

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт цифровых технологий
(наименование)

Департамент бакалавриата

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения для управления лояльностью клиентов»

Обучающийся

Д.А. Арсентьев

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.т.н., доцент, О.В. Аникина

ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Выпускная квалификационная работа посвящена актуальной задаче цифровой трансформации бизнес-процессов в сфере общественного питания, в частности повышению эффективности взаимодействия с клиентами через внедрение цифровых решений.

Целью данной работы является разработка программного обеспечения, направленного на автоматизацию процесса управления программой лояльности в кафе быстрого обслуживания.

Объектом исследования выступает деятельность современного кафе быстрого обслуживания.

Предметом исследования выступает процесс управления лояльностью клиентов, включая начисление и использование бонусов, персонализацию предложений.

Работа структурирована на три взаимосвязанные главы, каждая из которых решает отдельный блок задач, в совокупности обеспечивающих достижение поставленной цели.

В первой главе проводится анализ деятельности кафе быстрого обслуживания, рассматриваются его организационная структура, ключевые бизнес-процессы, взаимодействие с клиентами, а также выявляются проблемы, связанные с отсутствием или неэффективностью системы лояльности.

Во второй главе осуществляется логическое и архитектурное проектирование разрабатываемой системы. Обосновывается выбор архитектурного подхода, наиболее подходящего для масштабируемых решений.

Третья глава посвящена физической реализации программного обеспечения.

В заключении подведены итоги проделанной работы, сформулированы достигнутые результаты и подтверждено выполнение поставленной цели.

Оглавление

Введение.....	4
Глава 1 Функциональное моделирование деятельности предприятия	8
1.1 Анализ деятельности ресторана быстрого питания.....	8
1.2 Концептуальное проектирование процесса управления лояльностью клиентов.....	10
1.3 Анализ существующих разработок.....	15
1.4 Постановка задачи на разработку системы управления лояльностью клиентов.....	18
Глава 2 Логическое проектирование информационной системы управления лояльностью клиентов	22
2.1 Логическая модель информационной системы управления лояльностью клиентов.....	22
2.2 Проектирование базы данных системы.....	25
2.3 Выбор архитектуры информационной системы управления лояльностью клиентов.....	31
Глава 3 Физическое проектирование информационной системы управления лояльностью клиентов	36
3.1 Выбор инструментов разработки системы управления лояльностью клиентов.....	36
3.2 Описание программных модулей системы управления лояльностью клиентов.....	39
3.3 Разработка информационной системы управления лояльностью клиентов.....	43
3.4 Расчет показателей экономической эффективности проекта.....	50
Заключение.....	59
Список используемой литературы и используемых источников.....	61

Введение

Выпускная квалификационная работа посвящена актуальной и востребованной в условиях современной цифровой экономики задаче разработке программного обеспечения для автоматизации процесса управления лояльностью клиентов в заведениях общественного питания, в частности в кафе быстрого обслуживания.

В условиях растущей конкуренции и повышения ожиданий гостей в отношении персонализированного сервиса, отсутствие системного подхода к работе с лояльностью становится серьёзным ограничением для удержания клиентов и увеличения среднего чека. Ручное ведение бонусных программ, отсутствие аналитики поведения посетителей, несвоевременные или нерелевантные предложения всё это снижает удовлетворённость клиентов и напрямую влияет на прибыльность заведения.

Уверенность в том, что каждый гость получает персонализированный опыт, а его действия в рамках программы лояльности учитываются точно и в реальном времени, обеспечивается только за счёт систематизированного и автоматизированного подхода. Ответственность за этот процесс, как правило, ложится на администраторов или маркетологов кафе. Им необходимо видеть полную историю взаимодействия с каждым клиентом, отслеживать динамику посещений, начисление и использование бонусов, а также оперативно реагировать на изменения в поведении гостей.

Для эффективного управления лояльностью требуется централизованное хранилище данных, в котором фиксируется вся информация о клиентах: их профиль, история заказов, накопленные бонусы, использованные акции, предпочтения и обратная связь. Такая система позволяет персоналу быстро принимать обоснованные маркетинговые и операционные решения, повышая вовлечённость и частоту визитов.

Разрабатываемое программное обеспечение автоматизирует ключевые этапы взаимодействия с клиентом в рамках программы лояльности и позволяет

управлять десятками, а в перспективе сотнями участников программы одновременно, не теряя контроля над деталями и обеспечивая высокий уровень персонализации.

Целью данной работы является разработка программного обеспечения, предназначенного для автоматизации процесса управления лояльностью клиентов в кафе быстрого обслуживания.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ существующих подходов к управлению лояльностью в сфере общественного питания;
- изучить бизнес-процессы кафе быстрого обслуживания и выявить точки взаимодействия с клиентами, пригодные для интеграции системы лояльности;
- выполнить выбор технологического стека и инструментов для разработки программного обеспечения;
- спроектировать и реализовать систему управления лояльностью с поддержкой регистрации клиентов, начисления и списания бонусов, персонализированных уведомлений и аналитики;
- обосновать экономическую эффективность внедрения разработанного решения.

Объектом исследования выступает деятельность современного кафе быстрого обслуживания, функционирующего в условиях высокой конкуренции и растущих требований к качеству клиентского опыта. В качестве практического примера рассматривается реальное или модельное кафе, ориентированное на массового потребителя и стремящееся повысить повторные визиты через цифровые инструменты.

Предметом исследования является процесс управления лояльностью клиентов, от регистрации гостя в программе до анализа его поведения и формирования персонализированных маркетинговых акций. Особое внимание уделяется автоматизации таких операций, как: идентификация клиента при

заказе, начисление бонусов, проверка условий акций, отправка уведомлений и генерация отчётов по эффективности программы.

Работа структурирована на три взаимосвязанные главы, каждая из которых направлена на решение отдельного блока задач, в совокупности обеспечивающих достижение цели исследования.

В первой главе проводится детальный анализ деятельности кафе быстрого обслуживания. Рассматриваются его организационная структура, ключевые бизнес-процессы, точки взаимодействия с клиентами, а также выявляются проблемы, связанные с отсутствием или неэффективностью системы лояльности.

На основе собранной информации выполняется моделирование текущего состояния процесса управления лояльностью с использованием методологии IDEF0. Диаграммы наглядно демонстрируют этапы, где теряется вовлечённость клиентов или возникают операционные издержки, и обосновывают необходимость их автоматизации.

По итогам анализа формулируются функциональные и нефункциональные требования к разрабатываемой системе, включая требования к удобству интерфейса, безопасности персональных данных и масштабируемости.

Вторая глава посвящена логическому и архитектурному проектированию будущей системы. Рассматриваются возможные архитектурные подходы и обосновывается выбор оптимального решения для условий кафе быстрого обслуживания.

Особое внимание уделено проектированию реляционной базы данных, в которой будут храниться сведения о клиентах, их заказах, бонусных баллах, акциях, истории взаимодействий и сотрудниках. Приводятся диаграммы «сущность–связь», описывается нормализация таблиц, ключевые индексы и связи между сущностями.

Также в главе представлены диаграммы прецедентов и последовательностей, отражающие взаимодействие пользователей с системой и логику работы основных модулей.

Третья глава содержит описание физической реализации программного обеспечения. Для разработки выбраны современные и востребованные технологии.

В главе представлены основные интерфейсы системы. Подробно описаны алгоритмы работы ключевых модулей, автоматическое начисление бонусов при оплате, проверка условий участия в акциях.

В заключении подведены итоги проделанной работы, сформулированы достигнутые результаты и подтверждено выполнение поставленной цели. Создана полнофункциональная система управления лояльностью, способная интегрироваться в инфраструктуру кафе быстрого обслуживания и значительно повысить эффективность маркетинговых коммуникаций, удержание клиентов и общую прибыльность бизнеса.

Глава 1 Функциональное моделирование деятельности предприятия

1.1 Анализ деятельности ресторана быстрого питания

Ресторан быстрого питания – это место, где люди могут быстро перекусить. Как правило рестораны быстрого питания находятся в торгово-развлекательных центрах, куда приезжают отдыхать семьями.

К функциям ресторана быстрого питания можно отнести закупку продуктов и ингредиентов для приготовления блюд, непосредственное приготовление блюд, указанных в меню, расчет клиентов на кассе.

Основной целью ресторана быстрого питания является качественное обслуживание клиентов [2].

Ресторан быстрого питания осуществляет приготовление блюд на заказ.

Цель работы ресторана быстрого питания – обеспечить быстрое приготовление блюд достойного качества за конкурентную цену. Если клиенту нравится ассортимент блюд, качество продуктов из которых они приготовлены, качество работы персонала и прочие критерии ресторана, он посетит его вновь и вновь.

Следовательно, прибыль компании будет расти, что очень важно. Контроль надлежащего качества продуктов и прибыль компании, именно эти цели являются актуальными в данной организации.

Структура компании [14] больше похожа на линейно-функциональную, потому что предприятие разделено на отделы в каждом отделе есть начальник, которому подчиняется линейный персонал, рассмотрим более подробно каждый из них:

- отдел маркетинга;
- отдел закупок;
- ресторан;
- отдел бухгалтерии.

Упрощенная организационная схема компании показана на рисунке 1.

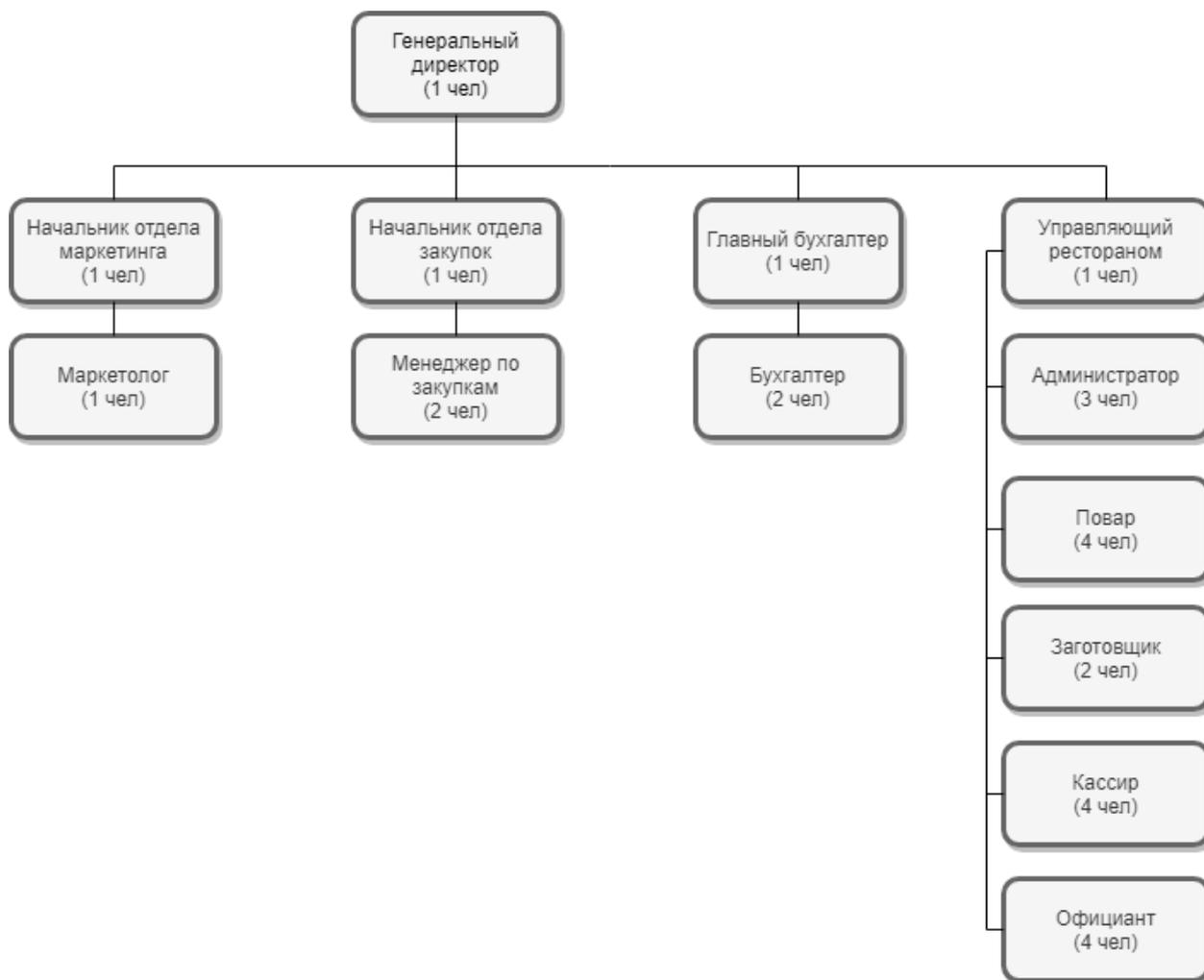


Рисунок 1 – Организационная структура ресторана быстрого питания

Представим краткое описание каждого из отделов.

Отдел закупок занимается закупками качественных продуктов и поиском новых поставщиков.

Отдел маркетинга занимается продвижением бренда.

Отдел бухгалтерии – занимается расчетами стоимости блюд, расчета прибыли предприятия, расчетом заработных плат для выплаты персоналу внутри компании.

В самом ресторане быстрого обслуживания происходят основные процессы, связанные с обслуживанием клиентов и повышения лояльности

покупателя. Для мотивации клиентов вернуться в ресторан следующий раз в ресторане постоянно действуют скидки и акции на блюда в определенное время посещения.

За каждым из отделов стоит ответственное лицо. Во главе всего предприятия стоит генеральный директор.

1.2 Концептуальное проектирование процесса управления лояльностью клиентов

Диаграмма IDEF0, описывающая деятельность ресторана быстрого питания, показана на рисунке 2.

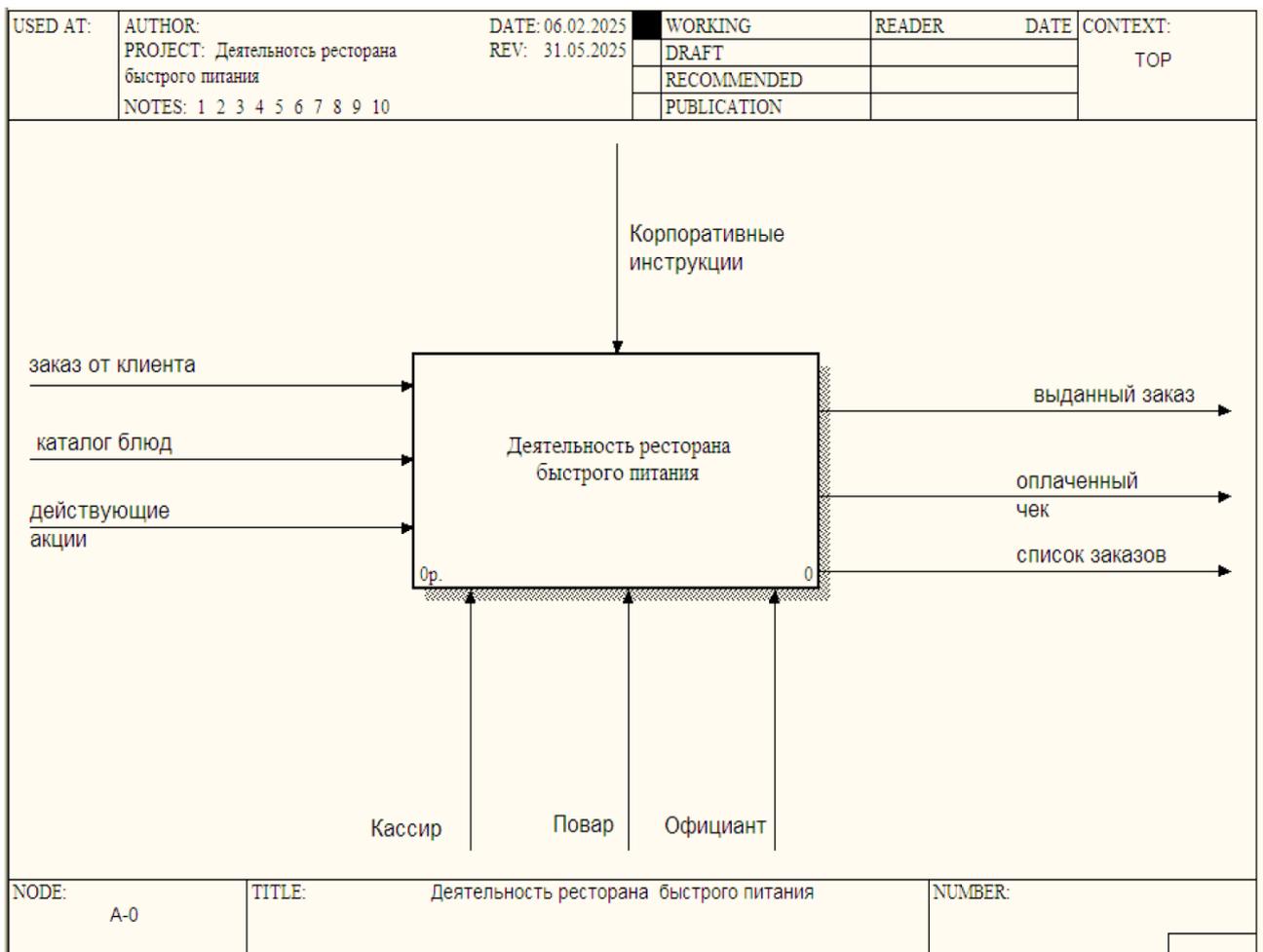


Рисунок 2 – Контекстная диаграмма работы ресторана быстрого питания

Проведем декомпозицию контекстной диаграммы [13] разбив работу ресторана быстрого питания на процессы:

- принять и обработать заказ;
- приготовить и выдать заказ;
- принять оплату за заказ.

Диаграмма второго уровня показана на рисунке 3.

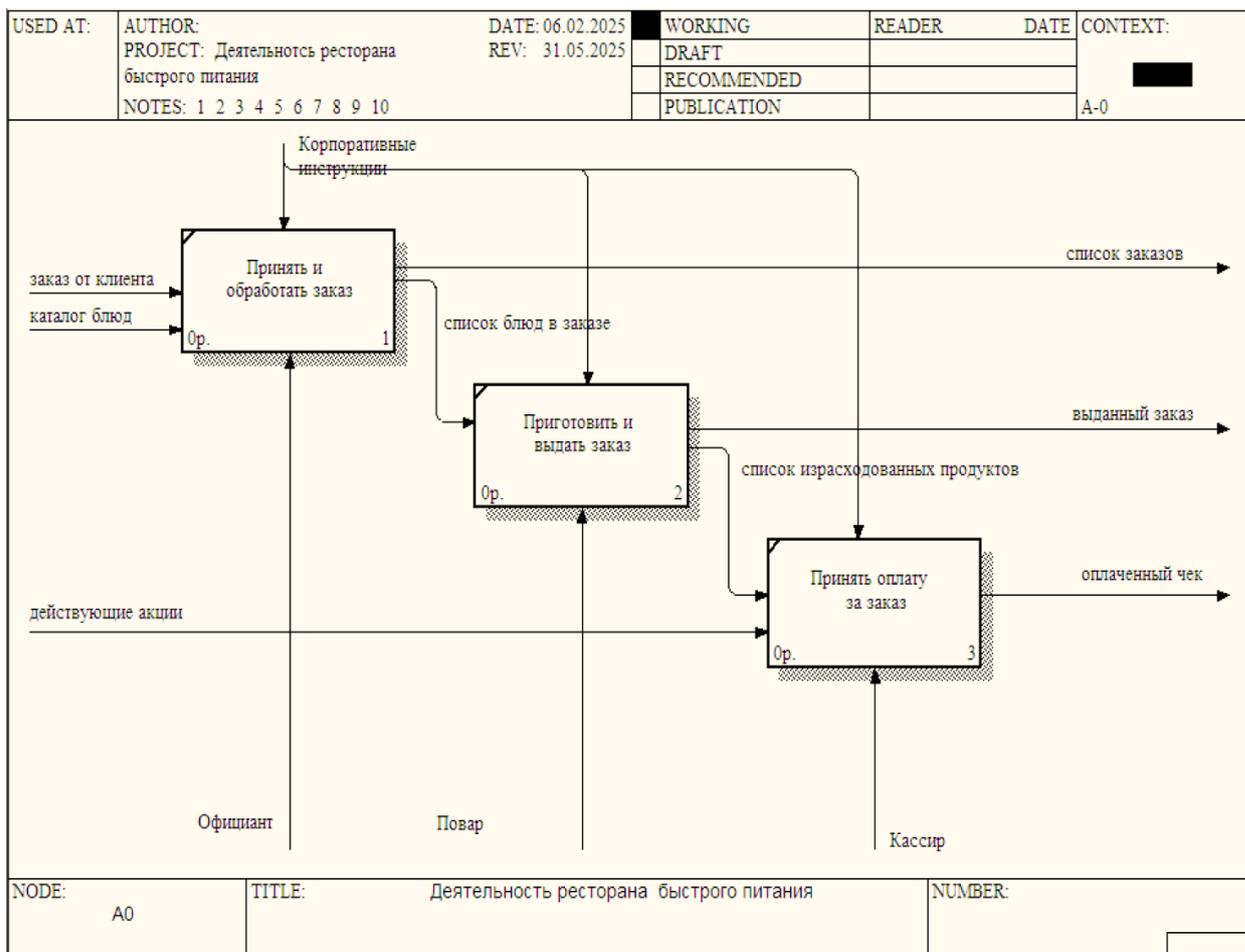


Рисунок 3 – Декомпозиция работы ресторана быстрого питания

На основе диаграмм информационной модели, построенных в нотации IDEF0 [12], можно сделать вывод и определить какие информационные потоки нуждаются в автоматизации:

- ведение данных заказа (блюдо, количество);
- ведение списка заказов (данные заказа, количество, дата заказа, блюдо);

- ведение данных о картах лояльности клиентов (номер карты, клиент, дата выдачи, процент допустимой скидки).

Рассмотрим более подробно функциональный блок «Принять оплату за заказ», рисунке 4.

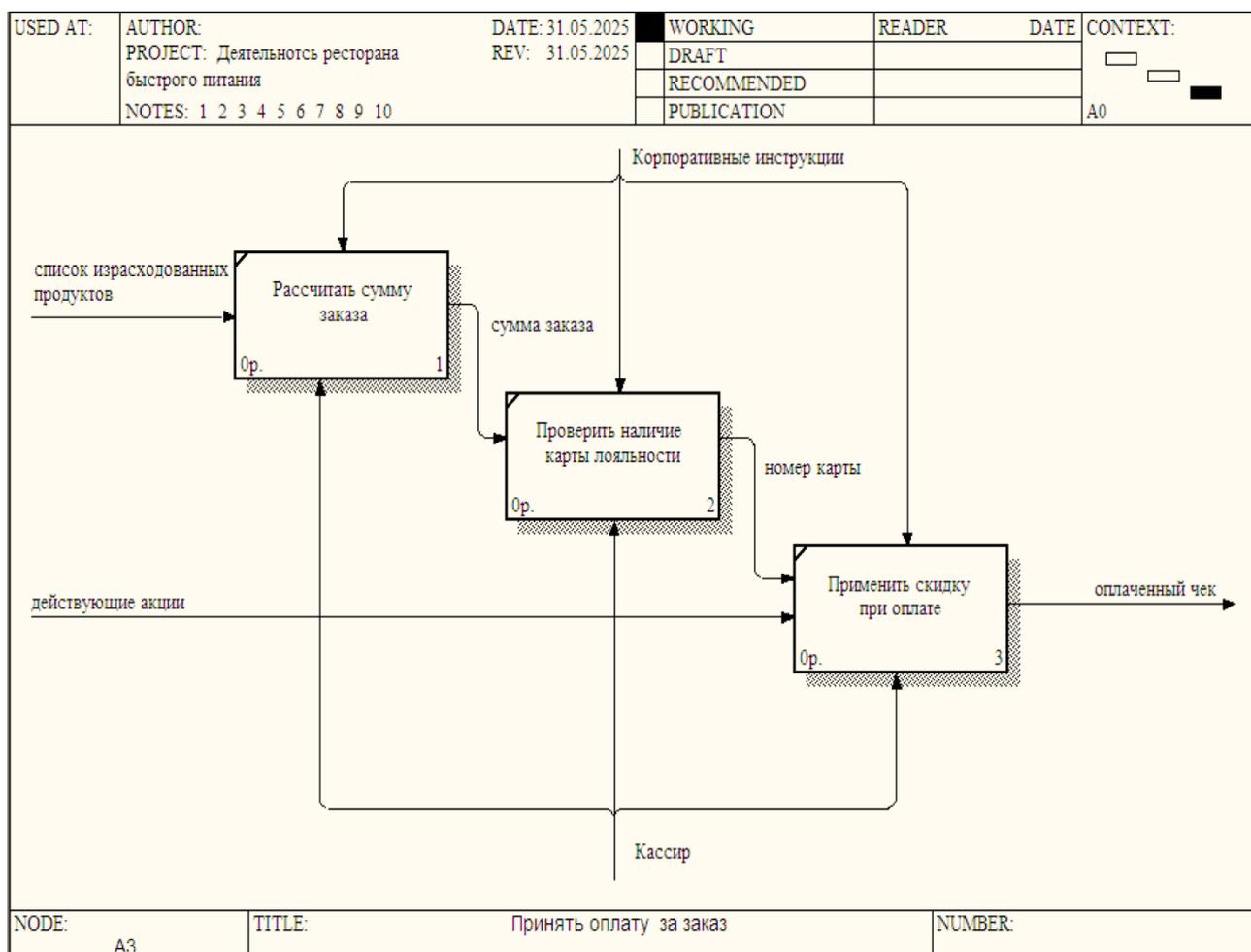


Рисунок 4 – Декомпозиция функционального блока «Принять оплату за заказ»

На кассе кассир при расчёте клиента за заказ должен проверить наличие у клиента карты лояльности, если карта у клиента есть, то кассир может посмотреть скидку согласно типу карты лояльности, если у клиента нет карты и он желает ее оформить, то кассир обязан зарегистрировать и выдать карту клиента для предоставления скидки при посещении ресторана быстрого питания.

Обновленные бизнес-процессы [20] позволят сотрудникам одновременно обращаться к одной базе данных и получать нужную информацию, что значительно сократит время обслуживания клиентов и повысит качество обслуживания, также с помощью базы данных будет вестись управление лояльностью клиентов.

Представим процесс после внедрения информационной системы для управления лояльностью клиентов на рисунке 5.

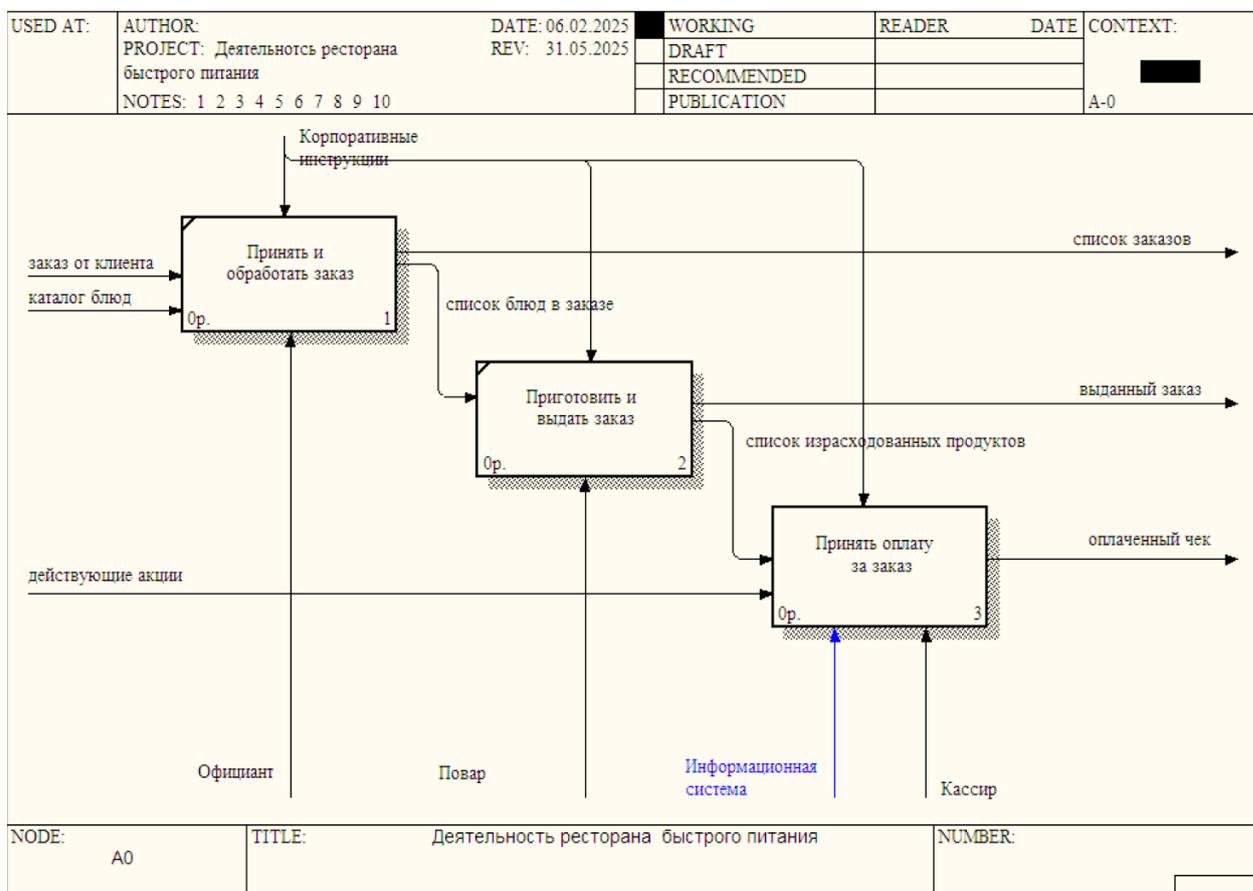


Рисунок 5 – Декомпозиция деятельности ресторана быстрого питания после внедрения системы

На диаграмме показано, что информационная система будет внедряться в процесс кассира, когда он рассчитывает клиентов за заказы. Декомпозиция функционального блока «Принять оплату за заказ» позволит увидеть более

подробно как изменится процесс после автоматизации. На рисунке 6 показана декомпозиция процесса [15].

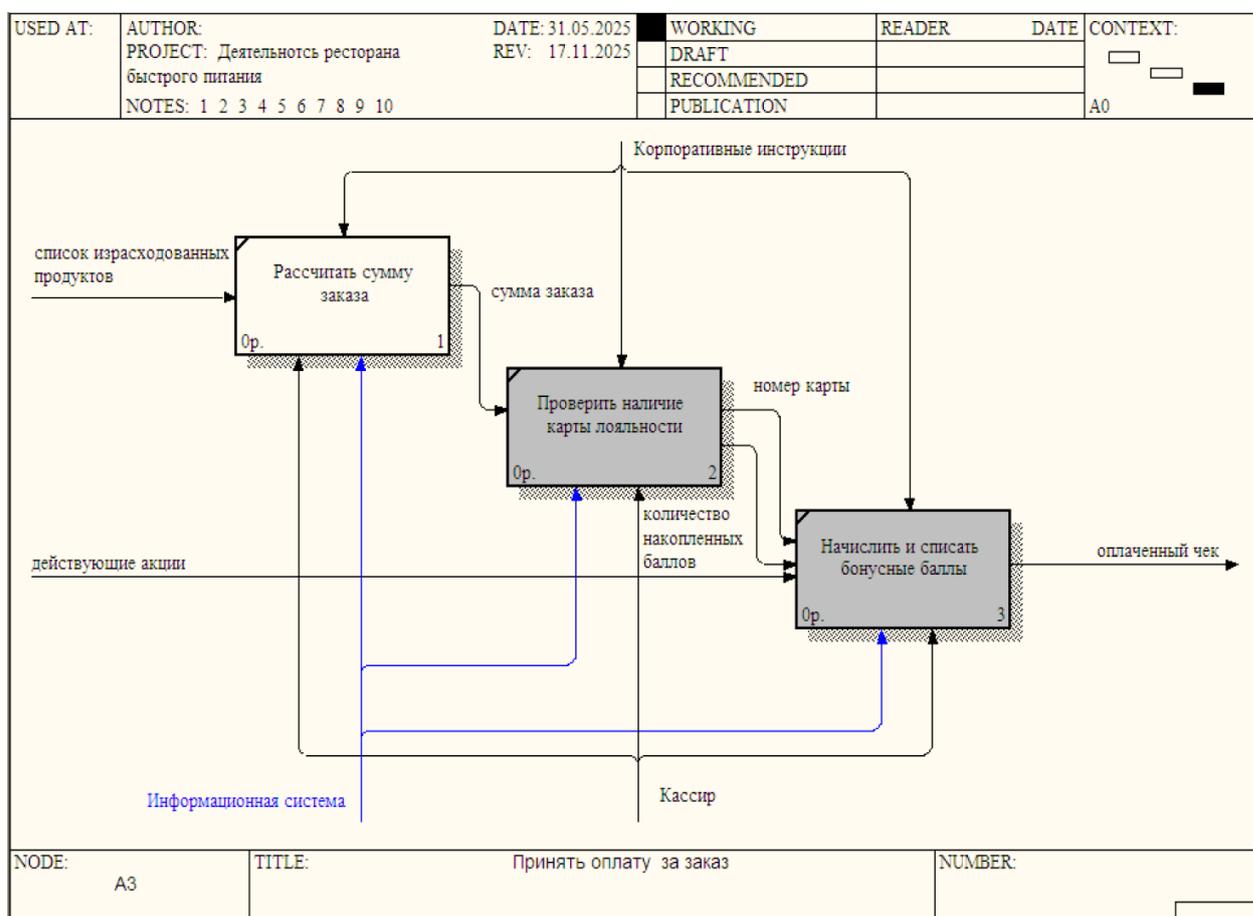


Рисунок 6 – Декомпозиция функционального блока «Принять оплату за заказ»

После создания программного обеспечения для управления лояльностью клиентов кассиру будет доступна вся информация о клиентах, которые приходят в ресторан на постоянной основе, количестве накопленных баллов при оплате за заказ, а также появится возможность списывать накопленные баллы, тем самым снижая сумму оплаты.

Если у клиента нет накопленных баллов и он пришел в ресторан впервые, кассир также сможет предложить ему завести карту лояльности, при этом физическая карта не обязательна при посещении ресторана, кассир

сможет найти данные клиента по номеру телефона на которую будет оформляться новая карта лояльности клиента.

1.3 Анализ существующих разработок

Имеется довольно большое количество различных программ для управления лояльностью клиентов, однако некоторые из них являются узконаправленными программами, кроме того, стоимость таких программ является довольно высокой.

На рынке существуют следующие виды программ для управления лояльностью клиентов: «Loymax Loyalty», «Locards», «10 cups».

«Программа «Loymax Loyalty» гозволяет создавать и управлять программами лояльности любой сложности в едином интерфейсе и без использования программирования» [2].

«Программа позволяет:

- увеличивать трафик и средний чек, используя скидки в виде % от суммы чека или товара, в виде спеццены или фиксированной суммы;
- возвращать клиентов с помощью виртуальных баллов за точечные покупки, отвечающие нужным критериям;
- настраивать уровни достижений и поощрений, дополнять их акциями с адаптацией условий для разных статусов;
- поддерживать купоны на следующую покупку, купоны с защитной полоской и скрытой наградой, электронные купоны и т. п.;
- использовать счетчики для подсчета действий или ограничения преференций» [2].

«Программа «Locards» — Электронные карты лояльности для ваших клиентов. Все управление электронными картами происходит через удобный web-интерфейс — личный кабинет. На основе ваших пожеланий мы

подготовим для вас личный кабинет, настроим программу лояльности и подготовим карты.

Благодаря интеграции с системой учета продаж баланс бонусных баллов обновляется автоматически после проведения покупки.

Дизайн и текстовое содержимое карты вы можете изменить в личном кабинете в любой необходимый момент. Все изменения отображаются на карте в режиме реального времени» [3].

«10 cups – лёгкий инструмент для увеличения продаж на 30%. Всё это благодаря сервисам и продуманной системе лояльности, в которой нет ничего лишнего» [4].

«Приложение 10Cups позволяет заведениям сделать гостей постоянными клиентами с помощью понятной и простой механики. Гость становится вашим клиентом, с которым вы можете всегда взаимодействовать – рассылать уведомления, рассказывать об акциях и новинках, продавать абонементы и получать предзаказы» [4].

«Вся коммуникация происходит через приложение, где реализован удобный механизм оценок и обратной связи» [4].

«Преимущества:

- клиенту больше не придется предъявлять чеки / карты лояльности или оставаться без подарка, если он забыл визитку с отметками — смартфон всегда с собой;
- никаких дополнительных устройств или программ. Только служебный телефон на iOS или Android, где установлено приложение. Такой подход позволит сократить расходы на типографию;
- анализировать продажи и их динамику, частоту использования приложения и количество начисляемых бонусов станет проще. А подарить услугу «просто так» уже не выйдет;
- функция обратной связи через приложение поможет устранить недочеты, замеченные клиентами. Рейтинг лучших сотрудников будет строиться на основании оценок покупателей» [4].

Таблица 1 отражает основные характеристики представленных программных средств.

Таблица 1 - Сравнительный анализ программных продуктов

Программа/Критерии	Loymax Loyalty	Locards	10 cups
Доступность	-	-	+
Единая базы для хранения данных	+	+	+
Применение суммарной скидки на кассе	-	-	-
Сопровождение программы	-	+	+

Рассмотренные платформы и информационные системы, позволяющие выполнять управление лояльностью клиентов по аналогичным критериям не совсем отвечают требованиям, которые необходимы для потребностей ресторана быстрого питания. Представленные программные продукты [9] не обеспечивают быстрое освоение функционала и достаточно дорогостоящие, также не все системы гарантируют дальнейшее сопровождение, чтократно затруднит ее использование в будущем.

Учитывая вышесказанное, возникает необходимость разработки системы, полностью адаптированной к процессу управления лояльностью клиентов.

Учитывая возможности программного и аппаратного обеспечения компании, требуется разработать современное программное обеспечение, избегая таких минусов имеющихся разработок, как высокая стоимость внедрения и сопровождения и низкая ориентированность на пользователя с наличием разного уровня профессиональной подготовки [5].

Также необходимо уделить особое внимание надежности создаваемому программному продукту и простоте его интерфейса.

1.4 Постановка задачи на разработку системы управления лояльностью клиентов

Разрабатываемая система позволит вести управление лояльностью клиентов.

Требования к ИС представлены в таблице 2.

Таблица 2 – Классификация требований к ИС

Требование	Статус	Полезность	Риск
Функциональные требования			
Регистрация и авторизация клиентов	Обязательно	Позволяет идентифицировать клиентов и отслеживать их активность	Утечка персональных данных при недостаточной защите
Начисление и списание бонусных баллов	Обязательно	Стимулирует повторные покупки и повышает вовлеченность	Ошибки в начислении/списании баллов могут вызвать недовольство
Интеграция с системой оплаты	Обязательно	Обеспечивает автоматическое начисление баллов при оплате	Уязвимость данных при сбоях безопасности
Управление акциями и персонализированными предложениями	Обязательно	Повышает конверсию и удовлетворенность клиентов	Неверная настройка акций может привести к финансовым потерям
Удобство использования			
Понятный интерфейс	Обязательно	Упрощает взаимодействие клиента с программой лояльности	Некачественный дизайн снижает вовлеченность и удержание

Продолжение таблицы 2

Требование	Статус	Полезность	Риск
Поддержка нескольких языков	Желательно	Увеличивает охват среди туристов и многоязычных клиентов	Увеличение сроков и стоимости разработки
Быстрый доступ к текущему балансу и акциям	Обязательно	Повышает прозрачность и мотивацию клиента	Задержки при отображении данных могут вызывать раздражение
Надежность			
Резервное копирование данных	Обязательно	Защищает данные при сбоях системы	Дополнительные затраты на инфраструктуру
Защита от несанкционированного доступа	Обязательно	Обеспечивает безопасность данных	Уязвимости в системе безопасности
Производительность			
Обработка большого числа транзакций в часы пик	Обязательно	Гарантирует стабильную работу в загруженные периоды	Перегрузка серверов может вызвать сбои в начислении баллов
Мгновенное обновление информации об акциях	Обязательно	Обеспечивает актуальность предложений для клиентов	Ошибки синхронизации могут привести к неверной информации
Поддерживаемость			
Возможность обновления системы	Обязательно	Позволяет вносить улучшения	Проблемы совместимости с новыми модулями
Техническая поддержка	Обязательно	Решение проблем пользователей	Зависимость от квалификации команды поддержки
Ограничения			
Работа только в ресторанах сети	Реальное ограничение	Упрощает управление программой, но ограничивает гибкость	Снижение ценности программы для клиентов без доступа к партнёрам
Зависимость от интернета	Реальное ограничение	Невозможность начисления баллов или участия в акциях без подключения	Потребность в стабильном соединении может ограничить использование

Основная цель разрабатываемого программного продукта – упростить процесс для кассира, который занимается применением скидки на кассе и регистрацией новых карт лояльности.

Система создается для ресторана быстрого питания как вспомогательный программный продукт ведения лояльности клиентов. Правильное и своевременное управление лояльностью клиентов обеспечит быстрое обслуживание клиентов и повышение репутации ресторана быстрого питания.

Пользовательский интерфейс [6] должен обладать функциями, помогающими пользователям выполнять все действия, связанные с регистрацией и ведением лояльностью клиентов.

Пользователями информационной системы будут сотрудники ресторана быстрого питания, которые будут обслуживать клиентов и вести расчеты с клиентами на кассе.

Архитектура информационной системы [7] должна позволять масштабировать рабочее место сотрудника, работающего в ней с учетом расширения компании и списка пользователей.

Вывод по первой главе

В первой главе проведён всесторонний анализ деятельности ресторана быстрого питания с акцентом на процессы управления лояльностью клиентов.

Рассмотрена организационная структура предприятия, выделены ключевые функциональные подразделения и их роли в обеспечении эффективной работы.

На основе методологии IDEF0 выполнено функциональное моделирование основных бизнес-процессов, таких как приём и обработка заказа, приготовление блюд и расчёт с клиентом, проведен их детальный анализ с целью выявления узких мест, что позволило выявить информационные потоки, требующие автоматизации. Составлены требования к программному обеспечению в нотации FURPS+/

Особое внимание уделено процессу расчёта на кассе, где реализуется взаимодействие с программой лояльности.

Анализ существующих решений в области управления лояльностью показал, что большинство коммерческих продуктов либо не отвечают специфике работы ресторана быстрого питания, либо обладают высокой стоимостью и недостаточной простотой использования.

Это обосновало необходимость разработки собственной системы, ориентированной на удобство персонала, надёжность, масштабируемость и интеграцию в существующие бизнес-процессы.

Таким образом, первая глава заложила теоретическую и аналитическую основу для проектирования и последующей реализации специализированной информационной системы управления лояльностью клиентов, соответствующей реальным потребностям ресторана быстрого питания.

Глава 2 Логическое проектирование информационной системы управления лояльностью клиентов

2.1 Логическая модель информационной системы управления лояльностью клиентов

Логическое моделирование программного обеспечения для управления лояльностью клиентов осуществляется с использованием разных методологий, которые помогают понять логику [8] программного обеспечения.

UML диаграммы используются при проектировании системы проектировщиками, и на этапе реализации программного продукта программистами. UML помогает правильно построить логику системы при проектировании [16].

ВPMN используется при моделировании бизнес-процессов или деятельности определенного подразделения или компании в целом [17].

Логическое моделирование позволит представить работу системы изнутри и показать взаимодействие пользователей с системой, определить границы системы и функциональность ее.

Выделим границы системы и пользователей, которые будут работать с ней для управления лояльностью клиентов в ресторане быстрого обслуживания. Диаграмма вариантов использования представлена на рисунке 7.

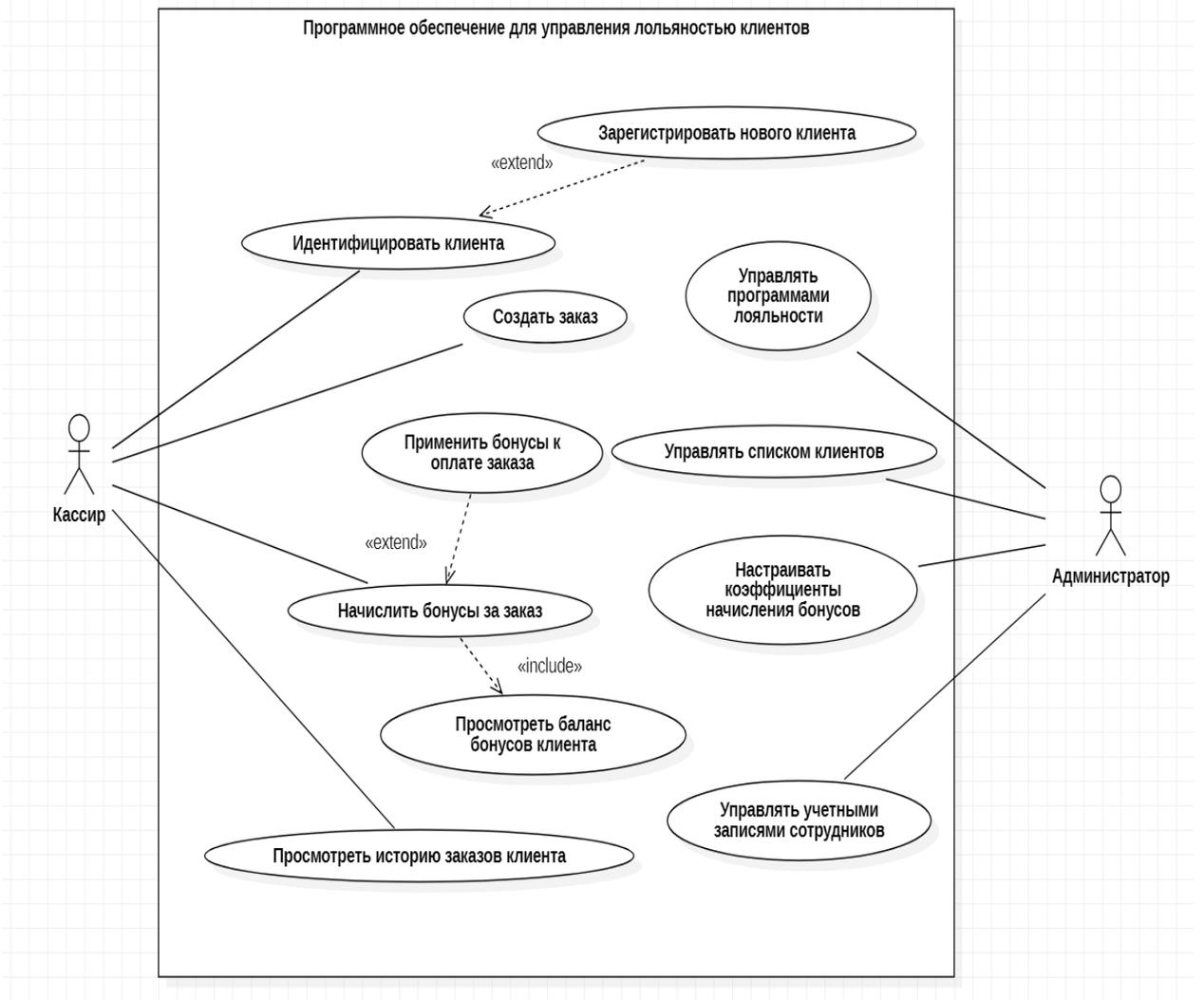


Рисунок 7 – Диаграмма вариантов использования системы для управления лояльностью клиентов в ресторане быстрого обслуживания

Основным пользователем системы для управления лояльностью клиентов является кассир. У кассира будет возможность регистрировать данные нового клиента и идентифицировать клиента по данным, которые уже имеются в информационной системе.

После создания заказа кассир имеет возможность начислять бонусы за заказ, также есть возможность просматривать баланс бонусов клиента и применять в последующем эти бонусы при оплате заказа. Кассир, управляя лояльностью клиентов просматривает историю заказов клиента при необходимости.

Администратор системы сможет выполнять действия в системе, связанные с управлением данных пользователей, в которые входят управление учетными данными сотрудников ресторана. Администратор также настраивает коэффициенты начисления бонусов и управляет программами лояльности. Управлять списком клиентов администратор сможет в системе при необходимости, когда нужно внести корректировки или добавить новые поля или функции [18].

Управление лояльностью клиента на кассе позволяет кассиру осуществлять процесс расчета скидки и применения ее на кассе. Процесс представлен более подробно и пошагово на рисунке 8.

На диаграмме показаны три объекта взаимодействующие при процессе управления лояльностью клиента.

Кассир при получении оплаты за заказ обязан проверить накопленные баллы у клиента и при желании клиента применить их при оплате. Кассир вводит данные номер телефона клиента чтобы найти его данные в базе данных, если клиент уже был зарегистрирован, система выводит данные клиента и данные о его накопленных баллах, если клиент пришел в ресторан впервые кассир должен предложить клиенту и внести его данные в систему для его идентификации при повторном приходе и накоплении баллов.

Найденные данные клиента позволяют просмотреть весь список клиентов, также открыть данные клиента и просмотреть накопленные баллы у клиента. Если у клиента есть баллы, то кассир может применить их при оплате за заказ снизив сумму заказа на количество накопленных баллов.

Таким образом ресторан быстрого обслуживания мотивирует клиентов вернуться в ресторан.

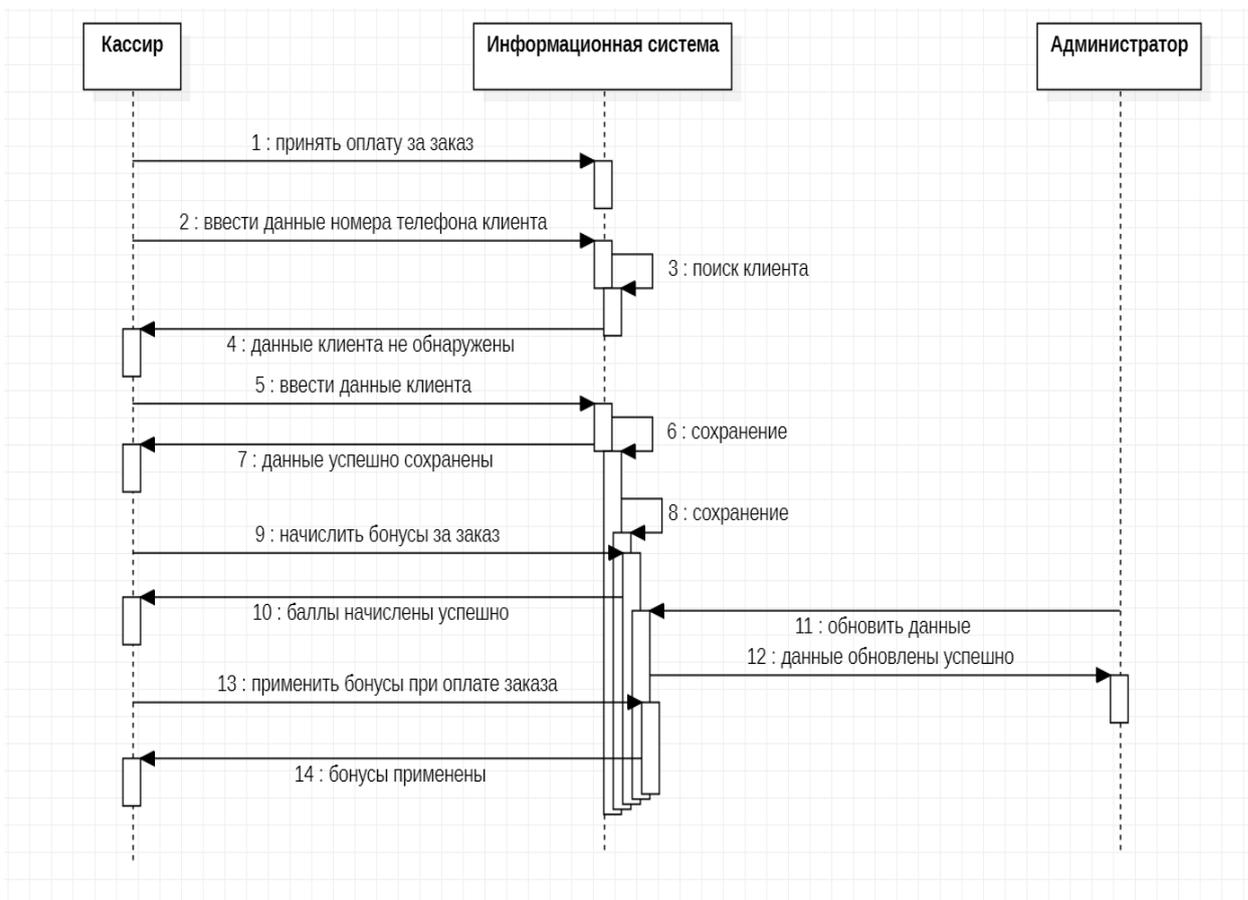


Рисунок 8 – Диаграмма последовательности процесса «Прием оплаты за заказ»

Представленные диаграммы вариантов использования и последовательности позволяют выделить основные объекты для хранения и управления данными [19].

2.2 Проектирование базы данных системы

Управление лояльностью клиентов затрагивает хранение личных данных клиентов, таких как фамилия и имя, также телефон, который будет являться критерием поиска клиента в базе данных. Для начисления баллов клиенту необходимо вести данные о заказах, так как баллы будут начисляться согласно сумме заказа и программе лояльности, о которой также нужно хранить данные в базе данных.

Для управления лояльностью клиента необходимы данные о:

- клиентах;
- заказах;
- бонусных баллах;
- программах лояльности.

На рисунке 9 показана концептуальная модель данных информационной системы управления лояльностью клиента.

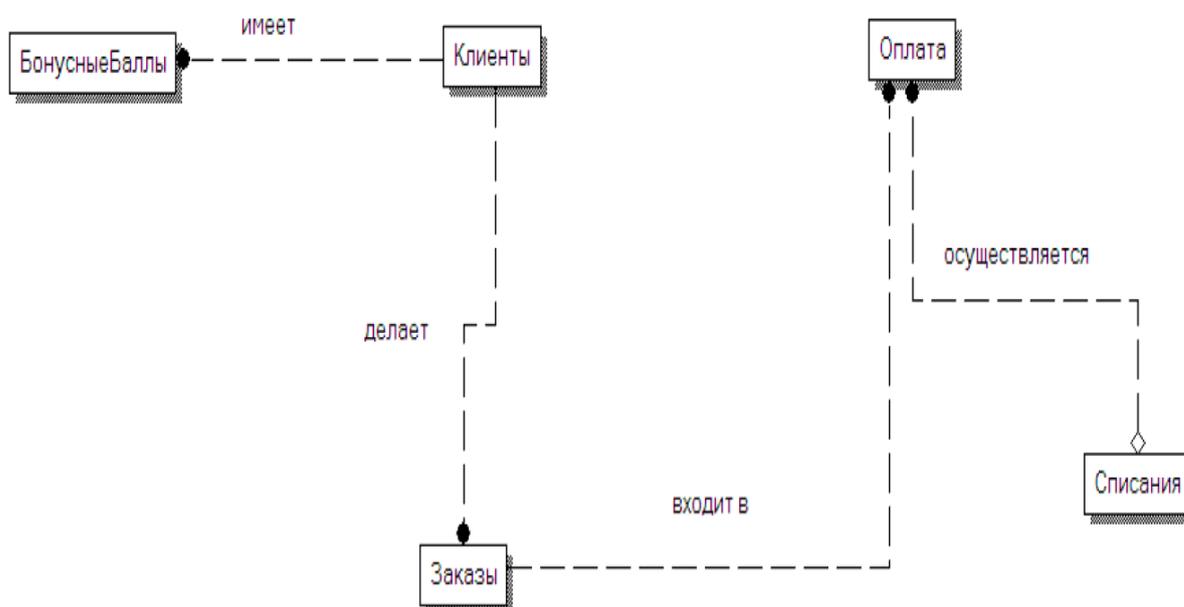


Рисунок 9 — Концептуальная модель данных информационной системы управления лояльностью клиента

Каждая таблица должна иметь атрибуты, определяющие характеристику данных хранящихся в базе данных [10]. В таблице 3 представлены атрибуты сущностей базы данных.

Таблица 3 – Атрибуты сущностей базы данных ИС управления лояльности клиентов

Название сущности	Название атрибута	Тип поля
БонусныеБаллы	id_баланса	Первичный ключ (Счетчик)
	клиент	Внешний ключ
	начислено	Короткий текст
	использовано	Короткий текст
	дата_обновления	Дата
Заказы	id_заказа	первичный ключ (Счетчик)
	дата_заказа	Дата
	клиент	Внешний ключ
	сумма	Денежный
Клиенты	id_клиента	Первичный ключ (Счетчик)
	фамилия	Короткий текст
	имя	Короткий текст
	телефон	Короткий текст
	дата_регистрации	Дата
Оплата	id_оплаты	Первичный ключ (Счетчик)
	дата_оплаты	Дата
	id_списания	Внешний ключ
	id_заказа	Внешний ключ
	сумма	Денежный
Списания	id_списания	Первичный ключ (Счетчик)
	сумма	Денежный
	остаток	Денежный
	дата_списания	Дата

На рисунке 10 показана логическая модель данных ИС управления лояльностью клиентов.

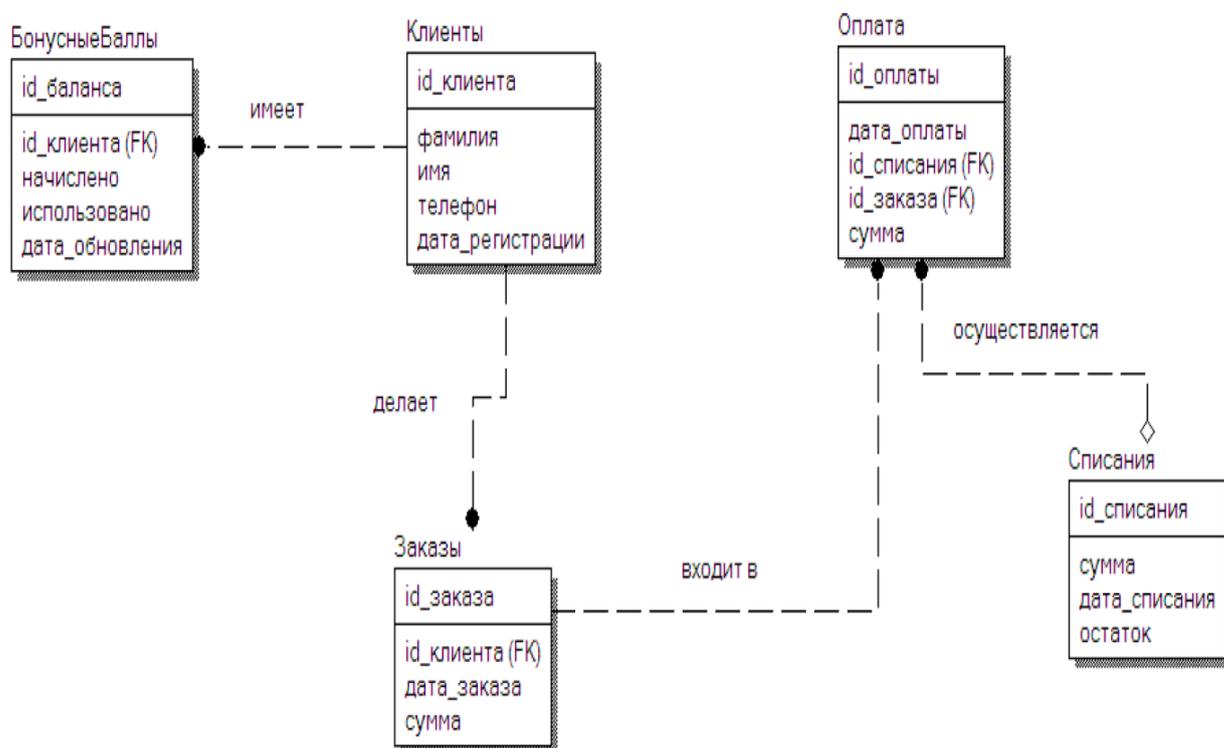


Рисунок 10 – Логическая модель данных информационной системы управления лояльностью клиента

На рисунке 11 показана физическая модель данных информационной системы управления лояльностью клиента.

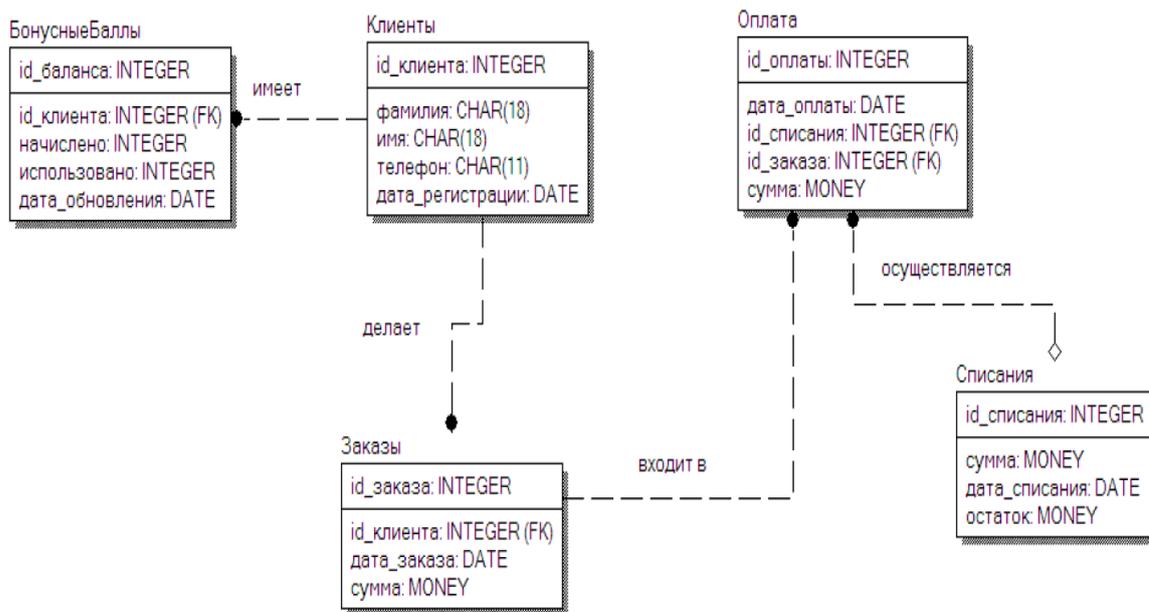


Рисунок 11 – Физическая модель базы данных информационной системы управления лояльностью клиента

В СУБД создание таблиц происходит с помощью команды CREATE TABLE.

```

CREATE TABLE Клиенты (
  id_клиента AUTOINCREMENT PRIMARY KEY,
  фамилия TEXT NOT NULL,
  имя TEXT NOT NULL,
  телефон TEXT NOT NULL,
  дата_регистрации DATE NOT NULL
);
  
```

```

CREATE TABLE Заказы (
  id_заказа AUTOINCREMENT PRIMARY KEY,
  id_клиента INTEGER NOT NULL,
  дата_заказа DATE NOT NULL,
  сумма MONEY NOT NULL,
  );
  
```

```

FOREIGN KEY (id_клиента) REFERENCES Клиенты(id_клиента)
);
CREATE TABLE БонусныеБаллы (
id_баланса AUTOINCREMENT PRIMARY KEY,
id_клиента INTEGER NOT NULL,
начислено INTEGER,
использовано INTEGER,
дата_обновления DATE NOT NULL,
FOREIGN KEY (id_клиента) REFERENCES Клиенты(id_клиента)
);
CREATE TABLE ПрограммыЛояльности (
id_программы AUTOINCREMENT PRIMARY KEY,
название TEXT NOT NULL,
минимальная_сумма MONEY,
коэффициент_начисления TEXT,
активна BOOLEAN
);
CREATE TABLE УчастиеВПрограмме (
id_участия AUTOINCREMENT PRIMARY KEY,
id_клиента INTEGER NOT NULL,
id_программы INTEGER NOT NULL,
дата_начала DATE NOT NULL,
FOREIGN KEY (id_клиента) REFERENCES Клиенты(id_клиента),
FOREIGN KEY (id_программы) REFERENCES
ПрограммыЛояльности(id_программы)
);

```

С помощью команд INSERT INTO и VALUES происходит вставка строк внутри той или иной таблицы.

Все данные, которые вносятся впервые в систему будут автоматически сохранены в базу данных, где будут защищены от постороннего вмешательства и будут храниться в одном месте [16].

Администратор системы обеспечивает сохранность данных и защиту их от внешнего вмешательства и устранение потери данных в процессе обработки и обмена.

Физическая реализация основывается на данных проектирования базы данных.

2.3 Выбор архитектуры информационной системы управления лояльностью клиентов

Архитектура программного обеспечения должна быть спроектирована таким образом, чтобы обеспечивать эффективное и быстрое взаимодействие пользователя с данными системы. Это предполагает не только надёжное хранение информации в базе данных, но и возможность оперативной её обработки и выдачи по запросу. Пользователь должен иметь возможность формировать выборки данных с помощью гибких и интуитивно понятных запросов, настроенных под конкретные критерии, фильтрацию по дате последнего визита, накопленным бонусам, участию в акциях или типу заказа. При этом время отклика системы должно быть минимальным, чтобы поддерживать удобство и плавность пользовательского опыта.

Особое внимание при проектировании архитектуры уделяется её способности к масштабированию. Поскольку система управления лояльностью клиентов разрабатывается для ресторана быстрого обслуживания, она должна быть готова к росту как в плане увеличения числа пользователей, так и в расширении функциональности (например, добавление новых типов вознаграждений, интеграция с внешними сервисами доставки или маркетплейсами). Выбранная архитектурная модель должна гибко

адаптироваться к меняющимся бизнес-требованиям и техническим нагрузкам, обеспечивая стабильную работу даже в пиковые часы [11].

В ходе анализа были рассмотрены две основные архитектурные парадигмы монолитная и микросервисная.

Монолитная архитектура предполагает разработку всего приложения как единого целого, все компоненты от обработки пользовательских запросов до взаимодействия с базой данных собраны в один исполняемый модуль. Такой подход отличается простотой в разработке, отладке и развёртывании, особенно на ранних этапах проекта. Тестирование также упрощено, поскольку вся логика сосредоточена в одном месте. Для пользователя это означает единый точечный доступ ко всем функциям системы: отправка данных и получение информации происходят через один интерфейс, что повышает удобство использования. Однако при масштабировании монолита могут возникнуть сложности любое изменение требует пересборки и повторного развёртывания всего приложения, а рост кодовой базы усложняет поддержку.

Микросервисная архитектура, напротив, строится на принципе разделения приложения на множество независимых, слабосвязанных сервисов, каждый из которых отвечает за отдельную бизнес-функцию такую как, управление профилями клиентов, расчёт бонусов, обработка акций или интеграция с кассовой системой. Такая структура обеспечивает высокую гибкость: отдельные сервисы можно разрабатывать, обновлять, масштабировать и развёртывать независимо друг от друга. Это особенно ценно для динамично развивающегося бизнеса, где требуется быстрая адаптация к новым требованиям. Однако микросервисный подход несёт и определённые риски: усложняется обеспечение согласованности данных между сервисами, возрастает потребность в надёжной сетевой инфраструктуре, а также увеличиваются затраты на мониторинг и управление всей системой в целом.

Таким образом, выбор между монолитом и микросервисами зависит от стратегических целей проекта, текущих ресурсов и ожидаемых темпов роста. Для начального этапа развёртывания системы лояльности в ресторане может

быть целесообразно использовать монолитную архитектуру с возможностью последующей эволюции в микросервисную модель по мере масштабирования.

В разрабатываемой системе управления лояльностью клиентов для ресторана быстрого обслуживания была выбрана монолитная архитектура в качестве основной модели построения программного обеспечения. Такой выбор обусловлен рядом практических и организационных факторов, характерных для текущего этапа развития проекта.

Во-первых, система изначально ориентирована на решение узкого круга задач: регистрация клиентов, начисление и списание бонусных баллов, управление персонализированными акциями, интеграция с кассовой системой и отображение информации о лояльности как сотрудникам ресторана, так и самим клиентам. Объединение всех этих компонентов в единое монолитное приложение позволило значительно упростить как этап разработки, так и последующее тестирование и развёртывание. Все модули, включая пользовательский интерфейс, бизнес-логику и доступ к базе данных реализованы в рамках одной кодовой базы и развернуты как единый сервис, что обеспечивает низкую задержку при внутреннем взаимодействии компонентов.

Во-вторых, монолитная архитектура позволила создать единый и последовательный пользовательский интерфейс, где как кассир ресторана, так и клиент могут мгновенно получать доступ ко всей необходимой информации от баланса бонусов до текущих акций без необходимости взаимодействия с несколькими независимыми системами. Запросы к данным обрабатываются локально в рамках одного приложения, что минимизирует время отклика и повышает удобство эксплуатации.

Кроме того, с учётом того, что система разрабатывается для одного ресторана или небольшой сети точек, отсутствует острая необходимость в высокой степени горизонтального масштабирования на данном этапе. Монолитная структура полностью удовлетворяет требованиям по

производительности и надёжности, особенно при грамотной оптимизации базы данных и использовании кэширования.

Тем не менее, при проектировании монолита были заложены принципы модульности и слабой связанности внутри кодовой базы, что в будущем облегчит возможную декомпозицию системы на микросервисы, при расширении сети или интеграции с внешними платформами. Таким образом, выбранная архитектура представляет собой сбалансированное решение, сочетающее простоту внедрения и поддержки с потенциалом для дальнейшей эволюции.

На рисунке 12 показана монолитная архитектура.

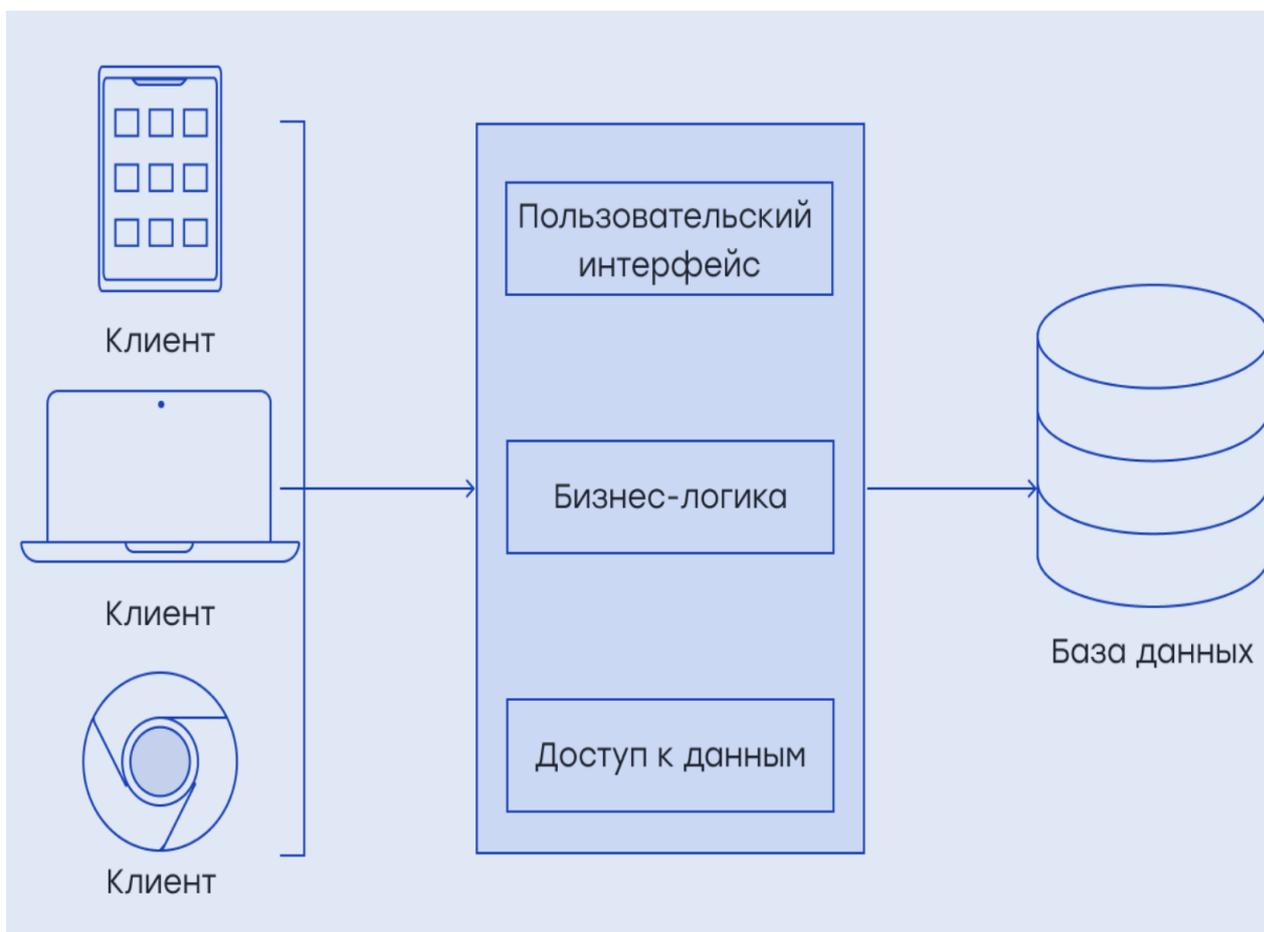


Рисунок 12 – Монолитная архитектура

Монолитная архитектура позволит пользователям быстро и наиболее удобно получать данные для управления лояльностью клиента ресторана быстрого обслуживания.

Вывод по второй главе

Во второй главе выполнено логическое проектирование информационной системы управления лояльностью клиентов для ресторана быстрого питания. На основе анализа требований и бизнес-процессов, описанных в первой главе, была разработана логическая модель системы с использованием стандартных методологий UML и BPMN. Диаграммы вариантов использования и последовательности позволили чётко определить роли пользователей, их взаимодействие с системой и основные сценарии работы, включая регистрацию клиентов, начисление и списание бонусных баллов, а также применение скидок на кассе.

Особое внимание уделено проектированию базы данных. Выделены ключевые сущности и определены их атрибуты, связи и ограничения. Представлены концептуальная, логическая и физическая модели данных, а также SQL-скрипты для создания таблиц, что обеспечивает целостность и надёжность хранения информации.

Кроме того, обоснован выбор архитектуры информационной системы. Учитывая масштаб и специфику деятельности ресторана быстрого питания, была выбрана монолитная архитектура как наиболее простая в разработке, внедрении и эксплуатации, при этом обеспечивающая достаточную производительность и удобство взаимодействия пользователя с системой.

Таким образом, вторая глава заложила прочную логическую и структурную основу для последующей реализации программного продукта, обеспечив соответствие функциональным требованиям, удобство использования и техническую реализуемость.

Глава 3 Физическое проектирование информационной системы управления лояльностью клиентов

3.1 Выбор инструментов разработки системы управления лояльностью клиентов

Разработка информационной системы осуществляется с помощью различных инструментов, которые необходимо подобрать перед реализацией проекта. Рассмотрим язык программирования высокого уровня C#.

«C Sharp – это объектно-ориентированный язык программирования» [21].

«Он предусматривает следующие преимущества:

- строгую типизацию;
- сохранение концепций объектно-ориентированного программирования;
- функциональность;
- достаточно мощный инструментарий;
- стабильную работу через Visual Studio;
- компактный и легко читаемый код;
- понятный даже новичкам синтаксис.

Минусы:

- синтаксис, безграмотное применение которого провоцирует разного рода ошибки утилит;
- библиотеки, которые не подойдут для выполнения специфических задач;
- читаемость кода, конкуренты смогут с легкостью «разобрать» программное обеспечение и изучить его состав» [21].

Являясь объектно-ориентированным языком, он много перенял у Java и C++. Как и Java, C# изначально предназначался для веб-разработки, и примерно 75% его синтаксических возможностей такие же, как у Java.

Объектно-ориентированный подход позволяет строить с помощью C# крупные, но в то же время гибкие, масштабируемые и расширяемые приложения» [22].

«C# уже давно поддерживает много полезных функций:

- инкапсуляция,
- наследование,
- полиморфизм,
- перегрузка операторов,
- статическая типизация.

При этом он всё ещё активно развивается, и с каждой новой версией появляется всё больше интересного – например лямбды, динамическое связывание, асинхронные методы и т.д.

Первая версия языка вышла вместе с релизом Microsoft Visual Studio .NET в феврале 2002 года» [22].

«Под C#, нередко имеют в виду технологии платформы .NET (Windows Forms, WPF, ASP.NET, Xamarin)» [22].

«В основе .NET – общезыковая среда исполнения Common Language Runtime (CLR), благодаря чему платформа поддерживает несколько языков: наряду с C# это VB.NET, C++, F#, а также различные диалекты других языков, привязанные к .NET, например, Delphi.NET. Код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) – своего рода ассемблер платформы .NET.

Общезыковая среда исполнения CLR и базовая библиотека классов – это основа для целого стека технологий, которые разработчики могут задействовать при создании разных приложений» [22].

«Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML. Переняв многое от своих предшественников –

языков C++, Java, Delphi, Модула и Smalltalk – C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++)» [22].

«C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем. (Однако эта закономерность была нарушена с выходом C# 3.0, представляющим собой расширения языка, не опирающиеся на расширения платформы .NET.) CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.» [22].

Сравнение языков программирования представлено в таблице 4.

Таблица 4 – Сравнение языков программирования

Критерий сравнения	C++	C#	Java
Наличие фреймворков и библиотек	+	+	+
Строгая типизация данных	-	+	+

Продолжение таблицы 4

Критерий сравнения	C++	C#	Java
Постоянное развитие	-	+	-
Базовая библиотека классов	-	+	+
Популярность	-	+	-
Сложность изучения	+	-	-

Microsoft Visual Studio 2022 – представляет собой набор интегрированных сред разработки, он, как и многие был разработан программистами Microsoft как бесплатная и ограниченная по функциям версия несвободной Microsoft Visual Studio [18].

3.2 Описание программных модулей системы управления лояльностью клиентов

Для успешного внедрения разработанной системы управления лояльностью клиентов потребуется выделить дополнительных организационных и технических ресурсов. В частности, будет привлечена команда специалистов по внедрению информационных систем, которые обеспечат подготовку необходимой ИТ-инфраструктуры, которая включает установку и настройку серверного оборудования, развёртывание программного обеспечения, а также обеспечение стабильного и безопасного сетевого взаимодействия.

Для разрабатываемой системы управления лояльностью клиентов в ресторане быстрого обслуживания, построенной на монолитной архитектуре, диаграмма компонентов отражает основные логические модули приложения, их интерфейсы и зависимости. Ниже приведено описание диаграммы компонентов.

Пользовательский интерфейс отвечает за взаимодействие с пользователями, кассирами в точке продаж.

Слой бизнес-логики реализует правила программы лояльности: начисление баллов, списание, участие в акциях, проверка условий.

Подкомпоненты:

- BonusCalculator расчёт бонусов на основе суммы заказа и текущей программы лояльности;
- LoyaltyProgramManager управление активными программами и коэффициентами;
- TransactionProcessor обработка операций (начисление/списание);
- PromotionEngine применение персонализированных акций.

Расчет бонусов и начисление их обеспечивается в классе BonusBalance, рисунок 13.

```
public readonly struct BonusBalance
{
    public int Value { get; }

    public BonusBalance(int value)
    {
        if (value < 0)
            throw new ArgumentOutOfRangeException(nameof(value), "Баланс не может быть отрицательным.");
        Value = value;
    }

    public BonusBalance Increase(int amount) => new(Value + amount);

    public BonusBalance Decrease(int amount)
    {
        if (amount > Value)
            throw new InvalidOperationException("Недостаточно бонусов.");
        return new(Value - amount);
    }

    public bool CanSpend(int amount) => Value >= amount;

    public override string ToString() => $"{Value} баллов";
}
```

Рисунок 13 – Класс, обрабатывающий операции по начислению баланса

Обработка всех новых транзакций по оплате за заказ обеспечивается в классе BonusAccountRepository, код, описывающий операции с транзакциями показан на рисунке 14.

```

public sealed class BonusAccountRepository : IBonusAccountRepository
{
    private readonly LoyaltyDbContext _context;

    public BonusAccountRepository(LoyaltyDbContext context)
    {
        _context = context ?? throw new ArgumentNullException(nameof(context));
    }

    /// <inheritdoc>
    public async Task AddAsync(BonusAccount account, CancellationToken cancellationToken = default)
    {
        await _context.BonusAccounts.AddAsync(account, cancellationToken);
        await _context.SaveChangesAsync(cancellationToken);
    }

    /// <inheritdoc>
    public async Task AddRangeAsync(IEnumerable<BonusAccount> accounts, CancellationToken cancellationToken)
    {
        await _context.BonusAccounts.AddRangeAsync(accounts, cancellationToken);
        await _context.SaveChangesAsync(cancellationToken);
    }

    /// <inheritdoc>
    public async Task UpdateAsync(BonusAccount account, CancellationToken cancellationToken = default)
    {

```

Рисунок 14 – Класс, обрабатывающий транзакции по оплате заказов

Слой доступа к данным обеспечивает взаимодействие с базой данных: выполнение запросов, транзакций, обновление записей.

Подкомпоненты:

- ClientRepository;
- OrderRepository;
- BonusBalanceRepository;
- LoyaltyProgramRepository.

База данных это централизованное хранилище данных.

На рисунке 15 показана диаграмма компонентов.

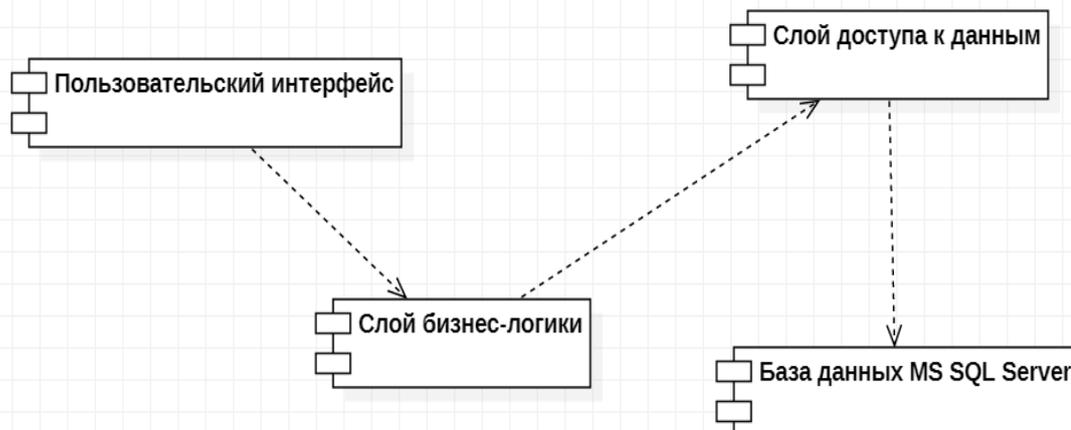


Рисунок 15 – Диаграмма компонентов

Диаграмма показывает, из каких основных частей состоит система лояльности и как эти части взаимодействуют друг с другом, чтобы всё работало плавно и без сбоев.

Внедрение системы позволит значительно повысить операционную эффективность сотрудников. Автоматизация рутинных процессов таких как начисление бонусов, формирование персонализированных предложений и анализ метрик лояльности сократит время на выполнение задач и снизит вероятность человеческих ошибок.

В результате сотрудник сможет обрабатывать больший объём клиентских взаимодействий, запускать больше маркетинговых кампаний и реализовывать больше проектов по удержанию и вовлечению клиентов за тот же период времени.

Настройка автоматизированной системы на рабочих местах пользователей будет осуществляться администратором информационной

системы, который будет официально назначен приказом руководства в соответствии со штатным расписанием организации.

Администратор отвечает за создание учётных записей, назначение ролей и прав доступа, управление паролями, а также за обеспечение соответствия системы политике информационной безопасности компании. Доступ к функционалу системы будет предоставлен исключительно тем сотрудникам, чья профессиональная деятельность напрямую связана с управлением лояльностью клиентов.

Все пользователи системы обязаны соблюдать внутренние регламенты и требования по охране труда, в том числе правила работы с вычислительной техникой, эргономические нормы при длительной работе за компьютером, а также положения по защите персональных данных и конфиденциальной информации.

Обучение работе с системой и инструктаж по технике безопасности будут проведены до начала эксплуатации программного обеспечения, что обеспечит не только эффективность, но и безопасность её использования в повседневной деятельности.

3.3 Разработка информационной системы управления лояльностью клиентов

Загрузка информационной системы не требует прохождения аутентификации методом ввода личных данных, которые присваивает администратор системы.

Форма загрузки информационной системы показана на рисунке 16.

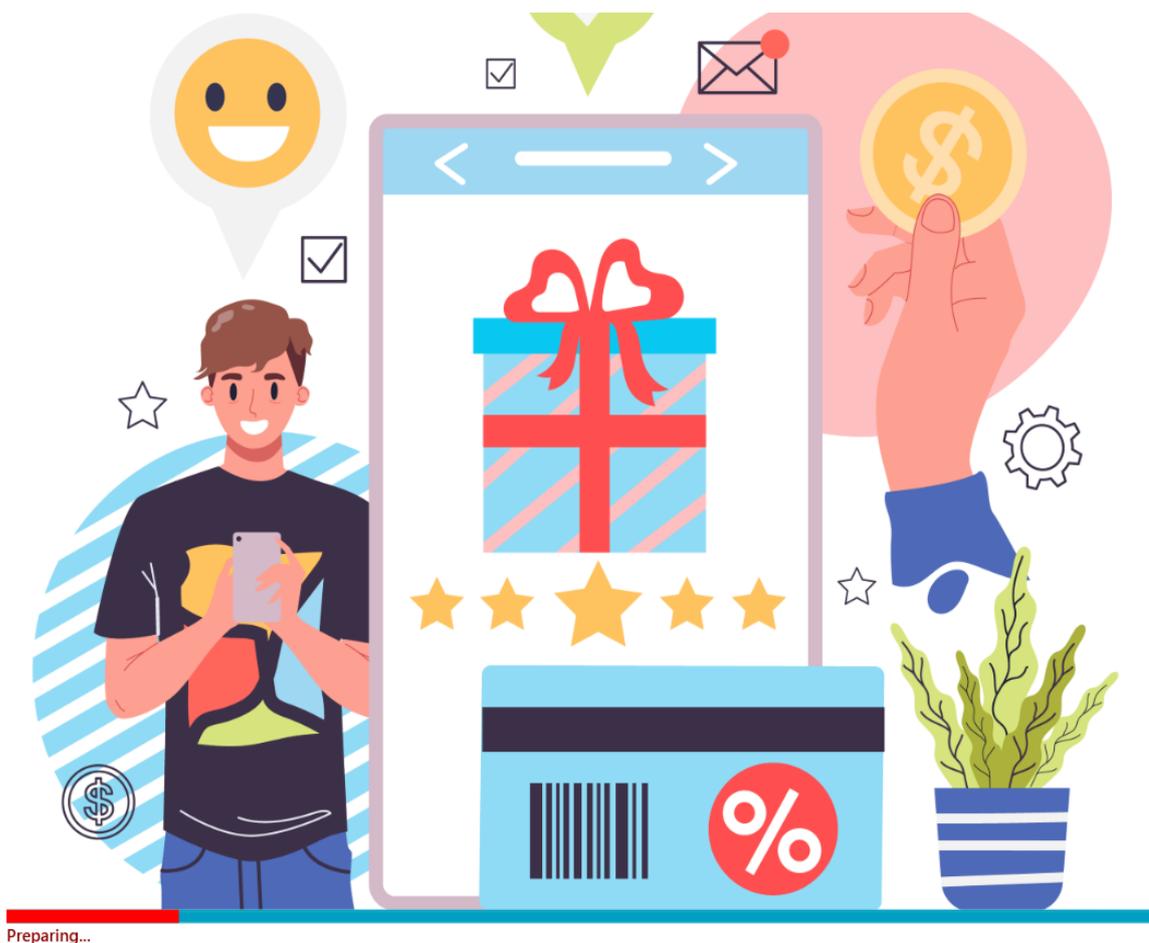


Рисунок 16 – Форма запуска информационной системы управления лояльностью клиентов

В информационной системе зачастую работает сотрудник, который принимает платежи от клиента в ресторане. Перед тем как принять оплату от клиента после посещения сотруднику необходимо проверить наличие регистрации клиента в системе лояльности ресторана. При отсутствии данных о клиенте ему предлагают зарегистрироваться для накопления баллов.

Информационная система проверяет наличие данных, введенных пользователем в каждое поле регистрационной формы.

При отсутствии данных в поле система выводит подсказку красным цветом о необходимости введения информации. Форма показана на рисунке 17.

Рисунок 17 – Форма регистрации нового клиента с подсказками при вводе данных

После заполнения всех необходимых полей при регистрации нового клиента система делает активной кнопку для сохранения данных в базу данных системы. Форма для с данными о новом клиенте показана на рисунке 18.

Рисунок 18 – Добавление новой клиента

Если клиент уже зарегистрирован ранее в системе, ему доступны баллы, которые начисляются автоматически системой с суммы чека, потраченной в ресторане при посещении.

Сотрудник может найти данные клиента из списка, рисунок 19.

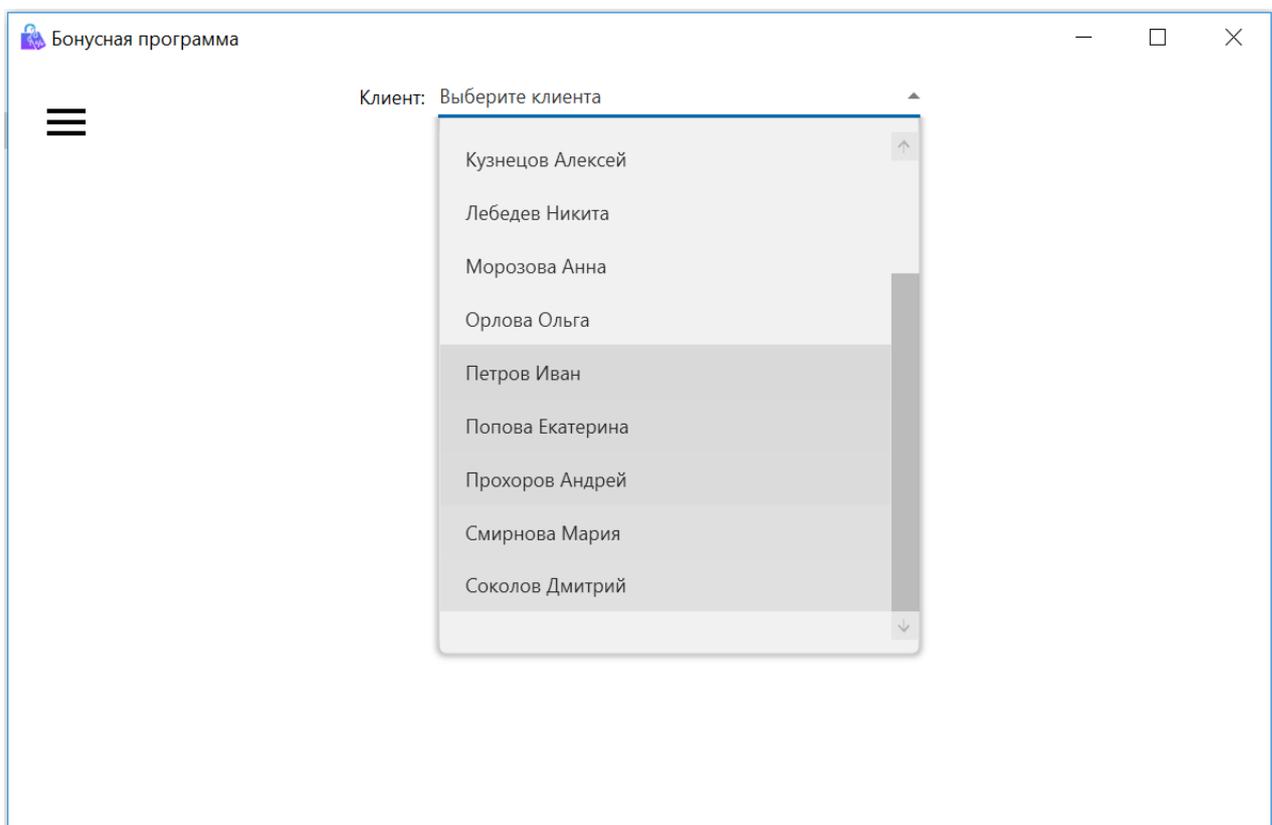


Рисунок 19 – Выбор клиента в системе

После выбора клиента данные о клиенте и начисленных баллах выводятся в отдельное окно, где сотрудник видит информацию о клиенте, а также историю начисления баллов и все операции с ними, это могут быть начисления и списания баллов при расчете. Форма показана на рисунке 20.

Бонусная программа

Клиент: Морозова Анна

Информация о клиенте:
Дата рождения: 30.01.1995
Телефон: +79990000006
Email: anna.morozova@example.com

Сумма покупки: Списать бонусы

Бонусный баланс:
83

История операций:

Дата и время	Тип операции	Количество бонусов
05.10.2025 20:09	Начисление	66
03.10.2025 23:42	Начисление	17
03.10.2025 00:58	Списание	118
17.09.2025 07:36	Начисление	118

Рисунок 20 – Полная информация о клиенте и его лояльности

На форме видно, что у клиента есть накопленные баллы в количестве 83. Баллы накоплены от суммы чека в разные дни и видно, что клиент уже применял прежде баллы при расчете, в истории операция есть списание в сумме 118 баллов, которые приравниваются к 118 рублям.

Сотрудники при расчете клиента вводят в систему сумму чека при посещении, если клиент не хочет применять и списывать накопленные баллы, то сотруднику выходит сообщение о сумме к полной оплате. При подтверждении видно, что сумма баллов обновилась и стала больше, как показано на рисунке 21.

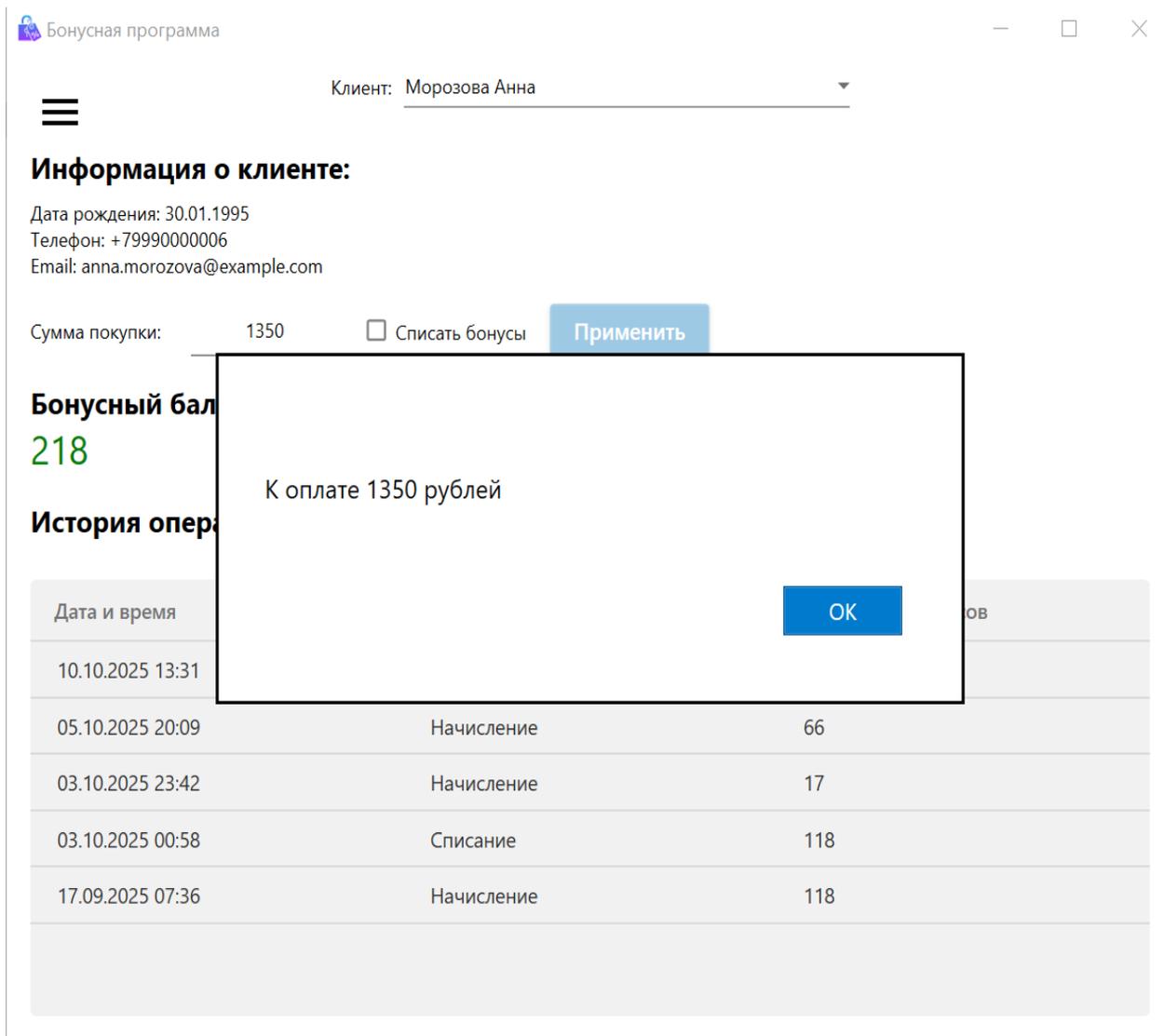


Рисунок 21 – Формирование баллов у клиента

Если клиент захочет при оплате списать накопленные баллы, сотрудник ресторана ставит галочку в пункте «Списать бонусы», тогда система автоматически уменьшает сумму для расчета на кассе. Применение баллов при расчете показано на рисунке 22.

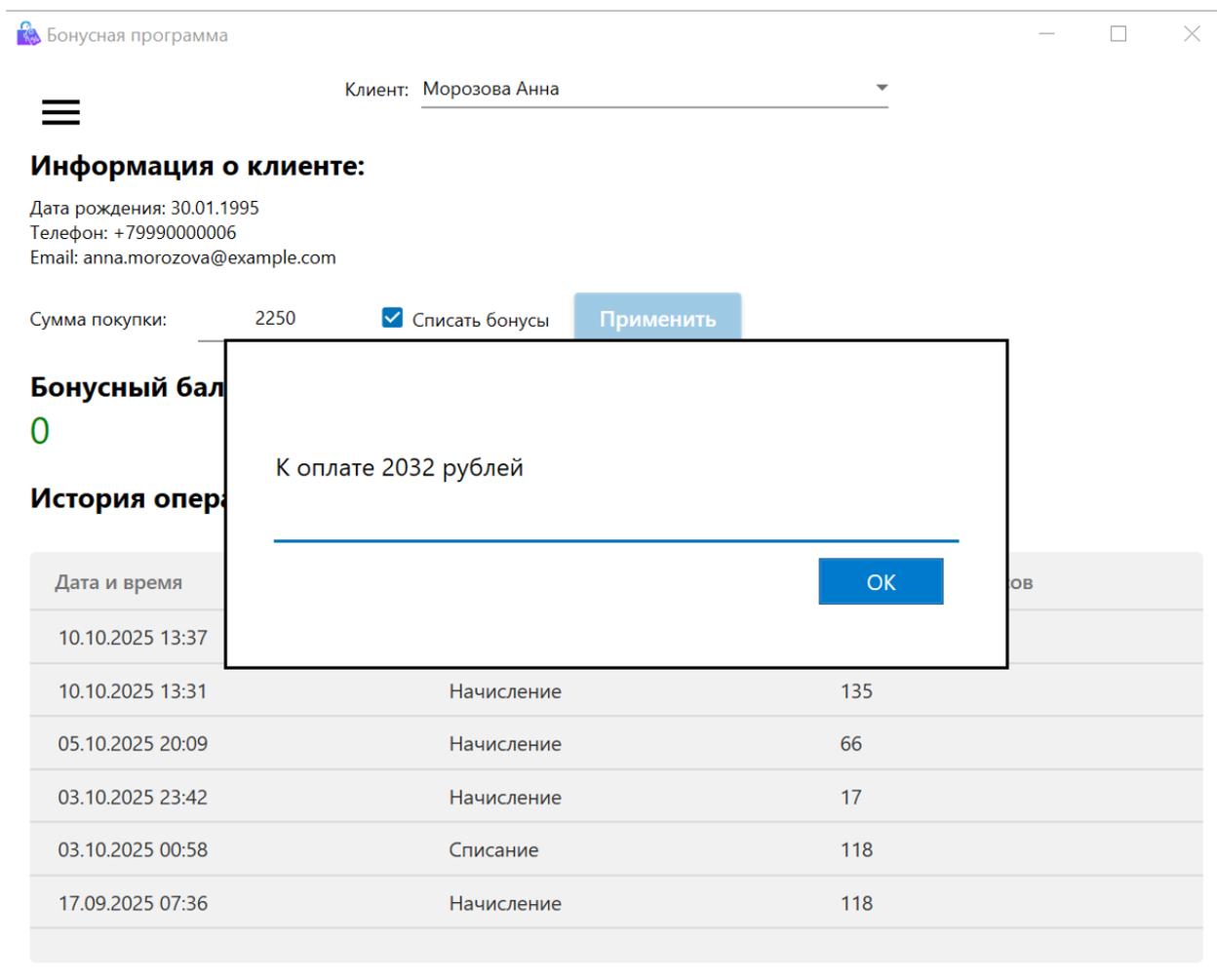


Рисунок 22 – Применение баллов для списания при расчете на кассе

В карточке клиента видно, что стоимость заказа составила 2250 рублей, у клиента прежде было накоплено 218 баллов которые он посчитал нужным списать при оплате чека. После подтверждения применения баллов сотрудников, система вывела сумму для оплаты, уменьшенную на количество применяемых баллов, к оплате вышло 2032 рубля.

Все операции по начислению списанию баллов хранятся в базе данных и доступны для обновления в любое время.

Разработанная система представляет собой комплексный программный модуль, включающий в себя клиентскую часть и механизм интеграции с внешними сервисами. В силу своей архитектурной сложности и функциональной насыщенности установка, первоначальная настройка и

последующая поддержка системы осуществляются исключительно квалифицированным администратором информационных систем, обладающим необходимыми техническими компетенциями и доступом к корпоративной ИТ-инфраструктуре.

В процессе внедрения могут возникнуть непредвиденные обстоятельства, способные повлиять на сроки реализации проекта. К таким рискам относятся: несовместимость с уже используемым программным обеспечением, недостаточная пропускная способность сетевой инфраструктуры, а также необходимость дополнительной адаптации функционала под специфику бизнес-процессов по управлению лояльностью клиентов. Для минимизации подобных рисков рекомендуется заранее провести аудит текущей ИТ-среды и разработать детальный план внедрения с резервными временными буферами.

Администратор системы совместно со специалистами по внедрению программного обеспечения обязаны заблаговременно подготовить как общую ИТ-инфраструктуру отдела, так и индивидуальные рабочие места сотрудников. Это включает в себя проверку соответствия технических характеристик компьютеров минимальным требованиям, настройку сетевых параметров, обеспечение стабильного доступа к базе данных, а также конфигурирование прав доступа в соответствии с ролевой моделью. Такой комплексный подход гарантирует бесперебойную установку, корректную работу системы на всех этапах эксплуатации и комфортное взаимодействие с ней конечных пользователей.

3.4 Расчет показателей экономической эффективности проекта

Оценка экономической эффективности играет важную роль в оценке проекта внедрения программного продукта для управления лояльностью клиентов в ресторане быстрого питания. В данном разделе проводится анализ

экономической эффективности внедрения информационной системы управления лояльностью клиентов.

Анализ себестоимости системы проведен используя данные о разных показателях, описанных в таблице, учитывая внедрение разработанной информационной системы в процессы ресторана быстрого обслуживания.

Таблица 5 содержит информацию, на основе которых производится расчет себестоимости информационной системы управления лояльностью клиентов.

Таблица 5 – Показатели для вычисления себестоимости программного обеспечения

Наименование показателя	Единицы измерения	Обозначение	Значение
Амортизация ПК	%	H_A	20
Затраты на покупку компьютера	руб.	C_K	250000
Стоимость 1 кВт электроэнергии	руб.	C_{KB}	5,20
Мощность компьютера	кВт/ч	M_K	0,6
Оклад программиста	руб.	C_{Tpc}	175000
Фонд рабочего времени в месяц	ч	Φ_{BM}	164,9
Фонд рабочего времени в год	ч	Φ_{BG}	1979

Стоимость амортизации и затраты электроэнергии при работе на компьютере рассчитывается по формуле 1.

$$C_{M/ч} = Am_K + C_{эл}, \quad (1)$$

где Am_K – амортизация компьютера за 1 м/ч (машино-час), руб.;

$C_{эл}$ – стоимость электроэнергии за 1 час работы, руб.

На первоначальном этапе вычисляется амортизация компьютера по формуле 2.

$$A_{M_K} = \frac{C_K \cdot N_A}{\Phi_{ВГ} \cdot 100\%}, \quad (2)$$

где C_K – стоимость компьютера, руб.;

N_A – норма издержки ПК;

$\Phi_{ВГ}$ – фонд годового рабочего времени, ч.

Норма издержки ПК рассчитывается при подстановке значений по формуле.

$$A_{M_K} = \frac{250000 \cdot 20\%}{1979 \cdot 100\%} = 25,26 \text{ руб.}$$

Затраты на электроэнергию затраченную за 1 час работы можно посчитать по формуле 3.

$$C_{ЭЛ} = M_K \cdot C_{КВ}, \quad (3)$$

где M_K – мощность, кВт/ч.;

$C_{КВ}$ – стоимость 1 кВт, руб.;

$$C_{ЭЛ} = 0,6 \cdot 5,20 = 3,12 \text{ руб.}$$

Рассчитаем стоимость электроэнергии м/ч:

$$C_{M/ч} = 25,26 + 3,12 = 28,38 \text{ руб.}$$

После проведенных расчетов затрат на материальные расходы можно рассчитать расходы на фонд зарплаты разработчикам программного обеспечения.

Затраты рассчитываются с помощью данных, которые представлены в таблице 6.

Таблица 6 – Затраты времени на разработку системы

Наименование этапов работы	Время выполнения
Разработка системы	456 часов
Тестирование и исправление ошибок	90 часов

Продолжение таблицы 6

Опытная эксплуатация	70 часов
Доработка системы по результатам эксплуатации	52 часа
Разработка документации	72 часов
Всего	740 часов

Расчет зарплатного фонда на программистов разрабатываемым информационную систему управления лояльностью клиентов, рассчитывается по формуле 4.

$$C_{pc} = Z_{пр} \cdot T_{pc}, \quad (4)$$

где $Z_{пр}$ – оклад за рабочий час;

T_{pc} – часы затратившие на разработку системы.

$P_{клад}$ разработчика за рабочий час рассчитывается по формуле 5.

$$Z_{пр} = \frac{C_{Tpc}}{\Phi_{вм}}, \quad (5)$$

где C_{Tpc} – ставка, руб.;

$\Phi_{вм}$ – фонд рабочего времени в месяц, ч.

$$Z_{пр} = \frac{175000}{164,9} = 1061,25 \text{ руб.}$$

Затраты на оплату труда составили:

$$C_{pc} = 1061,25 \cdot 740 = 785325,00 \text{ руб.}$$

Себестоимость разработанной системы рассчитывается по формуле 6.

$$C_c = C_{м/ч} \cdot T_{pc} + C_{pc}, \quad (6)$$

где $C_{м/ч}$ – стоимость машинного часа работы, руб.;

$T_{рс}$ – количество времени, затратившее на разработку системы;

$C_{рс}$ – затраты на зарплатный фонд разработчика.

$$C_c = 28,38 \cdot 740 + 785325,00 = 806326,20 \text{ руб.}$$

Экономическая эффективность также включает затраты на предполагаемые риски при внедрении информационной системы.

Исходные данные представлены в таблице 7.

Таблица 7 – Затраты на предполагаемые риски при внедрении информационной системы

Наименование показателя	Единицы измерения	Обозначение	Значение
Время при вводе данных о баллах клиента	Ч	T_p	1,50
Время при вводе данных о баллах клиента после внедрения системы	Ч	T_c	0,20
Затраты на покупку компьютера	руб.	C_k	250000
Оклад сотрудника	руб.	$C_{тстс}$	60000
Фактическое время работы компьютера за год	Ч	$T_{фг}$	2000

Амортизация компьютера за 1 м/ч.

$$A_{мк} = 25,26 \text{ руб.}$$

Рассчитаем затраты на час рабочего времени кассира ресторана по формуле 7.

$$Z_c = \frac{C_{тстс}}{\Phi_{вм}}, \quad (7)$$

где $C_{тстс}$ – ставка, руб.;

$\Phi_{вм}$ – рабочие часы в месяц, ч.;

$$З_c = \frac{60000}{164,9} = 363,86 \text{ руб.}$$

Затраты за 1 час работы сотрудника на ПК рассчитаны по формуле 8.

$$C_{м/ч} = A_{мк} + З_c + C_{эл}, \quad (8)$$

где $A_{мк}$ – амортизация компьютера за 1 м/ч, руб.;

$З_c$ – часовая зарплата кассира;

$C_{эл}$ – стоимость электроэнергии за 1 ч;

$$C_{м/ч} = 25,26 + 363,86 + 3,12 = 392,24 \text{ руб.}$$

Экономический эффект внедрения информационной системы рассчитывается по формуле 9.

$$\mathcal{E} = C_1 - C_2, \quad (9)$$

где C_1 – стоимость процесса до автоматизации;

C_2 – стоимость процесса после автоматизации;

Стоимость первого варианта рассчитывается по формуле 10.

$$C_1 = (C_{т_{стс}} \cdot \frac{T_p}{\Phi_{вм}} + C_{м/ч} \cdot T_p) \cdot K_c \cdot 12, \quad (10)$$

где $C_{т_{стс}}$ – ставка сотрудника отдела;

T_p – затраченное время до автоматизации, ч;

$\Phi_{вм}$ – фонд рабочего времени в месяц, ч.;

$C_{м/ч}$ – стоимость одного часа, проведенного за компьютером, ч.;

K_c – количество сотрудников;

$$C_1 = (60000 \cdot \frac{1,5}{164,9} + 392,24 \cdot 1,5) \cdot 10 \cdot 12 = 136096,8 \text{ руб./год.}$$

Стоимость работы сотрудника с использованием новой системы вычисляется по формуле 11.

$$C_2 = \left(\frac{C_{T_{стс}} \cdot T_c}{\Phi_{вм}} + C_{м/ч} \cdot T_c \right) \cdot K_c \cdot 12 + C_c, \quad (11)$$

где $C_{T_{стс}}$ – ставка сотрудника;

$\Phi_{вм}$ – фонд рабочего времени в месяц, ч.;

T_c – затраченное время для анализа после внедрения системы, ч.;

$C_{м/ч}$ – стоимость одного машинного часа сотрудника отдела аналитики, ч.;

K_c – количество сотрудников компании;

C_c – стоимость внедренной системы, руб.;

$$C_2 = \left(\frac{60000 \cdot 0,20}{164,9} + 392,24 \cdot 0,2 \right) \cdot 10 \cdot 12 + 806326,20 = 824472,36$$

руб./год.

Экономический эффект от внедрения системы составляет:

$$\mathcal{E} = 136096,8 - 824472,36 = 688375,56 \text{ руб./год.}$$

Определим срок окупаемости капиталовложений по формуле 12.

$$T_{ср} = \frac{C_k + C_c}{\mathcal{E}}, \quad (12)$$

где C_k – стоимость компьютера, руб.;

C_c – стоимость системы, руб.;

\mathcal{E} – экономический эффект, руб./год.;

$$T_{ср} = \frac{250000 + 806326,20}{688375,56} = 1,54 \text{ года, т.е. приблизительно через 2 года.}$$

Определим экономическую эффективность от вложенных средств по формуле 13.

$$\begin{aligned} \text{Э}_\phi &= \frac{1}{T_{\text{ср}}}, & (13) \\ \text{Э}_\phi &= \frac{1}{1,54} = 0,65 \end{aligned}$$

Показатели, полученные после определения экономической эффективности вложенных средств и срока окупаемости, видно, что проект будет окупаться не быстро, но по истечении двух лет внедрение системы в ресторане полностью себя окупит.

Вывод по третьей главе

Третья глава посвящена физическому проектированию и реализации информационной системы управления лояльностью клиентов для ресторана быстрого питания. На этапе выбора инструментов разработки обосновано применение языка программирования C# в связке с платформой .NET и средой разработки Microsoft Visual Studio 2022. Выбор обусловлен высокой производительностью, поддержкой объектно-ориентированного подхода, богатой библиотекой классов, а также удобством разработки настольных приложений, соответствующих требованиям проекта.

В ходе проектирования были разработаны ключевые программные модули системы, обеспечивающие регистрацию клиентов, начисление и списание бонусных баллов, просмотр истории взаимодействий и применение скидок на кассе. Особое внимание уделено пользовательскому интерфейсу: реализованы формы с валидацией ввода данных, визуальной обратной связью и интуитивно понятной навигацией, что снижает вероятность ошибок персонала и ускоряет процесс обслуживания.

Описаны организационные аспекты внедрения: назначение администратора системы, настройка прав доступа, обучение сотрудников и обеспечение информационной безопасности. Также учтены потенциальные

риски несовместимости с существующим ПО, недостаточная сетевая инфраструктура и предложены меры по их минимизации.

Завершающим этапом стало проведение расчёта экономической эффективности проекта. Себестоимость разработки составила 806 326,20 руб., включая затраты на труд программиста и эксплуатацию оборудования. Внедрение системы позволяет сократить время обработки операций по лояльности с 1,5 до 0,2 часа на клиента, что обеспечивает годовую экономию операционных расходов. Несмотря на то, что чистый экономический эффект в первый год отрицателен из-за высоких первоначальных вложений, срок окупаемости проекта составляет около 1,54 года, а коэффициент эффективности 0,65, что свидетельствует о перспективности и целесообразности внедрения разработанной системы.

Таким образом, третья глава завершила цикл проектирования и реализации ИС, продемонстрировав её техническую осуществимость, функциональную полноту и экономическую обоснованность.

Заключение

Выпускная квалификационная работа успешно завершена, все поставленные задачи выполнены в полном объёме, что позволило достичь главной цели, разработать программное обеспечение для автоматизации процесса управления лояльностью клиентов.

На первом этапе было проведено детальное исследование предметной области, изучена типовая модель взаимодействия с клиентами, проанализированы ключевые бизнес-процессы, связанные с накоплением и использованием бонусов, сбором обратной связи и удержанием клиентов.

Полученные данные легли в основу моделирования текущего состояния системы управления лояльностью «как есть». В ходе анализа были выявлены существенные недостатки, фрагментированность данных о клиентах, отсутствие единой системы начисления и учёта бонусов, ручное формирование персонализированных предложений, низкая вовлечённость клиентов. Эти проблемы обосновали необходимость внедрения специализированной автоматизированной системы.

На основе модели «как есть» была разработана целевая модель процесса «как должно быть» с использованием нотации IDEF0, наглядно демонстрирующая, как автоматизация устраняет выявленные недостатки и повышает эффективность взаимодействия с клиентами. Параллельно были сформулированы функциональные и технические требования к системе, включая требования к пользовательскому интерфейсу, защите персональных данных, масштабируемости.

Для реализации программного решения были выбраны современные и надёжные технологии. В качестве языка программирования использовался C#, обеспечивающий высокую производительность и удобство разработки серверной логики. Среда разработки Microsoft Visual Studio 2022, предоставляющая полный набор инструментов для проектирования, отладки и тестирования. В качестве СУБД применён Microsoft SQL Server,

гарантирующий целостность, безопасность и эффективное хранение данных о клиентах, их действиях, бонусах и предпочтениях.

Разработанное программное обеспечение реализовано в виде интуитивно понятных веб-интерфейсов, адаптированных под задачи различных категорий пользователей.

Внедрение системы значительно повышает прозрачность и управляемость процессов удержания и мотивации клиентов. Сотрудники получают возможность в режиме реального времени отслеживать уровень вовлечённости каждого клиента, оперативно реагировать на снижение лояльности.

Благодаря автоматизации рутинных операций, таких как начисление баллов, формирование отчётов, освобождается значительное количество рабочего времени. Это позволяет команде сосредоточиться на поведении клиентов, разработке новых программ лояльности и повышении общего уровня удовлетворённости клиентской базы.

Список используемой литературы и используемых источников

1. Архангельский А. Проектирование программного обеспечения с использованием UML. – СПб.: Питер, 2008. – 352 с.
2. Баранов А. Основы организации продаж. КноРУС, 2019. – 256 с.
3. Бережной В.А. Электронная коммерция: создание и управление. СПб.: БХВ-Петербург, 2018. – 240 с.
4. Бизнес-процессное моделирование с помощью Bizagi Process Modeler. [Электронный ресурс]: <https://elibrary.ru/item.asp?id=43084546> (дата обращения: 15.05.2025)
5. Бизнес-процессы в ресторане. [Электронный ресурс]: <https://frost26.ru/blog/biznes/biznes-protsessy-v-restorane/> (дата обращения: 15.05.2025)
6. Будаков В.Ю., Мягков Н.К. Основы баз данных. – М.: ДМК Пресс, 2018. С. 28 – 45.
7. Инструмент 10cups. [Электронный ресурс]: <https://10cups.ru/> (дата обращения: 15.05.2025)
8. Меркулов А. Применение диаграмм вариантов использования UML в процессе разработки программных систем / А. Меркулов // Информационные технологии в программировании. – 2020. – №4. – С. 63-75.
9. Официальная документация BPMN от Object Management Group (OMG). [Электронный ресурс]: <http://www.omg.org/spec/BPMN> (дата обращения: 15.05.2025)
10. Павловская Т.В. Проектирование баз данных в среде Access. – СПб.: Питер, 2018. С. 73 – 92.
11. Пономарев С.В. Постпродажное обслуживание клиентов. – М.: Дашков и К, 2018. – 184 с.
12. Попов М.С. Электронная коммерция: технологии и практика. Москва: Дашков и К, 2018. – 352 с.

13. Программа лояльности Locard. [Электронный ресурс]: https://locards.ru/?utm_source=yandex&utm_medium=cpc&utm_campaign=118544264&type=search&source=none&block=premium&position=1&utm_term=locards&callibri=yd_c:118544264_gb:5584609454_ad:17032380470_ph:54855706556_st:search_s:none_dt:desktop&etext=2202.thabaMgR7OosCFNPl6S3dDT8J6lFTzXFJfIPRMcgmZd6cGlhZGF3dWtobG96bW9m.41fd20856a18c18be02f9dd999a3ef0647ecd9b&yclid=17831410286466957311 (дата обращения: 15.05.2025)
14. Разработка диаграмм процессов бизнеса. [Электронный ресурс]: https://elibrary.ru/download/elibrary_23359677_63633299.pdf (дата обращения: 15.05.2025)
15. Система Loymax Loyalty. [Электронный ресурс]: <https://loymax.ru/solutions/loymax-loyalty/> (дата обращения: 15.05.2025)
16. Степаненко А. Введение в информационные системы. Моделирование систем. – М.: НИУ ВШЭ, 2008. – 416 с.
17. Халлиган Т. Информационная модель базы данных. – Киев: Диалектика, 2019.
18. Ширман Я. Маркетинговая стратегия. – СПб.: БХВ-Петербург, 2018. – 336 с.
19. Язык моделирования UML 2.0 в действии. [Электронный ресурс]: <https://dmkpress.com/catalog/computer/programming/978-5-94074-518-3/> (дата обращения: 15.05.2025)
20. Bodybuilding: American online retailer based in Boise, Idaho, specializing in dietary supplements, sports supplements and bodybuilding supplements. [Электронный ресурс]: <https://www.bodybuilding.com/> (дата обращения 15.05.2025).
21. C#: описание и особенности языка. [Электронный ресурс]: <https://otus.ru/journal/si-sharp-opisanie-i-osobennosti-yazyka/> (дата обращения: 15.05.2025).
22. C Sharp. [Электронный ресурс]: https://ru.ruwiki.ru/wiki/C_Sharp (дата обращения: 15.05.2025).