

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

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование)

09.03.03 Прикладная информатика  
(код и наименование направления подготовки, специальности)

Бизнес-информатика  
(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему «Автоматизированная информационная система управления  
продажами (на примере ИП Павлюк Инна Сергеевна)»

Студент

У.Д. Мазитов  
(И.О. Фамилия)

(личная подпись)

Руководитель

Т.Г. Любивая  
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

## Аннотация

Тема выпускной квалификационной работы: Автоматизированная информационная система управления продажами (на примере ИП Павлюк Инна Сергеевна).

Ключевые слова: автоматизированная информационная система, логическая модель данных, физическая модель данных, управление продажами, бизнес-процесс.

Целью бакалаврской работы является разработка автоматизированной информационной системы управления продажами для ИП Павлюк И.С.

Объект исследования – деятельность отдела продаж ИП Павлюк И.С.

Предмет исследования – автоматизация деятельности отдела продаж ИП Павлюк И.С.

В бакалаврской работе проведен анализ деятельности ИП Павлюк И.С., определены задачи на разработку информационной системы, разработаны функциональные модели деятельности отдела продаж «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ».

Определены требования к программно-аппаратным средствам. Проведен выбор архитектуры приложения, определены технологии разработки программного обеспечения, выбрана система управления базой данных. Разработана логическая и физическая модели базы данных.

Проведена разработка программного обеспечения автоматизированной информационной системы. В ходе тестирования информационной системы выполнена проверка работоспособности информационной системы, при которой определена работоспособность всех функций.

Разработанная автоматизированная информационная система управления продажами может быть применена в отделе продаж ИП Павлюк И.С.

Работа включает: страниц 66, рисунков 33, таблиц 7, источников 21.

## Оглавление

Введение.....	4
Глава 1 Функциональное моделирование предметной области.....	6
1.1 Техничко-экономическая характеристика предприятия.....	6
1.2 Концептуальное моделирование предметной области.....	8
1.2.1 Выбор технологии концептуального моделирования предметной области.....	8
1.2.2 Моделирование бизнес-процессов предметной области для постановки задачи автоматизированного варианта решения.....	10
1.2.3 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ».....	11
1.2.4 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии.....	13
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям.....	15
1.4 Постановка задачи на разработку проекта АИС.....	19
1.5 Разработка модели бизнес-процесса ТО-ВЕ.....	21
Глава 2 Логическое проектирование АИС.....	24
2.1 Выбор технологии логического моделирования АИС.....	24
2.2 Логическая модель АИС: описание.....	28
2.3 Информационное обеспечение АИС.....	30
2.3.1 Классификаторы и системы кодирования.....	30
2.3.2 Характеристика нормативно-справочной и входной оперативной информации.....	31
2.3.3 Характеристика выходной информации.....	32
2.4 Проектирование базы данных АИС.....	36
2.4.1 Выбор технологии проектирования БД АИС.....	36
2.4.2 Разработка концептуальной модели данных АИС.....	36
2.4.3 Разработка логической модели данных АИС.....	37
2.5. Требования к аппаратно-программному обеспечению АИС.....	38
Глава 3 Физическое проектирование АИС.....	40
3.1 Выбор архитектуры АИС.....	40
3.2 Выбор технологии разработки программного обеспечения АИС.....	41
3.3 Выбор СУБД АИС.....	42
3.4 Разработка физической модели данных АИС.....	42
3.5 Разработка программного обеспечения АИС.....	44
3.5.1 Схема взаимосвязи модулей приложения АИС.....	44
3.5.2 Описание модулей приложения АИС.....	45
3.6 Описание функциональности АИС.....	55
3.7 Тестирование программного проекта.....	61
Заключение.....	63
Список используемой литературы.....	64
Приложение А Программный код.....	67

## Введение

Управление продажами – это многогранный термин, для которого еще не разработан уникальный алгоритм, позволяющий раскрыть его суть. В основном это связано с этапом развития экономики рынка и сменяющимися обстоятельствами хозяйствования, ставящими новые требования и создающими иной механизм управления продажами в пределах организации. Часть экспертов считает управление продажами составной частью системы управления компанией, другие рассматривают его как элемент управления торговым персоналом, осуществляющие реализацию продукции (в том числе набором, мотивацией, подготовкой, организацией и контролем работы).

В настоящее время для ускорения и упрощения процесса взаимодействия между продавцами и покупателями активно применяются средства автоматизации. В крупных компаниях клиентская база включает данные десятков тысяч человек, и отследить вручную, на какой стадии совершения покупки находится каждый из них, невозможно. На помощь сотрудникам отдела продаж приходят CRM-системы, благодаря которым все этапы работы с покупателями автоматизированы. При их использовании повышается уровень продаж, оптимизируется маркетинг, фиксируется каждый шаг взаимоотношений с клиентами для аналитики и корректировки планов.

Целью бакалаврской работы является разработка автоматизированной информационной системы управления продажами для ИП Павлюк И.С.

Для достижения цели ставились следующие задачи:

- осуществить технико-экономическую характеристику предприятия;
- разработать функциональные модели AS-IS и TO-BE деятельности отдела продаж;
- построить логическую и физическую модели данных исследуемой предметной области;

– спроектировать и реализовать модули автоматизированной информационной системы (АИС) для управления продажами в ИП Павлюк И.С.

Объект исследования – деятельность отдела продаж ИП Павлюк И.С.

Предмет исследования – автоматизация деятельности отдела продаж ИП Павлюк И.С.

В основу методологии были положены: структурный подход к анализу и проектированию информационных систем, методология объектно-структурного подхода к моделированию информационных систем, CASE-технологии структурного и объектно-ориентированного анализа и проектирования.

Структура работы: введение, три главы, заключение, список литературы, приложение.

В первой главе исследована организационная структура ИП Павлюк И.С., изучены бизнес-процессы, определены особенности предметной области, определена задача на разработку информационной системы, разработана новая модель бизнес-процесса.

Во второй главе приведено логическое проектирование информационной системы, построена логическая модель приложения и его базы данных (БД), определены требования к программно-аппаратным средствам.

В третьей главе представлены результаты физического проектирования информационной системы для ИП Павлюк И.С. При этом определена архитектура приложения, выбрана система управления базой данных (СУБД) и реализован функционал информационной системы.

Заключение содержит результаты, полученные в процессе выполнения бакалаврской работы.

## **Глава 1 Функциональное моделирование предметной области**

### **1.1 Техничко-экономическая характеристика предприятия**

ИП Павлюк И.С. является обществом с ограниченной ответственностью и 25 лет в городе Урая предоставляет следующие услуги [13]:

- внедрение и поддержка программного обеспечения семейства «1С», «Консультант Плюс», «Контур-Экстерн», а также решения собственных разработок;
- ремонт и техническое обслуживание компьютеров и оргтехники;
- предоставление услуг доступа в Интернет, разработку локальных сетей;
- полиграфические и мультимедийные услуги.

В настоящее время на предприятии работают более двухсот человек. Деятельность компании охватывает полный комплекс услуг по автоматизации учета и управления на предприятиях различных отраслей и любого масштаба – от индивидуальных предпринимателей до холдингов, производственных предприятий и бюджетных организаций.

Превышает десять тысяч названий техники товарный ассортимент магазинов: направлений (аудио- и видео-направления, цифрового направления); бытовой электроники; товаров, предназначенных для активного развлечения и различных аксессуаров.

Специалисты внедряют современные и многофункциональные программные продукты «1С», большой опыт и квалификация позволяют подобрать именно то решение, которое максимально подойдет под бизнес-процессы предприятия [13].

Юридический адрес: 628284, Ханты-Мансийский автономный округ-Югра, город Урай, мкр. Аэропорт, д.29.

Схема организационной структуры предприятия ИП Павлюк И.С. представлена на рисунке 1.

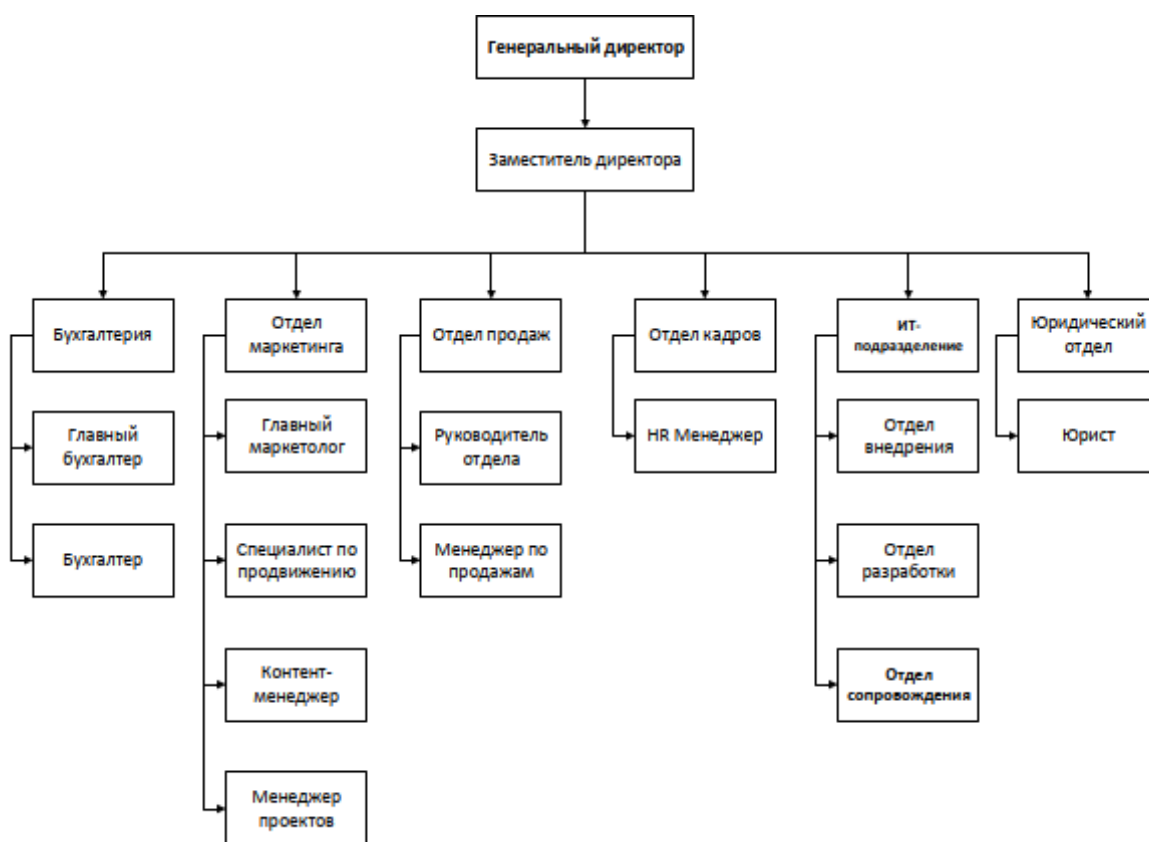


Рисунок 1 – Организационная структура предприятия ИП Павлюк И.С.

Руководство предприятием осуществляет генеральный директор. Ответственным за финансовую деятельность организации является главный бухгалтер. За вопросы закупки, маркетинга и продаж, а также формирования их планов несет ответственность коммерческий директор.

Ответственным за руководство отделом продаж является начальник отдела, обязанности которого сводятся к контролю и координированию продаж на предприятии, а также за разработку должностных инструкций для всех работников своего отдела, разработку мероприятий, направленных на расширение рынка сбыта и повышение продаж. Также он занимается формированием планов и проведением обучения сотрудников отдела.

Одна из важных задач отдела продаж – определение целевой аудитории. Для этого необходимо проанализировать интересы потребителей.

Результатом такой задачи является формирование базы данных клиентов.

Основные функции отдела продаж представлены на рисунке 2.



Рисунок 2 – Функции отдела продаж предприятия

Сотрудники отдела продаж занимаются самостоятельным поиском клиентов. Их деятельность направлена на привлечение целевой аудитории и реализацию товаров или услуг. В обязанности сотрудников отдела продаж входит контроль над циклом продаж как основных продуктов, так и дополнительных.

## 1.2 Концептуальное моделирование предметной области

### 1.2.1 Выбор технологии концептуального моделирования предметной области

Для моделирования бизнес-процессов выбрана методология IDEF0, предполагающая эффективное выполнение анализа, а также создание проекта



и предоставления процессов на деловом уровне. IDEF0 включает в свою структуру диаграммы, представляющие описание модели предметной области в графическом виде. Главный компонент IDEF0 – блок. Каждый блок показывает такие процессы, как работа, функции, а также задачи, осуществляемые в нужный срок [7].

В нотации IDEF0 модели являются нужными для описания ведения бизнеса фирмы высокоуровневого типа в функциональном отношении.

Нотация DFD дает возможность отражения последовательности работ, которые выполняются по ходу процесса, и потоков сведений, циркулирующих между данными работами.

Построение схемы системы – цель методологии IDEF0. Данная схема системы исследования обеспечивает точное описание каждого процесса. При этом самой точности достаточно для проведения моделирования работы системы однозначного типа. Моделируемую систему в IDEF0 представляют в качестве определенной совокупности тех или иных работ, связанных друг с другом.

IDEF0 – методология, которая применяет метод так называемого моделирования так называемого функционального типа в отношении систем, являющихся сложными по своей структуре, а также представляют собой часть методологии по осуществлению структурного анализа и создание проекта SADT. IDEF0 – это средство, позволяющая создавать модели функционального типа, описывающие процессы в форме системы функций. При этом данные функции являются связанными друг с другом.

Дуги и блоки включены в любую IDEF0. Блоки показывают то, какую работу выполняет моделируемая система. Дуги осуществляют совмещение друг с другом блоков, а также отображают между ними определённые связи. Создание модели осуществляется с представления системы в форме единственного блока и дуг, которые показывают вне самой системы интерфейсы с необходимыми и дополнительными функциями. Далее осуществляется детализирование блока, представляющего систему в виде

общего модуля. Детализация осуществляется на иной диаграмме посредством трёх и более блоков, которые связаны дугами интерфейсов. Декомпозицию блока непосредственно из самой диаграммы предыдущего уровня представляет собой любая диаграмма детального типа. Диаграмма предыд. уровня – родительская для более детальной диаграммы – на всех шагах декомпозиции [7].

Нотация IDEF0 применяется в целях реализации верхнего уровня модели процессов. Создание IDEF0 верхнего уровня дает возможность описывать объект моделирования в абстрактном или же общем виде. Подготовленную модель необходимо согласовывать с архитекторами и программистами, для подтверждения того, что структура бизнес-процессов наглядна, понятна и доступна. Модели должны быть согласованы со специалистами, которые являются ответственными и имеют наиболее полную информацию по определенному процессу [7].

### **1.2.2 Моделирование бизнес-процессов предметной области для постановки задачи автоматизированного варианта решения**

Бизнес-моделирование – процесс, предполагающий создание и дальнейшее задействование моделей самой организации (конкретная стратегия, процессы бизнес типа, структура организации, качество и другие.) для формализации работы организации и оптимизации [5].

Большинство сфер деятельности объекта исследования включают такую функцию как моделирование бизнес-процессов:

- изменение принципа проведения операций, а также корректировка внутренних документов;
- корректировка структуры самой организации;
- новые запросы к процессу автоматизации осуществляемых процессов;
- оптимизация функций сотрудников, а также подразделений самой организации;

- распределение прав руководства, а также полномочий по осуществляемой деятельности.

Возможно точно понять, каким образом организована работа на всех рабочих местах, как функционирует система в общем, а также как вступает во взаимодействие с внешними факторами посредством моделирования бизнес-процессов.

Моделирование бизнес-процессов является действенным инструментом по осуществлению поиска пути улучшения самой системы, а также средством по прогнозированию возможных рисков и их нивелированию.

Моделирование бизнес-процессов позволит определить особенности существующего бизнес-процесса и на основании этих данных сформировать новые требования и разработать новую бизнес-модель «КАК ДОЛЖНО БЫТЬ» [7].

### 1.2.3 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

На рисунке 3 изображена контекстная диаграмма деятельности отдела продаж ИП Павлюк И.С. [13].

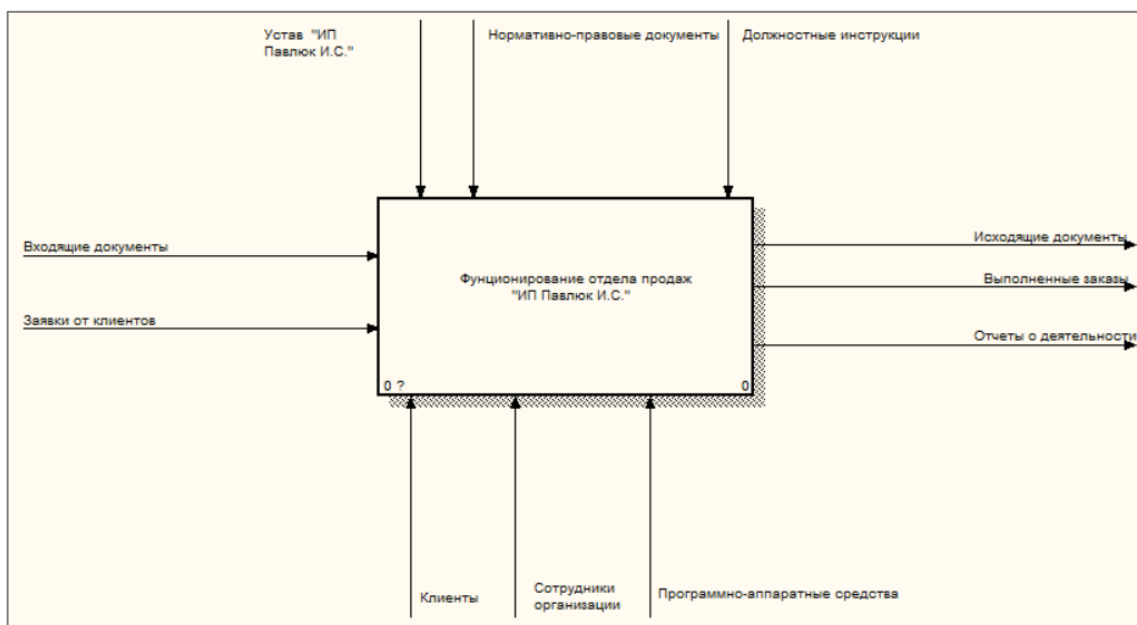


Рисунок 3 – Контекстная диаграмма деятельности отдела продаж ИП Павлюк И.С.

На рисунке 4 представлена декомпозиция процесса «Функционирование отдела продаж ИП Павлюк И.С.», состоящая из следующих процессов:

- обработка документов;
- прием заявок от клиентов;
- организация обеспечения;
- обработка заказа;
- выполнение заказа.

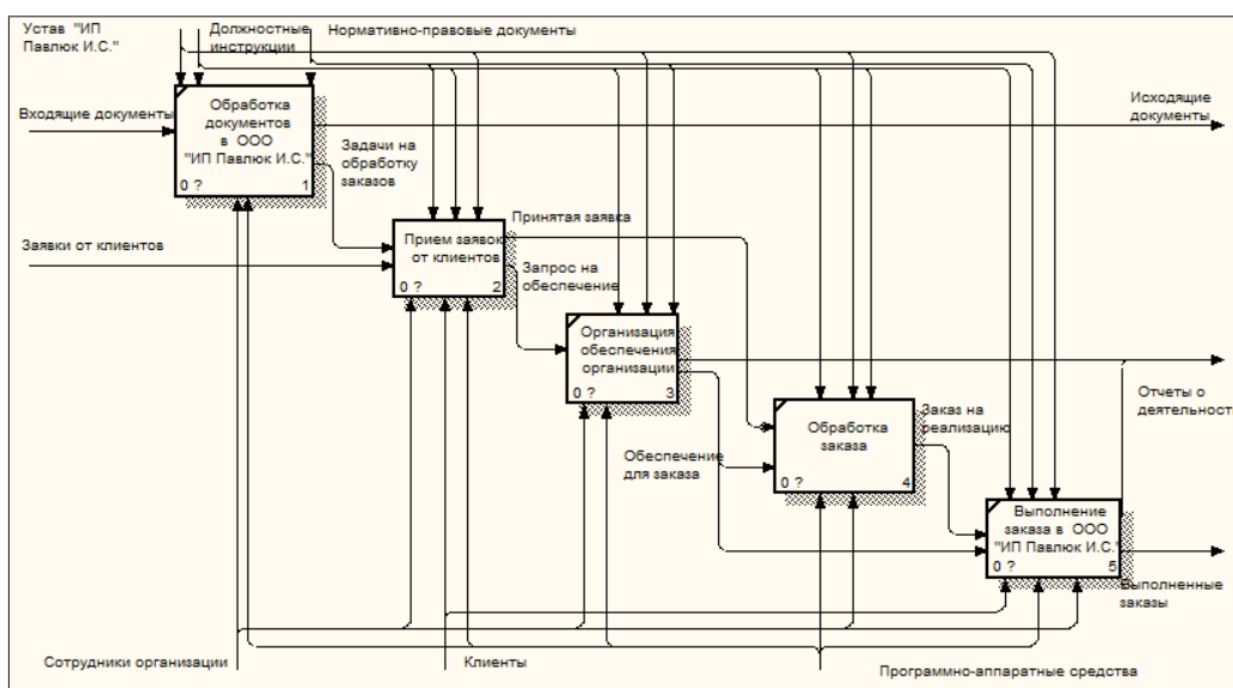


Рисунок 4 – Декомпозиция контекстной диаграммы

Входящими потоками являются входящие документы, заявки от клиентов. Управляющие потоки образуют: устав организации, должностные инструкции и нормативно-правовые документы. Механизмы представлены клиентами, сотрудниками организации и программно-аппаратными средствами. Выходные потоки: исходящие документы, отчеты о деятельности и выполненные заказы.

Обработка заказа начинается с выбора клиентом актуальной для него услуги (товара), обрабатывает заказ менеджер по продажам. После обработки

заказа информация о нем записывается в базу данных и в процессе выполнения заказа, информация берется из базы данных. После выполнения заказа создается счет представленных услуг (товаров).

При анализе существующих процессов в отделе продаж можно выделить следующие недостатки:

- отсутствие автоматизации процесса учета заявок клиентов;
- отсутствие базы данных клиентов и заказов;
- отсутствие механизмов контроля деятельности отдела продаж и удобной статистики продаж;
- отсутствие систематизации в базе данных заказов;
- отсутствие возможностей масштабирования процессов компании.

Представленные недостатки снижают продуктивность работы отдела продаж и эффективность деятельности предприятия. Увеличивается количество времени, которое менеджер тратит на обработку заказов, снижается количество обработанных заказов в день.

Для решения представленных выше недостатков наиболее подходящим вариантом является создание автоматизированной системы для обработки заказов, которое позволит обеспечить:

- автоматизацию процесса учета заявок клиентов;
- механизмы контроля деятельности отдела продаж;
- систематизацию базы данных заказов;
- наличие базы данных клиентов.

#### **1.2.4 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии**

Цели внедрения системы по осуществлению управления продажами:

- повышение качества удовлетворённости клиентов посредством анализа всех полученных данных по клиентам;
- регулирование политики по тарифам;
- настройки средств маркетинга.

За счёт использования АИС есть возможность наиболее качественно и успешно, а также с наименьшим участием работников собирать сведения по запросам заказчиков в индивидуальном порядке, а посредством оперативности обработки – выполнять в сжатые сроки определение различных рисков и возможностей, которые могут быть с определенной долей вероятности.

Существуют основные задачи в процессе внедрения системы по управлению продажами в работе организации [11]:

- анализ рынка;
- аргументация потребности создания новой продукции, непосредственно её последующего вывода;
- использование в работе эффективных методов по продвижению услуг, а также товаров;
- осуществление контроля различных показателей работы сотрудников;
- обучение и увеличение профессионального уровня сотрудников организации, которые совместно работают непосредственно с клиентами, партнерами;
- увеличение практического интереса сотрудников компании в использовании новейших средств работы для того, чтобы достигать целей, которые были изначально заданы;
- сбор данных по конкурентам, а также клиентам, партнерам, систематизация всех данных в общей базе.

Главные функции используемой системы при работе с данными и клиентами:

- регистрация данных;
- хранение данных и их дальнейшая обработка;
- представление данных и их экспорт;
- идентификация заказчика;
- дифференциация заказчиков;

- взаимодействие с заказчиком;
- персонализация.

### 1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Чтобы определить задачи, которые должна выполнять информационная система предприятия, необходимо рассмотреть известные решения, целью которых является взаимодействие с клиентами. В таблице 1 представлена сравнительная характеристика системы управления продажами.

Таблица 1 – Сравнительная оценка информационных систем

Сравнительные характеристики	Inv Gate	Bpm' online	Insightly CRM	AmoCRM	Salesforce
Интуитивный интерфейс	1	4	2	3	5
Скорость внедрения	1	4	3	2	5
Ведение проектов	3	1	4	2	5
Настройка ограничений пользователей	4	2	3	1	5
Клиентский модуль	3	4	2	1	5
Ведение сделок	4	2	3	1	5
Выставление счетов	4	1	2	3	5
Управленческий учет фин. средств	1	3	2	4	5
Скорость работы	3	4	1	2	5
Стоимость	1	4	5	2	3
Средняя оценка	2,5	2,9	2,7	2,1	4,8

Результаты анализа показали, что лучшие показатели имеет система разработчика Salesforce.

BPM-online позволяет компаниям управлять полным потребительским жизненным циклом при помощи единой платформы CRM, в которую интегрировали три продукта: маркетинг, продажи, сервис [19].

Маркетинг Bpm-online – многоканальное маркетинговое программное обеспечение, которое дает возможность специалистам осуществлять продажи с последующим взаимодействием с клиентами.

Продажи Bpm-online – инструмент, автоматизирующий систему продаж. Плюсы данной функции – это возможность объединить в единую систему продажи, финансовые операции, счета, коммуникации и т.п.

Сервис Bpm-online – это инструмент для обслуживания и привлечения клиентов. Может работать как в режиме облачной системы, так и локальной, где данные размещены на серверах компании. В обоих случаях пользователи могут получить доступ к bpm-online как с веб-браузера, так и с мобильного приложения.

Дополнительные возможности системы:

- управление любыми полями в карточке клиента;
- установление режимов доступа позволяет править карты клиентов только тем сотрудникам, которые создали контрагентов, остальные только просматривают;
- хранение документов;
- формирование базы знаний с сохранением всего обучающего или необходимого для работы материала;
- формирование стандартных отчетов с визуализацией.

Исходя из всего вышеперечисленного, можно сделать вывод, что CRM-система BPM-online функциональна. Но для того, чтобы комфортно осуществлять работу в ней, необходимо загрузить дополнительные надстройки [19].

Служба управления сбытом AutoCRM – это веб-платформа, доступная в любом месте, где есть Интернет-соединение. Пользователи системы могут управлять продажами, персоналом и получать аналитическую информацию и отчеты [17].



### Функции системы AmoCRM:

- интегрируется с почтой;
- поддерживает поле для создания новых пользователей;
- уникальные теги позволяют пользователям создавать новые соглашения и контакты;
- существующие клиенты могут быть выгружены из баз данных контактов компаний, таких как Outlook и Gmail;
- возможность контролировать актуальные задачи при помощи вызовов и электронных писем.

AmoCRM обеспечивает полную видимость конвейера продаж: показывает количество продаж, доходы торговых представителей на основе тегов.

Платформы как Facebook, MailChimp, Zendesk, Dropbox и Xero могут интегрировать непосредственно с AmoCRM. Мобильное приложение доступно для скачивания в App Store или Google Play.

Помимо плюсов определены недостатки данной системы. Во-первых, отсутствует возможность разграничения доступа к информации. Во-вторых, недоступны схемы сделок, управленческие и финансовые отчеты [17].

Insightly CRM позволяет небольшим компаниям управлять проектами, контактами, продажами и документами при помощи единой платформы, которая доступна через сеть интернет и на мобильных устройствах. Соединяет пользователей с онлайн приложениями, такими как Google Календарь, Gmail, MailChimp, Evernote, Dropbox, QuickBooks, Xero и другие. [20].

Приборная панель Insightly предоставляет информацию в реальном времени по текущим задачам. Продвинутые отчеты Insightly позволяют создать таблицы и графики. Программа сообщает такие детали, как ответственный пользователь, стадия задачи, крайняя дата, вероятность завоевания клиента.

Insightly также предлагает интегрированную функцию управления проектом. Как только сделка заключена, пользователи могут отследить и управлять последующими обязательствами проекта непосредственно из CRM. Боковая панель Insightly также сохраняет электронные письма непосредственно с Gmail, предоставляет пользовательский доступ к истории разговоров, соединяет всю корреспонденцию с проектами и событиями [20].

InvGate – это платформа управления услугами, которая предлагает помощь регистрации запросов, обслуживание клиентов и техническую поддержку, самостоятельного управления знаниями и многое другое. Система совместима с Mac, Windows и iPad. Предприятия практически любой отрасли и любого размера могут использовать InvGate. Данное ПО масштабируемо и настраивается в зависимости от потребностей компании.

InvGate позволяет пользователям запускать расширенные отчеты и просматривать аналитику, автоматизировать рабочие процессы и многое другое. Детализация позволяет иметь самые последние данные, которые доступны для всех членов команды [21].

Основным недостатком данной системы является отсутствие интеграции с социальными сетями.

Система Salesforce предлагает широкий набор CRM приложений для всех видов бизнеса, с уклоном на продажи и поддержку. Предлагает вертикальные решения для управления благосостоянием и сегментами финансовых услуг. Её партнеры предлагают широкий выбор дополнительных отраслевых решений. Приложения, построенные на платформе force.com, современная архитектура которой обеспечивает повышенную гибкость и масштабируемость для организаций любого размера.

Приложение Salesforce имеет возможности управление продажами, автоматизации маркетинга, управления отношениями с партнерами и обслуживания клиентов. Они помогают организациям управлять счетами клиентов, отслеживать уровень продаж, проводить и контролировать маркетинговые кампании и обеспечить техническое обслуживание.

Система Salesforce взаимодействует с социальными сетями и осуществляет совместную работу в масштабах организации. Для специализированных организаций платформа force.com предоставляет возможность разрабатывать пользовательских приложений. Разработчики могут получить доступ к среде разработки приложений и доступ к инструментам и ресурсам, необходимым для проектирования, создания и пользовательских приложений для организации.

Salesforce позволяет управлять неограниченным количеством контактов, отслеживать сделки, управлять задачами и событиями, а также отслеживать производительность и формировать отчетности [21].

#### **1.4 Постановка задачи на разработку проекта АИС**

Основное назначение системы управления продажами состоит в организации учета продаж. Система позволит упростить и ускорить обслуживание заявок клиентов за счет автоматизации этого процесса. Разрабатываемая система предназначена для работников отдела продаж ИП Павлюк И.С.

Для создания системы необходимо решить следующие задачи:

- выбор подхода, технологии, методов, стандартов и средств разработки;
- проведение анализа требований тех.задания относительно выбранной технологии, а также уточнение запросов к ИС: инструментам, интерфейсам внешнего типа;
- исследование предметной области – выбор способов по решению данных задач;
- установление архитектурной части ИС: создание её структурной части; определение набора нужного оборудования и ПО;
- проведение анализа запросов по заданию и создание определенных спецификаций ПО, который находится на стадии проектирования;

- создание структурной части ПО и установление особенностей его элементов (спецификаций) программного обеспечения и определение спецификаций его компонентов;
- создание проекта элементов продукта программы: сервер, клиент, базы данных;
- создание элементов с применением выбранных инструментов, их тестирование в автономном режиме;
- сборка программного обеспечения и его комплексное тестирование.

Требования к программному изделию: управление менеджерами, управление заявками.

Исходные данные:

- ФИО работника;
- отдел;
- описание проблемы.

Результаты: добавление заявки клиента.

Запросы к такому свойству как надежность:

- предусмотреть проведение контроля информации, которая вводится;
- предусмотреть блокировку действий, которые являются некорректными;
- обеспечить целостность данных в БД.

Необходимо, чтобы ПО работало на IBM-совместимых ПК.

У технических средств должна быть следующая минимальная конфигурация:

- процессор (тип): Pentium 4;
- объем ОЗУ: 1 Гб;
- свободное пространство жесткого диска: 8 Гб.

Приложение должно работать под управлением операционных систем семейства Windows (Windows 7, Windows 10 и Windows 11).

## 1.5 Разработка модели бизнес-процесса ТО-ВЕ

Разработка стратегии реализации любого крупного проекта предполагает наличие ряда взаимосвязанных между собой последовательных действий – этапов, на каждом из которых решается определенная задача. В качестве примера можно привести следующие этапы:

- анализ бизнеса;
- анализ стратегии развития бизнеса;
- определение стратегических свойств ИС;
- определение функциональности ИС в целом;
- выбор стратегии автоматизации (по направлениям);
- формирование комплексного проекта;
- определение архитектуры информационной системы;
- разработка и внедрение системы.

Целью создания системы является увеличение производительности труда и возможностей масштабирования в процессах взаимодействия с клиентами. Критериями эффективности данной системы являются [5]:

- скорость и удобство обработки новых заявок;
- скорость поиска заявок, а также количество критериев для поиска;
- гибкость настройки системы под существующие бизнес-процессы в организации;
- надежность хранения информации в системе;
- простота эксплуатации;
- возможности интеграций с другим ПО.

На рисунках 5 и 6 представлены диаграммы процессов с учетом автоматизации управления продажами в ИП Павлюк И.С.

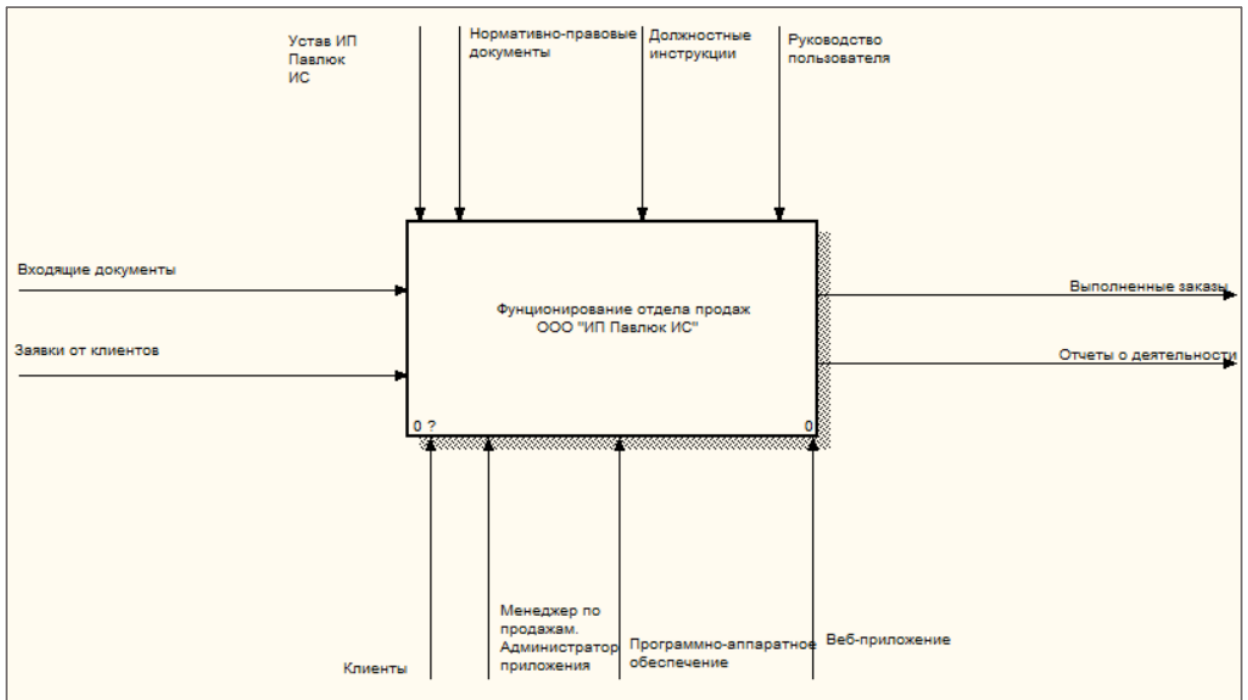


Рисунок 5 – Контекстная диаграмма (ТО-ВЕ)

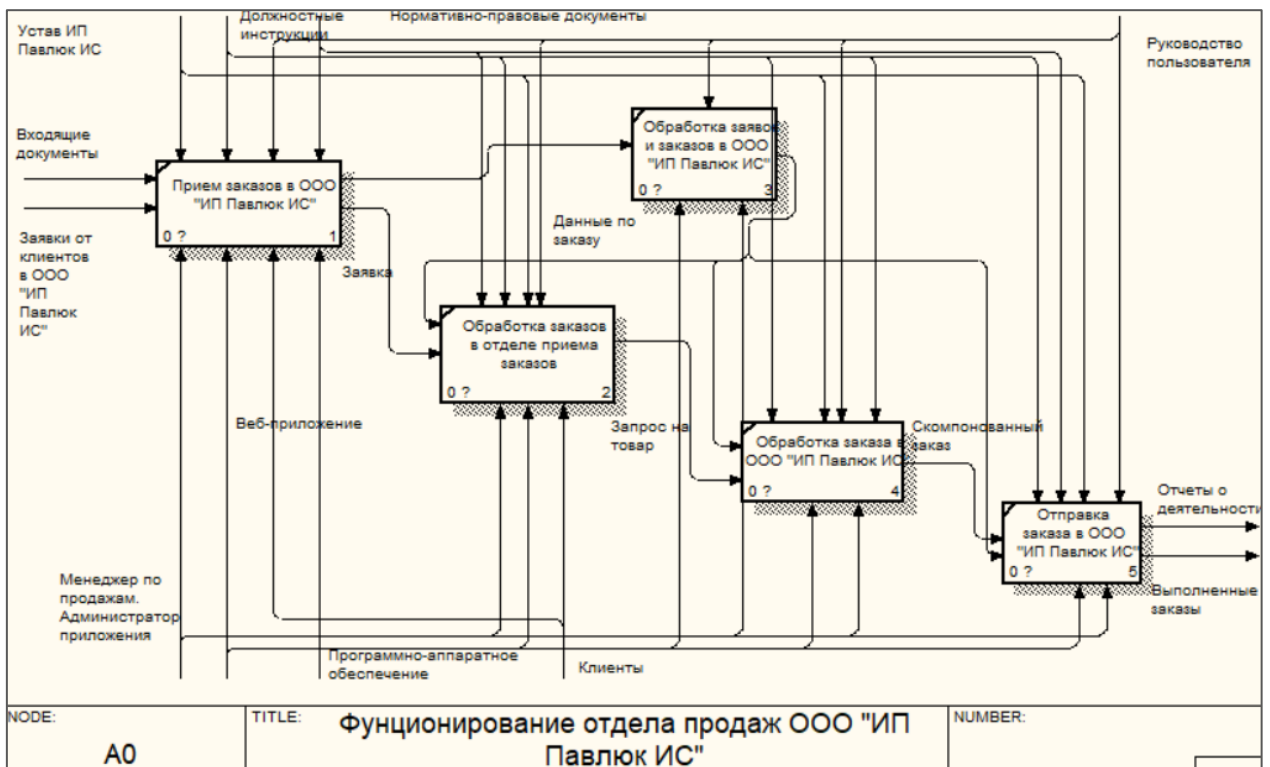


Рисунок 6 – Декомпозиция контекстной диаграммы (ТО-ВЕ)

Входящими потоками являются входящие документы, заявки от клиентов.

Управляющие потоки представлены уставом ИП Павлюк И.С., должностными инструкциями и нормативно-правовыми документами, руководством пользователя информационной системой (веб-приложением и административной панелью).

Механизмы представлены клиентами, менеджером по продажам (администратором приложения), веб-приложением (серверным и клиентским компонентами).

Выходные потоки представлены отчетами о деятельности и выполненными заказами.

В ходе декомпозиции контекстной диаграммы были определены ряд блоков, которые отображают определенные действия, которые осуществляются в отделе продаж. Они представлены следующими элементами: «прием заказов», «обработка заказов в отделе приема заказов», «обработка заявок и заказов в ИП Павлюк И.С.», «отправка заказа в ИП Павлюк И.С.».

## Глава 2 Логическое проектирование АИС

### 2.1 Выбор технологии логического моделирования АИС

Для логического моделирования информационной системы выбрана нотация UML. UML – один из самых распространённых языков моделирования. Данный язык применяется в ходе создания программного обеспечения.

Язык UML является унифицированным и прагматичным, таким образом он является графическим, наиболее простым языком общего типа. Предназначен данный язык программирования для визуализации (точнее спецификации), а также для проектирования и непосредственно документирования каждого компонента, создаваемого для разработки системы программного типа.

UML – объектно-ориентированный язык. Его применение для построения моделей информационных систем требует понимания общих принципов методологии объектно-ориентированного анализа и проектирования (ООАП). В соответствии с ООАП методология моделирования включает следующие принципы (в отношении сложных по типу систем):

- абстрагирование. Это означает, что в модель должны быть включены только те аспекты проектируемой системы, которые имеют непосредственное отношение к функции или назначению системы. Все несущественные детали опускаются, чтобы не усложнять процесс анализа и изучения полученной модели;
- множественные модели. Это означает, что одного представления сложной системы недостаточно, чтобы адекватно выразить все её особенности;
- иерархическое моделирование для сложных систем. Требуется, чтобы процесс построения модели рассматривался в контексте



фиксированных представлений на различных уровнях абстракции или детализации.

Таким образом, процесс разработки проекта данного типа можно рассматривать как непосредственно «спуск» к представлениям более детальным и сложным, логическим и физическим от наиболее общих концептуальных моделей и представлений. Добавляются постепенно и все более подробно данные и характеристики моделей на каждом этапе разработки проекта, чтобы лучше отразить различные аспекты конкретной реализации сложной системы.

Язык UML имеет определённые характеристики:

- осуществляет поддержку непосредственно каждого этапа ЖЦ ПО;
- считается языком, предназначенным для выполнения визуального моделирования, и он обеспечивает создание моделей репрезентативного типа;
- одна из стандартных нотаций по осуществлению моделирования систем, поддерживается большинством CASE-продуктов объектно-ориентированного типа;
- включает механизмы по расширению основных концепций программирования языка, а также их специализации.

Словарь UML, который содержит три типа блоков: сущности, отношения и диаграммы предполагает проведение анализа объектно-ориентированного типа и проектирование самой системы. Всего четыре типа сущностей (структурного, группирующего, поведенческого, а также аннотационного типа), четыре типа отношений (обобщение, ассоциация, реализация, зависимость) и восемь диаграмм подразумевается в UML.

UML представляет собой набор компонентов в графическом виде, который включает в себя связанный граф с так называемыми рёбрами, сущностями, что примеряется непосредственно в целях визуализации с разных позиций самой системы.

Все диаграммы UML можно условно разбить на «две группы: общие диаграммы и специальные диаграммы. Общие диаграммы практически не зависят от предмета моделирования и могут применяться в любом программном проекте. Специальные диаграммы служат для дополнения какой-либо общей диаграммы, например, являются её частным случаем или играют вспомогательную роль, уточняя некоторые детали» [12].

В языке программирования чаще всего в целях создания моделей ИС применяют следующие диаграммы:

- взаимодействия – эти диаграммы представляют собой модель процессов обмена данными непосредственно между объектами. Взаимодействие бывает двух типов: в виде последовательности или диаграмм кооперативного типа;
- состояний – это диаграммы в виде модели динамического поведения системы и её компонентов во время перехода из одного состояния в другое;
- прецедентов – модель работы системы в среде обобщенного типа;
- классов – модель логического типа, показывающая статическую структурную часть системы и коммуникации непосредственно между её компонентами;
- БД – структуры базы данных;
- компонентов – модель подчинения низших высшим в рамках подсистемы, показывающая нахождение БД, а также интерфейсов ИС и непосредственно приложений;
- развертывания – это модель архитектуры самой системы в её физическом представлении, которая показывает конкретно конфигурацию ИС;
- деятельности – это модель осуществляемых бизнес-процессов или же поведения системы в пределах непосредственно самого прецедента.

Диаграммы представляются в свернутом виде компонентов. Из этих элементов состоит создаваемая система (кроме моделей наиболее простого типа). Каждый компонент может находиться в каждой диаграмме или в нескольких.

UML-диаграммы обеспечивают возможность осуществления с разной позиции описание системы. В то же время отдельной диаграммы недостаточно, чтобы полностью описать систему, так как фокусирование диаграммы осуществляется на определенном аспекте работы системы, и диаграмма на определенном уровне абстракции описывает определенный аспект системы.

Частный и ограниченный взгляд на разрабатываемую систему представляет каждая диаграмма. Моделью системы одна диаграмма сама по себе не может быть. Только совокупность нескольких различных типов диаграмм может быть моделью системы и описывать её достаточно полно.

Для описания различных аспектов функционирования и структуры на разных этапах её разработки и, следовательно, на разных этапах её моделирования применяют UML.

Исходные модели и диаграммы деятельности на этапе разработки концептуальной модели ИС используются для описания процессов, подлежащих автоматизации, а модели бизнес-объектов и диаграммы последовательностей – для описания бизнес-объектов.

В виде модели прецедентов системного типа проходит процесс проектирования, а предварительный этап осуществляется с применением следующих диаграмм: классов, а также последовательностей, состояний. Требования к системе описываются на этапе проектирования логической модели.

Проектирование системы начинается с анализа и моделирования деятельности организации. Модель прецедентов описывает процессы с позиции непосредственно внешнего пользователя, таким образом отражая внешний взгляд на организационную деятельность. Этот этап завершается

после создания диаграмм деятельности для всех идентифицированных прецедентов. На более поздних этапах анализа и проектирования ИС можно определить дальнейшие детали деятельности блока автоматизации. Таким образом, разработанная модель прецедента может быть приспособлена в дальнейшем несколько раз.

## **2.2 Логическая модель АИС: описание**

Для представления логического поведения системы используются диаграммы вариантов использования, которые представляют пользователей (субъектов) системы и функции, выполняемые каждым субъектом. Диаграмма примеров использования – это первоначальное концептуальное представление системы во время проектирования и разработки. Эта диаграмма состоит из действующих лиц, вариантов использования и отношений между ними. Диаграмма может быть структурирована с использованием общих элементов нотации: аннотаций и механизмов расширения.

Значение программы заключается в следующем: система, которая проектируется, включает в себя множество субъектов (актер), которые посредством вариантов применения работают с системой. В то же время актер (действующее лицо) – это система, объект или субъект. Действующее лицо извне работает с моделируемой системой. Вариант использования – это спецификация услуг (функций). Данные функции система предоставляет непосредственно действующему лицу. Другими словами, каждый вариант использования определяет набор операций, которые система должна выполнить при взаимодействии с актером. Модель никоим образом не отражает того, как этот набор действий должен быть реализован. Эта модель представляет актеров с различными вариантами использования. Основными элементами диаграммы являются актер и вариант. Участник – это набор логически связанных ролей, которые выполняются во взаимодействии с

предшественниками или сущностями (система, подсистема или класс). Участником может быть человек или другая система, подсистема или класс, которые представляют что-то вне сущности. Графически участник представлен в виде «человека».

Диаграмма бизнес-прецедентов предназначена для графического описания взаимодействия участников в процессе реализации некоторых вариантов использования. Она помогает представить действия участников процессов и в дальнейшем определить, какие из функций участников могут быть возложены на проектируемую информационную систему. На рисунке 7 показана диаграмма бизнес-прецедентов, созданная в программе MS Visio.

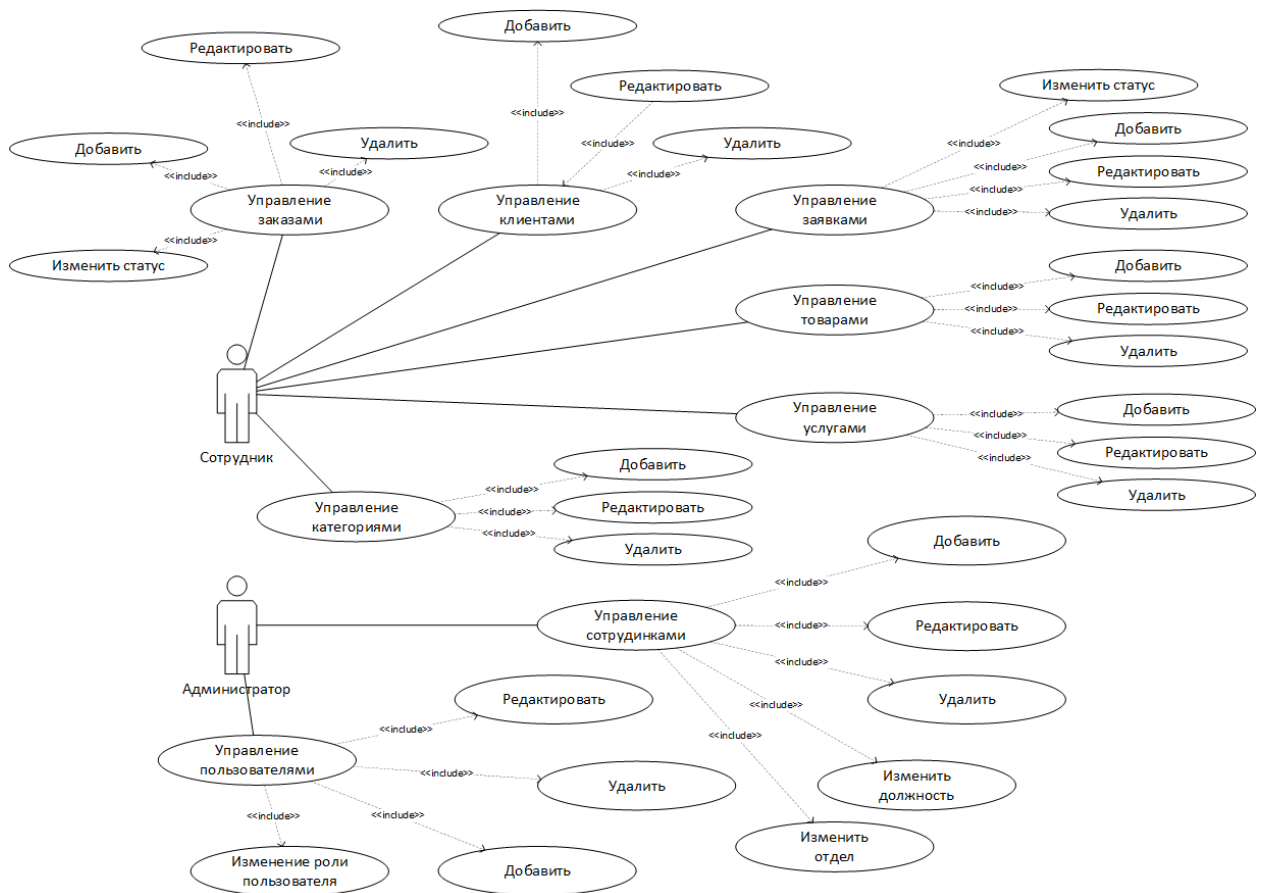


Рисунок 7 – Диаграмма прецедентов

Прецедент представляет собой описание последовательности событий (включая варианты), осуществляемых системой, которые в результате являются эффективными. Прецедент выражает поведение объекта, описывая

взаимодействие между участниками и системой. В случае прецедента не описывается «как» достигается результат, а только «что» выполняется. Прецедент можно представить очень просто: в виде эллипса с его названием внутри.

## 2.3 Информационное обеспечение АИС

### 2.3.1 Классификаторы и системы кодирования

Чтобы адаптировать бизнес-информацию для поиска, обработки на компьютере и по каналам, она должна быть представлена непосредственно в цифровой форме. Для этого её необходимо сначала упорядочить (классифицировать), а затем формализовать (закодировать) с помощью классификатора. Наиболее важными объектами классификации и кодирования являются эталонные признаки, описывающие процессы, место и время протекания процессов, а также субъекты и объекты действия, отраженные в показателе. В документации применяется кодирование по атрибутам, в соответствии с которыми информация группируется в ПК. Разработчики информационной системы имеют локальный классификатор, использующий иерархический метод классификации.

Структура нормативно-справочной информации, используемой для разработки базы данных для информационной системы показана в таблице 2.

Таблица 2 – Структура нормативно-справочной информации

Наименование кодируемого множества объектов	Значность кода	Система кодирования	Вид классификатора
Код клиента	XXXXX	порядковая	локальный
Код объекта в заказе	XXXXX	порядковая	локальный
Код категории	XXXXX	порядковая	локальный
Код услуги	XXXXX	порядковая	локальный
Код товара	XXXXX	порядковая	локальный
Код отдела	XXXXX	порядковая	локальный
Код должности	XXXXX	порядковая	локальный
Код сотрудника	XXXXX	порядковая	локальный

Длина кода определяется максимально возможным количеством объектов определенного типа.

### **2.3.2 Характеристика нормативно-справочной и входной оперативной информации**

К нормативно-справочной информации, используемой при работе информационной системы, относятся:

- справочник должностей;
- справочник отделов.

К входной информации, используемой для работы системы, относится:

- данные о клиентах;
- данные о сотрудниках;
- данные о категориях;
- данные заявок;
- данные заказов;
- данные о товарах;
- данные об услугах;
- информация о товарах и услугах в заказах.

Вводится информация в систему посредством HTML-форм, которые находятся на страницах информационной системы. Для ввода информации в информационную систему предназначены специально разработанные формы.

Справочник «Должность» включает следующие категории:

- идентификатор;
- название должности;
- описание должности.

Справочник «Отделы» включает следующие категории:

- идентификатор;
- название отдела;
- описание отдела.

Описание нормативно-справочной информации представлено в таблице 3.

Таблица 3 – Перечень нормативно-справочных документов

Название документа	Содержание	Периодичность
Список клиентов	Клиенты	Перед внедрением
Список заявок	Заявки клиентов	Перед внедрением
Список услуг	Услуги	Перед внедрением
Список товаров	Товары	Перед внедрением
Список категорий товаров и услуг	Категории товаров и услуг	Перед внедрением
Список отделов	Отделы компании	Перед внедрением
Список должностей	Должности сотрудников	Перед внедрением

Входной информацией являются данные о заявках, заказах, клиентах, услугах, товарах, ролях пользователей. Передаются указанные данные на веб-сервер информационной системы и после обработки отправляются на сервер базы данных, где сохраняются для последующего использования.

### 2.3.3 Характеристика выходной информации

Выходная информация формируется в приложении, отправляется в веб-браузер и отображается пользователям в виде HTML-страниц. Получается информация сервером приложения посредством обращения к серверу базы данных.

Для получения данных на сервере базы данных выполняются следующие SQL-запросы:

```
SELECT DISTINCT c0_.id AS id_0, c0_.fullname AS fullname_1 FROM customer c0_ ORDER BY c0_.fullname DESC LIMIT 20
```

```
SELECT c0_.id AS id_0, c0_.fullname AS fullname_1, c0_.address AS address_2, c0_.phone AS phone_3, c0_.email AS email_4, c0_.access_info AS access_info_5 FROM customer c0_ WHERE c0_.id IN (?) ORDER BY c0_.fullname DESC
```

```
SELECT count(c0_.id) AS sclr_0 FROM customer c0_
```

```
SELECT DISTINCT o0_.id AS id_0 FROM `order` o0_ LIMIT 20
```



```
SELECT o0_.id AS id_0, o0_.ordertitle AS ordertitle_1, o0_.date AS date_2, o0_.summ AS summ_3, o0_.paystatus AS paystatus_4, o0_.customer_id AS customer_id_5, o0_.user_id AS user_id_6 FROM `order` o0_ WHERE o0_.id IN 1
```

```
SELECT t0.id AS id_1, t0.aplformtitle AS aplformtitle_2, t0.appdate AS appdate_3, t0.app_status AS app_status_4, t0.orders_id AS orders_id_5 FROM application_form t0 WHERE t0.orders_id = 1
```

```
SELECT count(o0_.id) AS sclr_0 FROM `order` o0_
```

```
SELECT t0.id AS id_1, t0.fullname AS fullname_2, t0.address AS address_3, t0.phone AS phone_4, t0.email AS email_5, t0.access_info AS access_info_6 FROM customer t0 WHERE t0.id = 1
```

```
SELECT DISTINCT s0_.id AS id_0 FROM service s0_ LIMIT 20
```

```
SELECT s0_.id AS id_0, s0_.title AS title_1, s0_.description AS description_2, s0_.cost AS cost_3, s0_.service_articul AS service_articul_4 FROM service s0_ WHERE s0_.id IN (1)
```

```
SELECT count(s0_.id) AS sclr_0 FROM service s0_
```

```
SELECT DISTINCT p0_.id AS id_0 FROM product p0_ LIMIT 20
```

```
SELECT p0_.id AS id_0, p0_.title AS title_1, p0_.description AS description_2, p0_.cost AS cost_3, p0_.prod_articul AS prod_articul_4, p0_.image AS image_5 FROM product p0_ WHERE p0_.id IN 1
```

```
SELECT count(p0_.id) AS sclr_0 FROM product p0_
```

```
SELECT DISTINCT a0_.id AS id_0 FROM application_form a0_ LIMIT 20
```

```
SELECT a0_.id AS id_0, a0_.aplformtitle AS aplformtitle_1, a0_.appdate AS appdate_2, a0_.app_status AS app_status_3, a0_.orders_id AS orders_id_4 FROM application_form a0_ WHERE a0_.id IN 1
```

```
SELECT count(a0_.id) AS sclr_0 FROM application_form a0_
```

```
SELECT DISTINCT o0_.id AS id_0 FROM object_in_app_form o0_ LIMIT 20
```

```
SELECT o0_id AS id_0, o0_product_count AS product_count_1,
o0_service_count AS service_count_2, o0_application_form_id AS
application_form_id_3, o0_service_id AS service_id_4, o0_product_id AS
product_id_5 FROM object_in_app_form o0_ WHERE o0_id IN (?)
```

```
SELECT count(o0_id) AS sclr_0 FROM object_in_app_form o0_
```

```
SELECT t0.id AS id_1, t0.aplformtitle AS aplformtitle_2, t0.appdate AS
appdate_3, t0.app_status AS app_status_4, t0.orders_id AS orders_id_5 FROM
application_form t0 WHERE t0.id = 1
```

```
SELECT t0.id AS id_1, t0.title AS title_2, t0.description AS description_3,
t0.cost AS cost_4, t0.prod_articul AS prod_articul_5, t0.image AS image_6 FROM
product t0 WHERE t0.id = 1
```

Представленные запросы формируются с помощью ORM Doctrine. На основании результатов выполнения представленных запросов формируются выходные документы.

Выходной информацией являются следующие наборы данных:

- клиенты компании;
- заявки клиентов;
- заказы;
- перечень услуг ИП Павлюк И.С.;
- перечень товаров ИП Павлюк И.С.;
- данные пользователей;
- профиль пользователя;
- данные клиентов;
- данные заказов;
- роли пользователей;
- разрешения пользователя.

В таблице 4 представлены характеристики выходной информации.

Таблица 4 - Характеристика выходной информации

Наименование	Назначение реквизита	Тип реквизита
Клиент		
Номер клиента	Номер клиента	Числовой
ФИО	Фамилия, Имя, Отчество	Текстовый
Номер телефона	Номер телефона клиента	Текстовый
Адрес	Адрес клиента	Текстовый
Email	Электронный адрес	Текстовый
Дополнительно	Дополнительная информация	Текстовый
Заявка		
Номер	Номер заявки	Числовой
Дата	Дата создания заявки	Дата
Название	Название заявки	Текстовый
Пользователь	Номер пользователя	Числовой
Статус	Статус заявки	Логический
Заказ		
Номер	Номер заказа	Числовой
Статус оплаты	Статус оплаты	Числовой
Клиент	Номер клиента	Числовой
Сумма	Сумма заказа	Числовой
Наименование	Наименование заказа	Текстовый
Дата	Дата создания заказа	Дата/время
Услуга		
Номер	Номер услуги	Числовой
Краткое описание	Краткое описание услуги	Текстовый
Наименование	Наименование услуги	Текстовый
Стоимость	Стоимость услуги	Числовой
Товар		
Номер	Номер услуги	Числовой
Краткое описание	Краткое описание товара	Текстовый
Наименование	Наименование товара	Текстовый
Стоимость	Стоимость товара	Числовой
Артикул	Артикулярный номер	Текстовый
Категория		
Номер	Номер категории	Числовой
Наименование	Наименование категории	Текстовый
Описание	Описание категории	Текстовый
Отдел		
Номер	Номер отдела	Числовой
Наименование	Наименование отдела	Текстовый
Описание	Описание отдела	Текстовый
Должность		
Номер	Номер должности	Числовой
Наименование	Наименование отдела	Текстовый
Описание	Описание должности	Текстовый

#### Продолжение таблицы 4

Наименование	Назначение реквизита	Тип реквизита
Товар (услуга) в заявке		
Клиент	Номер клиента	Числовой
Товар	Номер товара	Числовой
Услуга	Номер услуги	Числовой
Заявка	Номер заявки	Числовой
Номер	Номер записи о товаре (услуге) в заявке	Числовой

## 2.4 Проектирование базы данных АИС

### 2.4.1 Выбор технологии проектирования БД АИС

В настоящее время есть разные приемы выделения сущностей, а также связей – IDEF1X, а также нотации Чена. Сущность является реальным или воображаемым объектом, который имеет важное значение для анализируемой предметной области.

В обозначение IDEF1X все сущности подразделяются на зависимые и независимые. Самостоятельная сущность, которая является независимой, изображается в виде простого прямоугольника, а зависимая - с округленными углами прямоугольник.

В случае если экземпляр сущности-потомка единственным образом идентифицируется по его связи с сущностью-родителем, то такая связь именуется идентифицирующей, в ином случае – не идентифицирующей. Идентифицирующая представляется в виде сплошной линии, а не идентифицирующая непосредственно в виде пунктирной линии.

### 2.4.2 Разработка концептуальной модели данных АИС

Логический тип представления данных описывает концептуальную модель непосредственно самой предметной области.

На рисунке 8 изображена концептуальная модель данных информационной системы.

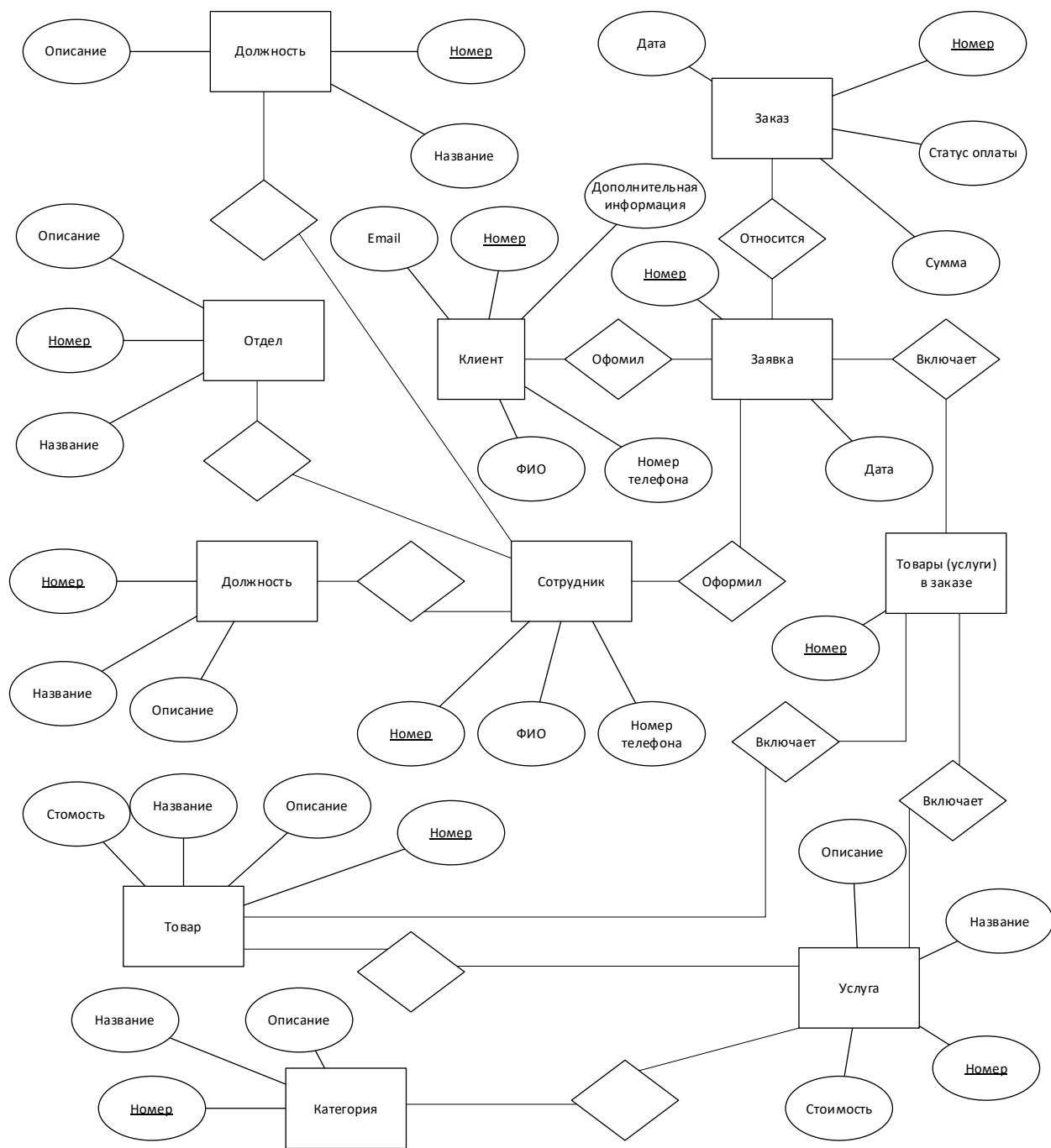


Рисунок 8 – Концептуальная модель данных информационной системы

Концептуальная модель может быть определена как модель, которая состоит из концепций и их отношений.

### 2.4.3 Разработка логической модели данных АИС

Логическая модель базы данных АИС для ИП Павлюк И.С. показана на рисунке 9.

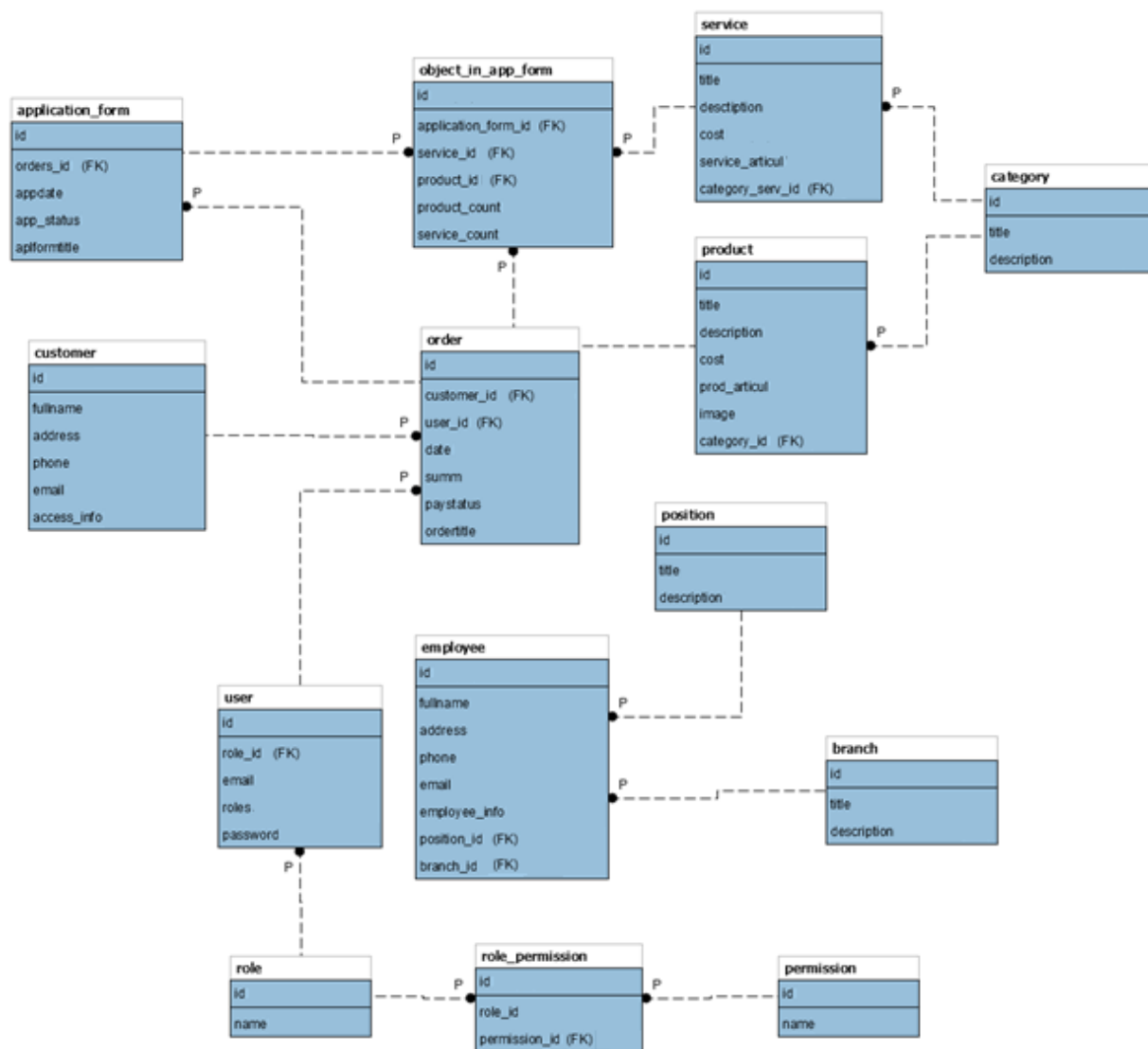


Рисунок 9 – Логическая модель данных

## 2.5. Требования к аппаратно-программному обеспечению АИС

Требования к аппаратному и программному обеспечению АИС приведены в таблице 5.

Таблица 5 – Требования к аппаратному и техническому обеспечению

Параметр/Характеристика	Значение
Процессор	2,2 GHz Intel Core i7
Кол-во ядер	8
Скорость процессора	2.8Ghz
ОЗУ	16Gb
Дисковое пространство	SSD256Gb
Видеосистема и монитор	Разрешающая способность 1920x1080
Операционная система	Winsows 10 Pro 1809

Необходим сервер, который обрабатывает запросы и отправляет данные из БД непосредственно в браузер. Применяется программная среда Docker, которая преимущественно используется для запуска сервера базы данных. Запуск приложения осуществлялся с помощью Symfony CLI, при этом есть возможность запускать Symfony с Docker-контейнера, но в процессе разработки этот способ развертывания приложения не отличается высокой скоростью.

Docker является платформой, предназначенной для разработки, а также для проведения тестов и внедрения приложений. Данная платформа осуществляет упаковку программного обеспечения непосредственно в блоки стандартизованного типа, которые являются контейнерами. Непосредственно каждый контейнер включает приложения, которые нужны для работы: библиотеки, средства системы, код и рабочую среду. Посредством Docker можно быстро развертывать приложения и масштабировать их в любой среде и оставаться уверенным в работоспособности кода.

В настройках программы предоставляется возможность изменить конфигурацию приложения, определить ресурсы для Docker, включить использование платформы для управления конвейеризованными сервисами.

За счет Docker возможно получение объекта, работающего с высоким уровнем защиты в независимости от вида платформы. Доступный и легкий в восприятии синтаксис Docker дает возможность для контроля осуществляемых операционных действий.

## Глава 3 Физическое проектирование АИС

### 3.1 Выбор архитектуры АИС

Разрабатываемая система базируется на трехуровневой архитектуре (рисунок 10).



Рисунок 10 – Трехуровневая архитектура системы

Назначение добавочного уровня – руководство выполнением приложений и управлением базой данных. Подобно двухуровневой модели, отдельные уровни могут быть расположены либо на разных ПК, либо в режиме тестирования на одном.

Архитектура клиент-сервер используется для разработки веб-приложений (программных продуктов), которые взаимодействуют через стандартный браузер для выхода в интернет. Взаимодействие с пользователем происходит через страницы и формы, из которых пользователь получает необходимые данные и отправляет часть из них обратно.

Формирование запросов к БД осуществляется непосредственно сервером, далее происходит отправка запросов серверу БД. Там происходит обработка, а также выполняются действия, возвращается конечный результат.

Браузером обеспечивается взаимодействие пользователя с приложением. За счет использования браузера можно управлять всеми возможностями приложения.



### 3.2 Выбор технологии разработки программного обеспечения АИС

В качестве языка программирования интерфейса приложения будет использован PHP. Средой разработки веб-приложения выбрана программа PHPStorm. Функции веб-сервера выполняет Simfony CLI и Docker.

Сравнительная характеристика PHP с другими языками программирования приведена в таблице 6.

Таблица 6 – Сравнение языков программирования

Критерий	Создание динамических веб-приложений	Создание веб-приложений и компьютерах программ	Создание динамических веб-приложений и компьютерных программ
	PHP	Ruby	Python
Год создания	1995	1995	1991
Сложность освоения	6	4	5
ООП	Да	Да	Да
Инструкция break	Да	Да	Да
Многомерные массивы	Да	Да	Да
Цикл foreach	Да	Да	Да
Множественное наследование	Нет	Нет	Да
Макросы	Нет	+/-	Нет
Именованные параметры	Нет	Да	Да
Наличие библиотек для работы с графикой и мультимедиа	Да	Да	Да
Работа с MySQL	Да	Да	Да
Наличие удобных и доступных средств разработки	Да (NetBeans)	Нет	Да (Visual Studio)
Поддержка популярными веб-серверами	Да (по умолчанию)	Нет (дополнительные службы)	Нет (дополнительные службы)

Анализ показал, что PHP имеет определенные положительные качества в направлении доступности средств разработки и наличия на распространенных веб-серверах. Поэтому в качестве языка программирования для создания ИС будет использоваться PHP.

Для разработки и управления веб-приложением выбран фреймворк Symfony. Работает с PHP 5 и выше. Имеет поддержку множества баз данных (MySQL, PostgreSQL или другая PDO-совместимая СУБД).

### **3.3 Выбор СУБД АИС**

MySQL была выбрана в качестве системы по управлению базами данных для проекта. Это высокопроизводительная многопоточная и многопользовательская реляционная СУБД.

Тесты производительности, доступные на сайте MySQL, показывают, что MySQL превосходит почти все другие системы по управлению БД, доступные в настоящее время.

MySQL осуществляет поддержку таких операционных систем, как: UNIX, Linux, FreeBSD, OS/2, MacOS. Работает на разных типах архитектуры, среди них: Intelx86, Alpha, SPARC, PowerPC и IA64.

MySQL может работать с разными языками, в частности, с C, C++, Java, Eiffel, Perl, Python, PHP, Ruby, также есть коннекторы для приложений ODBC, JDBC и .NET.

### **3.4 Разработка физической модели данных АИС**

На основе логической модели данных создана физическая модель данных для СУБД MySQL.

Физическое проектирование определяет особенности хранения данных, методов доступа и т.д. Физический уровень проектирования БД (с точки зрения администратора) – это группировка данных, индексов, методов доступа.

Особенности конкретной СУБД должны включать некоторые лимиты на именование объектов БД, а также на поддерживаемые типы данных.

На рисунке 11 построена физическая модель данных для СУБД MySQL.

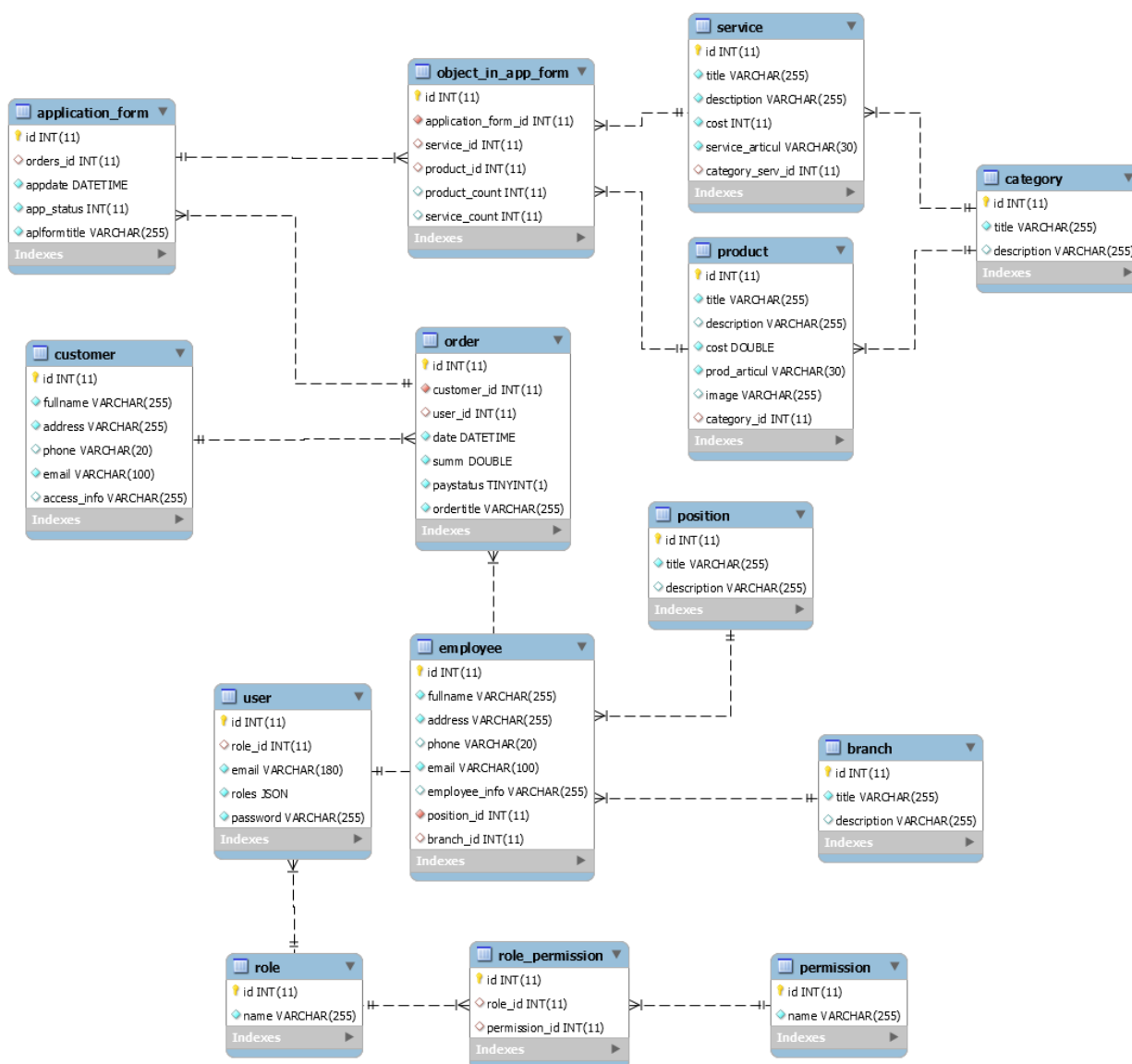


Рисунок 11 – Физическая модель данных

В результате создана нормализованная база данных на сервере MySQL, что показано в приложении RHPMyAdmin (рисунок 12).

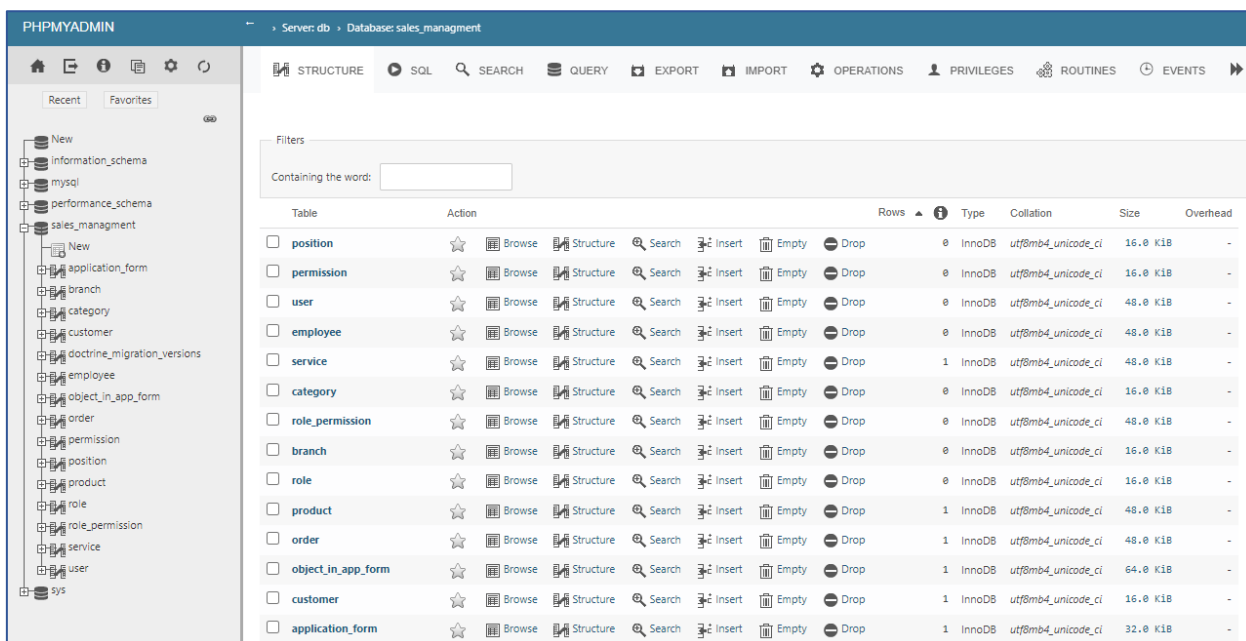


Рисунок 12 – База данных на сервере MySQL

Разработана физическая модель данных для конкретной СУБД – MySQL.

### 3.5 Разработка программного обеспечения АИС

#### 3.5.1 Схема взаимосвязи модулей приложения АИС

В разрабатываемом проекте применялась иерархическая структура модулей приложения.

Подобное отражение данных оптимально соответствует переработанной, структурированной и организованной информации.

Разрабатываемая система состоит из модулей, которые отвечают за функциональность системы. На рисунке 13 показана модульная структура разрабатываемой системы.

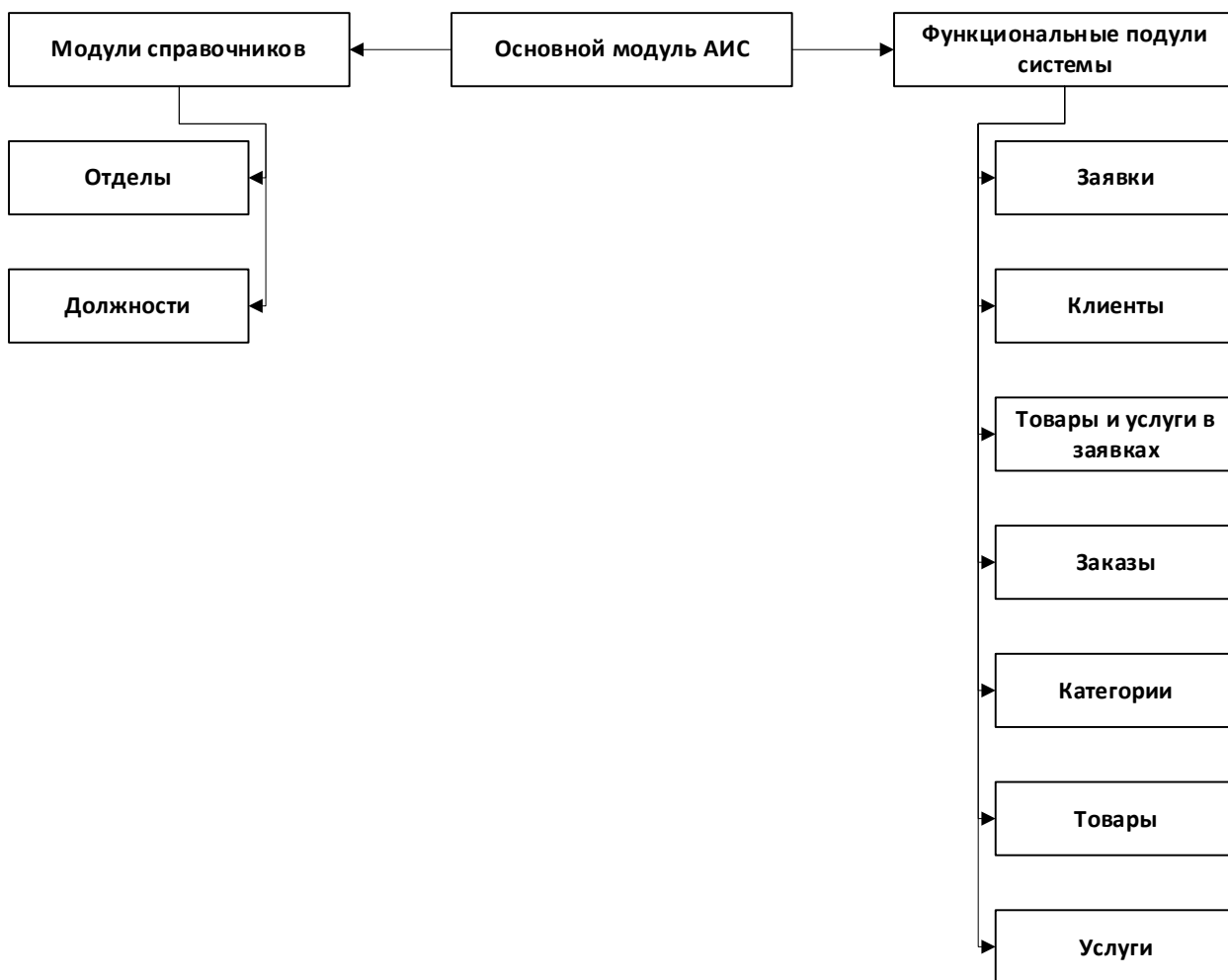


Рисунок 13 – Модули информационной системы

### 3.5.2 Описание модулей приложения АИС

Модуль «Отделы» обеспечивает возможность работы с таблицей базы данных «Отделы»: добавлять, редактировать и удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 14.

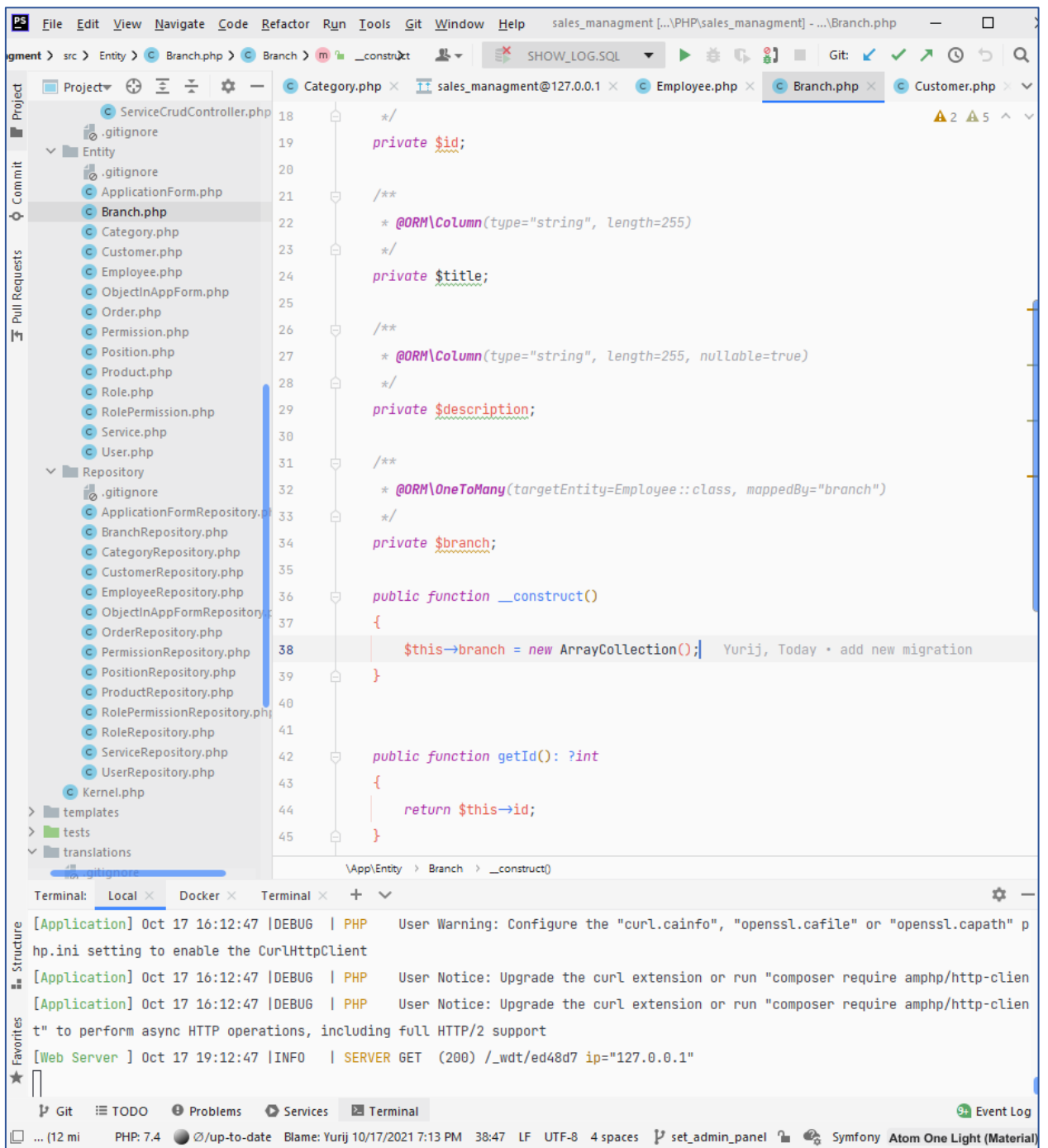


Рисунок 14 – Работа с модулем «Отделы»

Модуль «Должности» обеспечивает возможность работы с таблицей базы данных «Должности»: добавлять, редактировать и удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 15.

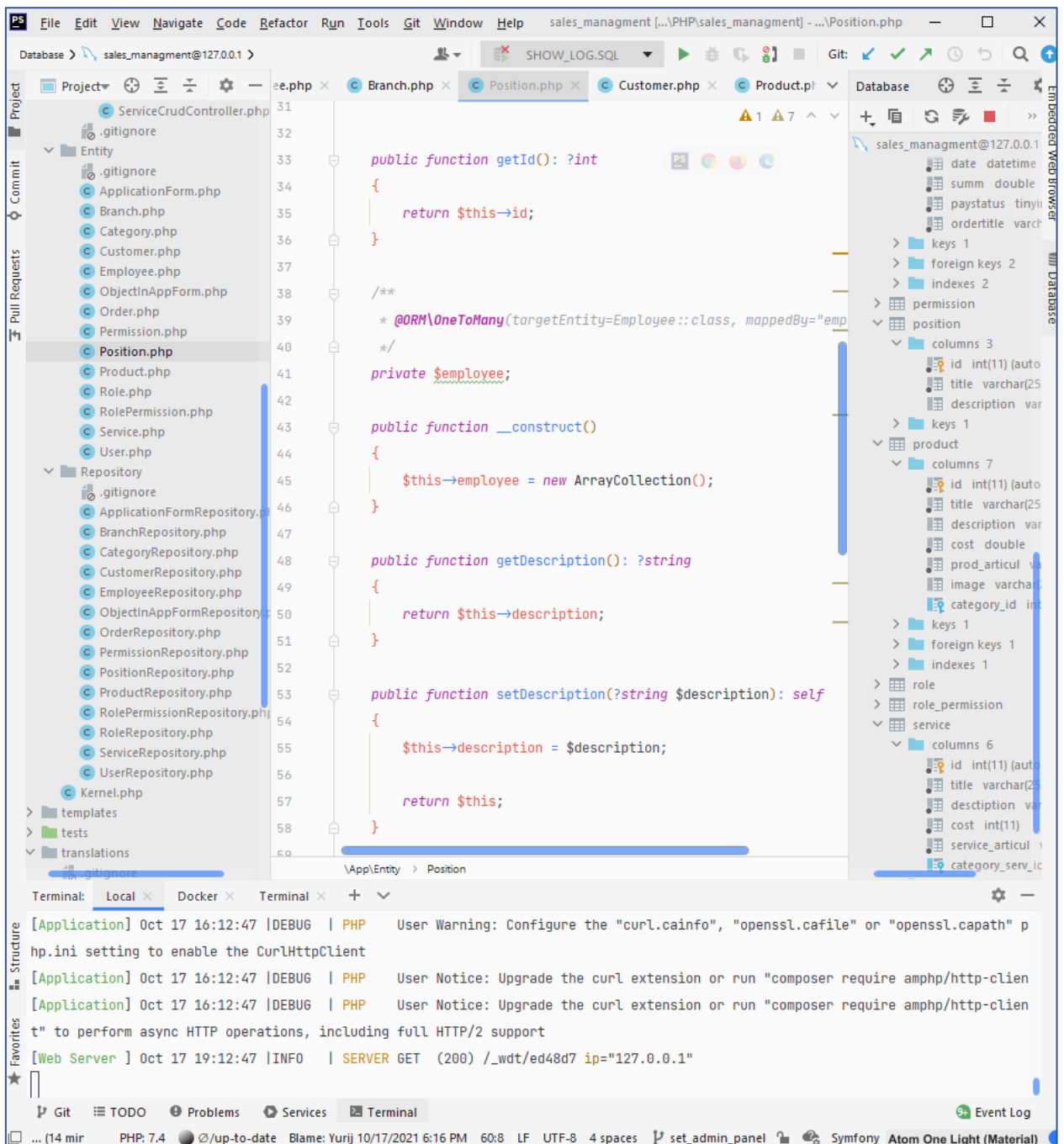


Рисунок 15 – Работа с модулем «Должности»

Модуль «Заявки» обеспечивает возможность работы с таблицей базы данных «Заявки»: добавлять, редактировать и удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 16.

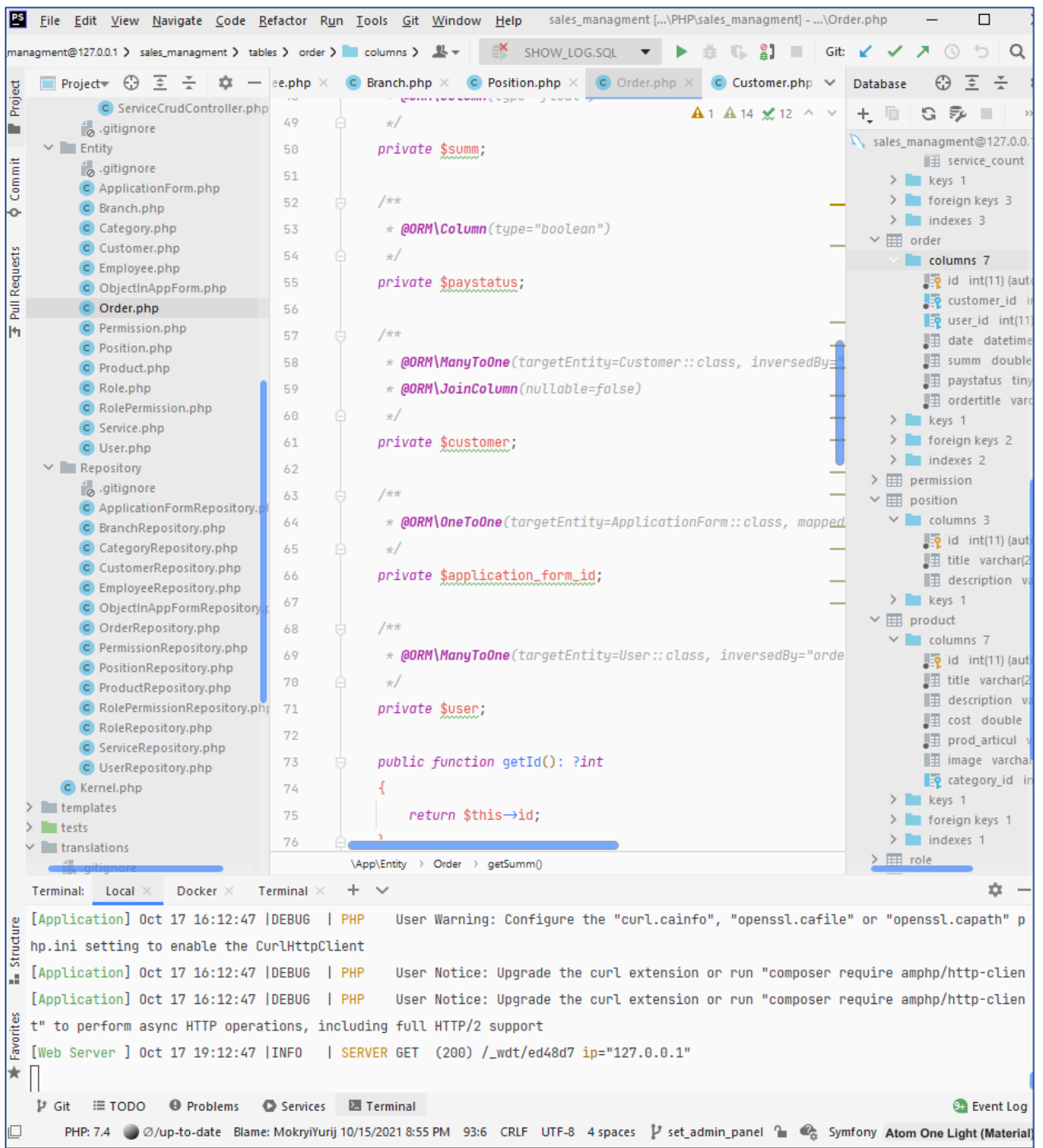


Рисунок 16 – Работа с модулем «Заявки»

Модуль «Клиенты» обеспечивает возможность работы с таблицей базы данных «Клиенты»: добавлять, редактировать и удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 17.



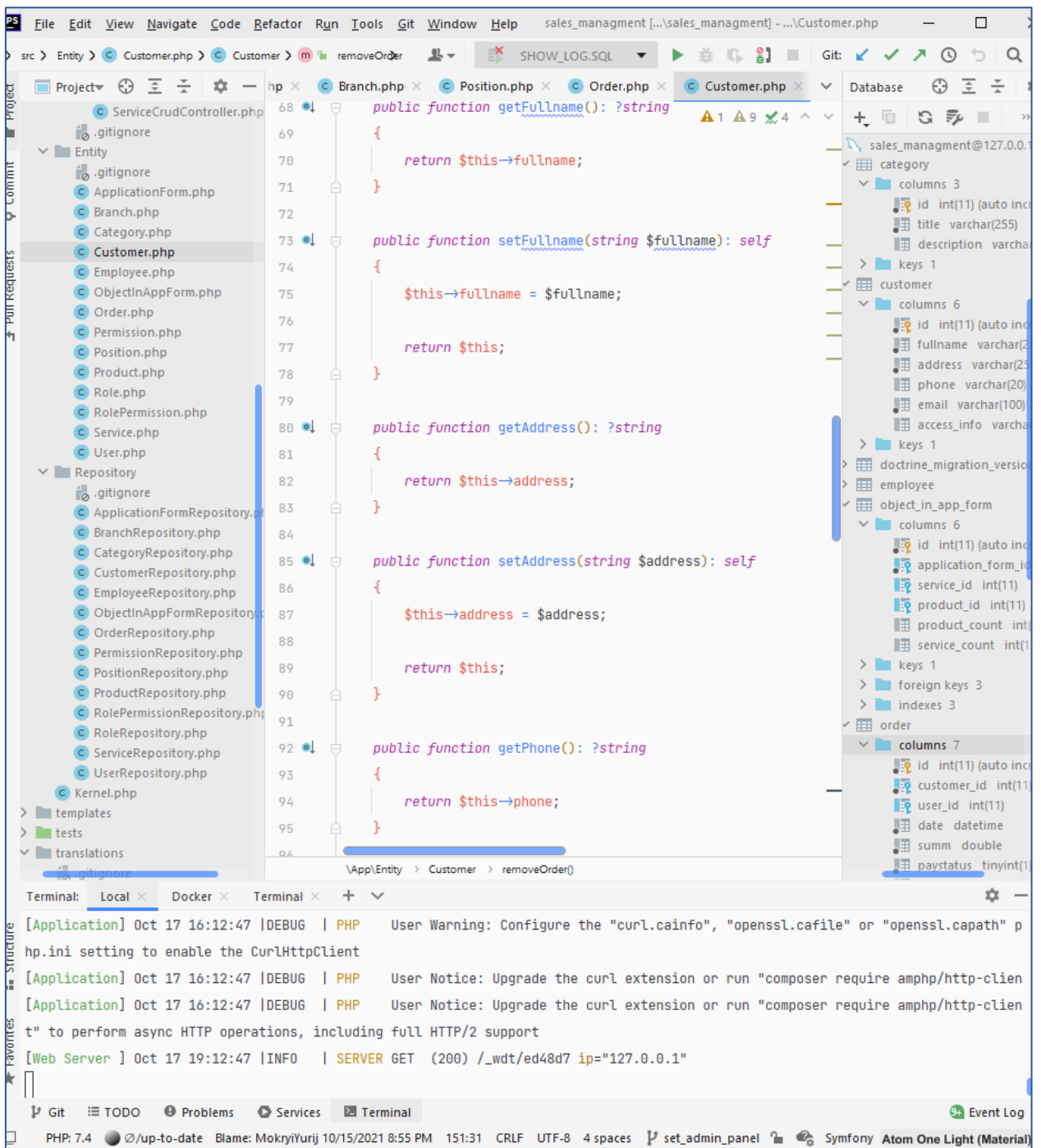


Рисунок 17 – Работа с модулем «Клиенты»

Модуль «Товары и услуги в заявках» обеспечивает возможность работы с таблицей базы данных «Товары и услуги в заявках»: добавлять, редактировать, удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 17.

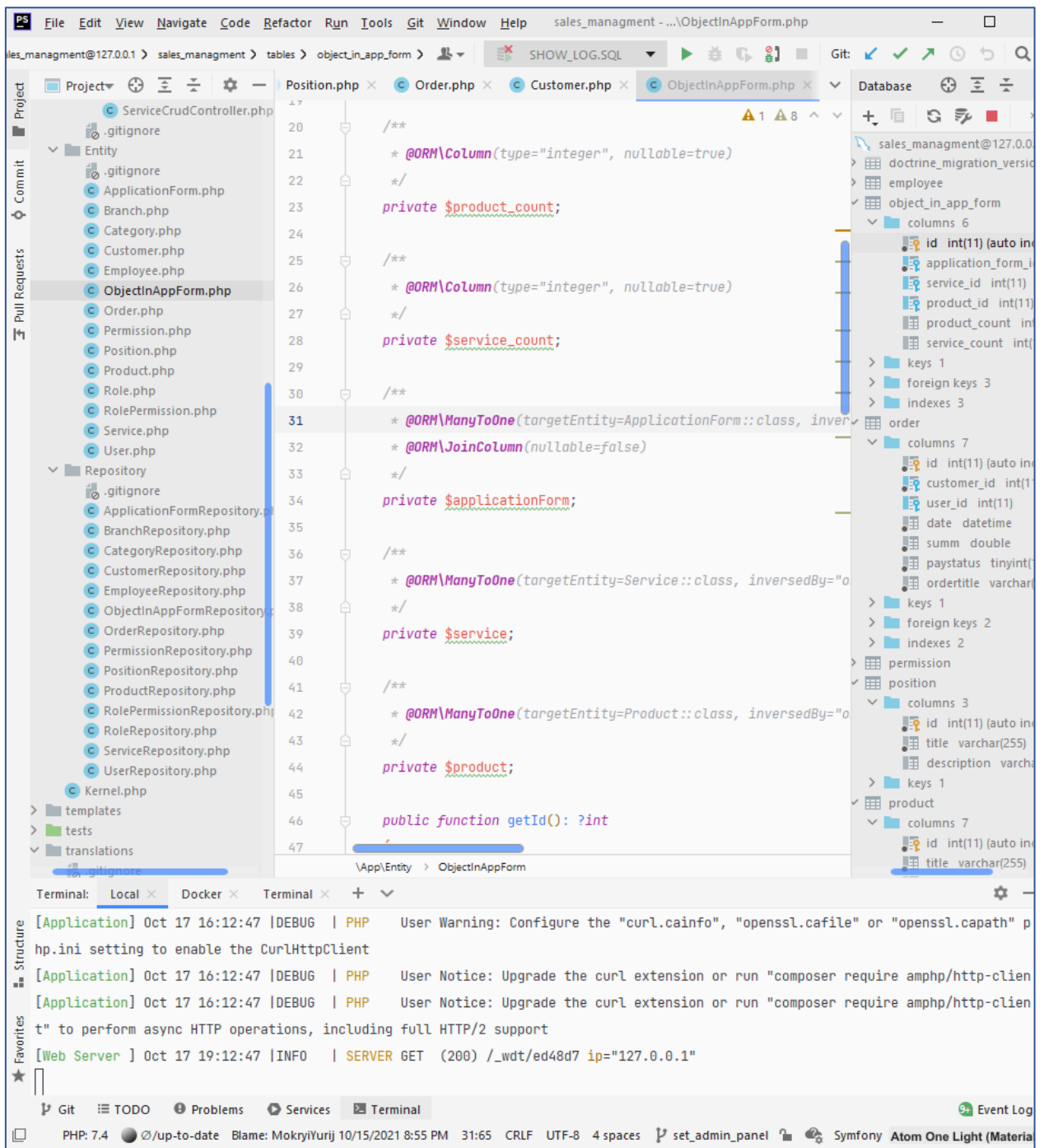


Рисунок 17 – Работа с модулем «Товары и услуги в заявках»

Модуль «Заказы» обеспечивает возможность работы с таблицей базы данных «Заказы»: добавлять, редактировать и удалять записи в таблице, а также изменять статус заказов. Процесс работы с данным модулем показан на рисунке 18.

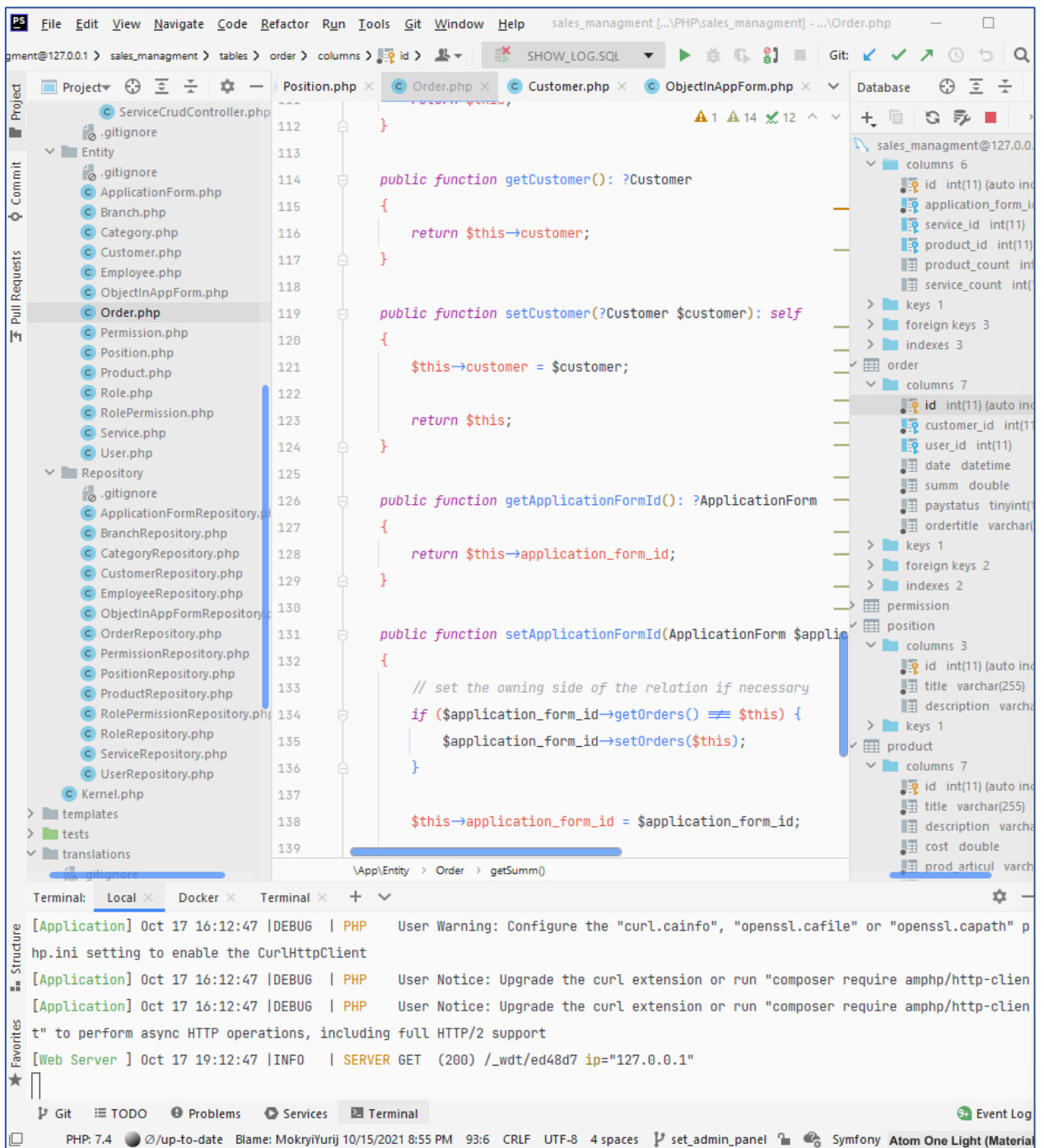


Рисунок 18 – Работа с модулем «Заказы»

Модуль «Категории» обеспечивает возможность работы с таблицей базы данных «Категории»: добавлять, редактировать и удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 19.

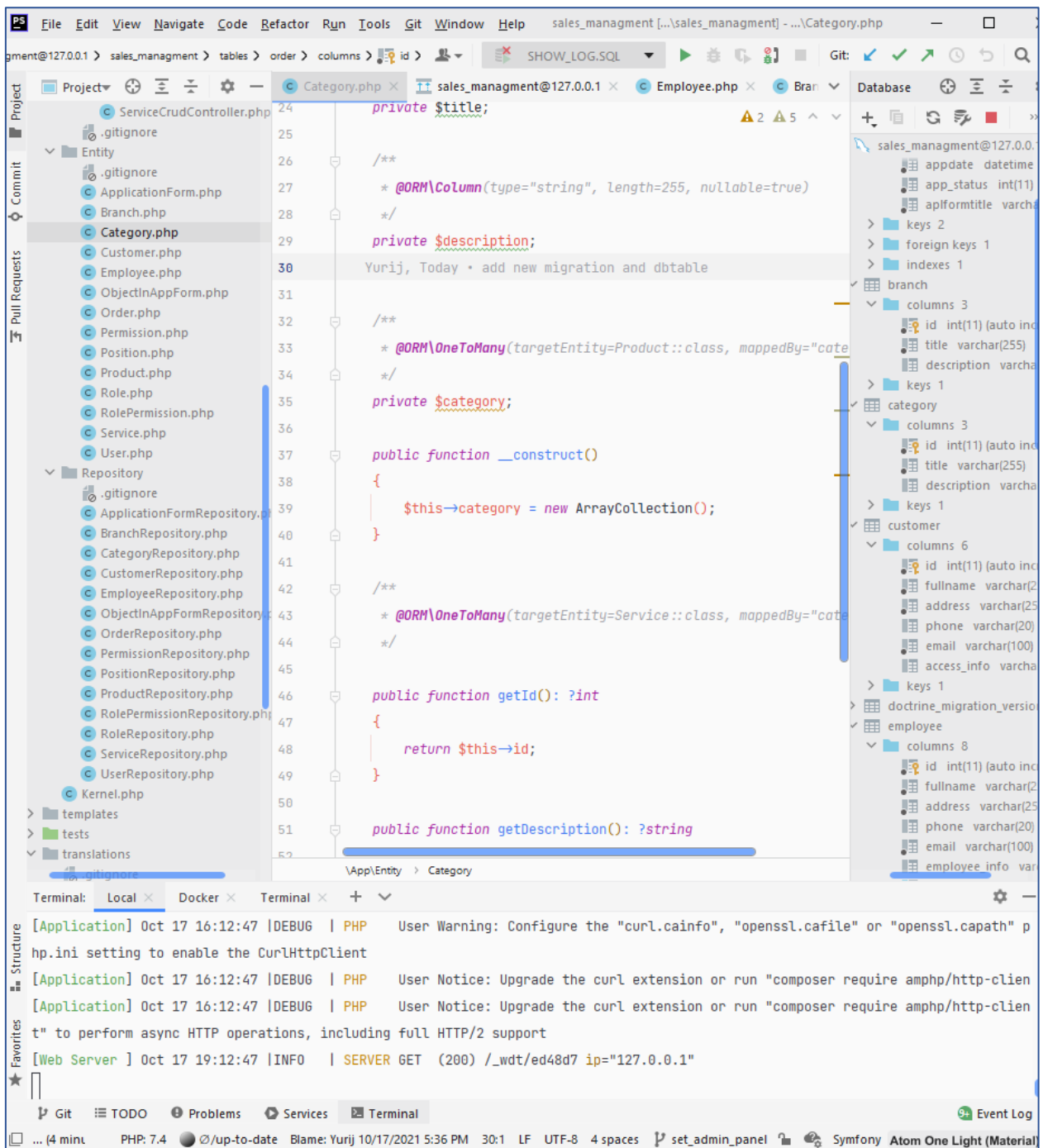


Рисунок 19 – Работа с модулем «Категории»

Модуль «Товары» обеспечивает возможность работы с таблицей базы данных «Товары»: добавлять, редактировать и удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 20.

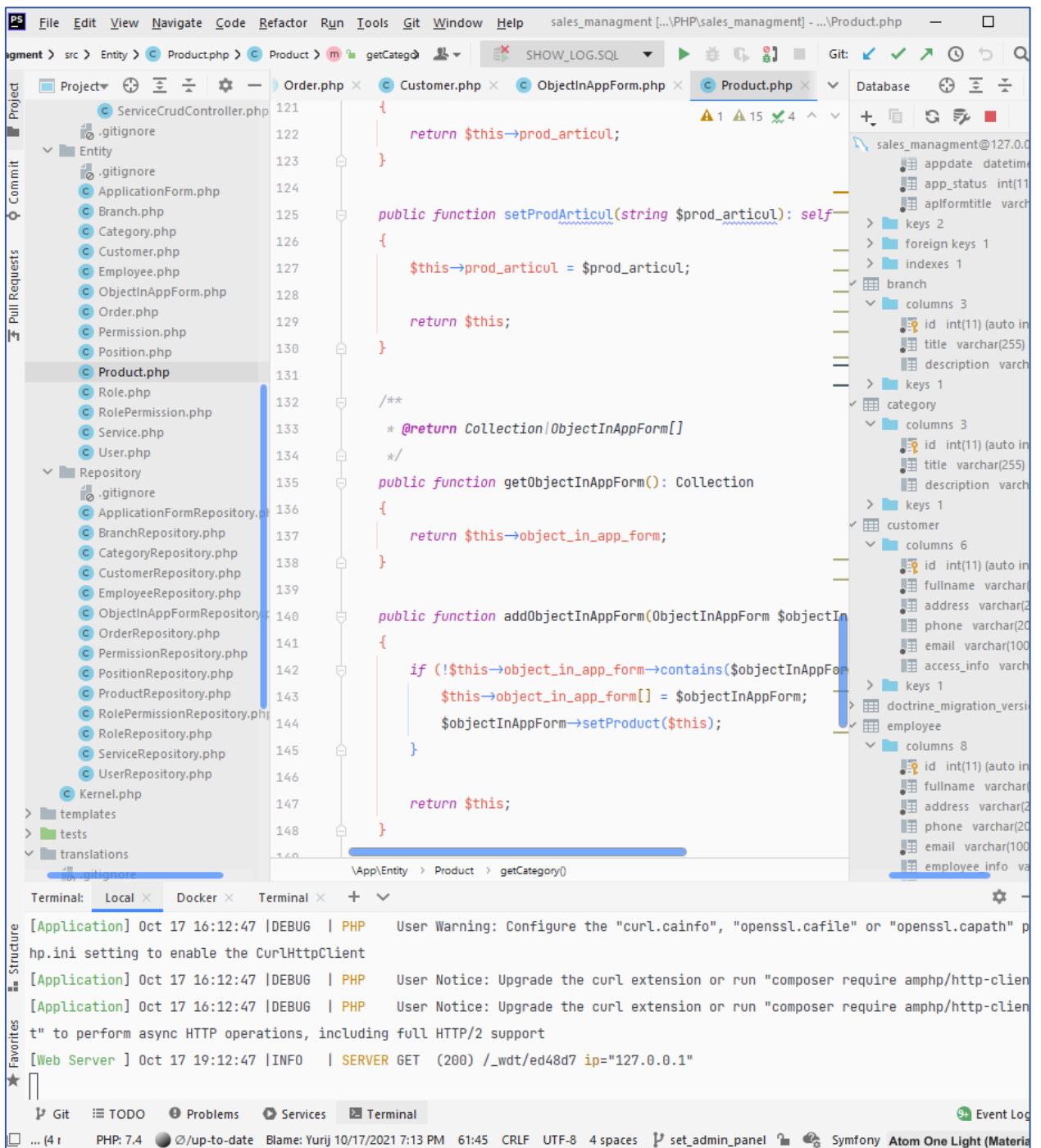


Рисунок 20 – Работа с модулем «Товары»

Модуль «Услуги» обеспечивает возможность работы с таблицей базы данных «Услуги»: добавлять, редактировать и удалять записи в таблице. Процесс работы с данным модулем показан на рисунке 21.

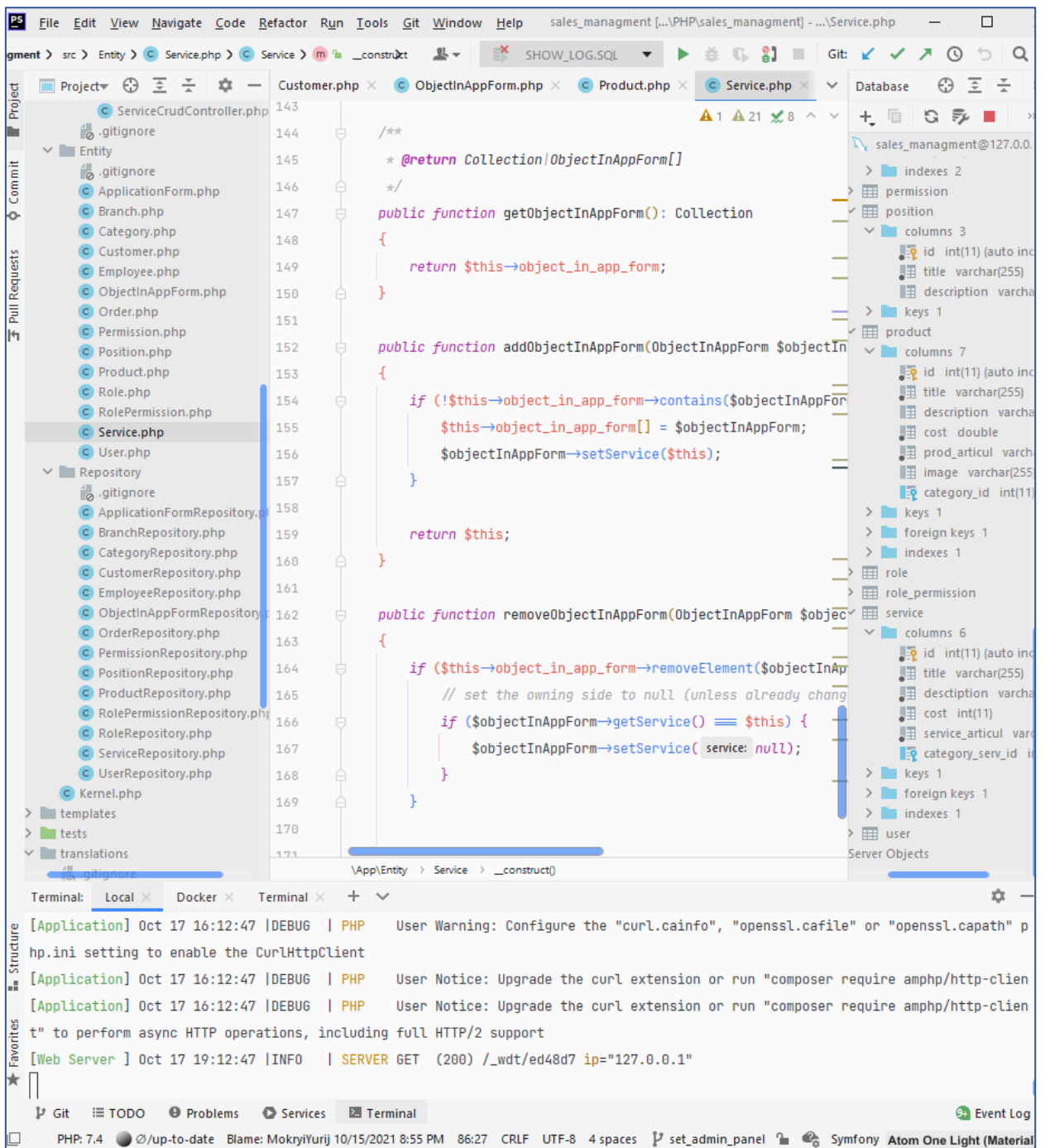


Рисунок 21 – Работа с модулем «Услуги»

Таким образом были созданы модули информационной системы. Исходный код модулей информационной системы приведен в Приложении А.

### 3.6 Описание функциональности АИС

Описание функциональности автоматизированной информационной системы управления продажами в ИП Павлюк И.С. представлено на рисунках 22-33.

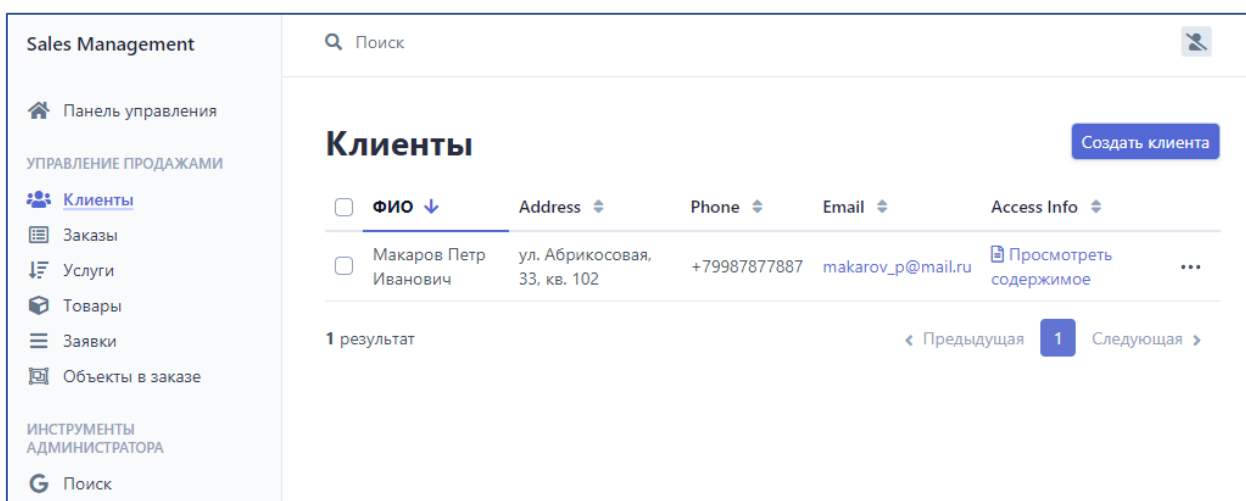


Рисунок 22 – Функциональный модуль «Клиенты». Просмотр списка клиентов

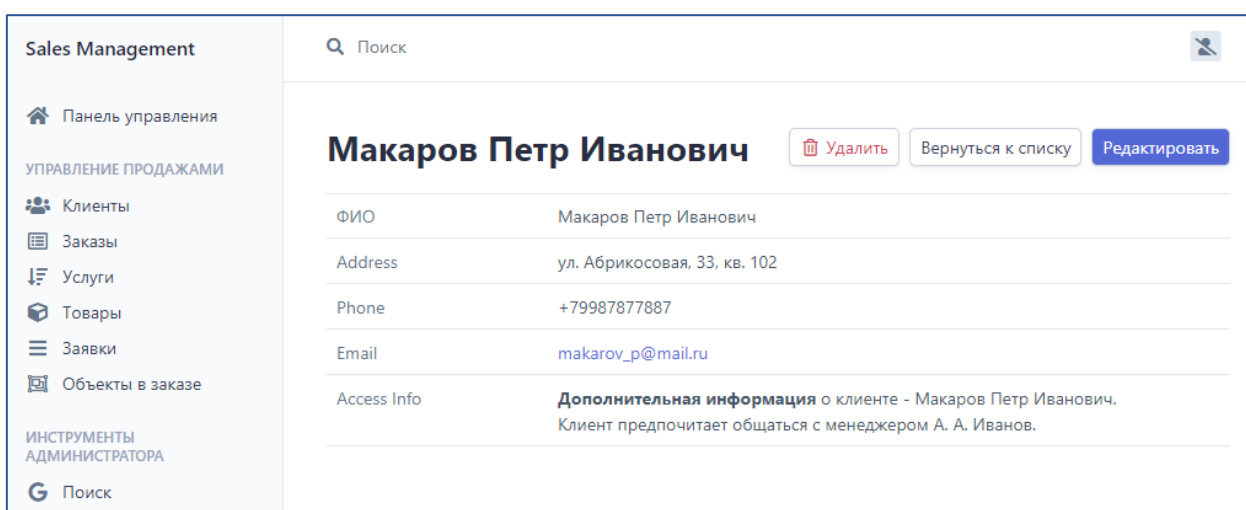


Рисунок 23 – Функциональный модуль «Клиенты». Просмотр информации о выбранном клиенте

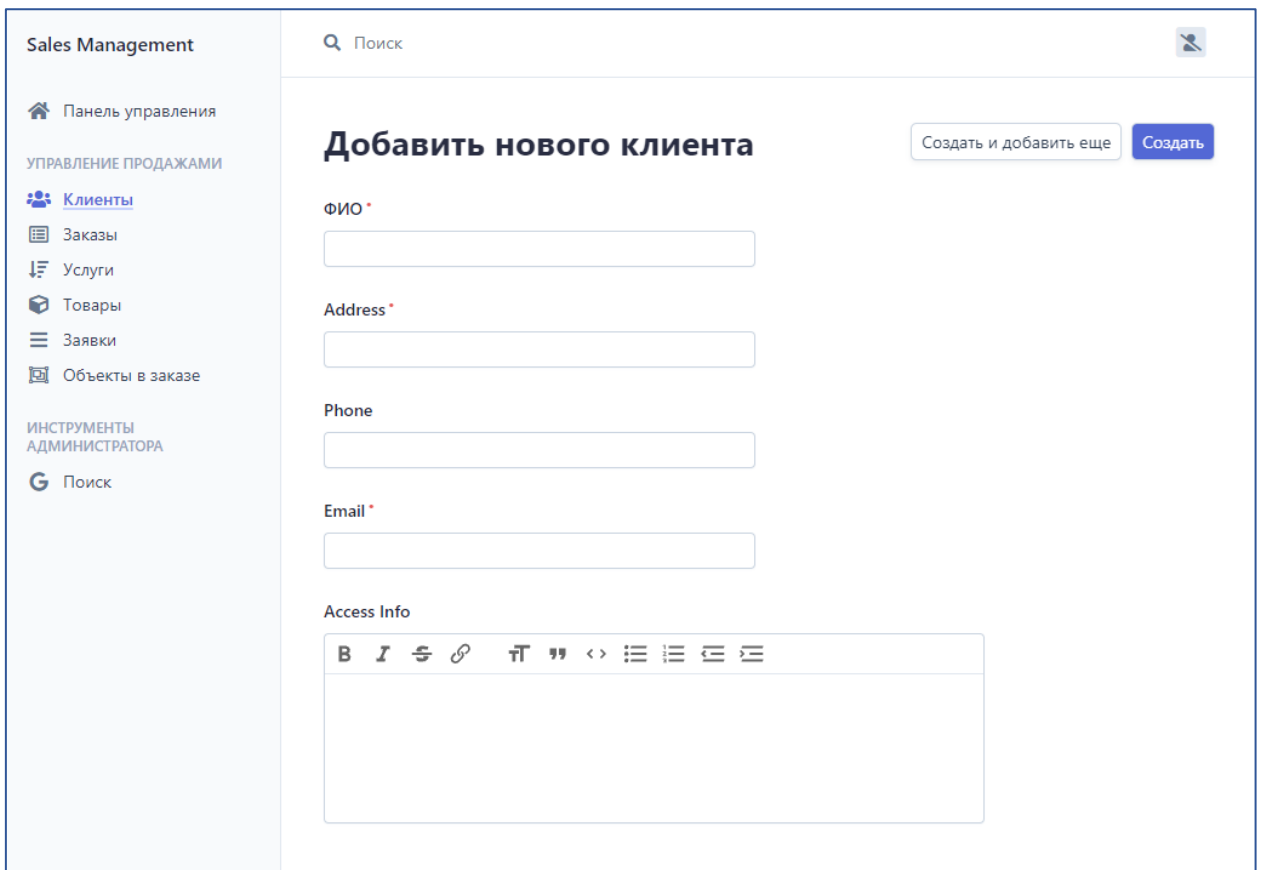


Рисунок 24 – Функциональный модуль «Клиенты». Добавление нового клиента

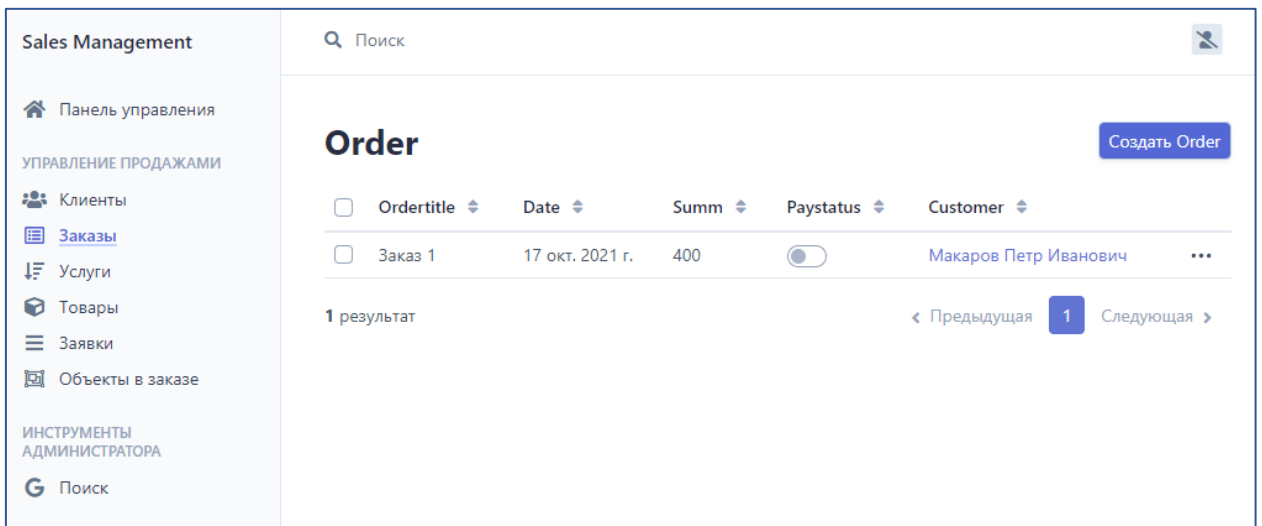


Рисунок 25 – Функциональный модуль «Заказы». Просмотр списка заказов и элементы управления заказами



The screenshot displays the 'Sales Management' dashboard with a sidebar on the left containing navigation options like 'Панель управления', 'УПРАВЛЕНИЕ ПРОДАЖАМИ', 'Клиенты', 'Заказы', 'Услуги', 'Товары', 'Заявки', and 'Объекты в заказе'. The main content area is titled 'Создать новый Order' and includes a search bar at the top right. The form contains the following fields: 'Ordertitle' (text input), 'Date' (calendar picker), 'Summ' (text input), a 'Paystatus' toggle switch, and a 'Customer' dropdown menu. Two buttons are located at the top right: 'Создать и добавить еще' and 'Создать'.

Рисунок 26 – Функциональный модуль «Заказы». Добавление нового заказа

The screenshot displays the 'Sales Management' dashboard with a sidebar on the left containing navigation options like 'Панель управления', 'УПРАВЛЕНИЕ ПРОДАЖАМИ', 'Клиенты', 'Заказы', 'Услуги', 'Товары', 'Заявки', and 'Объекты в заказе'. The main content area is titled 'Создать новый Service' and includes a search bar at the top right. The form contains the following fields: 'Title' (text input), 'Description' (text input), 'Cost' (text input), and 'Service Articul' (text input). Two buttons are located at the top right: 'Создать и добавить еще' and 'Создать'.

Рисунок 27 – Функциональный модуль «Услуги». Добавление новой услуги

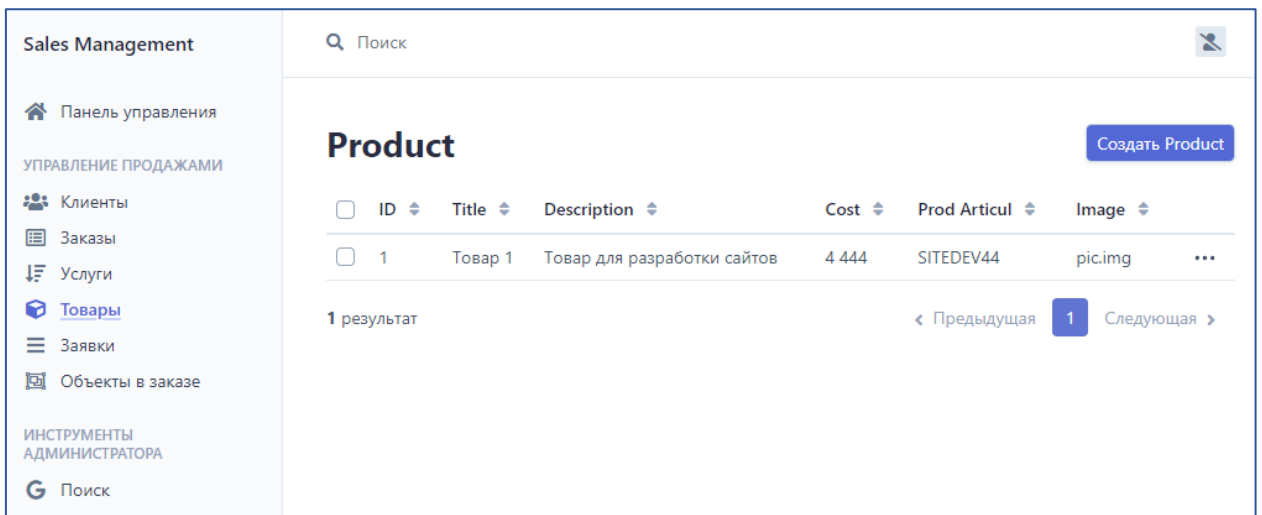


Рисунок 28 – Функциональный модуль «Товары». Просмотр списка и управление товарами

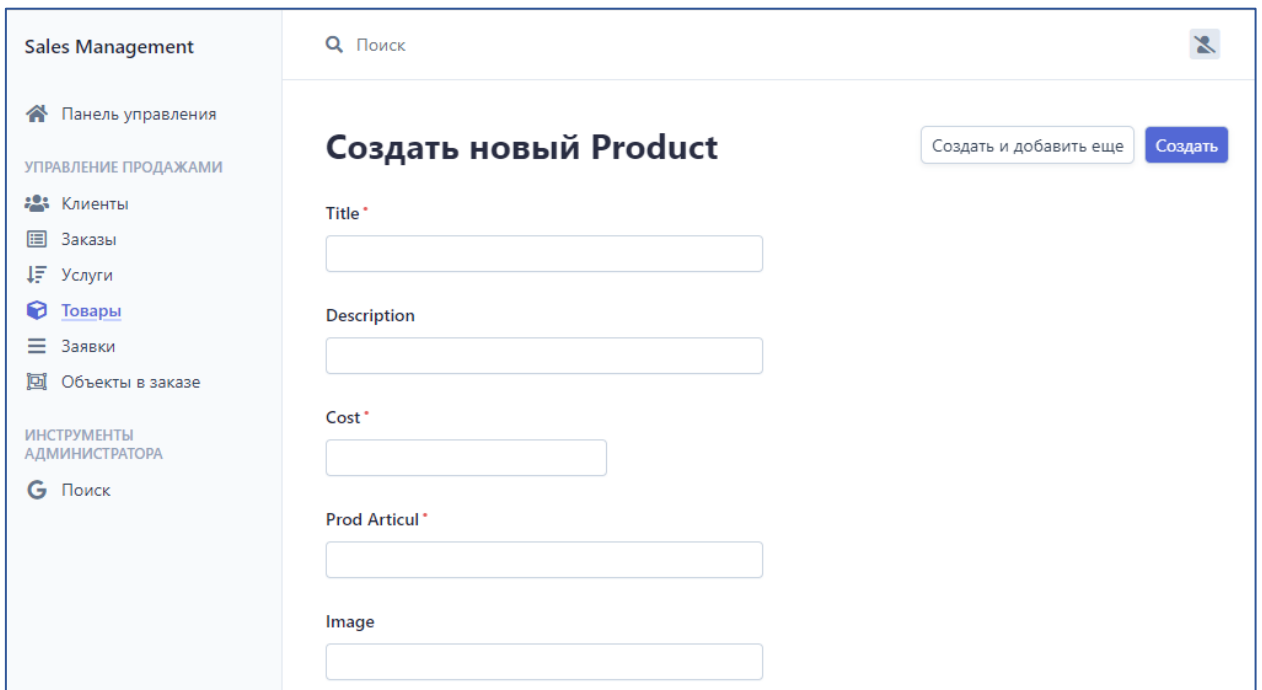


Рисунок 29 – Функциональный модуль «Товары». Создание нового товара

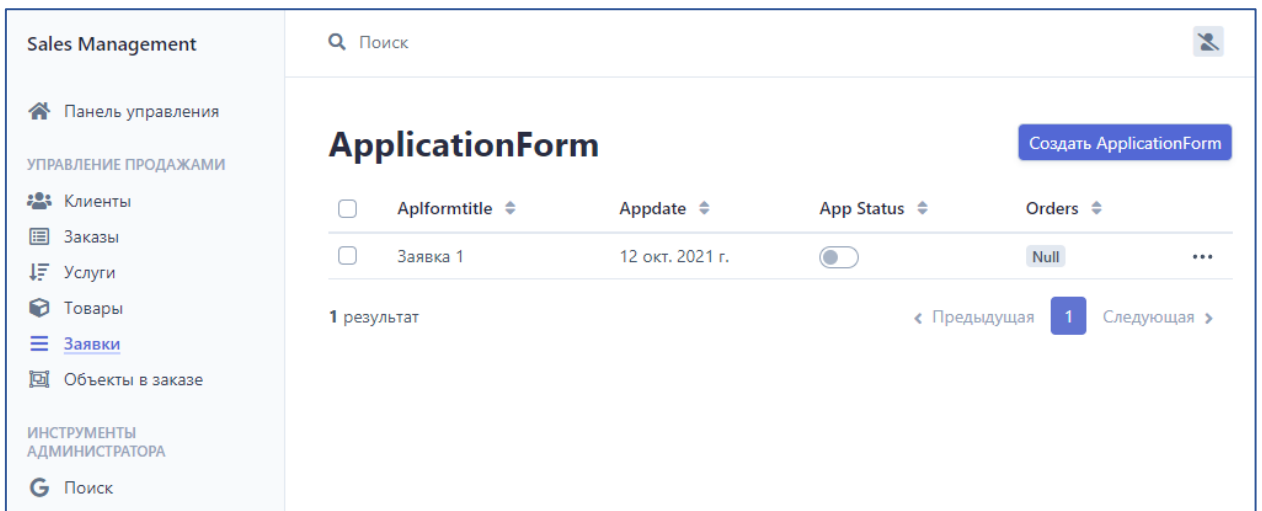


Рисунок 30 – Функциональный модуль «Заявки». Просмотр списка заявок и элементы управления для работы с заявками

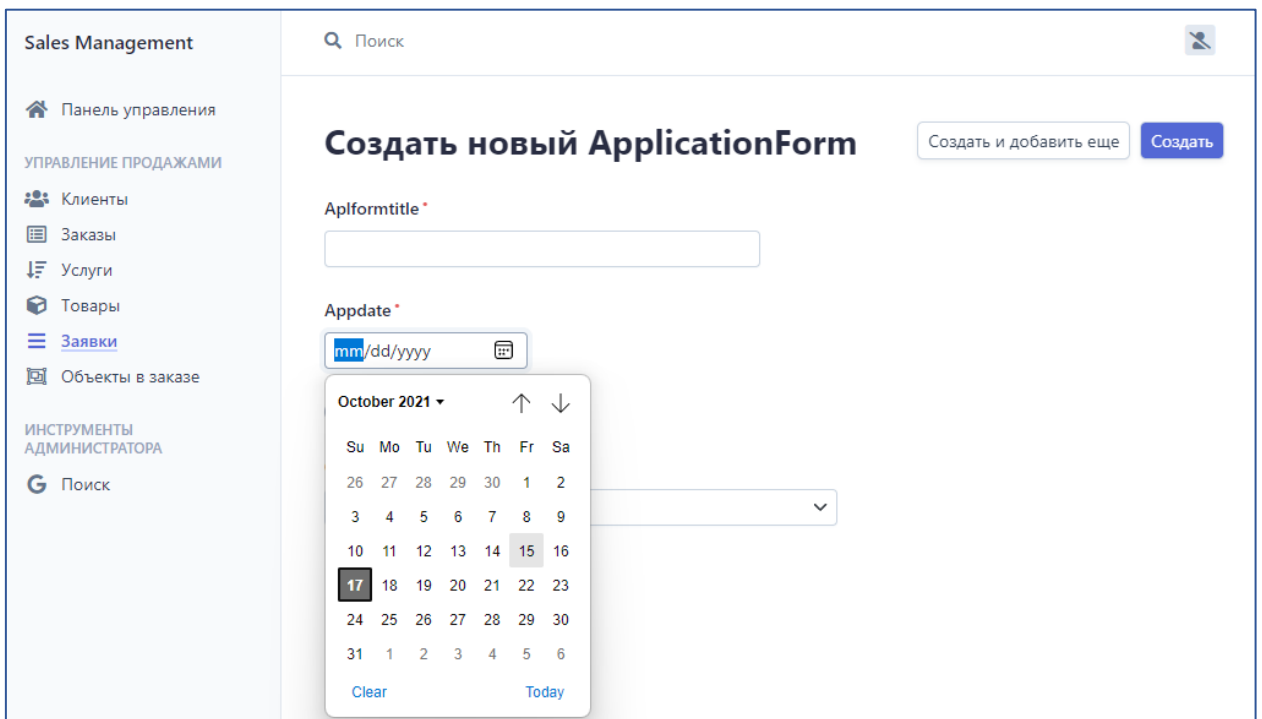


Рисунок 31 - Функциональный модуль «Заявки». Добавление новой заявки в систему

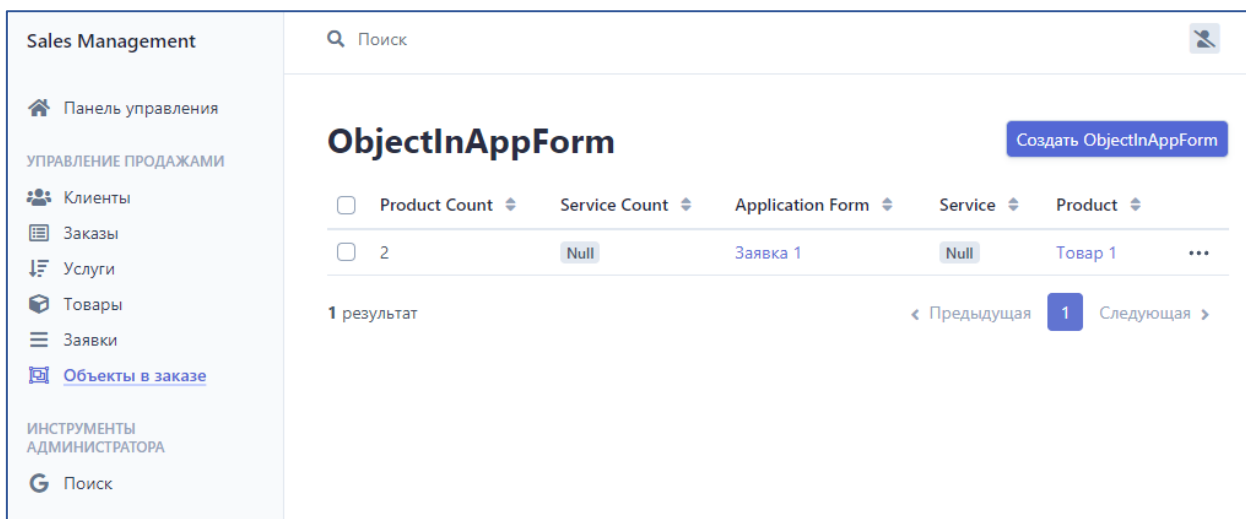


Рисунок 32 – Функциональный модуль «Товары и услуги в заявке». Просмотр содержимого заявок

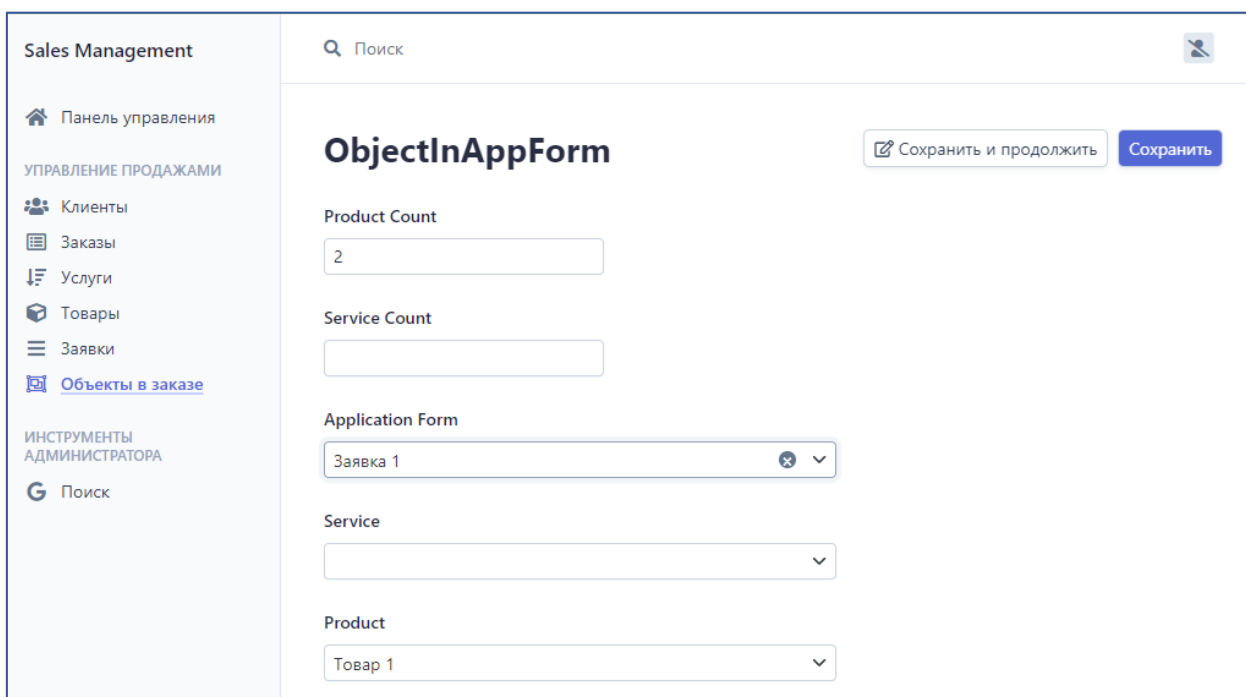


Рисунок 33 – Функциональный модуль «Товары и услуги в заявке». Добавление новой записи

Таким образом были рассмотрены функциональные возможности разработанной информационной системы для автоматизации процесса продаж в ИП Павлюк И.С.

### 3.7 Тестирование программного проекта

В процессе тестирования ПО должны быть выполнены следующие тесты:

- функциональный;
- конфигурация;
- удобство использования;
- проверка соответствия стандартам сборки.

Проверка функциональности заключается в проверке выполнения программой функций, указанных в спецификации, а также степени точности протестированных функций.

Тесты конфигурации проверяют, может ли созданное программное обеспечение работать на разных компьютерах и с разным установленным программным обеспечением.

Тест на удобство использования определяет практичность, комфорт при использовании приложения.

Тест сборки проводится специально для подтверждения того, что приложение выполняет все основные функции после его запуска. Тесты проводятся сразу же после создания программы. При отсутствии прохождения этих тестов нет смысла продолжать процесс тестирования, так как эти тесты показывают, что программное обеспечение не работает. Например, тест не проходит из-за ошибки подключения к БД.

Тестовый пример – это набор шагов, конкретных условий и параметров, необходимых для тестирования разработанной системы. Ниже в таблице 7 перечислены сценарии прохождения тестирования.

Тестирование программного обеспечения представляет собой процесс выявления наличия дефектов в программных системах.

Для тестирования данной системы было выбрано функциональное тестирование. В результате тестирования сформирована таблица тестовых случаев (таблица 7).

Таблица 7 – Тестирование сайта

№ тестового случая	Название теста	Ожидаемый результат	Полученный результат	Результат тестирования
1	Переход в модуль «Отделы»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
2	Переход в модуль «Должности»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
3	Переход в модуль «Заявки»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
4	Переход в модуль «Клиенты»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
5	Переход в модуль «Товары и услуги в заявках»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
6	Переход в модуль «Заказы»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
7	Переход в модуль «Категории»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
8	Переход в модуль «Товары»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно
10	Переход в модуль «Услуги»	Отображение списка записей и элементов управления модуля	Записи модуля и элементы управления загружены	Тест пройден успешно

В ходе тестирования информационной системы проведена проверка работоспособности АИС, при которой определена работоспособность всех функций.

## Заключение

В результате выполнения работы достигнута поставленная цель – разработана автоматизированная информационная система управления продажами для ИП Павлюк И.С. При достижении цели решены следующие задачи:

- проведена технико-экономическая характеристика предприятия;
- разработаны функциональные модели деятельности отдела продаж AS-IS и TO-BE;
- построена логическая и физическая модели данных исследуемой предметной области;
- спроектированы и реализованы модули автоматизированной информационной системы управления продажами в ИП Павлюк И.С.

В первой главе исследована организационная структура ИП Павлюк И.С., изучены бизнес-процессы, определены особенности предметной области, определена задача на разработку информационной системы, разработана новая модель бизнес-процесса.

Во второй главе приведено логическое проектирование информационной системы, построена логическая модель приложения и его базы данных, определены требования к программно-аппаратным средствам.

В результате физического проектирования автоматизированной информационной системы был проведен выбор архитектуры приложения, определены технологии разработки программного обеспечения, выбрана система управления базой данных, разработана логическая и физическая модели базы данных, проведена разработка программного обеспечения, определены взаимосвязи модулей и показан функционал созданной системы. В ходе тестирования информационной системы проведена проверка работоспособности информационной системы, при которой определена работоспособность всех функций.

## Список используемой литературы

1. Асланов, Тимур Арифметика продаж. Руководство по управлению продавцами / Тимур Асланов. - М.: Манн, Иванов и Фербер, 2016. - 160 с.
2. Буренин С. Н. Web-программирование и базы данных [Электронный ресурс] : учеб. практикум / С. Н. Буренин. - Москва : Моск. гуманит. ун-т, 2014. - 120 с.
3. Вдовин В. М. Предметно-ориентированные экономические информационные системы [Электронный ресурс] : учебное пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. - 3-е изд. - Москва : Дашков и К°, 2013. - 388 с.
4. Вертоградов В. Управление продажами. – 2-е изд. – СПб. : Питер, 2011. – 236 с.
5. Громов, А.И. Управление бизнес-процессами: современные методы. монография / А.И. Громов, А. Фляйшман, В. Шмидт. - Люберцы: Юрайт, 2016. - 367 с.
6. Джоббер Д., Ланкастер Дж. Продажи и управление продажами : учеб. пособие для вузов. – М. : Юнити-Дана, 2002. – 622 с.
7. Долганова, О.И. Моделирование бизнес-процессов: Учебник и практикум для академического бакалавриата / О.И. Долганова, Е.В. Виноградова, А.М. Лобанова. - Люберцы: Юрайт, 2016. - 289 с.
8. Дронов, В. А. PHP, MySQL, HTML5 и CSS3. Разработка современных динамических Web-сайтов / В. А. Дронов. – СПб.: ВHV, 2016. – 688 с.
9. Золотов С. Ю. Проектирование информационных систем [Электронный ресурс] : учеб. пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск: Эль Учебное пособие Контент, 2013. - 86 с.



10. Карпова И. П. Базы данных: курс лекций и материалы для практ. занятий : учеб. пособие для студентов техн. фак. / И. П. Карпова. - Санкт-Петербург: Питер, 2013. - 240 с. : ил. - (Учебное пособие). - Библиогр.: с. 233–234. - Прил.: с. 211-232. - Алф. указ.: с. 235-240.
11. Кондрашов, В. М. Менеджмент продаж / В.М. Кондрашов. - М.: Вузовский учебник, 2013. - 288 с.
12. Людоговский, А. Моделирование бизнес-процессов [Электронный ресурс] / А. Людоговский // Инф. портал «Script coding». - Режим доступа: <http://www.script-coding.com/bp.htm> 1 (дата обращения: 10.08.2021).
13. ООО ИП Павлюк И. С. [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. – Режим доступа: <http://pip.ru/> (дата обращения 10.09.2021).
14. Система Salesforce [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. – Режим доступа: <https://www.salesforce.com/eu/?ir=1> (дата обращения 10.08.2021).
15. Спиро Р. Л., Стэнтон У. Дж. Управление продажами. – М. : Издат. дом Гребенникова, 2004. – 704 с.
16. Фатрелл Ч. Управление продажами. –СПб. : Издат. дом «Нева», 2004. – 640 с.
17. AmoCRM [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. URL: <https://www.amocrm.ru/> (дата обращения 0708.2021).
18. Bringing MySQL to the web [Электронный ресурс] : phpMyAdmin. URL: <https://www.phpmyadmin.net/> (дата обращения: 15.10.2021).
19. BPM-online [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. URL: <https://habr.com/ru/company/trinion/blog/281192/> (дата обращения 10.09.2021).

20. Insightly CRM [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. URL: <https://www.insightly.com/> (дата обращения 10.08.2021).
21. InvGate [Электронный ресурс] / А. Е. Зашеловский, О. Ф. Абрамова. URL: <https://invgate.com/> (дата обращения 09.08.2021).

Приложение А  
**Программный код**

```
<?php
namespace App\Controller\Admin;
use App\Entity\ApplicationForm;
use App\Entity\ObjectInAppForm;
use App\Entity\Product;
use App\Entity\Service;
use EasyCorp\Bundle\EasyAdminBundle\Config\Dashboard;
use EasyCorp\Bundle\EasyAdminBundle\Config\MenuItem;
use
EasyCorp\Bundle\EasyAdminBundle\Controller\AbstractDashboardControll
er;
use EasyCorp\Bundle\EasyAdminBundle Router\AdminUrlGenerator;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;
use App\Entity\Customer;
use App\Entity\Order;
use Symfony\Contracts\Translation\TranslatorInterface;
class DashboardController extends AbstractDashboardController
{
    private TranslatorInterface $translator;
    public function __construct(TranslatorInterface $translator)
    {
        $this->translator = $translator;
    }
    public function index(): Response
    {
```

## Продолжение Приложения А

```
//     return parent::index();
    $routerBuilder = $this->get(AdminUrlGenerator::class);
    $url = $routerBuilder->setController(CustomerCrudController::class)-
>generateUrl();
    return $this->redirect($url);
}
public function configureDashboard(): Dashboard
{
    return Dashboard::new()
        ->setTitle('Sales Management')
        ->setFaviconPath('favicon.svg')
        ->generateRelativeUrls()
        ->disableUrlSignatures()
        ->setTranslationDomain('admin');
//     ->renderSidebarMinimized();
}
public function configureMenuItems(): iterable
{
    yield MenuItem::linktoDashboard($this->translator-
>trans('Dashboard'), 'fa fa-home');
    yield MenuItem::section($this->translator->trans('Sales management'));
    yield MenuItem::linkToCrud($this->translator->trans('Customers'), 'fas
fa-users', Customer::class);
    yield MenuItem::linkToCrud($this->translator->trans('Orders'), 'fa fa-
list-alt', Order::class);
    yield MenuItem::linkToCrud($this->translator->trans('Services'), 'fa fa-
sort-amount-asc', Service::class);
```

## Продолжение Приложения А

```
yield MenuItem::linkToCrud($this->translator->trans('Products'), 'fa fa-cube', Product::class);
yield MenuItem::linkToCrud($this->translator->trans('Applications'), 'fa fa-bars', ApplicationForm::class);
yield MenuItem::linkToCrud($this->translator->trans('Object in applications'), 'fa fa-object-group', ObjectInAppForm::class);
yield MenuItem::section($this->translator->trans('Administration tools'));
// yield MenuItem::linkToLogout('Logout', 'fa fa-exit');
yield MenuItem::linkToUrl($this->translator->trans('Search in Google'), 'fab fa-google', 'https://google.com');
// yield MenuItem::linkToCrud('The Label', 'fas fa-list', EntityClass::class);
}
}
```

<?php

```
namespace App\Controller\Admin;
use App\Entity\Customer;
use
EasyCorp\Bundle\EasyAdminBundle\Controller\AbstractCrudController;
use EasyCorp\Bundle\EasyAdminBundle\Config\Crud;
use EasyCorp\Bundle\EasyAdminBundle\Field\EmailField;
use EasyCorp\Bundle\EasyAdminBundle\Field\TextareaField;
use EasyCorp\Bundle\EasyAdminBundle\Field\TextEditorField;
use EasyCorp\Bundle\EasyAdminBundle\Field\TextField;
use Symfony\Contracts\Translation\TranslatorInterface;
```

## Продолжение Приложения А

```
class CustomerCrudController extends AbstractCrudController
{
    private $translator;
    public function __construct(TranslatorInterface $translator)
    {
        $this->translator = $translator;
    }
    public static function getEntityFqcn(): string
    {
        return Customer::class;
    }
    final
    public function configureCrud(Crud $crud): Crud
    {
        return $crud
            ->setEntityLabelInSingular($this->translator->trans('Customer'))
            ->setEntityLabelInPlural($this->translator->trans('Customers'))
            ->setSearchFields(['fullname, phone'])
            ->setDefaultSort(['fullname' => 'DESC']);
    }
    final
    public function configureFields(string $pageName): iterable
    {
        // return [
        //     IdField::new('id'), TextField::new('title'),
        //     TextEditorField::new('description'),
        // ];
        yield TextField::new('fullname', $this->translator->trans('FullName'));
```

## Продолжение Приложения А

```
yield TextField::new('address');
yield TextField::new('phone');
yield EmailField::new('email');
yield      TextEditorField::new('access_info')->formatValue(function
($value) {
    return $value;
});
}
}
```

```
<?php
namespace App\Controller\Admin;
use App\Entity\ObjectInAppForm;
use
EasyCorp\Bundle\EasyAdminBundle\Controller\AbstractCrudController;
use EasyCorp\Bundle\EasyAdminBundle\Field\AssociationField;
use EasyCorp\Bundle\EasyAdminBundle\Field\IdField;
use EasyCorp\Bundle\EasyAdminBundle\Field\NumberField;
use EasyCorp\Bundle\EasyAdminBundle\Field\TextEditorField;
use EasyCorp\Bundle\EasyAdminBundle\Field\TextField;
class ObjectInAppFormCrudController extends AbstractCrudController
{
    public static function getEntityFqcn(): string
    {
        return ObjectInAppForm::class;
    }
    public function configureFields(string $pageName): iterable
```

## Продолжение Приложения А

```
{
    return [
//      IdField::new('id'),
        NumberField::new('product_count'),
        NumberField::new('service_count'),
        AssociationField::new('applicationForm'),
        AssociationField::new('service'),
        AssociationField::new('product')
    ];
}
}
```

```
<?php
namespace App\Controller\Admin;
use App\Entity\Order;
use
EasyCorp\Bundle\EasyAdminBundle\Controller\AbstractCrudController;
use EasyCorp\Bundle\EasyAdminBundle\Field\AssociationField;
use EasyCorp\Bundle\EasyAdminBundle\Field\BooleanField;
use EasyCorp\Bundle\EasyAdminBundle\Field\DateField;
use EasyCorp\Bundle\EasyAdminBundle\Field\NumberField;
use EasyCorp\Bundle\EasyAdminBundle\Field\TextField;
class OrderCrudController extends AbstractCrudController
{
    public static function getEntityFqcn(): string
    {
        return Order::class;
    }
}
```



## Продолжение Приложения А

```
public function configureFields(string $pageName): iterable
{
    return [
        TextField::new('ordertitle'),
        DateField::new('date'),
        NumberField::new('summ'),
        BooleanField::new('paystatus'),
        AssociationField::new('customer')
    ];
}
}
```

```
<?php
namespace App\Entity;
use App\Repository\ApplicationFormRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\ORM\Mapping as ORM;
class ApplicationForm
{
    /**
     * @ORM\Id
     * @ORM\GeneratedValue
     * @ORM\Column(type="integer")
     */
    private $id;
```

## Продолжение Приложения А

```
private $aplformtitle;
public function __toString(){
    return $this->aplformtitle;
}
public function getAplformtitle(): ?string
{
    return $this->aplformtitle;
}
public function setAplformtitle(string $aplformtitle): self
{
    $this->aplformtitle = $aplformtitle;
    return $this;
}
/**
 * @ORM\Column(type="datetime")
 */
private $appdate;
/**
 * @ORM\Column(type="integer")
 */
private $app_status;
/**
 * @ORM\OneToOne(targetEntity=Order::class,
inversedBy="application_form_id", cascade={"persist", "remove"})
 * @ORM\JoinColumn(nullable=true)
 */
private $orders;
```

## Продолжение Приложения А

```
/**
 *      @ORM\OneToMany(targetEntity=ObjectInAppForm::class,
mappedBy="applicationForm", orphanRemoval=true)
 */
private $object_in_app_form;
public function __construct()
{
    $this->object_in_app_form = new ArrayCollection();
}
public function getId(): ?int
{
    return $this->id;
}
public function getAppdate(): ?\DateTimeInterface
{
    return $this->appdate;
}
public function setAppdate(\DateTimeInterface $appdate): self
{
    $this->appdate = $appdate;
    return $this;
}
public function getAppStatus(): ?int
{
    return $this->app_status;
}
public function setAppStatus(int $app_status): self
```

## Продолжение Приложения А

```
{
    $this->app_status = $app_status;
    return $this;
}
public function getOrders(): ?Order
{
    return $this->orders;
}
public function setOrders(Order $orders): self
{
    $this->orders = $orders;
    return $this;
}
/**
 * @return Collection|ObjectInAppForm[]
 */
public function getObjectInAppForm(): Collection
{
    return $this->object_in_app_form;
}
public function addObjectInAppForm(ObjectInAppForm
$objectInAppForm): self
{
    if (!$this->object_in_app_form->contains($objectInAppForm)) {
        $this->object_in_app_form[] = $objectInAppForm;
        $objectInAppForm->setApplicationForm($this);
    }
}
```

## Продолжение Приложения А

```
        return $this;
    }
    public function removeObjectInAppForm(ObjectInAppForm
    $objectInAppForm): self
    {
        if ($this->object_in_app_form->removeElement($objectInAppForm)) {
            // set the owning side to null (unless already changed)
            if ($objectInAppForm->getApplicationForm() === $this) {
                $objectInAppForm->setApplicationForm(null);
            }
        }
        return $this;
    }
}
```

```
<?php
namespace App\Entity;
use App\Repository\OrderRepository;
use Doctrine\ORM\Mapping as ORM;
/**
 * @ORM\Entity(repositoryClass=OrderRepository::class)
 * @ORM\Table(name="`order`")
 */
class Order
{
    /**
     * @ORM\Id
     * @ORM\GeneratedValue
```

## Продолжение Приложения А

```
* @ORM\Column(type="integer")
*/
private $id;
public function __toString(){
    return $this->ordertitle;
}
/**
 * @ORM\Column(type="string", length=255)
 */
private $ordertitle;
public function getOrdertitle(): ?string
{
    return $this->ordertitle;
}
public function setOrdertitle(string $ordertitle): self
{
    $this->ordertitle = $ordertitle;
    return $this;
}
/**
 * @ORM\Column(type="datetime")
 */
private $date;
/**
 * @ORM\Column(type="float")
 */
private $summ;
```

## Продолжение Приложения А

```
/**
 * @ORM\Column(type="boolean")
 */
private $paystatus;
/**
 * @ORM\ManyToOne(targetEntity=Customer::class,
inversedBy="orders")
 * @ORM\JoinColumn(nullable=false)
 */
private $customer;
/**
 * @ORM\OneToOne(targetEntity=ApplicationForm::class,
mappedBy="orders", cascade={"persist", "remove"})
 */
private $application_form_id;
/**
 * @ORM\ManyToOne(targetEntity=User::class,
inversedBy="orderitem")
 */
private $user;
public function getId(): ?int
{
    return $this->id;
}
public function getDate(): ?\DateTimeInterface
{
    return $this->date;
}
```

## Продолжение Приложения А

```
public function setDate(\DateTimeInterface $date): self
{
    $this->date = $date;
    return $this;
}
public function getSumm(): ?float
{
    return $this->summ;
}
public function setSumm(float $summ): self
{
    $this->summ = $summ;
    return $this;
}
public function getPaystatus(): ?bool
{
    return $this->paystatus;
}
public function setPaystatus(bool $paystatus): self
{
    $this->paystatus = $paystatus;
    return $this;
}
public function getCustomer(): ?Customer
{
    return $this->customer;
}
```



## Продолжение Приложения А

```
public function setCustomer(?Customer $customer): self
{
    $this->customer = $customer;
    return $this;
}
public function getApplicationFormId(): ?ApplicationForm
{
    return $this->application_form_id;
}
public function setApplicationFormId(ApplicationForm
$application_form_id): self
{
    // set the owning side of the relation if necessary
    if ($application_form_id->getOrders() !== $this) {
        $application_form_id->setOrders($this);
    }
    $this->application_form_id = $application_form_id;
    return $this;
}
public function getUser(): ?User
{
    return $this->user;
}
public function setUser(?User $user): self
{
    $this->user = $user;
    return $this;
}
}
```