МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра информационных систем управления

СИСТЕМА УЧЁТА И КОНТРОЛЯ IT-АКТИВОВ ПРЕДПРИЯТИЯ

Курсовая работа

Демина Владимира Олеговича обучающегося 4 курса специальности

«Прикладная информатика»

Научный руководитель: старший преподаватель

Е.С. Малашенко

Минск, 2024

**СОДЕРЖАНИЕ**

[**РЕФЕРАТ** 3](#_vtpg4ektigey)

[**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ**](#_v0xc05vu01k8)............................................................................................................6

**[ВВЕДЕНИЕ](#_vtpg4ektigey)**7

[**ГЛАВА 1**](#_dekhzrit00ll) **СОСТОЯНИЕ ПРОБЛЕМЫ, ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ**………………………………………………………………8

[1.1 Понятие и основные типы IT-активов](#_yqm3y6bx9g4u) 8

[1.2 Подходы к управлению IT-активами предприятия](#_yqm3y6bx9g4u) 10

[1.3 Обзор и анализ современных ITAM](#_yqm3y6bx9g4u) 12

[1.4 Цель и задачи исследования](#_42qzsrik74or) 14

[1.5 Выводы](#_3y2ify2epyhg) 15

[**ГЛАВА 2**](#_i23bp1j49m8m) **РАЗРАБОТКА И АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ** **ДЛЯ УЧЕТА И КОНТРОЛЯ IT-АКТИВОВ** **ПРЕДПРИЯТИЯ**…........................16

[2.1 Анализ деятельности предприятия и проблем в сфере управления IT-активами.](#_7zkfey3pnekh) 16

[2.2 Моделирование бизнес-процессов учета и контроля IT-активов](#_whzirxrp95o7) 18

[2.3 Выводы](#_whzirxrp95o7) 27

[**ГЛАВА 3**](#_i23bp1j49m8m) **ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗА-ЦИЯ БАЗЫ ДАННЫХ ДЛЯ УЧЕТА И КОНТРОЛЯ IT-АКТИВОВ**…………..28

[3.1 Концептуальное и логическое проектирование БД «Учёт и контроль IT-активов»…………………………………………………………………………..](#_7zkfey3pnekh)28

[3.2 Программная реализация БД «Учёт и контроль IT-активов»](#_7zkfey3pnekh) 37

[3.3 Выводы](#_3y2ify2epyhg) 40

[**ГЛАВА 4**](#_5zcpp98e0bwn) [**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**………...](#_rlw8usncp0a)41

[4.1 Выбор технологий для программной реализации системы учета и контроля IT- активов](#_x5b6sp94g5mx) 41

[4.2 Разработка элементов веб-сайта системы учета и контроля IT-активов предприятия](#_3y2ify2epyhg) 43

[4.3 Выводы](#_3y2ify2epyhg) 52

[**ЗАКЛЮЧЕНИЕ**](#_spvt8mumntay) 54

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**](#_nt79vsz1d8p) 55

[**ПРИЛОЖЕНИЕ А**](#_spvt8mumntay) 56

[**ПРИЛОЖЕНИЕ Б**](#_spvt8mumntay) 59

[**ПРИЛОЖЕНИЕ В**](#_spvt8mumntay) 67

**РЕФЕРАТ**

Курсовая работа содержит: 54 страниц, 24 рисунка, 10 использованных литературных источников, 3 приложения.

**Ключевые слова** – IT-АКТИВЫ, УПРАВЛЕНИЕ АКТИВАМИ, СИСТЕМЫ ITAM, БАЗА ДАННЫХ, ВЕБ-РАЗРАБОТКА, SQL, C#, JAVASCRIPT, XML, SQL, HTML, CSS, ASP.NET, DATATABLE.NET, GOOGLE CHART.

**Объект исследования –** процесс управления IT-активами предприятия.

**Предмет исследования –** методы и инструменты учета и контроля IT-активами предприятия.

**Цель работы –** разработка системы учета и контроля IT-активов предприятия и её программного обеспечения, использующиеся для автоматизации процессов управления IT-активами, обеспечения их прозрачного учета, повышения эффективности бизнес-процессов и минимизации ошибок, связанных с человеческим фактором, путем создания интуитивно понятного интерфейса, надежной базы данных и функциональных модулей.

**Методы исследования –** Анализ литературных источников по теме IT-активов и систем управления активами, сравнительный анализ существующих систем ITAM, методология проектирования баз данных, Agile-методологии в веб-разработке (SCRUM).

**Результат –** разработана система учета и контроля IT-активов предприятия, включающая интуитивно понятный пользовательский интерфейс, надежную базу данных и функциональные модули для управления активами, заказами, брендами, отчетами и корпоративной информацией.

**Область применения –** разработанная система может быть применена на предприятиях, занимающихся управлением IT-активами, для автоматизации учета, оптимизации бизнес-процессов, повышения прозрачности и эффективности управления ресурсами, а также обеспечения информационной безопасности и минимизации затрат.

**РЭФЕРАТ**

Курсавая праца змяшчае: 54 старонак, 24 малюнкаў, 10 выкарыстаных літаратурных крыніц, 3 дадаткі.

**Ключавыя словы** - IT-АКТЫВЫ, КІРАВАННЕ АКТЫВАМІ, СІСТЭМЫ ITAM, БАЗА ДАНЫХ, ВЭБ-РАЗРАБОТКА, SQL, C#, JAVASCRIPT, XML, SQL, HTML, CSS, ASP.NET, DATATABLE.NET, GOOGLE CHART.

**Аб'ект даследавання** - працэс кіравання IT-актывамі прадпрыемства.

**Прадмет даследавання** - метады і інструменты ўліку і кантролю IT-актывамі прадпрыемства.

**Мэта працы** - распрацоўка сістэмы ўліку і кантролю IT-актываў прадпрыемства і яе праграмнага забеспячэння, якія выкарыстоўваюцца для аўтаматызацыі працэсаў кіравання IT-актывамі, забеспячэння іх празрыстага ўліку, павышэння эфектыўнасці бізнес-працэсаў і мінімізацыі памылак, звязаных з чалавечым фактарам, шляхам стварэння інтуітыўна зразумелага інтэрфейсу , надзейнай базы дадзеных і функцыянальных модуляў.

**Метады даследавання** - Аналіз літаратурных крыніц па тэме IT-актываў і сістэм кіравання актывамі, параўнальны аналіз існуючых сістэм ITAM, метадалогія праектавання баз дадзеных, Agile-метадалогіі ў вэб-распрацоўцы (SCRUM).

**Вынік** - распрацавана сістэма ўліку і кантролю IT-актываў прадпрыемства, якая ўключае інтуітыўна зразумелы карыстацкі інтэрфейс, надзейную базу дадзеных і функцыянальныя модулі для кіравання актывамі, заказамі, брэндамі, справаздачамі і карпаратыўнай інфармацыяй.

**Вобласць прымянення** - распрацаваная сістэма можа быць прыменена на прадпрыемствах, якія займаюцца кіраваннем IT-актывамі, для аўтаматызацыі ўліку, аптымізацыі бізнес-працэсаў, павышэння празрыстасці і эфектыўнасці кіравання рэсурсамі, а таксама забеспячэння інфармацыйнай бяспекі і мінімізацыі затрат.

**ABSTRACT**

The coursework contains: 54 pages, 24 figures, 10 references, 3 appendices.

**Keywords** – IT ASSETS, ASSET MANAGEMENT, ITAM SYSTEMS, DATABASE, WEB DEVELOPMENT, SQL, C#, JAVASCRIPT, XML, SQL, HTML, CSS, ASP.NET, DATATABLE.NET, GOOGLE CHART.

**Object of the study** – the process of managing the IT assets of the enterprise.

**Subject of the study** – methods and tools for accounting and control of the IT assets of the enterprise.

**The purpose of the work** - to develop a system for accounting and control of the IT assets of the enterprise and its software used to automate the processes of managing IT assets, ensure their transparent accounting, improve the efficiency of business processes and minimize errors associated with the human factor by creating an intuitive interface, a reliable database and functional modules.

**Research methods** – Analysis of literary sources on the topic of IT assets and asset management systems, comparative analysis of existing ITAM systems, database design methodology, Agile methodologies in web development (SCRUM).

**Result** – a system for accounting and control of enterprise IT assets has been developed, including an intuitive user interface, a reliable database and functional modules for managing assets, orders, brands, reports and corporate information.

Scope – the developed system can be used in enterprises involved in IT asset management to automate accounting, optimize business processes, increase transparency and efficiency of resource management, as well as ensure information security and minimize costs.

# ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

IT – информационные технологии.

ITIL – технология разработки интерактивных карт OpenStreetMap.

ITAM – сервисы, которые анализируют логическую целостность данных OSM (топологию административных границ и т.п.).

ITSM – Integrated Development Environment (интегрированная среда разработки).

SQL – Structured Query Language (язык структурированных запросов). Данная аббревиатура часто используется при работе с базами данных.

IDEF0 – Integrated Definition for Function Modeling.

SSMS – SQL Server Management Studio.

SKU – Stock Keeping Unit.

CRUD – Create, Read, Update, Delete.

БД – база данных.

# ВВЕДЕНИЕ

В настоящее время эффективное управление IT-активами на предприятии становится ключевым фактором для достижения конкурентных преимуществ на рынке и обеспечения стабильной работы предприятия. Системы учета и контроля IT-активов (IT Asset Management, ITAM) играют также важную роль в обеспечении безопасности информационной инфраструктуры. Так, ITAM позволяют предприятию минимизировать риски, связанные с уязвимостями и утечками данных.

Таким образом, использование ITAM на предприятиях позволит предприятиям отслеживать свои ресурсы на протяжении всего их жизненного цикла начиная с приобретения и заканчивая утилизацией, что является актуальным в условиях растущей зависимости бизнеса от технологий и увеличения объема данных.

Целью данной курсовой работы является разработка системы учета и контроля IT-активов предприятия для повышения эффективности бизнес-процессов деятельности предприятия в сфере управления IT-активами и минимизации ошибок, связанных с человеческим фактором.

Для достижения поставленной цели необходимо выполнить ряд задач:

– исследовать современные подходы к управлению IT-активами предприятия и дать сравнительный анализ существующих ITAM;

– провести анализ бизнес-процессов деятельности предприятия в сфере управления IT-активами и выполнить функциональное моделирование бизнес-процессов учета и контроля IT-активов;

– разработать систему учета и контроля IT-активов предприятия для автоматизации процессов управления IT-активами, обеспечения их прозрачного учета, путем создания интуитивно понятного интерфейса, надежной базы данных и функциональных модулей.

В теоретической части курсовой работы проведен анализ проблемы управления IT-активами, существующих подходов к их учету и контролю, осуществлен обзор и анализ современных ITAM.

В практической части курсовой работы выполнено моделирование бизнес-процессов, связанных с учетом и контролем IT-активов предприятия, проектирование и программная реализация БД «Учёта и контроля IT-активов», осуществлена разработка программного обеспечения системы учета и контроля IT-активов предприятия.

# ГЛАВА 1

**СОСТОЯНИЕ ПРОБЛЕМЫ, ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ**

## 

## 1.1 Понятие и основные типы IT-активов

IT-актив — это любой IT-ресурс или совокупность возможностей, позволяющих организации эффективно осуществлять IT-деятельность и предоставляющих ценность для её основной деятельности.

В современном мире IT-активы являются фундаментальными элементами, которые поддерживают операционную работоспособность, способствуют стратегическому росту и стимулируют инновационное развитие организаций. Для успешного управления этими активами важно понимать их природу, виды и жизненный цикл [1].

Согласно ГОСТ Р ИСО/МЭК 19770-1—2021, IT-активы классифицируются следующим образом:

1. Цифровые активы. Эта категория охватывает нематериальные IT-ресурсы, существующие в цифровом виде. В свою очередь, Цифровые активы подразделяются на:

– Активы ПО, которые включают как исполняемое программное обеспечение (например, операционные системы, приложения), так и неисполняемые компоненты (например, исходный код, библиотеки). Это ключевой элемент IT -инфраструктуры, определяющий функциональность и производительность систем. Активы ПО также подразделяются на исполняемое ПО (которые включают обычно исходный код) и неисполняемое ПО (например, буквы, информация о конфигурации, словари и др., используемые исполняемым ПО). Туда же включают виртуальное оборудование (прошивку, виртуальная машина, встроенное ПО).

– Цифровой информационный актив: эта категория объединяет цифровые данные с информационным контентом. Сюда относятся документы, аудио- и видеофайлы, графические изображения, базы данных и автономные словари. Эти активы содержат ценную информацию, и их защита и управление ими имеют решающее значение

2. Система управления активами ITAM: эта категория включает инструменты и системы, используемые для управления ИТ-активами. Она делится на:

– ITAM-системы и инструменты: это программное обеспечение, предназначенное для инвентаризации, учета и мониторинга ИТ-активов. Они обеспечивают контроль над жизненным циклом активов.

– Метаданные для ITAM: включают данные о структуре, свойствах и взаимосвязях ИТ-активов, которые используются системами ITAM для эффективного управления.

3. ИТ-оборудование: эта категория включает физические компоненты ИТ-инфраструктуры:

– Физический носитель: это устройства хранения данных, такие как жесткие диски, твердотельные накопители, флеш-накопители и т. д., которые содержат цифровые активы.

– Физическое ИТ-оборудование: охватывает все остальные физические компоненты ИТ-инфраструктуры, включая серверы, рабочие станции, сетевое оборудование, периферийные устройства и т. д.

4. Лицензии на ИТ-активы: эта категория включает правовые документы, регулирующие использование программного обеспечения и других ИТ-активов. Управление лицензиями — ключевой аспект соответствия требованиям и снижения рисков.

5. Контракты по IT-активам: документы, определяющие договорные отношения с поставщиками IT-активов и услуг.

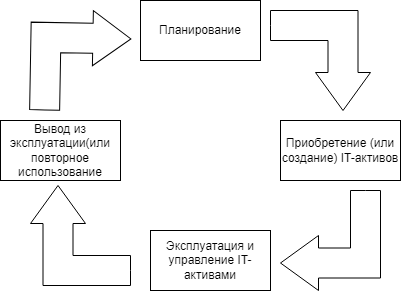
6. Сервисы IT-активов: услуги, связанные с поддержкой и обслуживанием IT-активов, включая техническое обслуживание, консультации и обучение.

Также нам необходимо понятие жизненного цикла IT-актива:

Жизненный цикл IT-актива — это комплекс этапов, через которые проходит IT-актив на протяжении своего существования, включая связанные с ним финансовые потоки.

Этот цикл охватывает все стадии жизни актива: от его планирования и покупки до эксплуатации и последующего вывода из использования, включая возможные варианты утилизации. В зависимости от подхода и источников информации, жизненные циклы активов могут значительно варьироваться [4].

Тем не менее, можно выделить четыре основные фазы жизненного цикла IT-актива, как показано на рисунке 1:



**Рисунок 1**. **Фазы жизненного цикла IT-актива**.

IT-активы играют ключевую роль в достижении как оперативных, так и стратегических целей компании. Они обеспечивают инфраструктуру для автоматизации рутинных задач, что позволяет повысить производительность и снизить ошибки, возникающие при выполнении операций вручную. Современные программные решения позволяют компании не только эффективно управлять своими бизнес-процессами, но и быстро адаптироваться к изменениям на рынке, что является важным конкурентным преимуществом в динамично развивающейся технологической среде.

Кроме того, IT-активы способствуют инновациям и развитию новых продуктов и услуг. Например, внедрение новых технологий может открыть перед компанией новые возможности для оптимизации внутренних процессов или улучшения взаимодействия с клиентами. Инвестиции в передовые IT-активы могут стать драйвером роста и инновационного развития компании.

## 1.2 Подходы к управлению IT-активами предприятия

Управление ИТ-активами представляет собой область деятельности, посвящённую планированию, учёту и мониторингу состояния ИТ-активов. Данная деятельность обеспечивает необходимую ценность для основной деятельности организации и предоставляет прозрачный контроль финансовых потоков на протяжении жизненного цикла ИТ-активов, учитывая все сопутствующие характеристики (артефакты).

ITAM — это непрерывный процесс, требующий регулярного выполнения и корректировки в ответ на изменения в активах, стратегических целях и технологическом окружении.

Первым этапом в управлении ИТ-активами является составление полной и точной инвентаризации всех ИТ-ресурсов. Это подразумевает знание о наличии активов, их местоположении, дате приобретения и стоимости.

На втором этапе производится расчёт полной стоимости владения каждым активом на протяжении его жизненного цикла. Данный расчёт включает затраты на обслуживание, обновления и утилизацию, что позволяет более чётко оценить реальную ценность каждого актива.

Третий шаг включает постоянный мониторинг IТ-активов с применением специализированных инструментов. Это необходимо для отслеживания сроков действия лицензий, контрактов и гарантий, а также для обеспечения своевременного проведения обслуживания активов. Все действия, связанные с обслуживанием, должны фиксироваться и анализироваться, чтобы оценить эффективность работы каждого актива и спланировать его дальнейшее использование.

Завершающий этап управления ИТ-активами заключается в планировании бюджета. На основе информации о стоимости и сроках службы активов можно эффективно прогнозировать расходы на их обслуживание и модернизацию, что способствует поддержанию высоких уровней обслуживания для ключевых активов и корректировке подхода к менее эффективным ресурсам.

Важным аспектом управления IT-активами является выбор программных решений, которые соответствуют актуальным требованиям и способны адаптироваться к возможным изменениям в будущем. Гибкие инструменты управления дадут возможность эффективно справляться с новыми задачами и улучшить управление ресурсами [5].

Для успешного управления ИТ-активами также необходимо обеспечить прозрачность и доступность информации для всех заинтересованных сторон.

При выборе программного обеспечения для управления ИТ-ресурсами важно чётко определить его цели и задачи. Одним из основных преимуществ является оптимизация затрат. Согласно данным компании Gartner, оптимизация лицензий на программное обеспечение может снизить расходы до 30%. Ручное выполнение данной задачи представляет собой сложный процесс, в то время как автоматизация значительно упрощает его и позволяет сэкономить средства.

Использование электронных таблиц для отслеживания активов может привести к ошибкам и устареванию данных, в результате чего компании могут терять значительные суммы на "призрачные активы", которые учтены, но физически отсутствуют.

В организациях часто происходят изменения, например, приобретение новых устройств или утрата оборудования. При отсутствии системы управления активами может потребоваться выделение отдельного сотрудника для упрощения данного процесса, что явно неэффективно.

Также наблюдается проблема с наличием так называемых "теневых" IT-активов. Часто программное обеспечение и лицензии приобретаются без ведома ИТ-отдела, что создаёт дополнительные риски и затраты. Специальные инструменты могут помочь ИТ-отделу контролировать такие ситуации и оптимизировать процессы.

Начинать управление активами следует с инвентаризации всех доступных ресурсов, что позволит оценить жизненные циклы и затраты в сотрудничестве с финансовым отделом.

Современные системы управления активами обеспечивают автоматизацию рутинных задач. Например, сотрудники могут самостоятельно запрашивать лицензии через портал самообслуживания, позволяя ИТ-специалистам сосредоточиться на более стратегических задачах.

Системы совместной работы также способствуют устранению теневых процессов. Интеграция с популярными инструментами, такими как Slack, позволяет сотрудникам легко отправлять запросы, что увеличивает контроль над ИТ-активами.

Таким образом, каждая организация уникальна, и стратегии управления активами должны учитывать её специфические потребности. Может потребоваться учёт не только физических, но и нематериальных активов, таких как лицензии и документы по соблюдению нормативных требований. Следовательно, существует разнообразие решений для управления активами — от простых инструментов до сложных систем, способных автоматически обнаруживать оборудование в сети.

## 1.3 Обзор и анализ существующих ITAM

Существует множество инструментов, предназначенных для управления ИТ-активами, каждый из которых предлагает уникальные возможности и функционал. При выборе подходящего решения важно учитывать различные критерии, такие как тип решения, гибкость, методы добавления активов, поддержка мобильных устройств, возможности отчетности и интеграции с другими системами.

Для более глубокого анализа существующих инструментов управления ИТ-активами был проведён сравнительный анализ нескольких популярных решений на рынке. В таблице будут представлены основные характеристики и функциональные возможности таких инструментов, как Asset Panda, ManageEngine AssetExplorer, SolarWinds Service Desk, Pulseway и LANsweeper. Сравнительная таблица ITAM приведена приложении А.

Анализируя представленные в таблице данные, можно сделать важные выводы о существующих решениях для управления ИТ-активами (ITAM) и их ключевых особенностях. Каждая система имеет свои преимущества и ограничения, и создание идеальной ITAM-системы требует учета всех выявленных факторов. Основными аспектами, которые заслуживают особого внимания, являются гибкость, доступность мобильного управления, интеграция с другими системами, функционал по сбору и обработке данных, а также простота использования для конечного пользователя.

Гибкость и настраиваемость — важные характеристики, позволяющие адаптировать ITAM под нужды предприятия. Например, Asset Panda и ManageEngine AssetExplorer обладают высокой гибкостью, в то время как системы, такие как SolarWinds Service Desk, могут быть ограничены, особенно в облачных версиях, подчеркивая необходимость модульной архитектуры.

Поддержка мобильных устройств и автоматическое обнаружение активов также критичны. Системы, такие как Pulseway, с полноценными мобильными приложениями, становятся необходимыми. Автоматическое обнаружение ускоряет учет и минимизирует ошибки, что повышает эффективность.

Интеграция с другими системами и возможность генерации настраиваемых отчетов влияют на эффективность работы ITAM. ManageEngine AssetExplorer демонстрирует, что автоматизация управления лицензиями и детализированная отчетность важны для успеха, подчеркивая необходимость мощных интеграционных инструментов и настраиваемых отчетов.

На основе проведенного анализа можно выделить основные требования к разрабатываемой системе учета и контроля ИТ-активов:

1. Гибкость и модульная архитектура. Система должна быть легко адаптируемой и настраиваемой под нужды предприятия, с возможностью добавления новых функций.

2. Мобильное управление. Поддержка Android и iOS с полным функционалом через мобильное приложение.

3. Автоматическое обнаружение активов. Использование технологий для выявления активов без ручного ввода.

4. Штрихкоды и мультимедиа. Возможность использовать сканеры штрихкодов и прикреплять мультимедийные файлы, такие как фото и заметки, для детального учета активов.

5. Отчетность. Настраиваемые шаблоны отчетов для анализа использования активов, лицензий и финансовых затрат.

6. Интеграция. Поддержка API для взаимодействия с другими системами.

7. Управление лицензиями. Автоматическое отслеживание использования ПО, лицензий и сроков их истечения.

8. Простота интерфейса. Интуитивный и понятный интерфейс для минимизации времени обучения пользователей.

Идеальная ITAM должна представлять собой облачное решение с гибкой модульной структурой, поддержкой мобильного управления и возможностью автоматического обнаружения активов. Она должна быть интуитивно понятной, легко интегрируемой с другими системами и обеспечивать пользователей мощными аналитическими инструментами.

## 1.4 Цель и задачи исследования

Целью работы является разработка системы учета и контроля IT-активов предприятия и ее программного обеспечения, которое будет использоваться для автоматизации процессов управления IT-активами. Данная система должна обеспечить прозрачный учет активов, повысить эффективность бизнес-процессов и минимизировать ошибки, связанные с человеческим фактором, путем создания интуитивно понятного интерфейса, надежной базы данных и функциональных модулей.

Для достижения этой цели необходимо:

1. Проанализировать литературные источники по теме IT-активов и систем управления активами, чтобы выявить лучшие практики и современные подходы в данной области.

2. Провести сравнительный анализ существующих систем управления IT-активами (ITAM), чтобы определить их сильные и слабые стороны, а также выявить функциональные требования к новой системе.

3. Провести анализ бизнес-процессов деятельности предприятия в сфере управления IT-активами

4. Выполнить функциональное моделирование бизнес-процессов учета и контроля IT-активов;

5. Разработать методологию проектирования базы данных, которая обеспечит надежное хранение и обработку данных об IT-активах.

6. Разработать программное обеспечение для системы учета и контроля IT-активов, используя Agile-методологии в веб-разработке (SCRUM), с акцентом на интуитивно понятный пользовательский интерфейс и функциональные модули.

В результате выполнения указанных задач будет разработана система учета и контроля IT-активов предприятия, которая включает интуитивно понятный пользовательский интерфейс, надежную базу данных и функциональные модули для управления активами, заказами, брендами, отчетами и корпоративной информацией. Система пройдет тестирование, будут выявлены ее сильные и слабые стороны, а также предложены рекомендации по улучшению управления IT-активами.

## 1.5 Выводы

В первой главе работы проведен анализ проблемы управления IT-активами, что позволило определить ключевые аспекты и задачи, требующие решения. Вначале рассмотрено понятие IT-актива, что дало понимание его роли в бизнес-процессах. Далее проанализированы современные подходы управления IT-активами, выявлены их преимущества и недостатки. После проведен сравнительный анализ существующих систем управления IT-активами (ITAM), что позволило определить их сильные и слабые стороны. В заключение, сформулирована постановка задачи, направленная на разработку системы учета и контроля IT-активов, что станет основой для дальнейших исследований и разработки эффективного программного обеспечения.

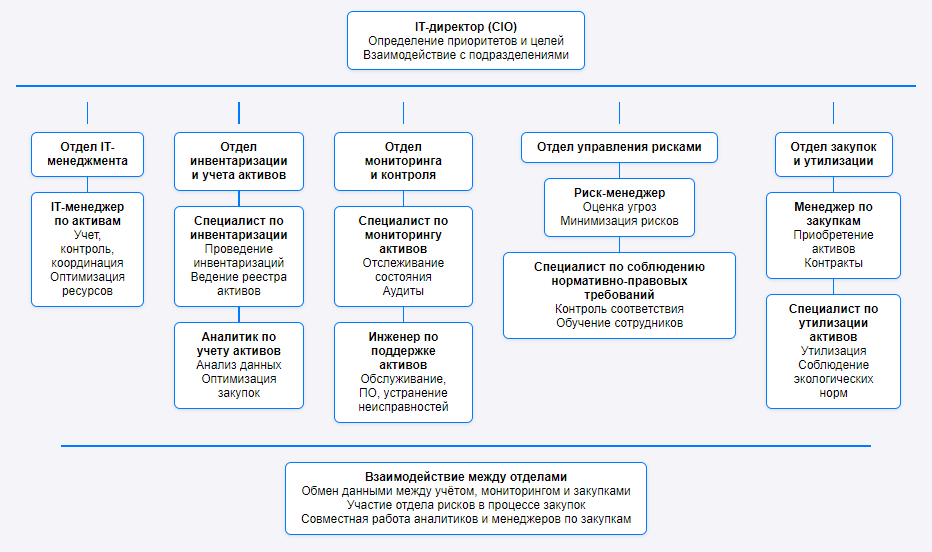
# ГЛАВА 2

# РАЗРАБОТКА И АНАЛИЗ БИЗНЕС-ПРОЦЕССОВ ДЛЯ УЧЕТА И КОНТРОЛЯ IT-АКТИВОВ ПРЕДПРИЯТИЯ

## 2.1 Анализ деятельности предприятия и проблем в сфере управления IT-активами

## Управление IT-активами в организации представляет собой многоуровневый процесс, который включает в себя несколько ключевых отделов, каждый из которых выполняет свои специфические функции.

Представим следующую обобщенную схему предприятия на рисунке 2.1, в котором описана структура управления IT-активами.



**Рисунок 2.1** **Схема организации предприятия по контролю и учёту IT-активов**.

Согласно этому рисунку, важнейшим элементом структуры является IT-директор (CIO), который определяет приоритеты работы с IT-активами и обеспечивает взаимодействие между IT-отделом и другими подразделениями компании, создавая единое информационное пространство. IT-менеджер по активам управляет учётом и контролем активов, координирует команду и отвечает за оптимизацию ресурсов.

Следующим ключевым подразделением является *отдел инвентаризации и учета активов*. Специалист по инвентаризации отвечает за идентификацию и классификацию активов, ведет реестр и следит за его актуальностью. Аналитик по учету активов анализирует данные, разрабатывает отчеты и участвует в планировании закупок и утилизации, что критично для развития организации.

*Отдел мониторинга и контроля* обеспечивает отслеживание состояния активов, проводя проверки и аудиты. Специалист по мониторингу следит за соответствием стандартам, а инженер по поддержке отвечает за техническое обслуживание и устранение неисправностей, что минимизирует время простоя.

*Отдел управления рисками* и соблюдением нормативных требований оценивает потенциальные угрозы и разрабатывает стратегии их минимизации. Специалист по соблюдению нормативных требований следит за соответствием законодательства и проводит обучение сотрудников по этим вопросам, повышая корпоративную ответственность.

Структура управления IT-активами включает также *отдел закупок и утилизации*. Менеджер по закупкам отвечает за приобретение новых активов и контроль контрактов, в то время как *специалист по утилизации* организует безопасную утилизацию устаревших активов, соблюдая экологические нормы.

*Отдел IT-менеджмента* занимается стратегическим планированием и координацией работы между подразделениями*.*

Эффективное управление IT-активами требует взаимодействия между отделами, например, отдел инвентаризации должен предоставлять данные в отдел мониторинга для своевременного реагирования на проблемы. Аналитики должны сотрудничать с менеджерами по закупкам для оптимизации процесса приобретения активов, что предотвращает избыточные закупки.

Также отдел управления рисками должен быть вовлечен в закупки для соблюдения нормативных требований и минимизации рисков. Это взаимодействие способствует более безопасному и эффективному управлению ресурсами. Интеграция процессов и обмен информацией между отделами являются ключевыми факторами успешного управления IT-активами.

Учёт и контроль IT-активов играют важную роль в управлении ресурсами. Недостаточная актуальность данных может затруднить принятие решений, поэтому регулярные инвентаризации и обновления реестра являются необходимыми.

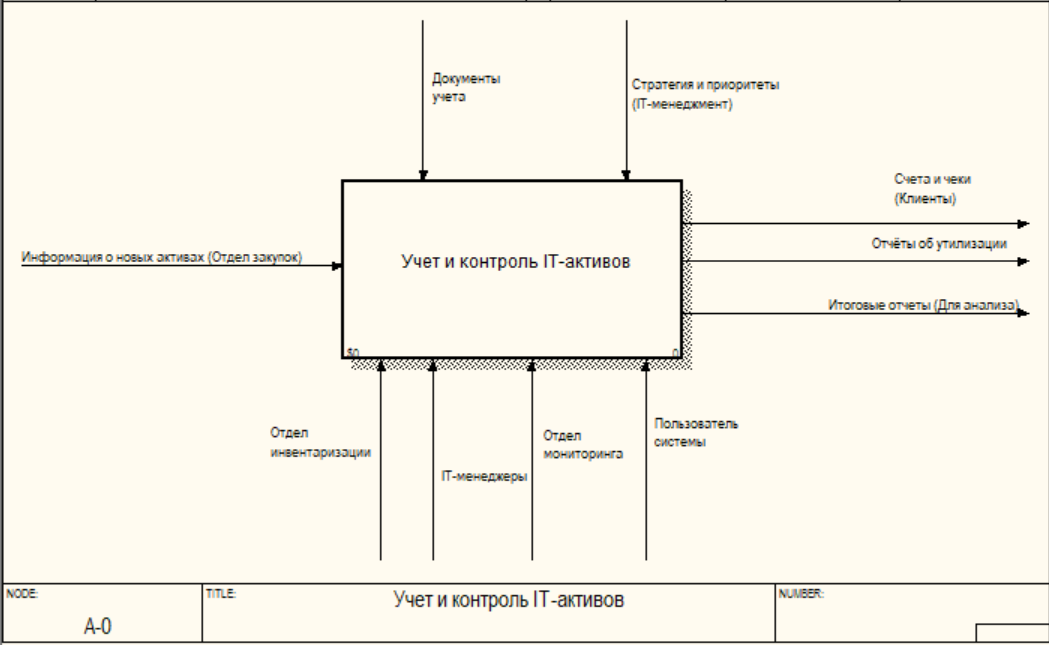
Отсутствие четких процедур контроля приводят к рискам для активов.

Таким образом, выявление проблем в учете и контроле IT-активов требует комплексного подхода. Это включает регулярные проверки, интеграцию систем, обучение и эффективное взаимодействие между подразделениями для повышения эффективности управления ресурсами и минимизации рисков.

## 2.2 Моделирование бизнес-процессов учета и контроля IT-активов

Моделирование бизнес-процессов учета и контроля IT-активов позволяет визуализировать текущие процессы и выявить области для улучшения. Для этого была использована нотация IDEF0 [7].

Контекстная диаграмма для системы «Учет и контроль IT-активов» отображает все взаимодействия между системой и внешними объектами. Она выполнена в соответствии с требованиями IDEF0, где стрелки распределены по четырем направлениям: вход (слева), управление (сверху), механизм (снизу) и выход (справа). Составленная диаграмма представлена на рисунке 2.2:



**Рисунок 2.2** **Контекстная диаграмма для системы «Учет и контроль IT-активов»**

Рассмотрим каждый элемент диаграммы на полученном рисунке: Центральный процесс (A0): "Учет IT-активов"

***Входные данные*** - информация о новых активах (Отдел закупок).

Этот компонент отвечает за передачу информации о приобретенных IT-активах в процесс учета. Отдел закупок предоставляет все необходимые документы и детали, включая спецификации оборудования, программного обеспечения и лицензий.

***Управление*** - документы учета, Стратегия и приоритеты (IT-менеджмент).

IT-менеджмент определяет общую стратегию управления IT-активами, указывает на приоритетные направления и разрабатывает документы учета для различных категорий активов, что служит основой для эффективного управления.

***Механизмы взаимодействия*** - отдел инвентаризации, IT-менеджеры, отдел мониторинга, пользователь системы.

Этот отдел отвечает за регистрацию и мониторинг IT-активов, включая их текущее состояние, использование и доступность. Он обеспечивает точность данных в учетной системе и проводит регулярные инвентаризации.

Менеджеры по IT играют ключевую роль в принятии решений, касающихся приобретения новых ресурсов, ухода за существующими активами и разработки стратегий обновления технологий.

Этот отдел отвечает за непрерывный контроль за состоянием активов, обеспечивает защиту данных и следит за соблюдением норм безопасности. Он также отслеживает производительность активов и генерирует отчеты о необходимости обновлений или замены.

Конечные пользователи системы учета IT-активов предоставляют обратную связь о работе инфраструктуры и активов, а также сообщают о любых проблемах или изменениях в использовании активов.

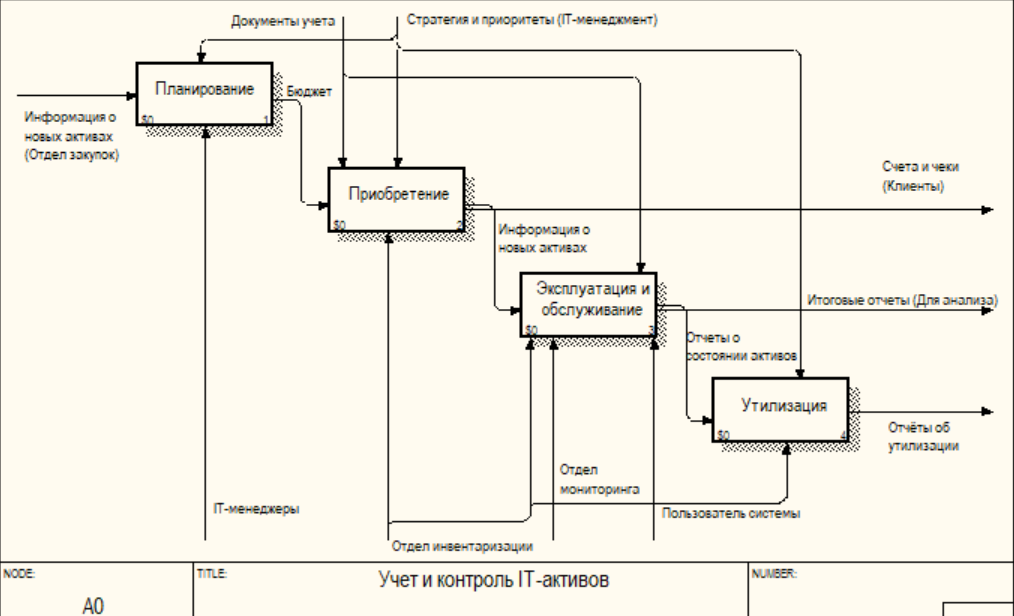
***Выходные данные*** - счета и чеки (Клиенты), отчёты об утилизации, итоговые отчеты (для анализа).

Финансовые документы, которые формируются на основе учётных данных активов. Эти документы обрабатываются и отправляются клиентам, отражая их покупку или аренду IT-ресурсов.

Эти документы содержат информацию о процессе утилизации устаревших или неиспользуемых активов после завершения их жизненного цикла. Они включают в себя все необходимые данные о порядке утилизации и соблюдении норм.

Конечные отчеты, которые анализируют данные по учету активов, их производительности и эффективности. Эти отчеты служат основой для принятия стратегических решений в будущем и помогают определить необходимость обновлений или изменений в управлении активами [4].

Диаграммы декомпозиции являются мощным инструментом для визуализации и структурирования сложных процессов. Для системы «Учет IT-активов» процесс декомпозируется на четыре этапа жизненного цикла: Планирование, Приобретение, Эксплуатация и обслуживание, Утилизация. На каждом этапе описаны подзадачи, которые будут реализованы в будущем программном продукте. Диаграмма представлена на рисунке 2.3:



**Рисунок 2.3** **Диаграмма декомпозиции A0 «Управление жизненным циклом IT-активов**

Рассмотрим подробнее диаграмму:

***Входные данные*** - информация о новых активах (Отдел закупок). Входные данные поступают из отдела закупок и передаются на этап планирования.

***Управление*** - документы учета (используется на этапах приобретения и эксплуатации для регистрации и отслеживания активов), стратегия и приоритеты (применяется ко всем четырем подпроцессам, обеспечивая общее направление и цели в управлении IT-активами).

***Механизмы взаимодействия:***

*IT-менеджеры.* Связаны с этапом планирования. Эти специалисты отвечают за стратегическое руководство и контроль за каждым из процессов.

*Отдел инвентаризации.* Обеспечивает мониторинг активов на этапах приобретения, эксплуатации и утилизации. Он отвечает за актуальность и точность данных.

*Отдел мониторинга.* Обеспечивает контроль за эксплуатацией активов и процессом утилизации, следя за их состоянием и эффективностью.

*Пользователь системы.* Связан с этапом эксплуатации, предоставляя ценную обратную связь о работе активов и их производительности.

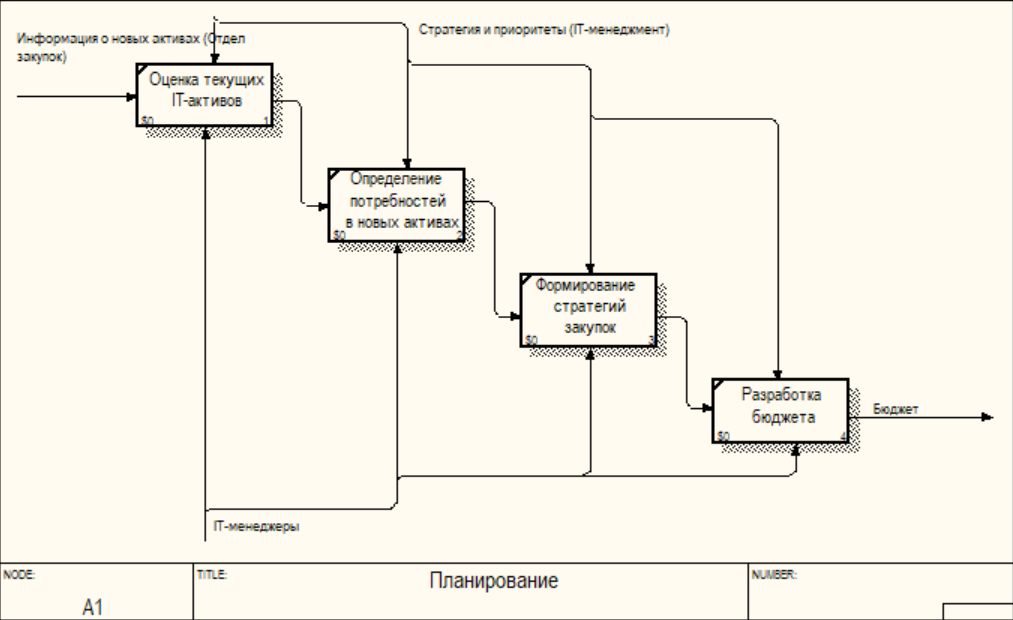
***Выходные данные:***

*Счета и чеки (Клиенты).* Формируются на этапе приобретения и предоставляются клиентам как доказательство покупки или аренды активов.

*Итоговые отчеты (Для анализа).* Выходят с этапа эксплуатации и предоставляют информацию для принятия управленческих решений и анализа эффективности активов.

*Отчёты об утилизации*. Генерируются на этапе утилизации и содержат данные о процессе утилизации старых активов, подтверждая соответствие нормам и документам.

Диаграмма декомпозиции A1: Этап "Планирование" представлена на рисунке 2.4:



**Рисунок 2.4** **Диаграмма декомпозиции A1: Этап «Планирование»**

Управление IT-активами включает несколько важных подпроцессов для эффективной оценки, планирования и приобретения ресурсов.

***Оценка текущих IT-активов*** (A1.1): На этом этапе анализируются существующие IT-активы компании, включая компьютеры, серверы и ПО. Важно оценить их состояние, производительность и соответствие задачам бизнеса, что поможет определить необходимость обновлений или замен, а также выявить дублирующие или неиспользуемые ресурсы.

***Определение потребностей в новых активах*** (A1.2). После оценки текущих активов необходимо выяснить потребности бизнеса в новых ресурсах.

***Формирование стратегий закупок*** (A1.3). Основываясь на информации о текущих активах и потребностях, разрабатываются стратегии закупок. Этот этап включает выбор поставщиков, анализ рыночной ситуации и установление оптимальных условий для приобретения, с акцентом на экономическую целесообразность и высокое качество закупаемых ресурсов [7].

***Разработка бюджета для приобретения активов*** (A1.4)  
Последний подпроцесс включает создание бюджета на основе стратегий закупок, устанавливающего финансовые рамки для каждой категории активов. Бюджет должен учитывать первоначальные затраты и будущие расходы на обслуживание.

***Связующие элементы процесса***:

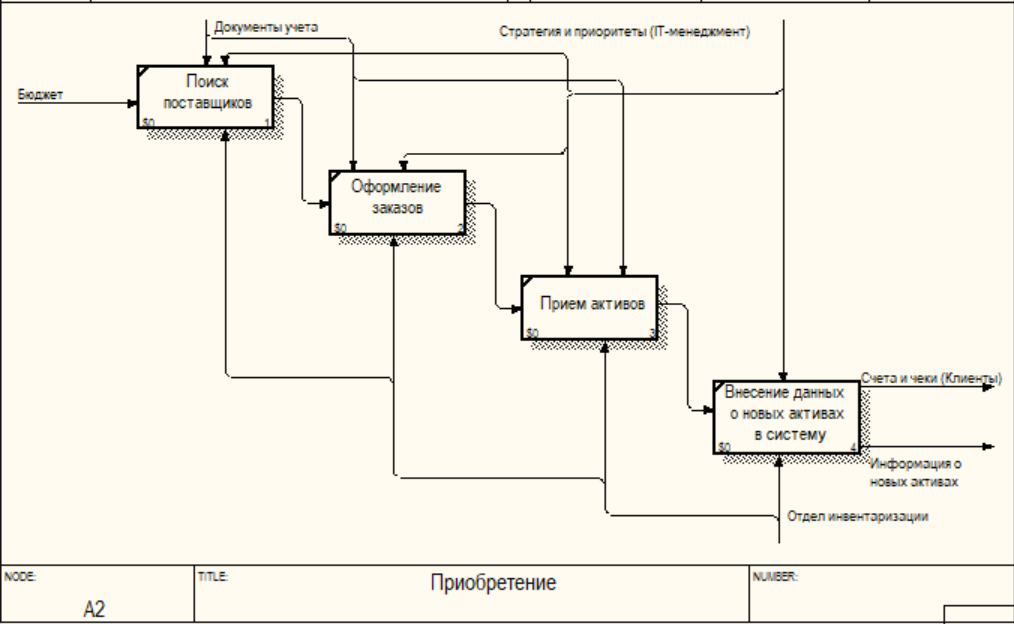
*Управление.* На всех этапах процесса важно учитывать стратегию и приоритеты IT-менеджмента.

*Вход.* Основной входной информацией для всех подпроцессов являются актуальные данные об инвентаризации и отчетности, поступающие от отдела инвентаризации. Эта информация помогает сформировать полное представление о текущих активах и их состоянии.

*Выход*. Результатом работы всех подпроцессов является план закупок и разработанный бюджет, а также рекомендации по внедрению новых активов.

*Механизм*. В выполнении всех процессов участвуют пользователи системы, среди которых — IT-менеджеры и сотрудники отдела планирования. Их роль заключается в анализе, планировании, согласовании и внедрении всех действий, направленных на эффективное управление IT-активами.

Диаграмма декомпозиции A2: Этап «Приобретение» представлена на рисунке 2.5:



**Рисунок 2.5** **Диаграмма декомпозиции A2: Этап «Приобретение»**

Процесс приобретения IT-активов — ключевой этап в управлении ресурсами компании, обеспечивающий необходимые технологии. Он включает следующие подпроцессы:

***Поиск поставщиков***. Анализ рынка и сбор информации о поставщиках, их репутации и условиях работы. Учитываются качество продукции, цена и сроки доставки. Команда закупок проводит тендеры или запросы предложений (RFP) для выбора оптимального варианта.

***Оформление заказов.*** После выбора поставщиков оформляются заказы на покупку с указанием активов, количества, цен и условий поставки. Заказы нужно согласовать с внутренними стандартами и получить одобрение руководства.

***Прием активов***. Проверка поступивших товаров на соответствие условиям заказа. Команда подтверждает соответствие спецификациям, отсутствие повреждений и комплектность, оформляя акты приемки.

***Внесение данных в систему***. После приемки информация о новых активах вводится в систему управления, включая метаданные (серийные номера, модели, даты приобретения). Это крайне важно для актуальности данных и планирования будущих закупок. Все данные должны быть задокументированы для удобного доступа и анализа [1].

***Связующие элементы процесса:***

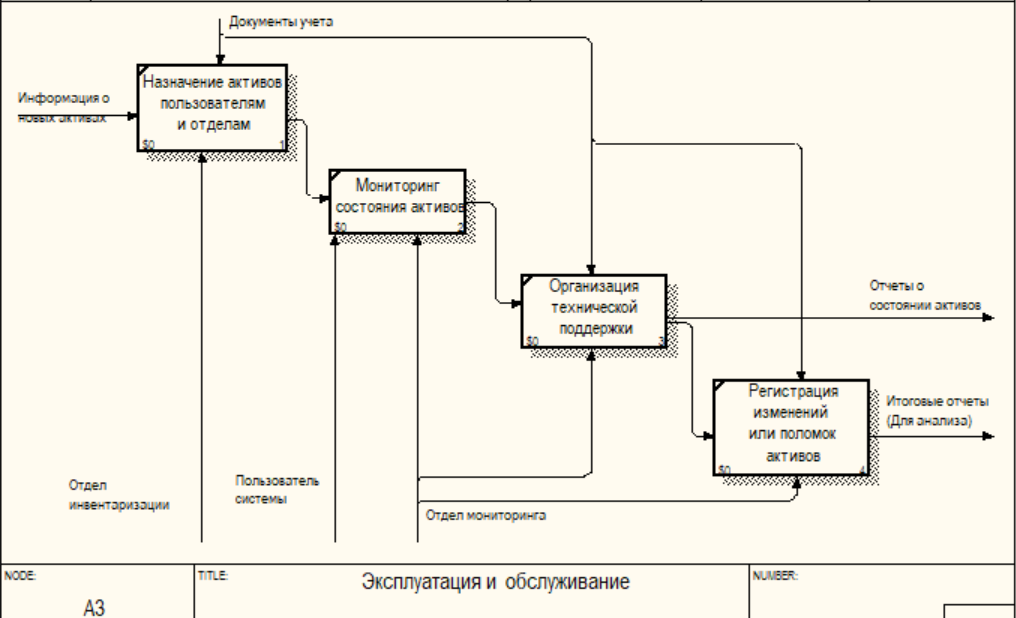
*Управление*: Важно, чтобы на каждом этапе происходила грамотная документация учета активов. Это могут быть карточки учета, акты приемки и другие документы, которые служат доказательством выполнения всех процедур.

*Вход*: Процесс начинается с утвержденного бюджета и плана закупок, которые обеспечивают необходимую финансовую основу и направление для осуществления закупок.

*Выход*: Результатом данного центрального процесса является информация о новых активах, которая затем будет использоваться для дальнейшего управления и оптимизации ресурсов компании.

*Механизм*: В процессе участвуют отдел закупок и конечные пользователи системы, которые взаимодействуют друг с другом на всех этапах. Участники процесса должны быть хорошо информированы и иметь необходимую подготовку для выполнения своих задач, что в итоге способствует более эффективному управлению IT-активами.

Диаграмма декомпозиции A3: Этап «Эксплуатация и обслуживание» представлена на рисунке 2.6:



**Рисунок 2.6** **Диаграмма декомпозиции A3: Этап «Эксплуатация и обслуживание»**

Центральный процесс A3 называется "Эксплуатация и обслуживание IT-активов" и включает несколько ключевых подпроцессов.

Подпроцессы:

***Назначение активов пользователям и отделам***. На этом этапе происходит распределение IT-активов между пользователями и подразделениями компании. Учитываются потребности каждого отдела и характеристики активов, что способствует повышению продуктивности.

***Мониторинг состояния активов***. Постоянный мониторинг производительности и работоспособности активов позволяет выявлять проблемы до их критического состояния. Используются инструменты для контроля загрузки системы и наличия обновлений.

***Организация технической поддержки***. Создается система для оказания помощи пользователям с IT-ресурсами, включая поддержку и обучение сотрудников. Быстрая и доступная поддержка минимизирует время простоя пользователей.

***Регистрация изменений или поломок активов***. Документируются все изменения и инциденты. Это аналитическая основа для улучшения обслуживания и поддержки актуальности данных в системе учета.

***Связующие элементы процесса:***

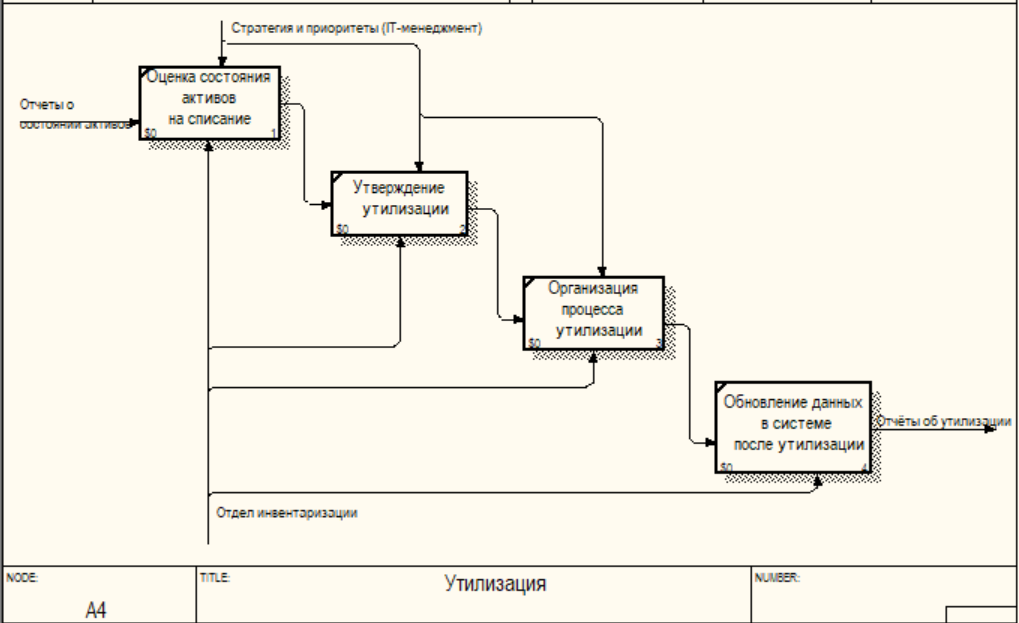
*Управление*. Регламенты и процедуры помогают установить стандарты эксплуатации активов, обеспечивая единство действий и минимизацию рисков.

*Вход*. Входной информацией служит база данных активов, включая их характеристики, назначение и состояние, что позволяет эффективно управлять и реагировать на проблемы.

*Выход*. Результат — отчеты о состоянии активов и технической поддержке, помогающие в управлении и анализе эффективности использования.

*Механизм*. В процессе участвуют отдел мониторинга и технические специалисты, которые обеспечивают функционирование активов и реагируют на запросы пользователей.

Диаграмма декомпозиции A4: Этап «Утилизация» представлена на рисунке 2.7:



**Рисунок 2.7** **Диаграмма декомпозиции A4: Этап «Утилизация»**

Центральный процесс A4 называется "Утилизация IT-активов" и представляет собой важный этап в жизненном цикле IT-ресурсов, включающий безопасное уничтожение устаревших или ненужных активов. Утилизация освобождает место для новых технологий и снижает риски, связанные с устаревшим оборудованием.

Подпроцессы:

***Оценка состояния активов на списание***. Здесь проводится оценка состояния IT-активов, предполагаемых для утилизации. Учитываются техническое состояние, функциональность и экономическая целесообразность. Анализ данных помогает определить, какие активы уже не подлежат ремонту, что снижает затраты и риски.

***Утверждение утилизации***. После оценки необходимо получить одобрение на утилизацию от руководства. Это включает согласование предложений, выбор способа утилизации и разработку плана действий для минимизации юридических последствий.

***Организация процесса утилизации***. Начинается реализация утилизации, включая выбор лицензированных компаний и контроль за соблюдением экологических норм. Разрабатывается схема утилизации, охватывающая логистику и документальное оформление.

***Обновление данных в системе после утилизации***. После утилизации обновляются данные о состоянии IT-активов в системе. Удаляются списанные активы и обновляются отчеты, что помогает избежать путаницы и обеспечивает прозрачность учета.

***Связующие элементы процесса***:

*Управление*: Процесс осуществляется в рамках политик утилизации, обеспечивающих соблюдение законодательных требований.

*Вход*: Входными данными служит информация о состоянии активов из систем мониторинга.

*Выход*: Итоговые отчеты об утилизации демонстрируют, какие активы были утилизированы и содержат рекомендации по улучшению процесса.

*Механизм*: В процессе участвуют отдел инвентаризации и лицензированные компании по утилизации, что гарантирует правильное исполнение процесса.

Таким образом, имеется одна контекстная диаграмма и пять диаграмм декомпозиции. Эти диаграммы полностью соответствуют требованиям IDEF0, отражают этапы жизненного цикла IT-активов, взаимосвязи между процессами и их взаимодействие с внешними объектами.

## 2.3 Выводы

В данной главе проведен анализ деятельности предприятия в управлении IT-активами, с акцентом на учет, контроль и оптимизацию ресурсов. Эффективное управление активно снижает затраты и повышает производительность, обеспечивая доступность ресурсов для бизнес-задач. Современные технологии автоматизируют учет, уменьшая ошибки и упрощая доступ к информации.

Обнаружены проблемы, такие как недостоверность данных, отсутствие единой системы учета, низкая прозрачность процессов и недостаток интеграции между подразделениями. Это затрудняет отслеживание активов и принятие обоснованных решений, а также увеличивает риски потери и недоиспользования ресурсов.

Моделирование бизнес-процессов учета и контроля IT-активов позволило оптимизировать существующие процессы. Модели показали, что внедрение единой информационной системы управления, автоматизация процессов и улучшение взаимодействия помогут повысить эффективность. Разработанные модели станут основой для дальнейшего совершенствования управления IT-активами и рационального использования ресурсов.

# ГЛАВА 3

**ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ДЛЯ УЧЕТА И КОНТРОЛЯ IT-АКТИВОВ**

## 3.1 Концептуальное и логическое проектирование БД «Учёт и контроль IT-активов»

При проектировании базы данных для учета и управления IT-активами, важно создать концептуальную модель данных, которая позволяет наглядно представить основную структуру информации и взаимосвязи между данными. Концептуальная модель данных не привязана к конкретной системе управления базами данных (СУБД) и служит абстрактным представлением данных, понятным для конечного пользователя. Такой подход позволяет сосредоточиться на логике предметной области, а не на технических деталях реализации [9].

Для построения концептуальной модели данных обычно используется модель "сущность-связь" (Entity-Relationship, ER), которая является стандартом в области проектирования баз данных. Данная модель позволяет выделить ключевые сущности предметной области, их атрибуты и связи между ними, что впоследствии упрощает разработку физической модели данных [10].

Проведем анализ описания предметной области базы данных IT-Assets и выделим основные сущности и их атрибуты:

***1. Bill (Счет)***

– Id (Идентификатор): int <<generated>>

– customer\_id (Идентификатор клиента): int <<FK>> : Customer

– date\_time (Дата и время): datetime

– gross\_amount (Валовая сумма): decimal(10, 2)

– service\_amount (Сумма услуги): decimal(10, 2)

– net\_amount (Чистая сумма): decimal(10, 2)

– discount (Скидка): decimal(10, 2)

– user\_id (Идентификатор пользователя): int <<FK>> : User

***2. Bill Item (Элемент счета)***

– Id (Идентификатор): int <<generated>>

– bill\_id (Идентификатор счета): int <<FK>> : Bill

– asset\_id (Идентификатор IT-актива): int <<FK>> : IT-Assets

– asset\_qty\_need (Необходимое количество): int

– asset\_total (Итоговая сумма): decimal(10, 2)

***3. Brand (Бренд)***

– Id (Идентификатор): int <<generated>>

– brand\_name (Название бренда): nvarchar(50)

– brand\_status (Статус бренда): int

***4. Таблица Software (Программное обеспечение)***

– Id (Идентификатор): int <<generated>>

– software\_name (Название программного обеспечения): nvarchar(100) NOT NULL

– version (Версия): nvarchar(50)

– software\_type (тип по: исполняемое или неисполняемое): nvarchar(100)

– installation\_date (Дата установки): date

***5. Таблица Media (Носители)***

– Id (Идентификатор): int <<generated>>

– media\_type (Тип носителя): nvarchar(50) NOT NULL -- цифровой или физический

– media\_description (Описание носителя): nvarchar(255)

– capacity (Вместимость): nvarchar(50) -- Вместимость, если применимо

– is\_digital (Цифровой носитель): bit -- Является ли носитель цифровым

***6. Таблица Hardware (ИТ-оборудование)***

– Id (Идентификатор): int <<generated>>

– hardware\_name (Название оборудования): nvarchar(100) NOT NULL

– hardware\_type (Тип оборудования): nvarchar(50) -- физическое или виртуальное

– manufacturer (Производитель): nvarchar(100)

– purchase\_date (Дата покупки): date

– warranty\_expiry (Срок гарантии): date

***7. Таблица License (Лицензии)***

– Id (Идентификатор): int <<generated>>

– license\_key (Ключ лицензии): nvarchar(100) NOT NULL

– confirmation\_date (Дата подтверждения): date

– valid\_until (Срок действия): date

***8. Таблица Contract (Контракты)***

– Id (Идентификатор): int <<generated>>

– contract\_number (Номер контракта): nvarchar(50) NOT NULL

– start\_date (Дата начала): date

– end\_date (Дата окончания): date

– details (Детали контракта): nvarchar(255)

– supplier (Поставщик): nvarchar(100)

***9. Таблица Category (Категория)***

– Id (Идентификатор): int <<generated>>

– category\_name (Название категории): nvarchar(50)

– category\_status (Статус категории): int

– software\_id (Id программного обеспечения): int REFERENCES Software(Id) -- Внешний ключ на Software

– media\_id (Id носителя): int REFERENCES Media(Id) -- Внешний ключ на Media

– hardware\_id (Id оборудования): int REFERENCES Hardware(Id) -- Внешний ключ на Hardware

– license\_id (Id лицензии): int REFERENCES License(Id) -- Внешний ключ на License

– contract\_id (Id контракта): int REFERENCES Contract(Id) -- Внешний ключ на Contract

***10. Instock (На складе)***

– Id (Идентификатор): int <<generated>>

– asset\_id (Идентификатор IT-актива): int <<FK>> : IT-Assets

– qty\_new (Новое количество): int

– purchase\_price (Цена покупки): decimal(10, 2)

– date\_time (Дата и время): datetime

– remark (Примечание): nvarchar(200)

***11. IT-Assets (ИТ-активы)***

– Id (Идентификатор): int <<generated>>

– asset\_name (Название IT-актива): nvarchar(255)

– asset\_sku (SKU IT-актива): nvarchar(255)

– asset\_inventory (инвентарный номер IT-актива): nvarchar(255)

– asset\_price (Цена IT-актива): decimal(10, 2)

– asset\_quantity (Количество IT-актива): int

– asset\_desc (Описание IT-актива): text

– asset\_brand (Идентификатор бренда): int <<FK>> : Brand

– asset\_category (Идентификатор категории): int <<FK>> : Category

– asset\_status (Статус IT-актива): int

– department\_id (Идентификатор отдела): int <<FK>> : Department

***12. User (Пользователь)***

– Id (Идентификатор): int <<generated>>

– username (Имя пользователя): nvarchar(20)

– password (Пароль): nvarchar(30)

– role (Роль): nvarchar(30) // пользователь или администратор

– customer\_phone (Телефон клиента): nvarchar(50)

– department\_id (Идентификатор отдела): int <<FK>> : Department

***13. Department (Отдел)***

– Id (Идентификатор): int <<generated>>

– department\_name (Название отдела): nvarchar(100)

– department\_head (Руководитель отдела): int <<FK>> : User

– number\_employees (количество сотрудников): int

***Далее определим связи между сущностями:***

1. Счет (Bill) принадлежит к Клиенту (Customer) по полю customer\_id. Один клиент может иметь несколько счетов, но каждый счет связан только с одним клиентом. Это связь один-ко-многим между сущностями Bill и Customer.

2. Счет (Bill) связан с Пользователем (User ) по полю user\_id. Один пользователь может создавать множество счетов, но каждый счет связан только с одним пользователем. Это связь один-ко-многим между сущностями Bill и User.

3. Счет (Bill) содержит Элементы счета (Bill Item) по полю bill\_id. Один счет может содержать несколько элементов счета, но каждый элемент счета связан только с одним счетом. Это связь один-ко-многим между сущностями Bill и Bill Item.

4. Элемент счета (Bill Item) связан с IT-активами (IT-Assets) по полю asset\_id. Один IT-актив может быть связан с несколькими элементами счета, но каждый элемент счета связан только с одним IT-активом. Это связь один-ко-многим между сущностями Bill Item и IT-Assets.

5. IT-активы (IT-Assets) связаны с Брендом (Brand) по полю asset\_brand. Один бренд может иметь множество IT-активов, но каждый IT-актив связан только с одним брендом. Это связь один-ко-многим между сущностями IT-Assets и Brand.

6. IT-активы (IT-Assets) связаны с Категорией (Category) по полю asset\_category. Одна категория может содержать множество IT-активов, но каждый IT-актив связан только с одной категорией. Это связь один-ко-многим между сущностями IT-Assets и Category.

7. Элементы на складе (Instock) связаны с IT-активами (IT-Assets) по полю asset\_id. Один IT-актив может находиться на складе в разных экземплярах, но каждый элемент на складе связан только с одним IT-активом. Это связь один-ко-многим между сущностями Instock и IT-Assets.

8. Цифровые активы (Digital Assets) связаны с Категорией (Category) по полю category\_id. Одна категория может содержать множество цифровых активов, но каждый цифровой актив связан только с одной категорией. Это связь один-ко-многим между сущностями Digital Assets и Category.

9. Категория (Category) связана с активами ПО (Software) по полю software\_id. Каждая категория может содержать только один актив ПО, и каждый актив ПО связан только с одной категорией. Это связь один к одному между сущностями Category и Software.

10. Категория (Category) связана с носителями (Media) по полю media\_id. Каждая категория может содержать только один носитель, и каждый носитель связан только с одной категорией. Это связь один к одному между сущностями Category и Media.

11. Категория (Category) связана с оборудованием (Hardware) по полю hardware\_id. Каждая категория может содержать только одно оборудование, и каждое оборудование связано только с одной категорией. Это связь один к одному между сущностями Category и Hardware.

12. Категория (Category) связана с лицензиями (License) по полю license\_id. Каждая категория может содержать только одну лицензию, и каждая лицензия связана только с одной категорией. Это связь один к одному между сущностями Category и License.

13. Категория (Category) связана с контрактами (Contract) по полю contract\_id. Каждая категория может содержать только один контракт, и каждый контракт связан только с одной категорией. Это связь один к одному между сущностями Category и Contract.

14. Пользователи и Отдел. Многие пользователи могут относиться к одному отделу.

15. IT-активы и Отдел. Активы могут быть распределены по отделам.

***Выделим группы пользователей базы данных:***

1. Администраторы системы имеют полный доступ к данным, что позволяет им управлять информацией, настраивать параметры и контролировать безопасность. Они отслеживают состояние IT-активов и складских остатков, что помогает обнаруживать проблемы на ранних стадиях.

2. Сотрудники отделов (Пользователи) имеют доступ лишь к данным своих отделов. Они управляют IT-активами и создают счета, что позволяет эффективно выполнять свои задачи.

3. Менеджеры отдела отвечают за сбор и анализ информации о ресурсах и сотрудниках, оптимизируя распределение IT-активов и контролируя затраты.

4. Клиенты получают доступ только к своим счетам, чтобы просматривать историю операций. Это ограничение гарантирует безопасность данных и повышает доверие к системе.

Каждая группа играет важную роль в работе базы данных учета IT-активов, и четкое распределение прав и обязанностей способствует ее эффективности [2].

***Выделим требования для атрибутов базы данных «Учёт и контроль IT-активов»:***

1. Таблица Bill (Счет) включает атрибут ID покупателя как INT, автоинкрементируемый и обязательный, служащий первичным ключом. Это гарантирует уникальность каждого счета.

2. customer\_id тоже INT, является внешним ключом, связывающим счет с конкретным клиентом для обеспечения целостности данных.

3. Атрибут date\_time хранит дату и время в формате DATETIME и также обязателен для заполнения.

4. gross\_amount — валовая сумма, заданная как DECIMAL(10, 2) и обязательная, как и service\_amount. net\_amount — итоговая сумма, тоже обязательна. Скидка discount — необязательна.

5. user\_id — идентификатор пользователя, обязательный для заполнения, отслеживающий, кто создал счет.

6. В таблице Bill Item (Элемент счета) атрибут Id — INT <<generated>>, как прежние идентификаторы. bill\_id — внешний ключ, связывающий элемент со счетом.

7. asset\_id — обязательный идентификатор IT-актива, а asset\_qty\_need — необходимое количество актива, заданное INT.

8. asset\_total — итоговая сумма, обязательная и заданная как DECIMAL.

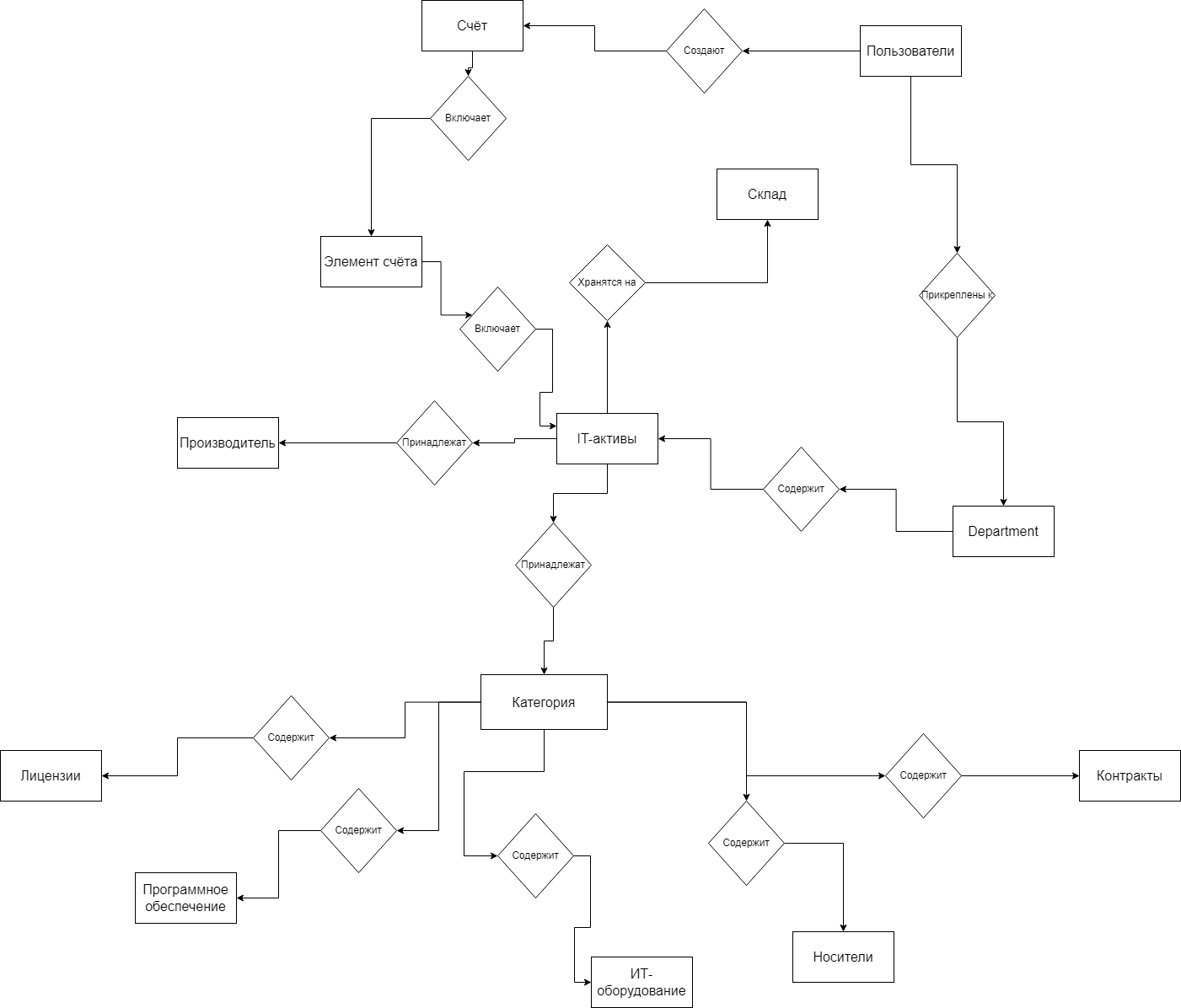
9. В таблице Brand (Бренд) Id — INT <<generated>>, brand\_name — NVARCHAR(50), обязательный, а brand\_status — целочисленный.

10. В остальных таблицах, таких как Software, Media, Hardware, License, Contract и Category, общие требования остаются схожими. Пример: software\_name — NVARCHAR(100) и обязательный, в то время как version и installation\_date — необязательные.

11. Для таблицы Media, атрибут media\_type обязателен, media\_description — необязателен.

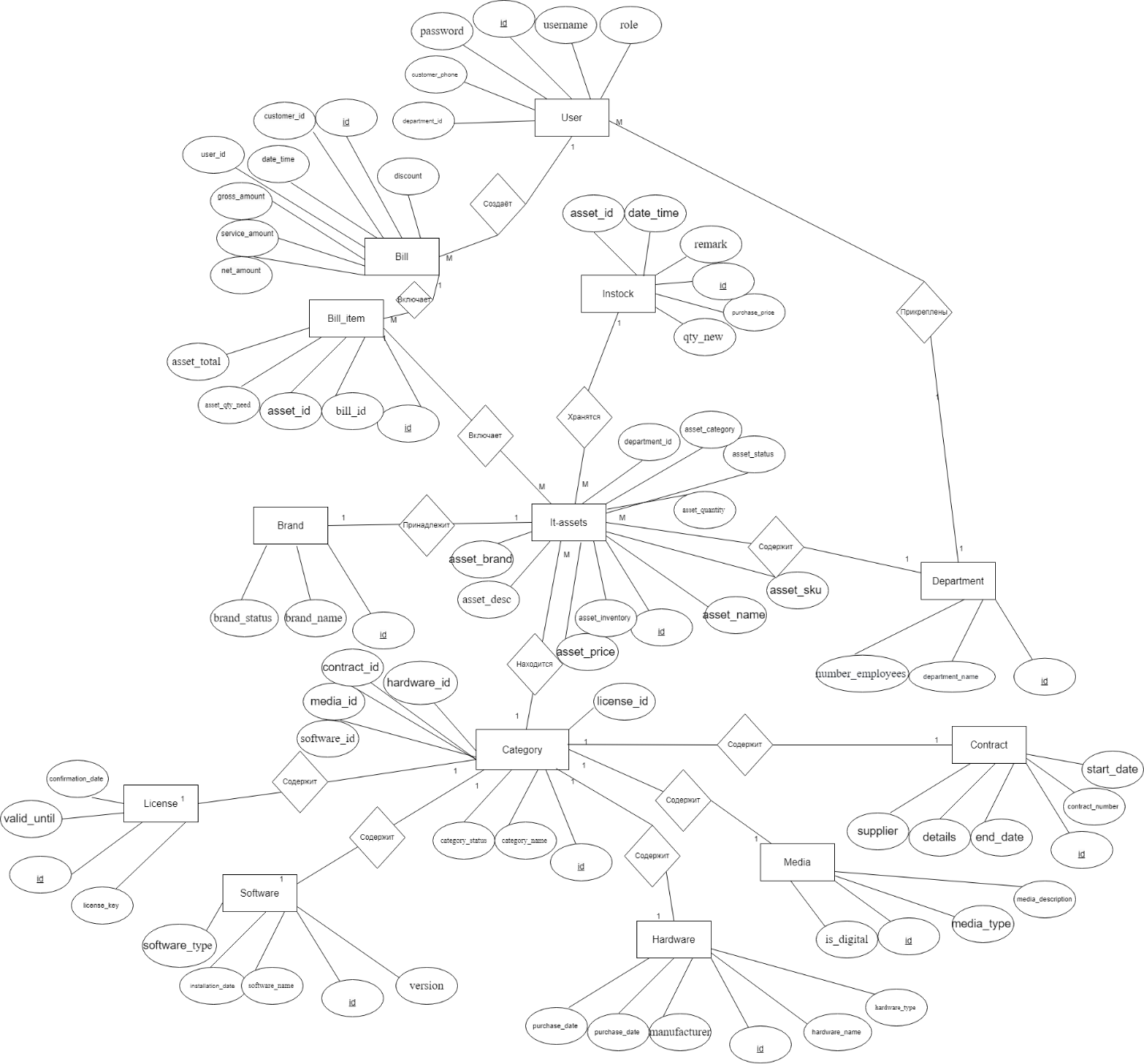
12. Таблицы Hardware, License, Contract и другие аналогичны, создавая гибкую структуру для управления данными и обеспечивая целостность базы данных.

На рисунке 3.1 представлена схема вариантов использования, где каждый прямоугольник символизирует отдельную сущность:



**Рисунок 3.1 Схема вариантов использования**

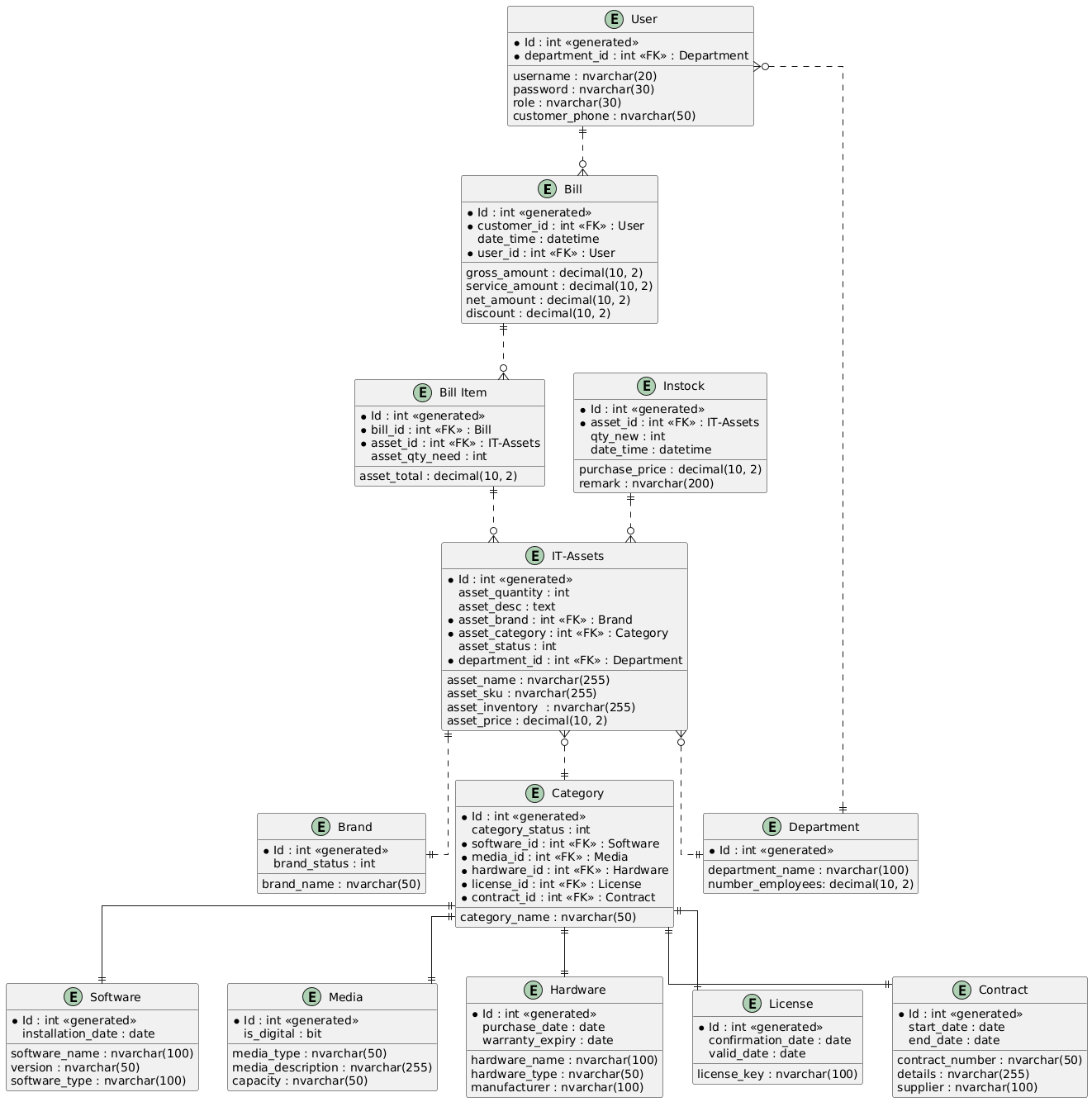
Исходя из схемы вариантов использования, представим нашу концептуальную модель базы данных в виде следующей ER-диаграммы на рисунке 3.2:



**Рисунок 3.2** **Диаграмма в нотации Чена (ER-диаграмма)**

Основная задача логического уровня проектирования заключается в детальном описании инфологической модели и преобразовании её в схему, где ранее определённые сущности, атрибуты и связи оформляются в соответствии с правилами моделирования, соответствующими выбранному типу базы данных.

Именно на логическом уровне формируется структура будущей базы данных, поэтому все принятые решения окажут влияние на удобство её использования, производительность и безопасность данных. На рисунке 3.3 представлена логическая модель БД «Учёт и контроль IT-активов».



**Рисунок 3.3 Логическая модель БД «Учёт и контроль IT-активов»**

Считается, что сущности находятся в третьей нормальной форме, когда каждый ключевой атрибут зависит исключительно от первичного ключа, и ни один ключевой атрибут не зависит от других неключевых атрибутов. Это означает, что все атрибуты сущностей должны состоять только из атомарных значений.

Реляционные таблицы базы данных представлены в таблицах Б1-Б13 в Приложении Б, которые содержат множество значительных деталей и связаны между собой с помощью ключей.

## 3.2 Программная реализация БД «Учёт и контроль IT-активов»

## Microsoft SQL Server Management Studio (SSMS) — это интегрированная среда для управления и разработки на платформе Microsoft SQL Server. SSMS предоставляет мощные средства для администрирования, проектирования и управления SQL Server и его базами данных. С его помощью пользователи могут создавать базы данных, управлять ими, выполнять запросы, настраивать безопасность, а также выполнять резервное копирование и восстановление данных.

## SSMS включает в себя графический интерфейс и позволяет работать с T-SQL, упрощая выполнение сложных SQL-команд. Он подходит как для новичков, так и для опытных разработчиков и администраторов. С помощью SSMS можно легко визуализировать структуры таблиц, создавать представления, сохранять запросы и создавать отчеты [6].

Следует начать с создания базы данных, необходимой для управления IT-активами. Мы создаем базу данных с именем IT\_Asset\_Management и переключаемся на нее с помощью команды USE. Затем последовательно создаем таблицы, каждая из которых отвечает за определенную часть системы управления активами.

Ниже приведён код создания базы данных и таблиц на языке Transact-Sql с обеспечением целостности данных.

Листинг 3.1 – Создание базы данных «Учёт и контроль IT-активов»

и таблиц в SSMS

***-- Создание базы данных***

*CREATE DATABASE IT\_Asset\_Management;*

*-- Использование созданной базы данных*

*USE IT\_Asset\_Management;*

***-- Таблица User***

*CREATE TABLE [User ] (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*username NVARCHAR(20) NOT NULL,*

*password NVARCHAR(30) NOT NULL,*

*role NVARCHAR(30) NOT NULL,*

*customer\_phone NVARCHAR(50),*

*department\_id INT);*

***-- Таблица Department***

*CREATE TABLE Department (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*department\_name NVARCHAR(100) NOT NULL,*

*department\_head INT,*

*number\_employees INT);*

***-- Таблица Bill***

*CREATE TABLE Bill (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*customer\_id INT,*

*date\_time DATETIME NOT NULL,*

*gross\_amount DECIMAL(10, 2) NOT NULL,*

*service\_amount DECIMAL(10, 2) NOT NULL,*

*net\_amount DECIMAL(10, 2) NOT NULL,*

*discount DECIMAL(10, 2) NOT NULL,*

*user\_id INT);*

***-- Таблица Bill Item***

*CREATE TABLE Bill\_Item (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*bill\_id INT,*

*asset\_id INT,*

*asset\_qty\_need INT NOT NULL,*

*asset\_total DECIMAL(10, 2) NOT NULL);*

***-- Таблица Brand***

*CREATE TABLE Brand (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*brand\_name NVARCHAR(50) NOT NULL,*

*brand\_status INT);*

***-- Таблица Software***

*CREATE TABLE Software (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*software\_name NVARCHAR(100) NOT NULL,*

*version NVARCHAR(50),*

*software\_type NVARCHAR(100),*

*installation\_date DATE);*

***-- Таблица Media***

*CREATE TABLE Media (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*media\_type NVARCHAR(50) NOT NULL,*

*media\_description NVARCHAR(255),*

*capacity NVARCHAR(50),*

*is\_digital BIT);*

***-- Таблица Hardware***

*CREATE TABLE Hardware (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*hardware\_name NVARCHAR(100) NOT NULL,*

*hardware\_type NVARCHAR(50),*

*manufacturer NVARCHAR(100),*

*purchase\_date DATE,*

*warranty\_expiry DATE);*

***-- Таблица License***

*CREATE TABLE License (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*license\_key NVARCHAR(100) NOT NULL,*

*confirmation\_date DATE,*

*valid\_until DATE);*

***-- Таблица Contract***

*CREATE TABLE Contract (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*contract\_number NVARCHAR(50) NOT NULL,*

*start\_date DATE NOT NULL,*

*end\_date DATE NOT NULL,*

*details NVARCHAR(255),*

*supplier NVARCHAR(100));*

***-- Таблица Category***

*CREATE TABLE Category (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*category\_name NVARCHAR(50) NOT NULL,*

*category\_status INT,*

*software\_id INT REFERENCES Software(Id),*

*media\_id INT REFERENCES Media(Id),*

*hardware\_id INT REFERENCES Hardware(Id),*

*license\_id INT REFERENCES License(Id),*

*contract\_id INT REFERENCES Contract(Id));*

***-- Таблица Instock***

*CREATE TABLE Instock (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*asset\_id INT REFERENCES IT\_Assets(Id),*

*qty\_new INT NOT NULL,*

*purchase\_price DECIMAL(10, 2) NOT NULL,*

*date\_time DATETIME NOT NULL,*

*remark NVARCHAR(200));*

***-- Таблица IT-Assets***

*CREATE TABLE IT\_Assets (*

*Id INT IDENTITY(1,1) PRIMARY KEY,*

*asset\_name NVARCHAR(255) NOT NULL,*

*asset\_sku NVARCHAR(255),*

*asset\_inventory NVARCHAR(255),*

*asset\_price DECIMAL(10, 2) NOT NULL,*

*asset\_quantity INT NOT NULL,*

*asset\_desc TEXT,*

*asset\_brand INT REFERENCES Brand(Id),*

*asset\_category INT REFERENCES Category(Id),*

*asset\_status INT,*

*department\_id INT REFERENCES Department(Id)*

*);*

Далее следует наполнить созданную базу данных данными, которые необходимы для функционирования системы. Используются команды INSERT INTO для добавления записей в каждую из таблиц. Заполнение таблиц представлено в листинге Б2 в Приложении Б.

## 3.3 Выводы

В данной секции главы было проведено концептуальное и логическое проектирование базы данных для системы управления IT-активами. На этапе концептуального проектирования была разработана модель данных, отражающая основные сущности, их атрибуты и связи между ними. Эта модель включала ключевые элементы, такие как пользователи, департаменты, активы и счета. Логическое проектирование позволило детализировать структуру базы данных, определить типы данных и отношения между таблицами, обеспечивая целостность и согласованность информации. В результате, созданная логическая схема легла в основу дальнейшей разработки и реализации базы данных, что обеспечит свою эффективность и легкость в использовании.

На этапе программной реализации была выполнена разработка и внедрение базы данных на платформе Microsoft SQL Server. В процессе реализации были созданы необходимые таблицы с соответствующими полями и ограничениями, определены индексы для повышения производительности запросов, а также реализованы механизмы для обеспечения целостности данных. Также были внедрены процедуры и триггеры для автоматизации обработки данных, что повысило оперативность управления активами. В результате успешной реализации база данных готова к эксплуатации и дальнейшему использованию в системе управления IT-активами, что позволит улучшить учет, контроль и аналитические возможности предприятия.

В соответствии с проведенными этапами проектирования и реализации базы данных была достигнута цель по созданию эффективной системы управления IT-активами. Разработанная база данных обеспечивает надежное хранение, быстрый доступ и удобное управление информацией о ресурсах. Она закладывает основу для улучшения процессов учета, контроля и принятия решений на предприятии. Работа, выполненная на данном этапе, является критически важной для дальнейших шагов по оптимизации управления IT-активами и повышения общей эффективности бизнеса.

# ГЛАВА 4

# РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 4.1 Выбор технологий для программной реализации системы учета и контроля IT- активов

В данной курсовой работе была выбрана комплексная архитектура для разработки веб-приложения.

Microsoft Visual Studio .NET была выбрана в качестве основной среды разработки для реализации веб-приложения благодаря своей богатой функциональности и удобному интерфейсу. Эта интегрированная среда разработки (IDE) предоставляет мощные инструменты, необходимые для разработки, отладки и развертывания приложений на платформе .NET. Visual Studio поддерживает различные языки программирования и позволяет разработчикам быстро создавать и тестировать приложения без необходимости переключаться между несколькими инструментами. Это значительно ускоряет процесс разработки, что особенно важно для соблюдения сроков проекта [1].

ASP.NET была выбрана для построения веб-приложения, так как это мощная платформа для создания интерактивных и динамических веб-решений. В процессе разработки использовались компоненты, такие как GridView и Repeater, для отображения таблиц из базы данных. ASP.NET Repeater позволил эффективно выводить повторяющиеся табличные данные, что сослужило хорошую службу при необходимости обработки форм. Также был использован SqlDataSource, что обеспечивает простое соединение с базой данных в реальном времени. Кнопки ASP.NET работают на сервере, что позволяет серверу контролировать действия клиента с использованием языка C# в качестве серверной логики. ASP.NET предоставляет богатый набор контролов, таких как текстовые поля, кнопки и метки, которые комбинируются для динамического создания HTML-страниц, что делает разработку более гибкой и удобной.

C# был выбран в качестве языка программирования для серверной части системы благодаря его сильной типизации и объектно-ориентированным возможностям, которые идеально подходят для построения бизнес-логики приложения. Этот язык программирования является частью платформы ASP.NET и управляет взаимодействием между веб-приложением и базой данных. В процессе реализации C# использовался в сочетании с SQL для выполнения операций CRUD (создание, чтение, обновление, удаление) на данных инвентаря, что крайне важно для корректного взаимодействия с базой данных. Кроме того, на C# реализованы механизмы управления поведением различных элементов пользовательского интерфейса, что облегчает обработку пользовательских действий [8].

HTML стал основным языком разметки для представления структуры веб-приложения. Он является стандартным языком разметки, используемым для отображения документов в веб-браузерах. HTML позволяет структурировать контент инвентарной системы, формируя базу для дальнейшей стилизации и функциональности. Каждая веб-страница должна быть представлена в HTML-формате, так как это обеспечивает совместимость со всеми браузерами и устройствами, что делает веб-приложение доступным для широкого круга пользователей.

CSS был использован в реализации дизайна системы, и в частности, были выбраны фреймворки Bootstrap и W3.CSS. Эти фреймворки позволяют ускорить разработку интерфейса за счет наличия обширной библиотеки компонентов. Дизайн интерфейса приложения выглядит аккуратно и современно благодаря стилям, предоставляемым Bootstrap и W3.CSS. CSS значительно улучшает взаимодействие человека с компьютером, делая веб-страницы более приятными и удобными для восприятия. Упрощение процесса стилизации элементов интерфейса позволяет сосредоточиться на функциональных аспектах приложения.

SQL был выбран в качестве языка, позволяющего взаимодействовать с реляционными базами данных. Язык SQL играет ключевую роль в реализации системы, так как он используется для проектирования и программирования данных, хранящихся в базе. В рамках данного проекта SQL используется для выполнения операций CRUD, связанных с инвентаризацией. C# поддерживает активное соединение и передает запросы в базу данных, что позволяет динамично извлекать необходимую информацию и обеспечивать постоянное обновление данных.

Datatables.net был интегрирован для улучшения функциональности таблиц в веб-приложении. Этот плагин для JavaScript добавляет функции сортировки, разбивки на страницы и фильтрации для HTML-таблиц с минимальными затратами времени на внедрение. Он облегчает работу с базой данных, позволяя пользователям легко находить и сортировать записи, что значительно повышает удобство работы с интерфейсом и обогащает функциональные возможности приложения.

JavaScript был выбран для разработки фронтенда системы из-за его важности для взаимодействия пользователей с веб-страницами. Он предоставляет возможности для динамических действий на стороне клиента, что уменьшает количество запросов к серверу и улучшает производительность. JavaScript используется в приложении для реализации различных клиентских сценариев, а также для визуализации данных с использованием специализированных библиотек. В частности, библиотека jQuery позволяет оптимизировать работу с динамическими элементами интерфейса, упрощая реализацию задач, таких как печать счетов и накладных.

Google Charts был выбран для визуализации данных в приложении, предоставляя интерактивные графические представления информации. Эта служба позволяет пользователям создавать графики на основе данных, предоставленных из базы данных, обеспечивая наглядное отображение результатов продаж. Использование Google Charts помогает аналитически подходить к оценке состояния инвентаря, что позволяет пользователям легко отслеживать и анализировать показатели продаж в удобном графическом формате.

## 4.2 Разработка элементов веб-сайта системы учета и контроля IT-активов предприятия

В рамках данной курсовой работы были разработаны: ***форма авторизации; форма регистрации; главная страница;* форма добавления актива*; форма управления категорий; форма управления брендами; форма управления заказами(заявки).***

*Форма авторизации* (реализацияс использованием ASP.NET Web For)предназначена для обеспечения безопасности доступа к системе и управления пользователями, включающая в себя два основных функционала. Ниже представлена часть кода для открытия соединения с базой данных:

*SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["connStr"].ConnectionString);*

В этом коде происходит подключение к базе данных, параметры которого берутся из конфигурационного файла Web.config, что позволяет обеспечить гибкость в управлении подключениями и настройках базы данных.

При нажатии на кнопку входа (рисунок 4.1) происходит выполнение метода btnLogin\_Click, который проверяет введенные пользователем данные.

## 

**Рисунок 4.1 Форма авторизации**

Листинг аутентификации кода пользователя приведён в приложении В, который реализует следующий алгоритм:

1) Открывает соединение с базой данных.

2) Выполняет SQL-запрос для проверки наличия пользователя с введенными логином и паролем.

3) Если пользователь найден, данные о сессии сохраняются, а пользователя перенаправляют на страницу управления.

4) При неудачной попытке входа выводится сообщение об ошибке.

*Форма регистрации* пользователей для веб-приложения реализована с использованием ASP.NET Web Forms и предназначена для обеспечения безопасности доступа к системе и управления пользователями. Этот интерфейс предоставляет пользователям возможность создавать новые аккаунты, что является важным элементом управления доступом в системе.

Форма регистрации включает в себя два основных функционала: регистрацию нового пользователя и возможность вернуться к форме входа. Эта функциональность реализована в классе Login, который управляет процессами аутентификации и регистрации.

Часть кода для открытия соединения с базой данных:

*SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["connStr"].ConnectionString);*

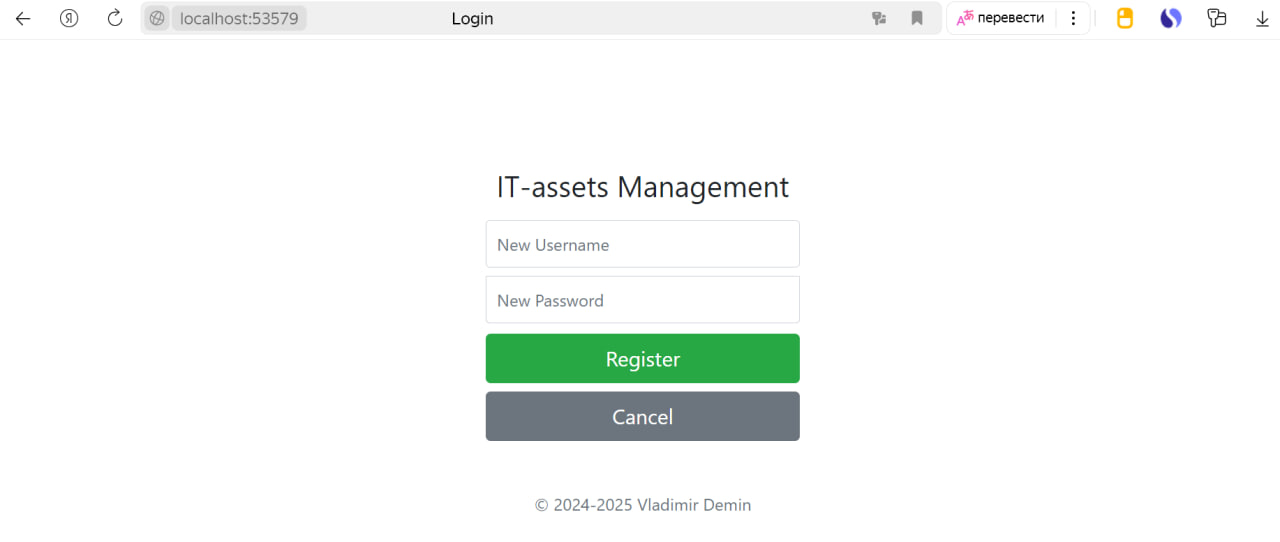
В этом коде происходит открытие подключения к базе данных, параметры которого берутся из конфигурационного файла Web.config. Это позволяет обеспечить гибкость в управлении подключениями и настройками базы данных, что делает приложение более устойчивым к изменениям в инфраструктуре.

Метод btnRegister\_Click(рисунок 4.2) отвечает за регистрацию новых пользователей. Этот код выполняет следующие действия:

1) Открывает подключение к базе данных.

2) Выполняет SQL-запрос для вставки нового пользователя в таблицу, включая его имя, пароль и роль (по умолчанию 'user'). Этот процесс также включает хеширование пароля для повышения безопасности.

3) Уведомляет пользователя об успешной регистрации или неудаче, выводя соответствующие сообщения на экран.



**Рисунок 4.2 Форма регистрации**

*Главная страница* веб-приложения отвечает за управление заказами и взаимодействие с базой данных. В этом разделе подробно описаны ключевые элементы класса im\_order\_Default, который реализует основную логику для обработки различных операций, связанных с заказами.

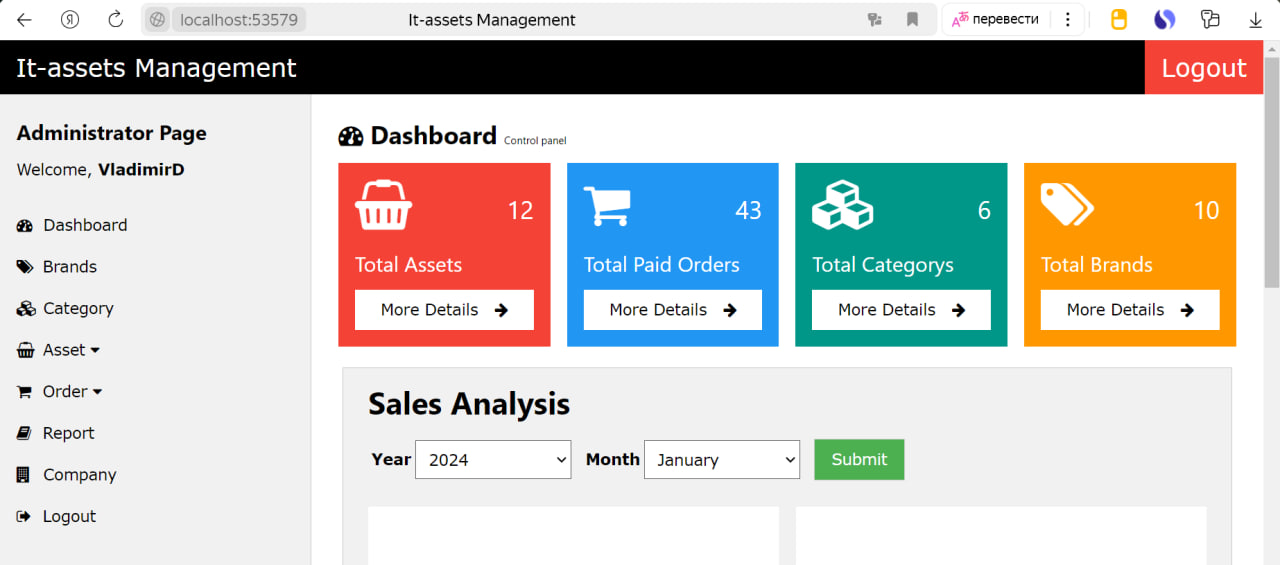
Класс im\_order\_Default наследуется от System.Web.UI.Page, что позволяет ему функционировать как веб-страница и обрабатывать события в рамках приложения.

Обработчик события Page\_Load:

*protected void Page\_Load(object sender, EventArgs e) { }*

Метод Page\_Load вызывается при загрузке страницы. В данном случае он не содержит функциональности, но может быть использован для инициализации данных или выполнения логики, которая зависит от состояния страницы, например, загрузки информации о заказах или обновления интерфейса в зависимости от пользовательских вводов.

Главная страница веб-сайта представлена на рисунке 4.3:



**Рисунок 4.3 Главная страница**

***Основные функции главной страницы:***

*Загрузка и отображение метрик*. При первом обращении к странице (не будучи обновленной) происходит инициализация интерфейса, включающая обновление текстовых меток с информацией о количестве активах, заказов, категорий и брендов. Это осуществляется через метод updateCount, который выполняет SQL-запросы к базе данных.

*Проверка и установка значений выпадающих списков*:

Проверяется, существуют ли значения года и месяца в выпадающих списках (ddlYear и ddlMonth). Если значения находятся, они выбираются, в противном случае устанавливаются значения по умолчанию.

*Получение данных о активах, категориях и брендах*:

Используя методы getProduct, getCategory, и getBrand, форма делает запросы к базе данных для получения информации об активах, категориях и брендах за указанный год и месяц.

Результаты запросов заполняют соответствующие списки и переменные строк (например, product, category, brand), которые затем отображаются на странице.

*Обработка нажатий кнопок*:

Метод btnDateSubmit\_Click обрабатывает событие нажатия на кнопку для смены параметров даты. Пользователь перенаправляется на главную страницу с новыми параметрами в строке запроса.

Метод btnReplenish\_Command обрабатывает команду пополнения запасов, перенаправляя пользователя на страницу управления запасами для конкретного продукта.

*Форма добавления актива* позволяет пользователю вводить данные о новом продукте, такие как название, артикул (SKU), цена, количество, описание, бренд, категория и статус. Рассмотрим ключевые элементы кода для реализации этой функциональности.

Ключевые строки кода:

*SqlCommand cmd = new SqlCommand("INSERT INTO IT\_assets(asset\_name, asset \_sku, asset \_price, asset \_quantity, asset \_desc, asset \_brand, asset \_category, asset \_status) " + "VALUES(@name,@sku,@price,@quantity,@desc,@brand,@category,@status)");*

Эта строка SQL выполняет вставку новой записи в таблицу актива. Параметры @name, @sku, и т.д. заменяются на значения, введенные пользователем в форму.

При нажатии кнопки "Сохранить" (btnSave), данные из формы считываются и передаются в базу данных:

*cmd.Parameters.AddWithValue("name", txtName.Text.Trim());*

Таким образом, заполняется информация о продукте из полей ввода на форме.

После добавления нового актива пользователь переходит к списку всех активов. Этот список может содержать таблицу со всеми активами, где отображаются данные, такие как название, SKU, цена и статус.

Вместо этого форма добавления и редактирования использует информацию об активе из базы данных и позволяет пользователю изменять существующие записи.

Форма используется для обновления существующей информации об активе. При загрузке формы we проверяем, передан ли идентификатор активах с помощью cookie:

*if (Request.Cookies["ProductId"] != null)*

При отправке формы (по нажатию кнопки "Сохранить") данные обновляются:

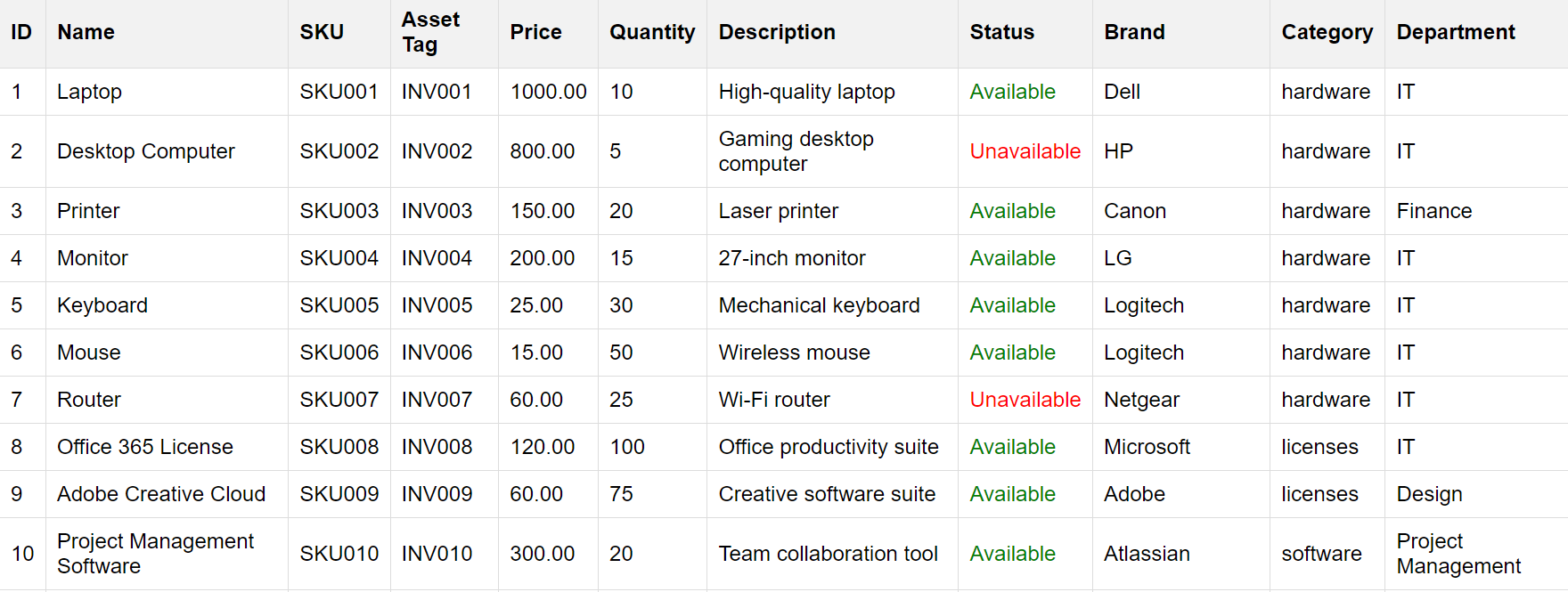
*SqlCommand cmd = new SqlCommand(edit, conn);*

Как при добавлении, так и при редактировании данных, предусмотрена обработка ошибок. Если возникает исключение, оно отображается пользователю, что позволяет понять, что что-то пошло не так:

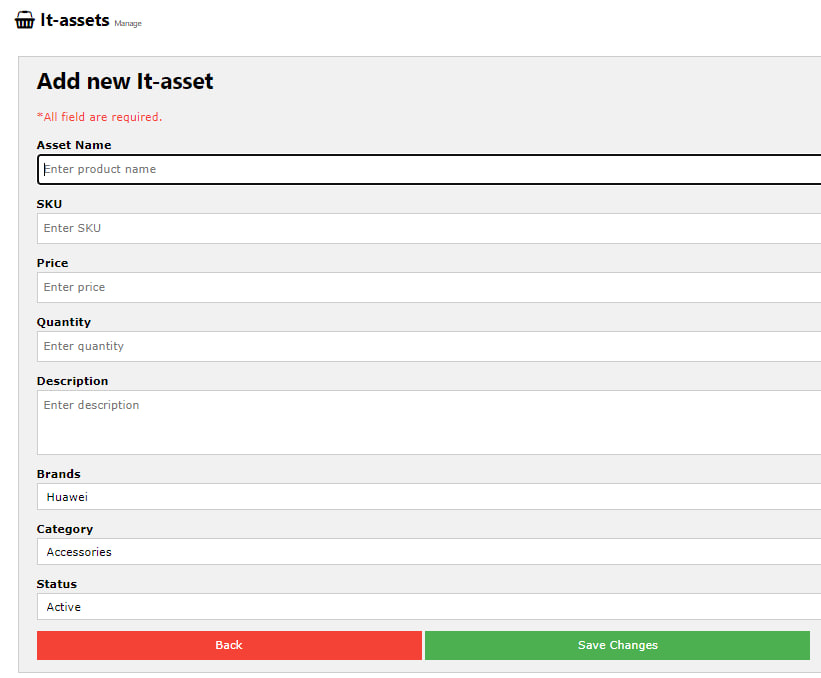
*lblError.Text = "ERROR: " + ex.ToString();*

*lblError.ForeColor = System.Drawing.Color.Red;*

Представления форм добавления и списка активов представлены на рисунках 4.4, 4.5:



**Рисунок 4.4 Список IT-активов**



**Рисунок 4.5 Добавление IT-активов**

*Форма учёта категорий* реализована с использованием ASP.NET Web Forms и предназначена для управления данными о категориях в системе. Этот интерфейс предоставляет пользователям возможность добавлять, обновлять и проверять существующие категории, что является важным элементом управления данными в приложении.

***Основные функции формы учёта категорий:***

1) Регистрация новой категории

2) Обновление существующей категории

Коды и описание функционала:

Регистрация новой категории:

*protected void btnAdd\_Click(object sender, EventArgs e)*

*{*

*divOn();*

*lblHeader.Text = "Add new category";*

*Response.Cookies["OperationId"].Value = "0";*

*txtCategory.Text = "";*

*}*

Эта функция сигнализирует о намерении пользователя добавить новую категорию. При нажатии кнопки "Добавить" устанавливаются необходимые параметры для ввода данных.

Сохранение новой категории:

*string queryCategory = "SELECT \* FROM category WHERE category\_name=@name";*

*if (rf){*

*lblCheck.Visible = true;*

*else{*

*lblCheck.Visible = false;*

*string add = "INSERT INTO category (category\_name, category\_status) VALUES(@name, @status)";}*

Здесь мы сначала проверяем, существует ли уже указанная категория, чтобы избежать дублирования данных. Если категория не найдена, происходит вставка новой записи в базу данных.

Обновление категории:

*string edit = "UPDATE [category] SET category\_name=@name, category\_status=@status WHERE Id=@Id";*

*cmd.Parameters.AddWithValue("Id", Convert.ToInt32(Request.Cookies["categoryId"].Value));*

Если пользователь редактирует существующую категорию, осуществляется поиск по идентификатору, хранящемуся в cookie, и обновляются данные в базе данных.

На рисунках В1 – В5 расположены таблицы с категориями IT-активов.

*Форма учёта брендов* также реализована с использованием ASP.NET Web Forms и предназначена для управления данными о брендах в системе. Она позволяет пользователям добавлять, обновлять и проверять существующие бренды, что является важным элементом управления данными в приложении.

***Основные функции формы учёта брендов***:

1) Регистрация нового бренда

2) Обновление существующего бренда

Регистрация нового бренда:

*protected void btnAdd\_Click(object sender, EventArgs e)*

*{*

*divOn();*

*lblHeader.Text = "Add new brand";*

*Response.Cookies["OperationId"].Value = "0";*

*txtBrand.Text = "";*

*}*

Эта функция аналогична функции добавления категории и инициирует режим добавления нового бренда при нажатии кнопки "Добавить".

Сохранение нового бренда:

*string queryBrand = "SELECT \* FROM brand WHERE brand\_name=@name";*

*if (rf){*

*lblCheck.Visible = true; // Проверка на наличие бренда}*

*else{*

*string add = "INSERT INTO brand (brand\_name, brand\_status) VALUES(@name, @status)";}*

В этом блоке происходит проверка на существование вводимого бренда. Если бренд отсутствует, новый бренд добавляется в базу данных.

Обновление бренда:

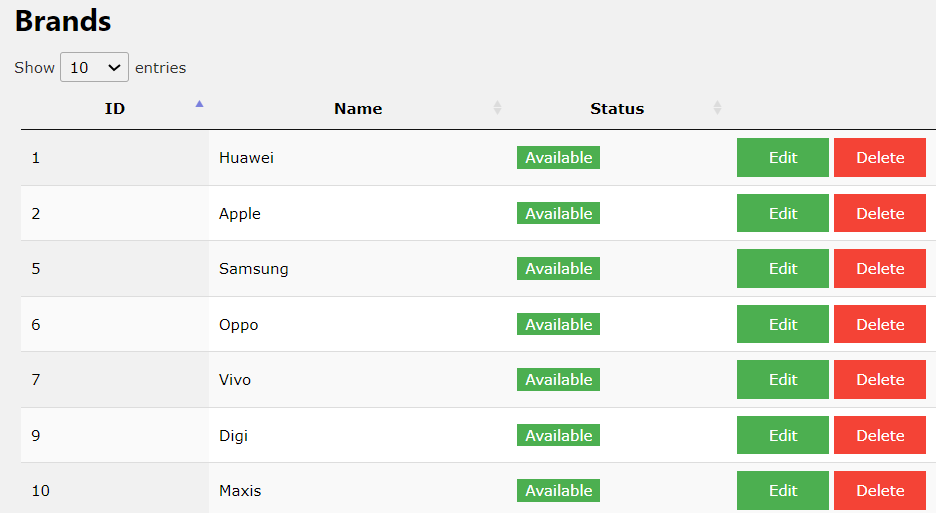
*string edit = "UPDATE [brand] SET brand\_name=@name, brand\_status=@status WHERE Id=@Id";*

*SqlCommand cmd = new SqlCommand(edit, conn);*

*cmd.Parameters.AddWithValue("Id", Convert.ToInt32(Request.Cookies["brandId"].Value));*

Редактирование бренда выполняется аналогично редактированию категории: данные обновляются по идентификатору, который хранится в cookie, что упрощает управление данными.

Форма учёта брендов представлена на рисунке 4.6:



**Рисунок 4.6 Список производителей (брендов активов)**

*Форма управления заказами* представляет собой страницу веб-приложения на ASP.NET, позволяющую пользователям управлять процессом оформления заказов на IT-активы. Основные функции включают добавление продуктов, редактирование и удаление из корзины, а также расчет итоговой суммы заказа.

***Основные элементы управления и их функции***:

1) Подключение к Базе Данных:

2) Инициализация страницы:

В методе Page\_Load происходит проверка на первый загрузку страницы (!IsPostBack), в которой также очищается корзина:

*string deleteCart = "DELETE FROM cart";*

Посредством этого предотвращается накапливание данных в корзине при повторной загрузке страницы.

3) Обновление данных в корзине:

Метод gvProduct\_RowUpdating обрабатывает обновления количества продуктов, позволяя пользователям изменять нужное количество:

*string add = "UPDATE cart SET product\_qty\_need = @qty\_need, product\_total\_price=@price WHERE product\_id = @id";*

Параметризация запроса помогает избежать SQL-инъекций.

4) Удаление продуктов из корзины:

Функция gvProduct\_RowDeleting отвечает за удаление выбранного продукта из корзины:

*string queryBrand = "DELETE FROM cart WHERE product\_id=@id";*

5) Добавление нового продукта в корзину:

Метод btnSelectProduct\_Click добавляет продукт в корзину, проверяя, существует ли он уже:

*string add = "INSERT INTO cart (product\_id,product\_name,product\_qty\_instock,product\_qty\_need,product\_price,product\_total\_price) VALUES(@id,@name,@qty\_instock,@qty\_need,@price,@total)";*

6) Расчет итоговой суммы:

Метод calcultateSum вычисляет общую сумму для заказа, включая распределение сервисного сбора и налогов:

*netTotal = gross + service;*

7) Сохранение заказа:

btnSave\_Click отвечает за сохранение информации о заказе. Он может добавлять информацию о новом клиенте, если клиент еще не зарегистрирован:

*string add = "INSERT INTO customer (customer\_name,customer\_phone,customer\_address) VALUES(@name,@phone,@address)";*

8) Обновление информации о продуктах:

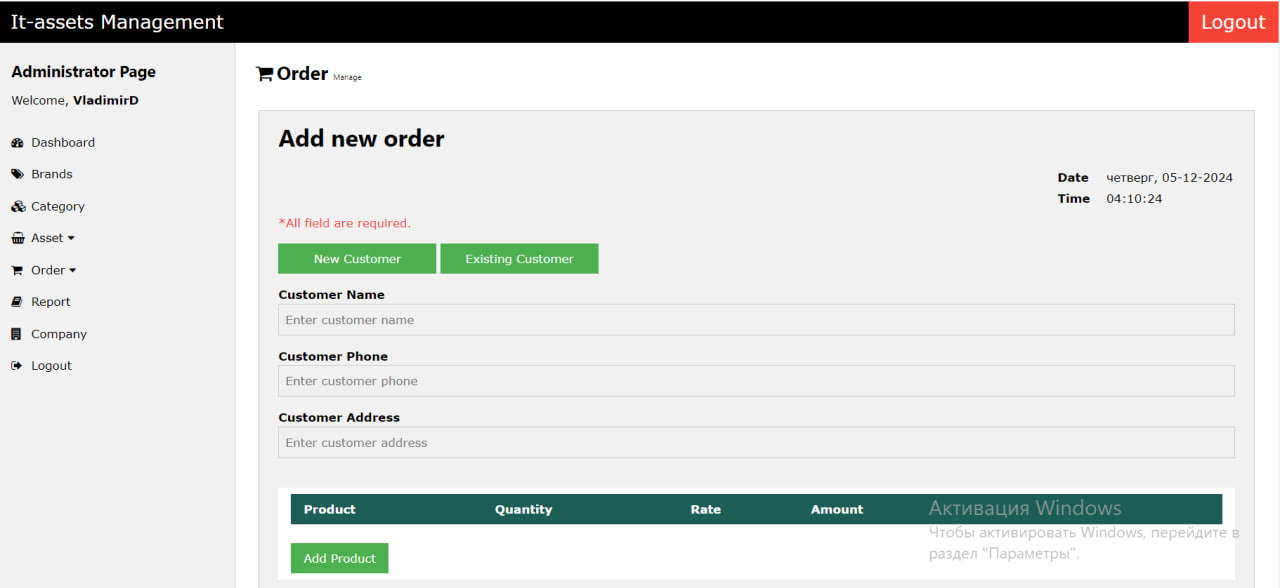
После оформления заказа обновляются оставшиеся количества продуктов:

*string update = "UPDATE product SET product\_quantity = @qty WHERE Id = @id";*

9) Визуальные Элементы:

GridView (gvProduct). Отображает текущий список продуктов в корзине, с возможностью редактирования и удаления.

Вид формы управления заказами представлен на рисунке 4.7:



**Рисунок 4.7 Добавление нового заказа**

## 4.3 Выводы

Таким образом, определим сильные и слабые стороны разработанной системы учёта и контроля IT-активов предприятия.

***Сильные стороны системы учёта и контроля IT-активов:***

1. Защита данных пользователей. Все пользователи обязаны входить в систему, используя свои имя пользователя и пароль. Это предотвращает несанкционированный доступ к системе и обеспечивает защиту данных инвентаризации, исключая возможность использования системы неавторизованными лицами.

2. Удобная навигация. Система имеет ясную и понятную навигационную панель. Все функции разбиты на несколько разделов, таких как бренд, категория, продукт, заказ и т.д. Это делает все функции доступными и понятными для пользователей, что помогает избежать путаницы в их использовании.

3. Гибкое управление данными (CRUD операции). Система поддерживает операции создания, чтения, обновления и удаления (CRUD), что позволяет пользователям гибко управлять данными. Пользователи имеют возможность легко добавлять новые записи, просматривать существующие, изменять их или удалять.

4. Поиск и сортировка данных. Все записи в системе имеют функции поиска и сортировки. Функция поиска помогает пользователям быстро находить необходимую информацию, а сортировка по возрастанию или убыванию позволяет удобно организовывать данные, что существенно экономит время на поиски.

***Слабые стороны системы учёта и контроля IT-активов:***

1. Негибкое соединение с базой данных. Взаимосвязь между базой данных и системой построена слишком жестко. Все соединения осуществляются напрямую между фронтендом и бэкендом, что создает некоторые ограничения. Если возникает необходимость переноса бэкенда на другую машину, потребуется изменить код или полностью заново настраивать бэкенд.

2. Зависимость от сетевого подключения. Для работы системы необходимо установленное сетевое соединение. Без доступа к сети анализ не может быть выполнен, и не предоставляются визуализации данных (например, графики продаж). Однако другие функции системы остаются доступными и могут использоваться даже при локальной установке.

# ЗАКЛЮЧЕНИЕ

# 

# В процессе выполнения курсовой работы были получены следующие результаты:

1. Проведен обзор и анализ современных систем управления IT-активами предприятий, в результате которого выявлены их преимущества и недостатки существующих на рынке ITAM.

2. Проведен анализ и выполнено функциональное моделирование бизнес-процессов учета и контроля IT-активов на предприятия.

3. Разработка и проектирование БД «Учёт и контроль IT-активов» системы управления IT-активами предприятия.

4. Выполнена программная реализация БД на платформе Microsoft SQL Server.

5. Разработан пользовательский интерфейс к БД с использованием технологий Microsoft Visual Studio .NET и ASP.NET.

6. Разработаны и реализованы программные модули авторизации, регистрации, управления активами, категориями и брендами, а также интерфейс для обработки заявок.

Таким образом, в результате выполнения курсовой работы была разработана система учета и контроля IT-активов предприятия, включающая интуитивно понятный пользовательский интерфейс, надежную базу данных и функциональные модули для управления активами, заказами, брендами, отчетами и корпоративной информацией.

Разработанная ITAM может быть применена на предприятиях, занимающихся управлением IT-активами, для автоматизации учета, оптимизации бизнес-процессов, повышения прозрачности и эффективности управления ресурсами, а также обеспечения информационной безопасности и минимизации затрат.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. - 3-е изд., стер. - Санкт-Петербург: Лань, 2022. - 244 с.

2. Системы управления базами данных: учебное пособие / Р.Э. Мамедли – Нижневартовск: изд. Нижневартовского государственного университета, 2021. – 214 с.

3. Уорд, Б. Инновации SQL Server 2019. Использование технологий больших данных и машинного обучения / Боб Уорд. – М.: ДМК Пресс, 2020. - 407 с.

4. Кэмпбелл, Л. Базы данных. Инжиниринг надежности = Database

Reliability Engineering / Л. Кэмпбелл, Ч. Мейджорс; [пер. с англ. Е. Сандицкая]. - Санкт-Петербург [и др.]: Питер, 2020. - 303 с. [Электронный ресурс] - https://ibooks.ru/bookshelf/367974

5. Грофф, Джеймс Р., Вайнберг, Пол Н., Оппель, Эндрю Дж. SQL: полное руководство, 3-е изд. - СПб.: ООО "Диалектика", 2019. - 960 с.

6. Ицик Бен-Ган, Microsoft SQL Server 2012. Основы T-SQL. - М: Эксмо, 2015, 400 с.

7. Технологии построения моделей бизнес-процессов / Нарваткина Н. С. Екатеринбург 2020

8. Разработка программного обеспечения: проектирование, конструирование и внедрение: методическое пособие / И.Н. Тонкович, А.В. Шелест. УО «Белорусский государственный университет информатики радиоэлектроники», 2023

9. Документация по Microsoft SQL. [Электронный ресурс] - <https://learn.microsoft.com/ru-ru/sql/sqlserver/?view=sql-server-ver15>.

10. Обучающие ресурсы по SQL [Электронный ресурс] - <https://learn.microsoft.com/ru-ru/sql/sql-server/educational-sql-resources?view=sql-server-ver16>.

# ПРИЛОЖЕНИЕ А

Таблица А1 – Сравнительная таблица ITAM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Критерии** | Asset Panda | ManageEngine AssetExplorer | SolarWinds Service Desk | Pulseway | LANsweeper |
| **Тип решения** | Облачное управление | Облачное браузерное управление | Интегрированное решение (облачное и локальное) | Облачное решение с мобильным управлением | Облачное решение для управления |
| **Гибкость и настраивае-мость** | Высокая, может адаптиро-ваться под потребности | Высокая, может создавать собственные категории | Ограниченная настраива-емость в облачной версии | Высокая, интуитивно понятный интерфейс | Ограничен-ная история программ-ного обеспечения |
| **Методы добавления активов** | Вручную / через мобильное приложения | Автоматически / вручную / мобильное приложение | Ручное добавление/ загрузка через CSV/ WMI (локально) | Автоматически / вручную / мобильное приложение | Автоматически / вручную / мобильное приложение |
| **Поддержка мобильных устройств** | Android и iOS | Android и iOS | Да | Да | Android и iOS |
| **Сканирование штрихкодов** | Да | Да | Нет | Да | Да |
| **Прикрепление мультимедийных файлов** | Да | Да, | Нет | Да | Да |

Продолжение таблицы А1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Отчеты** | Настраива-емые, детализиро-ванные | Полные отчеты по лицензиям и использова-нию ПО | Простой модуль отчетности, автоматиза-ция отчетов | Настраива-емые шаблоны отчетов | Встроенная отчетность |
| **GPS-отслеживание** | Да | Нет, но поддержива-ется отслеживание активов в реальном времени | Нет | Нет | Нет, но поддержива-ется отслеживание активов |
| **Интеграция с другими системами** | Да(но через API, может потребовать программирования) | Да | Да | Да | Да |
| **Обучение пользователей** | Необходимо, доступ к обучающим материалам | Необходимо, сложный пользовательский интерфейс | Необходимо, доступ к обучающим материалам | Необходимо, доступ к обучающим материалам | Необходимо, доступ к обучающим материалам |
| **Клиентская поддержка** | Высокое качество, оперативная реакция | Доступная, но может быть сложной из-за интерфейса | Высокое качество, оперативная реакция | Высокое качество, оперативная реакция | Доступная, но может быть сложной из-за интерфейса |
| **Управление лицензиями** | Нет автоматиче-ского управления, ручное управление | Полное управление лицензиями и ПО, автомати-  ческое отслеживание | Нет | Нет | Полное управление лицензиями и ПО, автомати-ческое отслежива-ние |
| **Цена** | $1500 в год (до 500 активов) | $795 за 250 активов | Гибкая модель, начальная стоимость за место | Гибкая цена, зависит от устройств | бизнес-план с фиксирован-ной стоимостью |

Продолжение таблицы А1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Пробная версия** | 14-дневная пробная версия | 30-дневная пробная версия | 14-дневная пробная версия | 14-дневная пробная версия | Бесплатно для 100 активов |
| **Оптимально для** | Компаний с ограниченным бюджетом, нуждающихся в гибком решении | Корпорации с большими потребностями | ИТ-компаний и MSP, ищущих простое доступное решение | Средних и крупных компаний с большим количеством устройств | Малых, средних и крупных компаний с недорогим решением |

# ПРИЛОЖЕНИЕ Б

Таблица Б1 – Схема отношения СЧЕТЫ (Bill)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | B\_ID | int(4) | первичный ключ |
| Идентификатор клиента | B\_CUSTOMER\_ID | int(4) | внешний ключ (к Customers) |
| Дата и время | B\_DATETIME | datetime | обязательное поле |
| Валовая сумма | B\_GROSS | decimal(10, 2) | обязательное поле |
| Сумма услуги | B\_SERVICE | decimal(10, 2) | обязательное поле |
| Чистая сумма | B\_NET | decimal(10, 2) | обязательное поле |
| Скидка | B\_DISCOUNT | decimal(10, 2) | обязательное поле |
| Идентификатор пользователя | B\_USER\_ID | int(4) | внешний ключ (к User) |

Таблица Б2 – Схема отношения ЭЛЕМЕНТЫ СЧЕТА (Bill Item)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | BI\_ID | int(4) | первичный ключ |
| Идентификатор счета | BI\_BILL\_ID | int(4) | внешний ключ (к Bill) |
| Идентификатор IT-актива | BI\_ASSET\_ID | int(4) | внешний ключ (к IT-Assets) |
| Необходимое количество | BI\_ASSET\_QTY | int(4) | обязательное поле |
| Итоговая сумма | BI\_TOTAL | decimal(10, 2) | обязательное поле |

Таблица Б3 – Схема отношения БРЕНДЫ (Brand)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | BR\_ID | int(4) | первичный ключ |
| Название бренда | BR\_NAME | nvarchar(50) | обязательное поле |
| Статус бренда | BR\_STATUS | int(4) | обязательное поле |

Таблица Б4 – Схема отношения ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ (Software)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | SW\_ID | int(4) | первичный ключ |
| Название ПО | SW\_NAME | nvarchar(100) | обязательное поле |
| Версия | SW\_VERSION | nvarchar(50) | обязательное поле |
| Тип ПО | SW\_TYPE | nvarchar(100) | обязательное поле |
| Дата установки | SW\_INSTALL\_DATE | date(3) | обязательное поле |

Таблица Б5 – Схема отношения НОСИТЕЛИ (Media)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | M\_ID | int(4) | первичный ключ |
| Тип носителя | M\_TYPE | nvarchar(50) | обязательное поле |
| Описание носителя | M\_DESCRIPTION | nvarchar(255) |  |
| Вместимость | M\_CAPACITY | nvarchar(50) |  |
| Цифровой носитель | M\_IS\_DIGITAL | bit(1) | обязательное поле |

Таблица Б6 – Схема отношения ИТ-ОБОРУДОВАНИЕ (Hardware)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | HW\_ID | int(4) | первичный ключ |
| Название оборудования | HW\_NAME | nvarchar(100) | обязательное поле |
| Тип оборудования | HW\_TYPE | nvarchar(50) | обязательное поле |
| Производитель | HW\_MANUFACTURER | nvarchar(100) |  |
| Дата покупки | HW\_PURCHASE\_DATE | date(3) |  |
| Срок гарантии | HW\_WARRANTY\_EXP | date(3) |  |

Таблица Б7 – Схема отношения ЛИЦЕНЗИИ (License)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | L\_ID | int(4) | первичный ключ |

Продолжение таблицы Б7

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ лицензии | L\_KEY | nvarchar(100) | обязательное поле |
| Дата подтверждения | L\_CONFIRM\_DATE | date(3) |  |
| Срок действия | L\_VALID\_UNTIL | date(3) |  |

Таблица Б8 – Схема отношения КОНТРАКТЫ (Contract)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | C\_ID | int(4) | первичный ключ |
| Номер контракта | C\_NUMBER | nvarchar(50) | обязательное поле |
| Дата начала | C\_START\_DATE | date(3) |  |
| Дата окончания | C\_END\_DATE | date(3) |  |
| Детали контракта | C\_DETAILS | nvarchar(255) |  |
| Поставщик | C\_SUPPLIER | nvarchar(100) |  |

Таблица Б9 – Схема отношения КАТЕГОРИИ (Category)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | CAT\_ID | int(4) | первичный ключ |
| Название категории | CAT\_NAME | nvarchar(50) | обязательное поле |
| Статус категории | CAT\_STATUS | int(4) | обязательное поле |
| Идентификатор ПО | CAT\_SW\_ID | int(4) | внешний ключ (к Software) |
| Идентификатор носителя | CAT\_MEDIA\_ID | int(4) | внешний ключ (к Media) |
| Идентификатор оборудования | CAT\_HW\_ID | int(4) | внешний ключ (к Hardware) |
| Идентификатор лицензии | CAT\_L\_ID | int(4) | внешний ключ (к License) |
| Идентификатор контракта | CAT\_C\_ID | int(4) | внешний ключ (к Contract) |

Таблица Б10 – Схема отношения НА СКЛАДЕ (Instock)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | IS\_ID | int(4) | первичный ключ |
| Идентификатор IT-актива | IS\_ASSET\_ID | int(4) | внешний ключ (к IT-Assets) |
| Количество новое | IS\_QTY\_NEW | int(4) | обязательное поле |
| Цена покупки | IS\_PURCHASE\_PRICE | decimal(10, 2) | обязательное поле |
| Дата и время | IS\_DATETIME | datetime(3) | обязательное поле |
| Примечание | IS\_REMARK | nvarchar(200) |  |

Таблица Б11 – Схема отношения ИТ-АКТИВЫ (IT-Assets)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | IA\_ID | int(4) | первичный ключ |
| Название IT-актива | IA\_NAME | nvarchar(255) | обязательное поле |
| SKU IT-актива | IA\_SKU | nvarchar(255) | обязательное поле |
| Инвентарный номер | IA\_INVENTORY | nvarchar(255) |  |
| Цена IT-актива | IA\_PRICE | decimal(10, 2) | обязательное поле |
| Количество IT-активов | IA\_QUANTITY | int(4) | обязательное поле |
| Описание IT-актива | IA\_DESC | nvarchar(255) |  |
| Идентификатор бренда | IA\_BR\_ID | int(4) | внешний ключ (к Brand) |
| Идентификатор категории | IA\_CAT\_ID | int(4) | внешний ключ (к Category) |
| Статус IT-актива | IA\_STATUS | int(4) | обязательное поле |
| Идентификатор отдела | IA\_DEP\_ID | int(4) | внешний ключ (к Department) |

Таблица Б12 – Схема отношения ПОЛЬЗОВАТЕЛИ (User)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | U\_ID | int(4) | первичный ключ |

Продолжение таблицы Б12

|  |  |  |  |
| --- | --- | --- | --- |
| Имя пользователя | U\_USERNAME | nvarchar(20) | обязательное поле |
| Пароль | U\_PASSWORD | nvarchar(30) | обязательное поле |
| Роль | U\_ROLE | nvarchar(30) | обязательное поле |
| Телефон клиента | U\_PHONE | nvarchar(50) |  |
| Идентификатор отдела | U\_DEP\_ID | int(4) | внешний ключ (к Department) |

Таблица Б13 – Схема отношения ОТДЕЛЫ (Department)

| Содержание поля | Имя поля | Тип, длина | Примечания |
| --- | --- | --- | --- |
| Идентификатор | D\_ID | int(4) | первичный ключ |
| Название отдела | D\_NAME | nvarchar(100) | обязательное поле |
| Руководитель отдела | D\_HEAD\_ID | int(4) | внешний ключ (к User) |
| Количество сотрудников | D\_EMP\_COUNT | int(4) | обязательное поле |

Листинг Б2 – Заполнение таблиц БД «Учёт и контроль IT-активов»

-- Заполнение таблицы User

INSERT INTO [User ] (username, password, role, customer\_phone, department\_id)

VALUES

('admin', 'password123', 'administrator', '123-456-7890', 1),

('jdoe', 'mypassword', 'user', '098-765-4321', 2),

('asmith', 'passw0rd', 'user', '555-555-5555', 1),

('bread', 'securepass', 'user', '666-777-8888', 3),

('mclay', 'pass1234', 'user', '444-222-1111', 4);

-- Заполнение таблицы Department

INSERT INTO Department (department\_name, department\_head, number\_employees)

VALUES

('IT', 1, 10),

('Finance', 2, 5),

('Sales', 3, 8),

('HR', 4, 7),

('Marketing', 5, 6);

-- Заполнение таблицы Bill

INSERT INTO Bill (customer\_id, date\_time, gross\_amount, service\_amount, net\_amount, discount, user\_id)

VALUES

(1, GETDATE(), 1000.00, 200.00, 800.00, 50.00, 1),

(2, GETDATE(), 500.00, 100.00, 400.00, 30.00, 2),

(3, GETDATE(), 1500.00, 300.00, 1200.00, 50.00, 3),

(4, GETDATE(), 750.00, 150.00, 600.00, 40.00, 4),

(5, GETDATE(), 2000.00, 400.00, 1600.00, 100.00, 5);

-- Заполнение таблицы Bill Item

INSERT INTO Bill\_Item (bill\_id, asset\_id, asset\_qty\_need, asset\_total)

VALUES

(1, 1, 2, 200.00),

(2, 2, 1, 100.00),

(3, 1, 3, 300.00),

(4, 3, 1, 250.00),

(5, 4, 2, 500.00);

-- Заполнение таблицы Brand

INSERT INTO Brand (brand\_name, brand\_status)

VALUES

('Brand A', 1),

('Brand B', 1),

('Brand C', 1),

('Brand D', 1),

('Brand E', 1);

-- Заполнение таблицы Software

INSERT INTO Software (software\_name, version, software\_type, installation\_date)

VALUES

('Software X', '1.0', 'Executable', '2023-01-01'),

('Software Y', '2.0', 'Web App', '2023-01-15'),

('Software Z', '3.1', 'Desktop App', '2023-02-01'),

('Software A', '4.0', 'Mobile App', '2023-02-10'),

('Software B', '5.0', 'Cloud', '2023-03-01');

-- Заполнение таблицы Media

INSERT INTO Media (media\_type, media\_description, capacity, is\_digital)

VALUES

('Digital', 'Cloud storage', '1 TB', 1),

('Physical', 'USB Drive', '64 GB', 0),

('Physical', 'DVD', '4.7 GB', 0),

('Digital', 'External HDD', '2 TB', 1),

('Digital', 'SSD', '500 GB', 1);

-- Заполнение таблицы Hardware

INSERT INTO Hardware (hardware\_name, hardware\_type, manufacturer, purchase\_date, warranty\_expiry)

VALUES

('Server 1', 'Physical', 'Dell', '2023-01-10', '2024-01-10'),

('Laptop 1', 'Physical', 'HP', '2023-01-15', '2024-01-15'),

('Desktop 1', 'Physical', 'Lenovo', '2023-01-20', '2024-01-20'),

('Printer 1', 'Physical', 'Canon', '2023-01-25', '2024-01-25'),

('Monitor 1', 'Physical', 'Samsung', '2023-02-15', '2024-02-15');

-- Заполнение таблицы License

INSERT INTO License (license\_key, confirmation\_date, valid\_until)

VALUES

('XYZ1234567', '2023-01-01', '2024-01-01'),

('ABC9876543', '2023-01-02', '2024-01-02'),

('LMN1357924', '2023-01-05', '2024-01-05'),

('QRS2468135', '2023-01-10', '2024-01-10'),

('TUV8642095', '2023-01-15', '2024-01-15');

-- Заполнение таблицы Contract

INSERT INTO Contract (contract\_number, start\_date, end\_date, details, supplier)

VALUES

('C-2023-1', '2023-01-01', '2024-01-01', 'Annual support contract', 'Supplier Co.'),

('C-2023-2', '2023-01-15', '2024-01-15', 'Monthly service agreement', 'Tech Services Inc.'),

('C-2023-3', '2023-02-01', '2024-02-01', 'Annual software licensing agreement', 'Software Corp.'),

('C-2023-4', '2023-02-15', '2024-02-15', 'Hardware maintenance contract', 'Hardware Solutions LLC'),

('C-2023-5', '2023-03-01', '2024-03-01', 'Cloud storage agreement', 'Cloud Storage Inc.');

-- Заполнение таблицы Category

INSERT INTO Category (category\_name, category\_status, software\_id, media\_id, hardware\_id, license\_id, contract\_id)

VALUES

('Software', 1, 1, NULL, NULL, 1, 1),

('Media', 1, NULL, 1, NULL, NULL, NULL),

('Hardware', 1, NULL, NULL, 1, NULL, NULL),

('Digital', 1, 2, NULL, NULL, 1, 2),

('Physical', 1, NULL, 2, NULL, NULL, 3);

-- Заполнение таблицы IT-Assets

INSERT INTO IT\_Assets (asset\_name, asset\_sku, asset\_inventory, asset\_price, asset\_quantity, asset\_desc, asset\_brand, asset\_category, asset\_status, department\_id)

VALUES

('Laptop', 'SKU-001', 'INV-001', 500.00, 5, 'Dell Latitude', 1, 1, 1, 1),

('Desktop', 'SKU-002', 'INV-002', 700.00, 3, 'HP Elite', 2, 1, 1, 2),

('Printer', 'SKU-003', 'INV-003', 300.00, 2, 'Brother Laser', 3, 2, 1, 3),

('Router', 'SKU-004', 'INV-004', 150.00, 10, 'Netgear AC1200', 4, 3, 1, 4),

('Monitor', 'SKU-005', 'INV-005', 200.00, 7, 'LG 24"', 5, 4, 1, 5);

-- Заполнение таблицы Instock

INSERT INTO Instock (asset\_id, qty\_new, purchase\_price, date\_time, remark)

VALUES

(1, 10, 450.00, GETDATE(), 'New shipment arrived'),

(2, 5, 650.00, GETDATE(), 'New shipment arrived'),

(3, 3, 300.00, GETDATE(), 'New shipment arrived'),

(4, 15, 120.00, GETDATE(), 'New shipment arrived')

# ПРИЛОЖЕНИЕ В

Листинг В.1 – Аутентификация пользователя

*protected void btnLogin\_Click(object sender, EventArgs e)*

*{*

*conn.Open();*

*SqlCommand cmd = new SqlCommand("SELECT \* FROM [user] WHERE username=@user AND password=@pass", conn);*

*cmd.Parameters.AddWithValue("user", txtUsrname.Text);*

*cmd.Parameters.AddWithValue("pass", txtPass.Text);*

*SqlDataReader dr = cmd.ExecuteReader();*

*if (dr.HasRows)*

*{*

*while (dr.Read())*

*{*

*Session["username"] = dr["username"].ToString();*

*Session["Id"] = dr["Id"].ToString();*

*Response.Redirect("/im/dashboard.aspx");*

*}*

*}*

*else*

*{*

*Response.Write("<script>alert('EMAIL AND PASSWORD IS NOT FOUND!');</script>");*

*}*

*conn.Close();*

*}*

Листинг В.2 – Аутентификация пользователя

*protected void btnRegister\_Click(object sender, EventArgs e)*

*{*

*conn.Open();*

*SqlCommand insertCmd = new SqlCommand("INSERT INTO [user] (username, password, role) VALUES (@user, @pass, @role)", conn);*

*insertCmd.Parameters.AddWithValue("user", txtRegUsrname.Text);*

*insertCmd.Parameters.AddWithValue("pass", txtRegPass.Text);*

*insertCmd.Parameters.AddWithValue("role", "user");*

*int rowsAffected = insertCmd.ExecuteNonQuery();*

*if (rowsAffected > 0)*

*{*

*Response.Write("<script>alert('Registration successful!');</script>");*

*pnlLogin.Visible = true;*

*pnlRegister.Visible = false;*

*}*

*else*

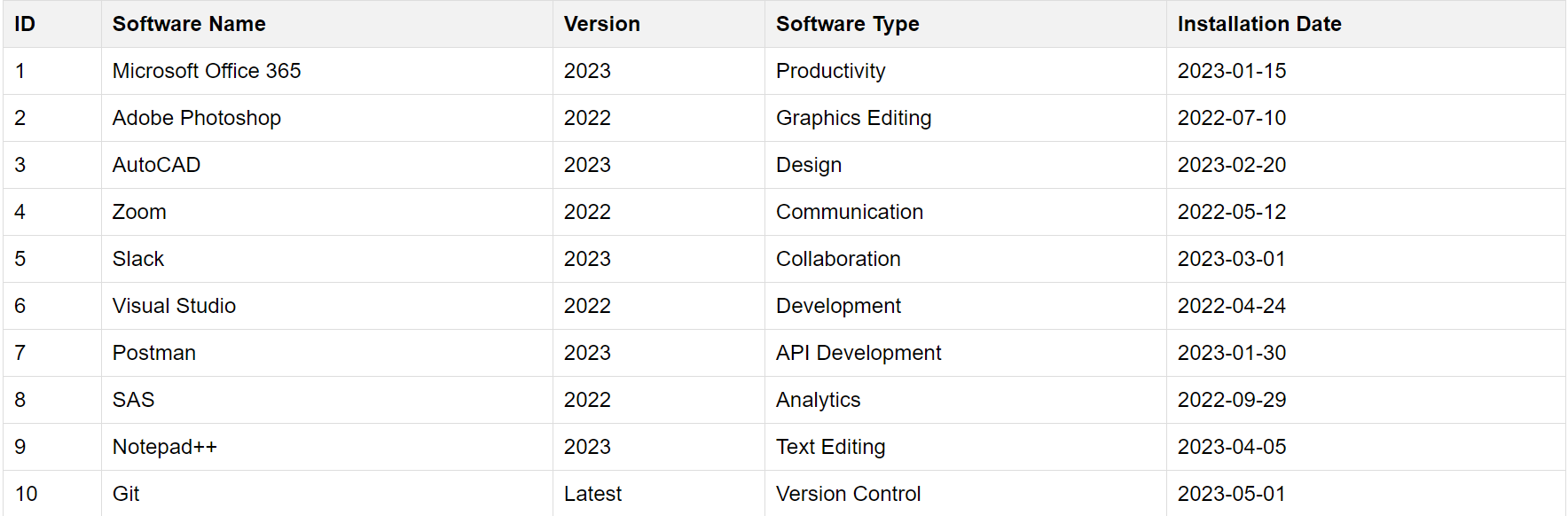
*{*

*Response.Write("<script>alert('Registration failed!');</script>");*

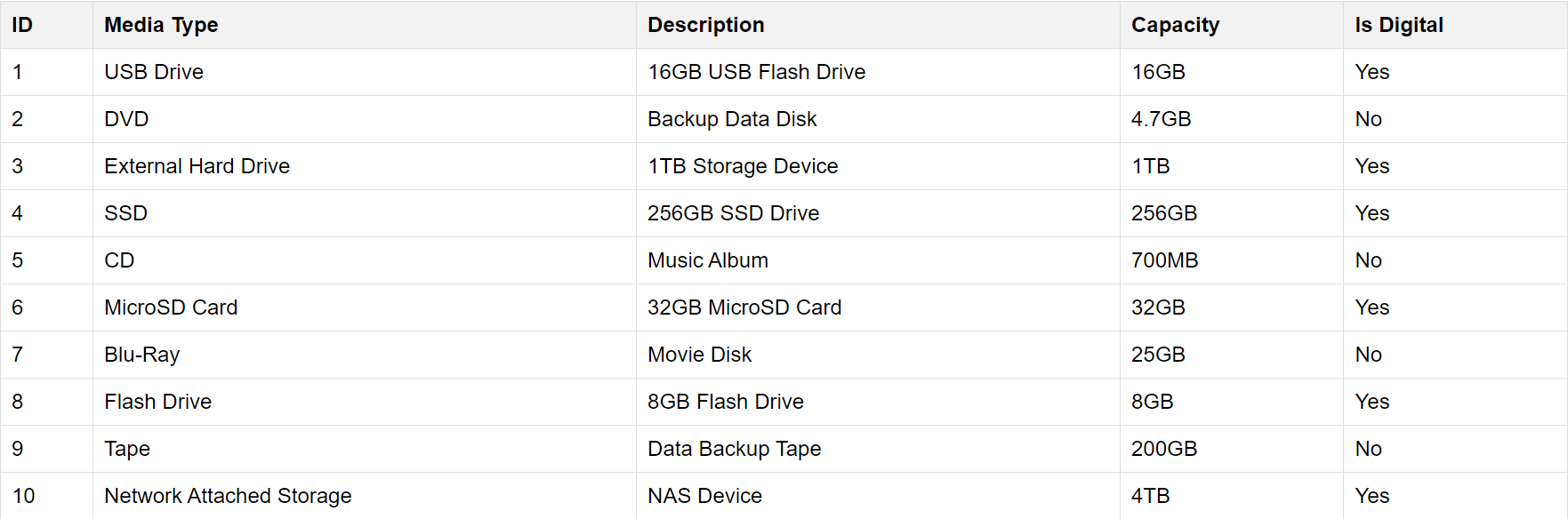
*}*

*conn.Close();*

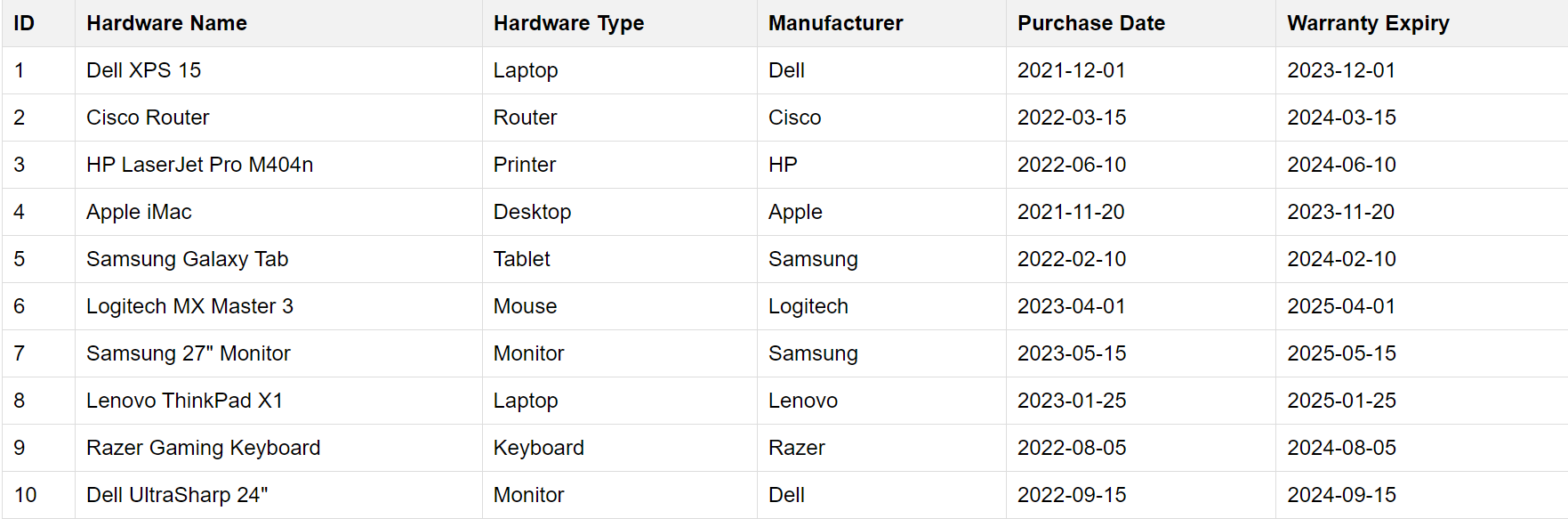
*}*



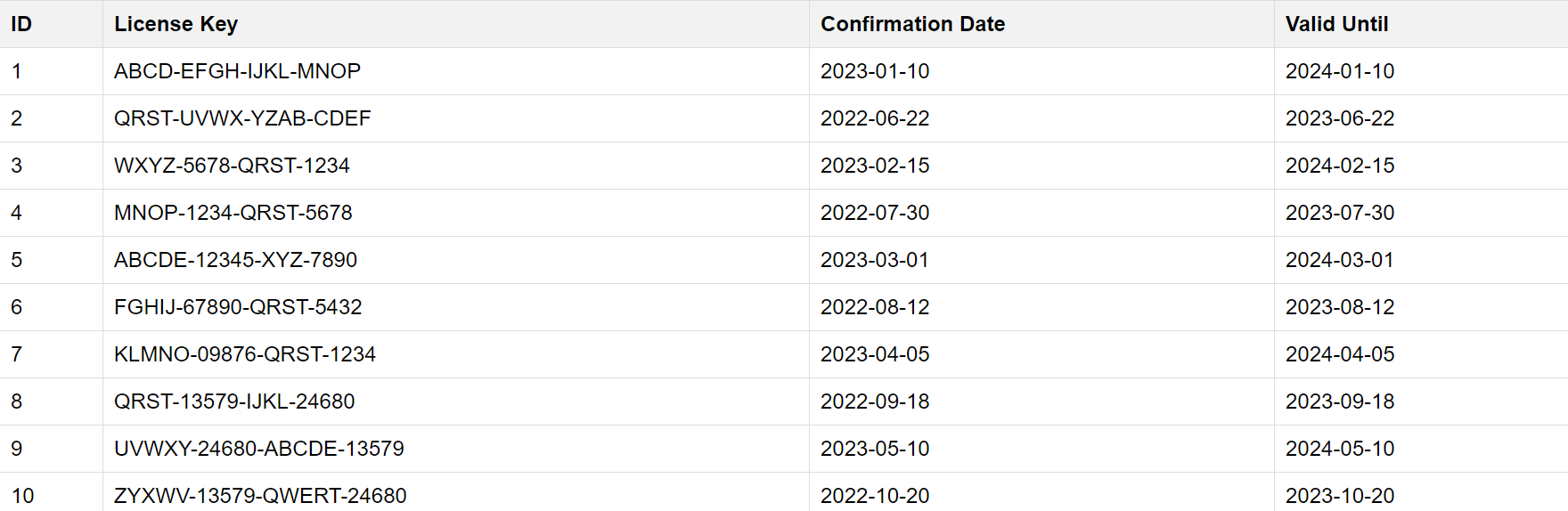
**Рисунок В1 - Список доступного программного обеспечения**



**Рисунок В2 - Список доступных носителей**



**Рисунок В3 - Список доступного оборудования**



**Рисунок В4 - Список доступных лицензий**



**Рисунок В5 - Список доступных контрактов**