников, владельцев заведения;
- восстановление – возврат системы к нормальному состоянию с использованием резервных копий, дублированного оборудования или альтернативных методов;
- анализ и улучшения – анализ причин ЧС, корректировка плана реагирования, внедрение дополнительных мер защиты.

План эвакуации и противопожарная подготовка

Для обеспечения безопасности людей в случае возникновения пожара необходимо:

- разработать и утвердить план эвакуации, включающий маршруты выхода, места сбора, обязанности сотрудников;
- обеспечить наличие аварийного освещения, указателей эвакуационных путей и знаков безопасности;
- проводить ежегодные противопожарные инструктажи и учения с персоналом;
- обеспечить наличие огнетушителей, дымовых извещателей и систем пожаротушения;
- обеспечить доступ сотрудников к информации о действиях при пожаре.

Итоговая сводка по основным категориям ЧС и мерам реагирования представлена в таблице 32

Таблица 32 – Итоговая таблица по основным категориям ЧС и мерам реагирования

Категория ЧС	Профилактика	Реагирование
Пожар	Пожарные датчики, огнетушители, обучение персонала	Отключение электричества, эвакуация, вызов МЧС
Отключение электроэнергии	ИБП, резервное питание, резервные копии	Переход на резерв, сохранение данных, восстановление
Выход из строя оборудования	Запасные устройства, облачные решения	Замена оборудования, восстановление из резерва
Киберугрозы	Обновления, шифрование, контроль доступа	Блокировка, отключение, восстановление, анализ уязвимостей
Сбои программного обеспечения	Тестирование, логирование, резервирование	Откат, восстановление, анализ логов

Таким образом, разработка и реализация комплексного плана реагирования на чрезвычайные ситуации позволяет обеспечить устойчивую работу веб-приложения MenuPRO, минимизировать риск потери данных и повысить уровень безопасности как для пользователей, так и для персонала заведения.

ЗАКЛЮЧЕНИЕ

В ходе исследования были изучены современные тенденции в использовании информационных технологий в сфере общественного питания, а также проанализированы существующие решения для цифровизации меню и автоматизации процессов взаимодействия с клиентами. Особое внимание было уделено бесконтактным технологиям, среди которых NFC–интеграция выделяется своей актуальностью, простотой использования и высокой степенью безопасности.

Разработанное мобильное приложение реализует комплексный подход к решению задач ресторанного бизнеса – от мгновенного доступа к меню через NFC до оформления заказа и оплаты. Оно обеспечивает удобство для пользователей за счёт минимального количества действий – всего одно прикосновение смартфона к метке. Для владельцев заведений приложение предоставляет инструменты управления меню и оперативной реакции на изменяющиеся предпочтения посетителей.

Анализ показал, что внедрение подобных решений требует не только технической проработки, но и учёта поведенческих особенностей пользователей, уровня цифровой грамотности и готовности заведений к внедрению мобильных технологий. Несмотря на ограниченную поддержку NFC в некоторых устройствах, преимущества технологии – быстрый доступ и безопасность данных – делают её одним из наиболее перспективных направлений в условиях роста спроса на бесконтактные сервисы.

Таким образом, создание мобильного приложения для заведений общественного питания является актуальным и востребованным решением, которое отвечает запросам современного рынка общепита.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1 Flutter Documentation [Электронный ресурс]. URL: <https://docs.flutter.dev/> (дата обращения: 10.04.2025).

2 Dart Programming Language Documentation [Электронный ресурс]. URL: <https://dart.dev/> (дата обращения: 10.04.2025).

3 Flutter API Reference [Электронный ресурс]. URL: <https://api.flutter.dev/> (дата обращения: 10.04.2025).

4 Getting Started with Flutter [Электронный ресурс]. URL: <https://flutter.dev/docs/get-started/install> (дата обращения: 10.04.2025).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Flutter Documentation [Электронный ресурс]. – Режим доступа: <https://docs.flutter.dev/>
- 2 Dart Programming Language Documentation [Электронный ресурс]. – Режим доступа: <https://dart.dev/>
- 3 Flutter API Reference [Электронный ресурс]. – Режим доступа: <https://api.flutter.dev/>
- 4 Getting Started with Flutter [Электронный ресурс]. – Режим доступа: <https://flutter.dev/docs/get-started/install>
- 5 Flutter Samples [Электронный ресурс]. – Режим доступа: <https://flutter.dev/docs/development/samples>
- 6 Flutter Community GitHub [Электронный ресурс]. – Режим доступа: <https://github.com/flutter/flutter>
- 7 Flutter Cookbook [Электронный ресурс]. – Режим доступа: <https://flutter.dev/docs/cookbook>
- 8 Flutter Widgets Catalog [Электронный ресурс]. – Режим доступа: <https://flutter.dev/docs/development/ui/widgets/catalog>
- 9 Flutter for Android Developers [Электронный ресурс]. – Режим доступа: <https://flutter.dev/docs/resources/android-to-flutter>.
- 10 Flutter for iOS Developers [Электронный ресурс]. – Режим доступа: <https://flutter.dev/docs/resources/ios-to-flutter>.
- 11 Udemy. Flutter & Dart – The Complete Guide [Электронный ресурс]. – Режим доступа: <https://www.udemy.com/course/learn-flutter-dart-from-scratch/>.
- 12 Coursera. Developing Mobile Apps with Flutter [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/specializations/flutter>
- 13 Ларман, К. Применение объектно–ориентированного анализа и проектирования с использованием UML и шаблонов: Введение в ООА/D / Крейг Ларман. – М.: ООО «И.Д. Вильямс», 2020. – 688 с.

14 Макконнелл, Стив. Совершенный код: практическое руководство по разработке программного обеспечения / Стив Макконнелл; пер. с англ. – 2–е изд. – Санкт–Петербург: ООО «Диалектика», 2020. – 900 с.

15 Мартин, Роберт К. Чистый код: создание, анализ и рефакторинг. / Роберт Кесил Мартин; пер. с англ – СПб.: Питер, 2021. – 464 с.

16 Фаулер, Мартин. UML. Основы: введение в язык стандартного объектно–ориентированного моделирования / Мартин Фаулер, Крис Уиллис. – М.: ДМК Пресс, 2021. – 192 с.

17 Буч, Гради. Объектно–ориентированный анализ и проектирование с примерами приложений / Гради Буч. – 3–е изд. – М.: Издательский дом «Вильямс», 2020. – 480 с.

18 Пособие по безопасной работе на персональных компьютерах / разраб. В. К. Шумилин. - М.: НЦ ЭНАС, 2005. - 28 с.

19 Шумилин, В.К. ПЭВМ. Защита пользователя / Шумилин В.К. - М: Охрана труда и социальное страхование, 2001. - 214с.

20 Кардаш, Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ: учеб. пособие / Т. А. Кардаш; АмГУ, ИФФ. - Благовещенск: Изд-во Амур. гос. ун-та, 2002. - 60 с. и др.

ПРИЛОЖЕНИЕ А

```
import 'package:flutter/material.dart';
import 'package:nfc_manager/nfc_manager.dart';
import 'package:flutter_nfc_kit/flutter_nfc_kit.dart';
import 'package:url_launcher/url_launcher.dart'; // Для открытия меню

class NFCReaderScreen extends StatefulWidget {
  @override
  _NFCReaderScreenState createState() => _NFCReaderScreenState();
}

class _NFCReaderScreenState extends State<NFCReaderScreen> {
  String _status = "Ожидание NFC-метки...";
  String? _urlFromTag;

  @override
  void initState() {
    super.initState();
    _checkNFC();
  }

  Future<void> _checkNFC() async {
    bool isAvailable = await NfcManager.instance.isSupported();
    if (!isAvailable) {
      setState(() {
        _status = "NFC не поддерживается на этом устройстве";
      });
      return;
    }

    bool isEnabled = await NfcManager.instance.isEnabled ?? false;
    if (!isEnabled) {
      setState(() {
        _status = "Включите NFC в настройках устройства";
      });
    }
  }
}
```

Рисунок А.1 – Пример кода для NFC–взаимодействия

ПРИЛОЖЕНИЕ Б

```
Future<void> sendOrderToFirebase(String venueId, Position venueLocation) async
    if (_cart.isEmpty || _tableId == null) return;

    // Проверка геолокации
    await getUserLocation();
    bool isInVenue = await isUserInVenue(venueLocation, 50); // радиус 50 метров
    if (!isInVenue) throw Exception("Вы вне зоны заведения");

    // Формирование заказа
    CollectionReference orders = FirebaseFirestore.instance.collection('orders');
    await orders.add({
        'table_id': _tableId,
        'venue_id': venueId,
        'created_at': FieldValue.serverTimestamp(),
        'status': 'new',
        'items': _cart.map((item) => {
            'product_id': item.product.id,
            'name': item.product.name,
            'price': item.product.price,
            'quantity': item.quantity,
        }).toList(),
    });
```

Рисунок Б.1 – Пример кода модуля заказов с геолокацией

ПРИЛОЖЕНИЕ В

```
Future<void> sendOrderToERP(Order order) async {  
    final url = 'https://api.menu-pro.ru/erp/order';  
    final headers = {'Content-Type': 'application/json'};  
    final body = jsonEncode({  
        'order_id': order.id,  
        'table_number': order.tableNumber,  
        'total_price': order.totalPrice,  
        'items': order.items.map((item) => {  
            'product_name': item.name,  
            'quantity': item.quantity,  
            'external_id': item.externalId  
        }).toList(),  
    });  
  
    final response = await http.post(  
        Uri.parse(url),  
        headers: headers,  
        body: body,  
    );  
  
    if (response.statusCode == 200) {  
        print('Заказ успешно синхронизирован');  
        updateSyncStatus(order.id, 'synced');  
    } else {  
        print('Ошибка синхронизации: ${response.body}');  
        updateSyncStatus(order.id, 'error');  
    }  
}
```

Рисунок В.1 – Пример кода: отправка заказов

ПРИЛОЖЕНИЕ Г

Таблица Г.1 – Бизнес–модель Остервальдера

Блок	Описание
1. Ценностное предложение	Система предоставляет клиентам моментальный доступ к меню через NFC, возможность выбора блюд и оформления заказа без участия официанта. Для заведений это повышает скорость обслуживания, снижает нагрузку на персонал и улучшает клиентский опыт.
2. Целевые клиенты	Кафе и рестораны, фуд–корт и точки общепита, малый и средний бизнес в сфере HoReCa, посетители заведений, ценящие удобство и скорость обслуживания
3. Каналы дистрибуции	Прямые продажи через личные встречи с владельцами заведений, партнёрские программы с поставщиками оборудования, облачный SaaS–сервис с подпиской, участие в выставках и мероприятиях в сфере цифровизации общепита
4. Отношения с клиентами	онлайн–поддержка пользователей, регулярные обновления функционала, сбор обратной связи и доработка продукта под запросы клиентов, обучение работе с системой
5. Ключевые виды деятельности	Разработка и поддержка клиентской части (веб и мобильного приложений), тестирование и оптимизация UX/UI, продвижение и поиск первых клиентов, поддержание безопасности и отказоустойчивости системы
6. Ключевые ресурсы	Команда разработчиков (веб разработка, мобильная разработка), серверное оборудование и облачная инфраструктура: домен, CMS MODX, API и база данных; юридическая поддержка и защита ИС
7. Ключевые партнёры	Поставщики NFC–меток, облачные провайдеры (например, Yandex Cloud или AWS), IT–интеграторы, промышленные партнёры (сети ресторанов)
8. Структура издержек	Разработка и тестирование программного обеспечения, закупка оборудования (NFC–меток, серверов), маркетинг и продвижение, администрирование и техническая поддержка
9. Источники дохода	Ежемесячная подписка за использование сервиса, интеграционные услуги

ПРИЛОЖЕНИЕ Д

Техническое задание

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Полное наименование системы

Веб-платформа с использованием NFC технологий для автоматизации процессов на предприятиях общественного питания.

Условное обозначение: MenuPRO.

1.2 Плановые сроки:

- начало работ: 01.10.2024;
- окончание работ: 30.05.2025.

2 НАЗНАЧЕНИЕ И ЦЕЛИ СИСТЕМЫ

2.1 Назначение системы

Платформа предназначена для автоматизации процессов обслуживания клиентов, управления заказами и оплатой на предприятиях общественного питания с использованием NFC-технологий. Она обеспечивает бесконтактное взаимодействие между клиентами и персоналом, а также повышает эффективность работы заведений.

2.2 Цели создания системы

Цели создания системы:

- сократить время обслуживания клиентов за счет внедрения бесконтактных решений;
- увеличить средний чек и лояльность клиентов благодаря персонализированным предложениям;
- снизить операционные затраты за счет автоматизации рутинных процессов.

Критерии оценки успеха:

- уменьшение времени обработки заказа на 30%;

Продолжение ПРИЛОЖЕНИЯ Д

- рост среднего чека на 15%;
- снижение количества ошибок при оформлении заказов на 50%.

3 ТРЕБОВАНИЯ К СИСТЕМЕ

3.1 Требования к системе в целом

Структура и функционирование: Модуль управления заказами; Модуль отправки заказа в мессенджер Telegram; Модуль администрирования.

Численность и квалификация персонала: Для работы с платформой требуется администратор с базовыми навыками работы с компьютером. Обучение проводится разработчиком.

Надежность: Система должна обеспечивать бесперебойную работу при нагрузке до 1000 одновременных пользователей.

Безопасность: Защита данных от несанкционированного доступа и утечек.

Эргономика: Интуитивно понятный интерфейс для клиентов и персонала.

3.2 Требования к функциям системы:

- автоматизация процесса оформления заказов через NFC-метки;
- поддержка бесконтактной оплаты через NFC;
- формирование аналитических отчетов по продажам, популярным блюдам и клиентским предпочтениям;
- возможность интеграции с существующими системами учета (например, iiko, R-Keeper).

3.3 Требования к видам обеспечения:

- математическое обеспечение: алгоритмы обработки данных о заказах;
- информационное обеспечение: использование базы данных MySQL;

Продолжение ПРИЛОЖЕНИЯ Д

- программное обеспечение: совместимость с Windows, macOS, iOS, Android;
- техническое обеспечение: поддержка NFC-устройств (смартфоны, терминалы).

4 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

4.1 Этапы работ:

- анализ предметной области;
- проектирование архитектуры платформы;
- разработка прототипа;
- тестирование и доработка;
- ввод в эксплуатацию.

5 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

5.1 Контроль качества:

- проверка соответствия функционала техническому заданию;
- тестирование производительности и безопасности.

5.2 Приемка системы

Платформа считается принятой после подписания акта приемки, подтверждающего выполнение всех требований.

6 ТРЕБОВАНИЯ К ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ

6.1 Подготовка персонала

Проведение обучения сотрудников работе с платформой.

6.2 Техническая подготовка

Установка необходимого оборудования (NFC-метки).

7 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

7.1 Перечень документов:

- техническое задание;
- руководство пользователя;

Продолжение ПРИЛОЖЕНИЯ Д

- руководство администратора;
- программный код.

8 ИСТОЧНИКИ РАЗРАБОТКИ

8.1 Нормативные документы:

- ГОСТ 34.602-89;
- ГОСТ 34.601-90.

8.2 Литература:

- Материалы по NFC-технологиям;
- Исследования рынка общественного питания.

ПРИЛОЖЕНИЕ Е

Логическая модель данных в нотации IDEF0 для платформы MenuPRO

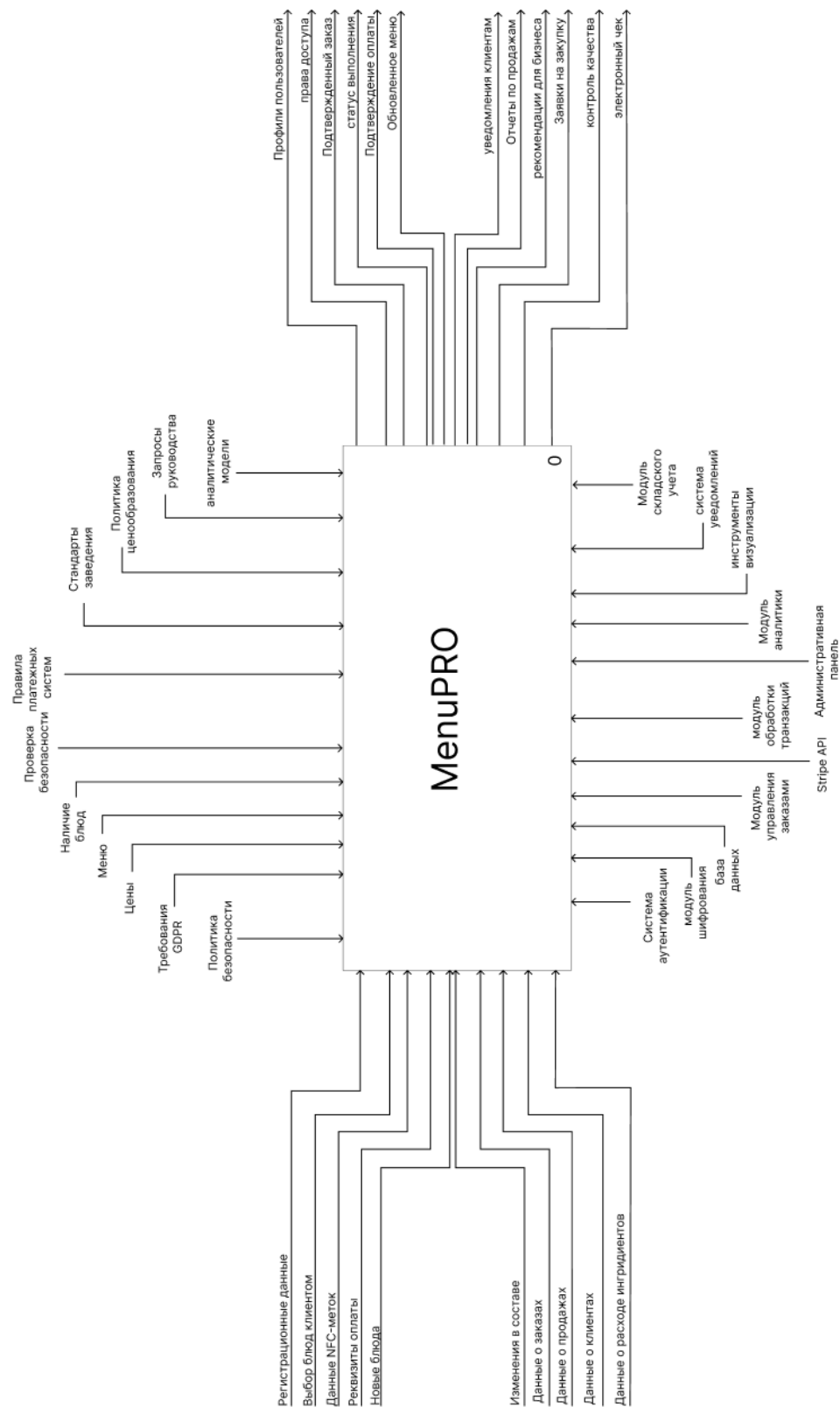


Рисунок Е.1 – Нотация IDEF0 декомпозиции 0 уровня

Продолжение ПРИЛОЖЕНИЯ Е

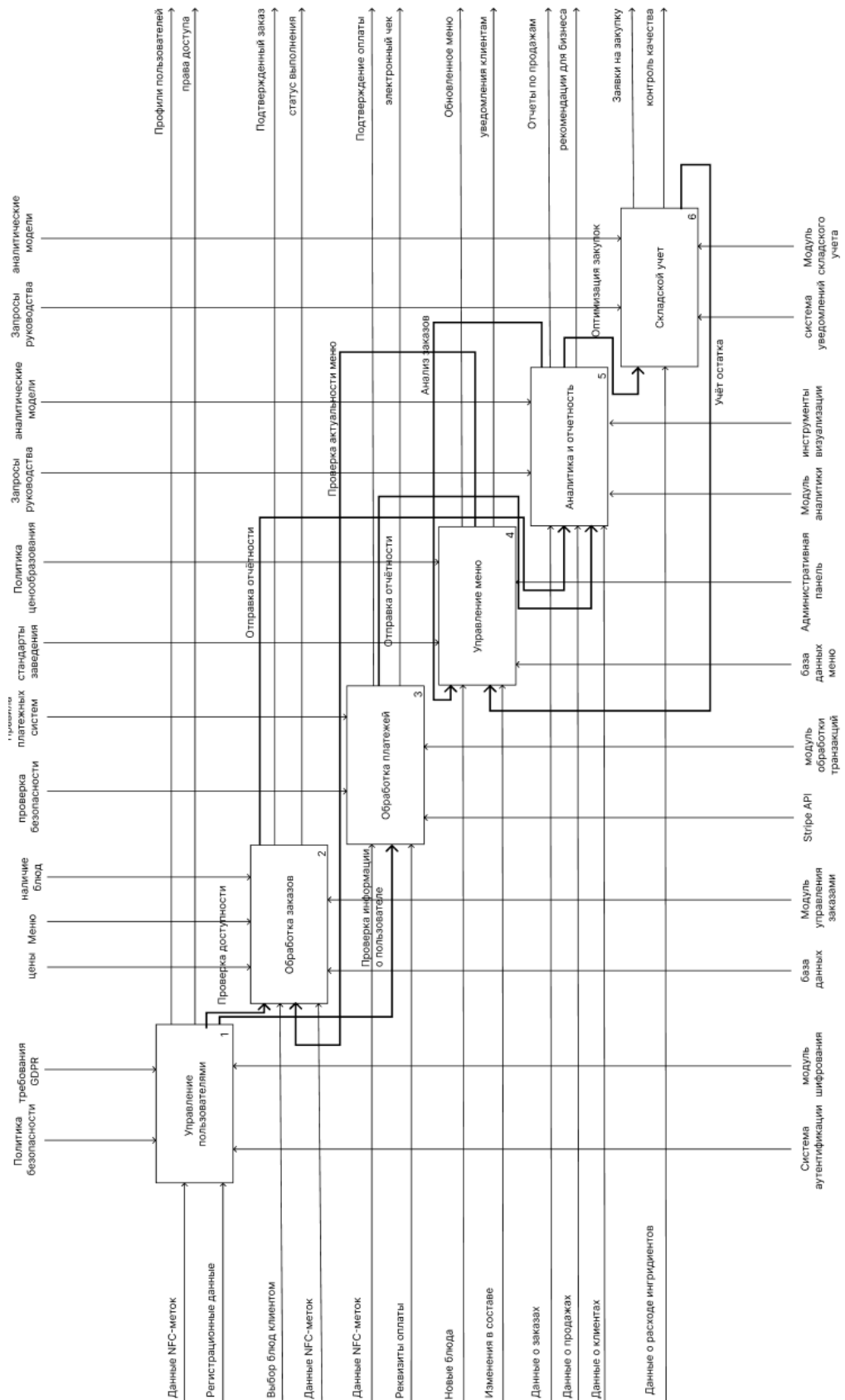


Рисунок Е.2 – Нотация IDEF0 декомпозиции 1 уровня