

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра информационных систем управления

ДЕМИН Владимир Олегович

СИСТЕМА УЧЁТА И КОНТРОЛЯ IT-АКТИВОВ ПРЕДПРИЯТИЯ

Дипломная работа

Научный руководитель:
старший преподаватель
Е.С.Малашенко

Допущена к защите

« ____ » _____ 20 ____ г.

Заведующий кафедрой информационных систем
управления доктор технических наук, профессор, доцент
А.М. Недзьведь

Минск, 2025

СОДЕРЖАНИЕ

РЕФЕРАТ	3
ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ	6
ВВЕДЕНИЕ	7
ГЛАВА 1 СОСТОЯНИЕ ПРОБЛЕМЫ, ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ	9
1.1 Принципы и подходы к управлению ИТ-активами предприятия	9
1.2 Обзор и анализ современных ИТАМ.....	11
1.3 Назначение и основные требования системы учета и контроля ИТ-активов.....	12
1.4 Цель и задачи исследования.....	14
1.5 Выводы	15
ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ УЧЁТА И КОНТРОЛЯ ИТ-АКТИВОВ	16
2.1 Анализ деятельности предприятия и проблем в сфере управления ИТ-активами.	16
2.2 Моделирование бизнес-процессов учета и контроля ИТ-активов	17
2.3 Разработка и проектирование информационной модели системы.....	23
2.4 Выводы	30
ГЛАВА 3 РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ УЧЕТА И КОНТРОЛЯ ИТ-АКТИВОВ	31
3.1 Выбор технологического стека для программной реализации системы учета и контроля ИТ- активов	31
3.2 Проектирование архитектуры веб-приложения системы.....	32
3.3 Проектирование и разработка пользовательского интерфейса и элементов веб-приложения.....	38
3.4 Описание действий пользователей системы учета и контроля ИТ-активов.....	54
3.5 Выводы	57
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	59
ПРИЛОЖЕНИЕ А	60
ПРИЛОЖЕНИЕ Б	62
ПРИЛОЖЕНИЕ В	69
ПРИЛОЖЕНИЕ Г	79

РЕФЕРАТ

Дипломная работа содержит: 58 страниц, 25 рисунков, 12 использованных литературных источников, 3 таблицы, 4 приложения.

Ключевые слова – ИТ-АКТИВЫ, УПРАВЛЕНИЕ АКТИВАМИ, ИТАМ, БАЗА ДАННЫХ, ВЕБ-РАЗРАБОТКА, SQL, C#, JAVASCRIPT, SQL, HTML, ASP.NET.

Объект исследования – процесс управления ИТ-активами предприятия.

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

Цель работы – разработка системы учета и контроля ИТ-активов предприятия и её программного обеспечения для автоматизации управления ИТ-активами, прозрачного учета, повышения эффективности бизнес-процессов и минимизации ошибок за счет интуитивно понятного интерфейса, надежной базы данных и функциональных модулей.

Методы исследования – анализ литературных источников по ИТАМ-системам и управлению активами, сравнительный анализ существующих решений, функциональное моделирование бизнес-процессов, проектирование реляционной базы данных с нормализацией, Agile (SCRUM) для поэтапной веб-разработки, тестирование системы на реальных сценариях.

Полученные результаты и их новизна – разработана система учета ИТ-активов с инновационными аспектами: интеграция модулей управления активами, заявками и аналитики в единый интерфейс; гибридная архитектура с использованием ASP.NET для серверной части и JavaScript-библиотек, таких как Datables.net и Google Charts.

Достоверность материалов и результатов дипломной работы подтверждается использованием промышленных стандартов (C#, ASP.NET, SQL), валидацией через сравнение с требованиями рынка ИТАМ-систем, защитой от уязвимостей через параметризованные SQL-запросы.

Область возможного практического применения – малый и средний бизнес, управляющий ИТ-инфраструктурой: ИТ-отделы, сервисные центры, отделы закупок.

Дипломная работа выполнена автором самостоятельно.

РЭФЕРАТ

Курсавая праца змяшчае: 58 старонак, 25 малюнкаў, 12 выкарыстаных літаратурных крыніц, 3 табліцы, 4 дадаткі.

Ключавыя словы - ІТ-АКТЫВЫ, КІРАВАННЕ АКТЫВАМІ, ІТАМ, БАЗА ДАНЫХ, ВЭБ-РАЗРАБОТКА, SQL, C#, JAVASCRIPT, SQL, HTML, ASP.NET

Аб'ект даследавання - працэс кіравання ІТ-актывамі прадпрыемства.

Прадмет даследавання - метады і інструменты ўліку і кантролю ІТ-актывамі прадпрыемства.

Мэта працы - распрацоўка сістэмы для ўліку і кантролю ІТ-актываў кампаніі і яе праграмнага забеспячэння. Гэта дапаможа аўтаматызаваць кіраванне ІТ-актывамі, зрабіць улік больш празрыстым, палепшыць бізнес-працэсы і зменшыць колькасць памылак дзякуючы простаму інтэрфейсу, надзейнай базе даных і карысным модулям.

Метады даследавання – аналіз літаратурных крыніц па ІТАМ-сістэмах і кіраванні актывамі, параўнальны аналіз існуючых рашэнняў, мадэляванне бізнес-працэсаў, праектаванне базы дадзеных з нармалізацыяй, выкарыстанне Agile (SCRUM) для вэб-распрацоўкі па этапах, праверка сістэмы на рэальных сцэнарах.

Атрыманыя вынікі і іх навізна - распрацавана сістэма ўліку ІТ-актываў з інавацыйнымі аспектамі: інтэграцыя модуляў кіравання актывамі, заяўкамі і аналітыкі ў адзіны інтэрфейс; гібрыдная архітэктурна на ASP.NET для сервернай часткі і JavaScript-бібліятэк (Datatables.net, Google Charts) для кліенцкага інтэрфейсу.

Дакладнасць матэрыялаў і вынікаў дыпломнай працы пацвярджаецца выкарыстаннем вядомых стандартаў, як C#, ASP.NET і SQL. Правялі тэставанне на рэальных дадзеных і праверку ў адпаведнасці з патрабаваннямі ІТАМ-сістэм. Таксама забяспечылі ахову ад уразлівасцяў з дапамогай параметрызаваных SQL-запытаў і хэшавання пароляў.

Вобласць магчымага практычнага прымянення - малы і сярэдні бізнес, які кіруе ІТ-інфраструктурай: ІТ-аддзелы, сэрвісныя цэнтры, аддзелы закупак.

Дыпломная праца выканана аўтарам самастойна.

ABSTRACT

The coursework contains: 58 pages, 25 figures, 12 references, 3 tables, 4 appendices.

Keywords – IT ASSETS, ASSET MANAGEMENT, ITAM, DATABASE, WEB DEVELOPMENT, SQL, C#, JAVASCRIPT, SQL, HTML, ASP.NET

The object of the study is the process of managing the IT assets of the enterprise.

The subject of the study is the methods and tools for accounting and control of the IT assets of the enterprise.

The objective of the work is to develop a system for accounting and control of enterprise IT assets and its software for automating IT asset management, transparent accounting, improving the efficiency of business processes and minimizing errors due to an intuitive interface, a reliable database and functional modules.

Research methods - analysis of literary sources on ITAM systems and asset management, comparative analysis of existing solutions, functional modeling of business processes, design of a relational database with normalization, Agile (SCRUM) for step-by-step web development, testing the system on real scenarios.

The results obtained and their novelty - a system for keeping track of IT assets. It brings together different asset management tools and analytics into one easy-to-use interface. The back end is built on ASP.NET, while the front end uses JavaScript libraries like Datatables.net and Google Charts.

The reliability of the materials and results of the thesis is confirmed by using industry standards like C#, ASP.NET, and SQL. We tested it with real data and compared it with what's needed in the ITAM systems market. We also made sure it's secure by using parameterized SQL queries and hashing passwords.

The area of possible practical application are small and medium businesses managing IT infrastructure: IT departments, service centers, purchasing departments.

The degree is performed by the author independently.

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).

IDEF0 – Integrated Definition for Function Modeling.

SSMS – SQL Server Management Studio.

SKU – Stock Keeping Unit.

CRUD – Create, Read, Update, Delete.

БД – база данных.

ВВЕДЕНИЕ

Эффективное управление ИТ-активами критически важно для современного предприятия, так как напрямую влияет на его конкурентоспособность и обеспечивает стабильную работу. Системы учета и контроля ИТ-активов (IT Asset Management, ITAM) являются неотъемлемой частью этой деятельности, также существенно повышая уровень безопасности информационной инфраструктуры за счет своевременного выявления уязвимостей и предотвращения утечек данных.

Таким образом, внедрение ITAM дает предприятию возможность отслеживать ИТ-ресурсы на всех этапах их жизненного цикла – от закупки до списания. «В данном контексте разработка веб-сайта, на котором размещена система учета и контроля ИТ-активов, представляется актуальной» [12].

Целью данной дипломной работы является разработка системы учета и контроля ИТ-активов предприятия и ее программного обеспечения, направленного на автоматизацию управления ИТ-активами, обеспечение прозрачности учета, повышение эффективности связанных бизнес-процессов, снижение количества ошибок за счет четкой структуры данных и специализированных функциональных модулей.

Для достижения поставленной цели необходимо выполнить ряд задач:

- исследование современных подходов к управлению ИТ-активами и сравнительный анализ существующих ITAM-систем;
- анализ и функциональное моделирование бизнес-процессов предприятия в сфере учёта ИТ-ресурсов;
- определение технических и функциональных требований к системе;
- проектирование архитектуры веб-приложения, включая структуру базы данных, пользовательский интерфейс и функциональные модули;
- разработка ключевых подсистем: управления активами, категориями, брендами, отделами и заявками;
- реализация форм взаимодействия с системой (авторизация, регистрация, каталог активов и др.);
- тестирование на корректность работы и отказоустойчивость.

Работа состоит из введения, четырёх глав, заключения и списка использованных источников.

Во введение обосновывается актуальность темы, определяется цель, задачи, теоретическая и практическая значимость исследования.

Глава 1 рассматривает принципы и подходы к управлению ИТ-ресурсами, анализирует современные ITAM-системы, выявляя их сильные и

слабые стороны. На основе анализа формулируются требования к системе и уточняются задачи исследования.

Глава 2 содержит анализ бизнес-процессов предприятия в сфере IT-активов и выявление ключевых проблем. В ней выполняется моделирование процессов с использованием методологий IDEF0 и UML для визуализации жизненного цикла активов. Представлено концептуальное и логическое проектирование базы данных (сущности, атрибуты, связи).

Глава 3 обосновывает выбор технологического стека (ASP.NET, C#, Microsoft SQL Server), описывает проектирование архитектуры системы и пользовательского интерфейса. Детализируется реализация модулей управления активами, категориями, брендами и обработки заявок. Приводятся результаты тестирования, подтверждающие работоспособность и соответствие требованиям.

Заключение содержит основные выводы и обобщает результаты работы.

ГЛАВА 1

СОСТОЯНИЕ ПРОБЛЕМЫ, ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

1.1 Принципы и подходы к управлению ИТ-активами предприятия

Понятия ИТ-активов и основные типы достаточно полно описаны в ГОСТ Р ИСО/МЭК 19770-1—2021, где прописано, что «ИТ-актив – это произвольный объект, предмет или сущность, которые имеют потенциальную или фактическую ценность для организации» [11].

Согласно ГОСТ Р ИСО/МЭК 19770-1—2021, ИТ-активы можно разделить на несколько категорий:

1. Цифровые активы. Эта категория включает нематериальные ИТ-активы, существующие в цифровом виде. Цифровые активы, в свою очередь, делятся на:

- Активы программного обеспечения (ПО), которые охватывают как исполняемое программное обеспечение (операционные системы и приложения), так и неисполняемые компоненты (например, исходный код и библиотеки).

- Информационные активы, включающие в себя цифровые данные с информационным содержанием. К этой группе относятся документы, аудио- и видеофайлы, графические изображения, базы данных и автономные словари.

2. Системы управления ИТ-активами (ИТАМ). В том же ГОСТ Р ИСО/МЭК 19770-1—2021 дано необходимое определение ИТАМ: «Система управления для управления активами, функция которой заключается в определении политики управления активами и целей управления активами» [11].

Данная категория подразделяется на следующие компоненты:

- ИТАМ-системы и инструменты: это программные решения, используемые для инвентаризации, учета и мониторинга ИТ-активов, которые обеспечивают контроль над их жизненным циклом.

- Метаданные для ИТАМ содержат информацию о структуре, характеристиках и взаимосвязях ИТ-активов, что позволяет системам ИТАМ осуществлять эффективное управление.

3. ИТ-оборудование. Данная группа охватывает следующие физические компоненты:

– Физические носители представляют из себя устройства для хранения данных (USB-накопители, HDD).

– Физическое ИТ-оборудование, включающее в себя остальные компоненты: серверы, персональные компьютеры и так далее.

4. Лицензии на ИТ-активы. К ним относятся юридические документы, определяющий правила распространения и использования ПО.

5. Контракты по ИТ-активам, которые включают соглашения с поставщиками активов.

6. Сервисы ИТ-активов. Сюда входят услуги по поддержке и функционированию ИТ-активов

Жизненный цикл ИТ-актива представляет из себя время существования актива, стадии его развития.

На рисунке 1 выделены четыре фазы, через которые проходит любой ИТ-актив.



Рисунок 1. Фазы жизненного цикла ИТ-актива

На первой фазе создаётся полный список всех доступных ИТ-активов, собирается информация, какие активы имеются, их расположение, когда они были куплены и их цены.

Во второй фазе вычисляется совокупная цена для каждого ИТ-актива, здесь будут учитываться траты на поддержку, модернизацию и утилизацию актива.

За время эксплуатации актив может использоваться сотрудниками предприятия для своих задач. В случае поломки, актив подлежит ремонту. ИТ-актив может также находиться в резерве. В таком случае, он со временем

подлежит выдачи сотруднику или пользователю, либо будет выведен из эксплуатации.

В финальной фазе, исходя из цены и сроках службы активов, принимается решение о выводе из эксплуатации или продолжении использования этого актива.

Согласно ГОСТ Р ИСО/МЭК 19770-1—2021 управление IT-активами представляет собой «скоординированную деятельность организации для получения ценности от активов» [11].

Для эффективного управления IT-активами необходимо выбрать такое программное решение, которое, помимо автоматизации процессов на предприятии, будет актуальным, адаптируясь к возможным изменениям в будущем, а также превосходить аналоги.

Использование современного программного обеспечения для управления IT-активами имеет множество преимуществ. Например, большинство современных компаний часто сталкиваются с проблемами в планировании ремонтов и обслуживания активов, также велики потери при непосредственно эксплуатации активов, а также при их списании. Существуют сложности с отсутствием прозрачности в расходах на активы и нехваткой актуальных данных. Программное обеспечение и лицензии могут приобретаться без ведома руководства.

1.2 Обзор и анализ существующих ITAM

На рынке представлено множество различных решений ITAM, которые отличаются функционалом и возможностями. При выборе подходящего варианта стоит обратить внимание на несколько важных моментов: тип платформы, гибкость, методы инвентаризации, совместимость с мобильными устройствами, инструменты для отчетности и интеграцию с уже существующими системами.

Для анализа существующих инструментов управления IT-активами был проведён сравнительный анализ наиболее популярных ITAM: Asset Panda, ManageEngine AssetExplorer, SolarWinds Service Desk, Pulseway и LANsweeper. Сравнительная таблица ITAM приведена в приложении А.

Анализируя представленные в таблице данные, заметим, что Lansweeper, будучи самой популярной из рассматриваемых систем, демонстрирует наивысшую эффективность в автоматизации, что делает её идеальной для крупных организаций, где критично минимизировать ручной труд. Lansweeper также обладает сравнительно небольшой ценой, учитывая

качество своего программного обеспечения. В системе используется автоматическое отслеживание лицензий программного обеспечения, автоматизированы процессы добавления активов.

Asset Panda представляет из себя бюджетное решение, подходящее для небольших компаний, но с ограниченными возможностями. Среди них особенно важным недостатком является отсутствие автоматического управления лицензиями, добавлять активы можно только вручную. Из положительных сторон следует отметить высокую гибкость и GPS-отслеживание активов.

Pulseway обладает удобным современным интерфейсом, мобильной версией, но, так же, как и Asset Panda, лишена автоматического управления лицензиями, добавлять активы также можно только вручную.

SolarWinds тоже имеет современный интерфейс. Система предлагает интегрированное решение: облачное и локальное. Недостатки схожи с Pulseway.

ManageEngine AssetExplorer является лучшим решением, если для предприятия наибольшую роль играет отчётность и гибкость. Эта система представляет полные отчёты и автоматизированное управление лицензиями.

Выделим наиболее важные критерии для современных ИТАМ:

1. Автоматизация
2. Стоимость
3. Мобильное управление.
4. Отчетность.
5. Интеграция с другими системами.
6. Простота интерфейса.

При разработке собственной системы в первую очередь будем обращать внимание на эти критерии.

1.3 Назначение и основные требования к системе учета и контроля ИТ-активов

Проанализировав наиболее популярные ИТАМ, определим требования к разрабатываемой системе.

Пользователями системы учета и контроля ИТ-активов являются:

- сотрудники, использующие ИТ-оборудование в своей работе;
- ИТ-специалисты, отвечающие за обслуживание технических средств;
- администраторы системы, обеспечивающие ее функционирование.

Эффективное внедрение системы учета IT-активов требует соблюдения нескольких принципиально важных условий:

- простота и удобство интерфейса для пользователей с разным уровнем технической подготовки;
- гибкость настроек для адаптации под конкретные нужды предприятия;
- надежность хранения данных и контроль доступа к информации;
- возможность оперативного получения актуальных данных о состоянии IT-инфраструктуры.

Основные требования к системе учета и контроля IT-активов:

1. Бизнес-требования

БТ-1: Оптимизация управления IT-инфраструктурой организации.

БТ-2: Повышение эффективности обслуживания IT-активов.

БТ-3: Обеспечение безопасности корпоративных данных.

БТ-4: Автоматизация бизнес-процессов технической поддержки.

2. Пользовательские требования

ПТ-1: Ведение реестра оборудования и ПО с классификацией по типам, категориям и статусам.

ПТ-2: Привязка активов к пользователям и подразделениям с учетом технических характеристик.

ПТ-3: Система создания и обработки заявок на поддержку с назначением ответственных.

ПТ-4: Формирование отчетов по состоянию IT-активов и аналитики использования ресурсов.

ПТ-5: Разграничение прав доступа по ролям (пользователь, IT-менеджер, администратор).

ПТ-6: Настройка индивидуальных прав доступа для сотрудников.

ПТ-7: Быстрый поиск и фильтрация данных по активам/заявкам.

ПТ-8: Автоматические уведомления о статусах заявок и важных событиях.

3. Нефункциональные требования

НТ-1: Интуитивно понятный интерфейс.

НТ-2: Адаптивный дизайн для работы с мобильных устройств.

НТ-3: Поддержка 200+ одновременных пользователей.

4. Технические требования

ТТ-1: Кросс-браузерная совместимость.

ТТ-2: Интеграция с Active Directory и корпоративными системами.

ТТ-3: Шифрование данных.

ТТ-4: Резервирование серверов (N+1).

ТТ-5: Поддержка API REST для внешних интеграций.

5. Административные требования

АТ-1: Ведение журнала аудита всех изменений данных.

АТ-2: Двухфакторная аутентификация для роли администратора.

АТ-3: Возможность отката изменений в системе за последние 24 часа.

6. Системные требования

СТ-1: Развертывание в приватном облаке организации

СТ-2: Мониторинг доступности системы

Спецификация функциональных возможностей:

1. Управление активами:

- иерархическая классификация оборудования/ПО;
- учет сроков эксплуатации и гарантий;

2. Рабочие процессы

- шаблоны заявок для типовых задач;
- эскалация невыполненных заявок;
- интеграция с почтовыми уведомлениями.

3. Аналитика

- дашборды с графиками использования ресурсов;
- прогнозирование потребности в обновлениях;
- экспорт отчетов в XLSX/PDF.

4. Безопасность

- ролевая модель доступа RBAC;
- валидация вводимых данных;
- архивирование неактивных записей.

Требования соответствуют критериям проверяемости, полноты и согласованности. Приоритизация реализована через систему вех (MVP v1.0 — базовый функционал управления активами и заявками).

1.4 Цель и задачи исследования

Целью исследования является разработка системы учета и контроля ИТ-активов предприятия и ее программного обеспечения, направленного на автоматизацию управления ИТ-активами, обеспечение прозрачности учета, повышение эффективности связанных бизнес-процессов, снижение количества ошибок за счет четкой структуры данных и специализированных функциональных модулей.

Для достижения этой цели необходимо:

1. Проанализировать литературные источники по предметной области и сравнить наиболее популярные ИТАМ, определить их преимущества и недостатки.
2. Определить требования к разрабатываемой системе.
3. Провести анализ деятельности предприятия в области управления ИТ-активами
4. Выполнить функциональное моделирование бизнес-процессов учета и контроля ИТ-активов;
5. Спроектировать информационную модель системы.
6. Разработать ИТАМ, которая включает интуитивно понятный пользовательский интерфейс, надежную базу данных и функциональные модули для управления активами, заказами, брендами, отчетами и информацией о компании.

1.5 Выводы

В первой главе дипломной работы были получены следующие результаты:

1. Проведен анализ проблемы управления ИТ-активами, определены ключевые аспекты и задачи, требующие решения.
2. Рассмотрено понятие ИТ-актива, его жизненный цикл и проанализированы современные подходы к управлению ИТ-активами.
3. Проведен сравнительный анализ существующих ИТАМ-систем, что позволило сформировать основные требования к системе учета и контроля ИТ-активов.
4. Сформулирована постановка задачи, направленная на разработку системы учета и контроля ИТ-активов.

ГЛАВА 2

ПРОЕКТИРОВАНИЕ СИСТЕМЫ УЧЁТА И КОНТРОЛЯ ИТ-АКТИВОВ

2.1 Анализ деятельности предприятия и проблем в сфере управления ИТ-активами

Управление ИТ-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 ИТ-активами.

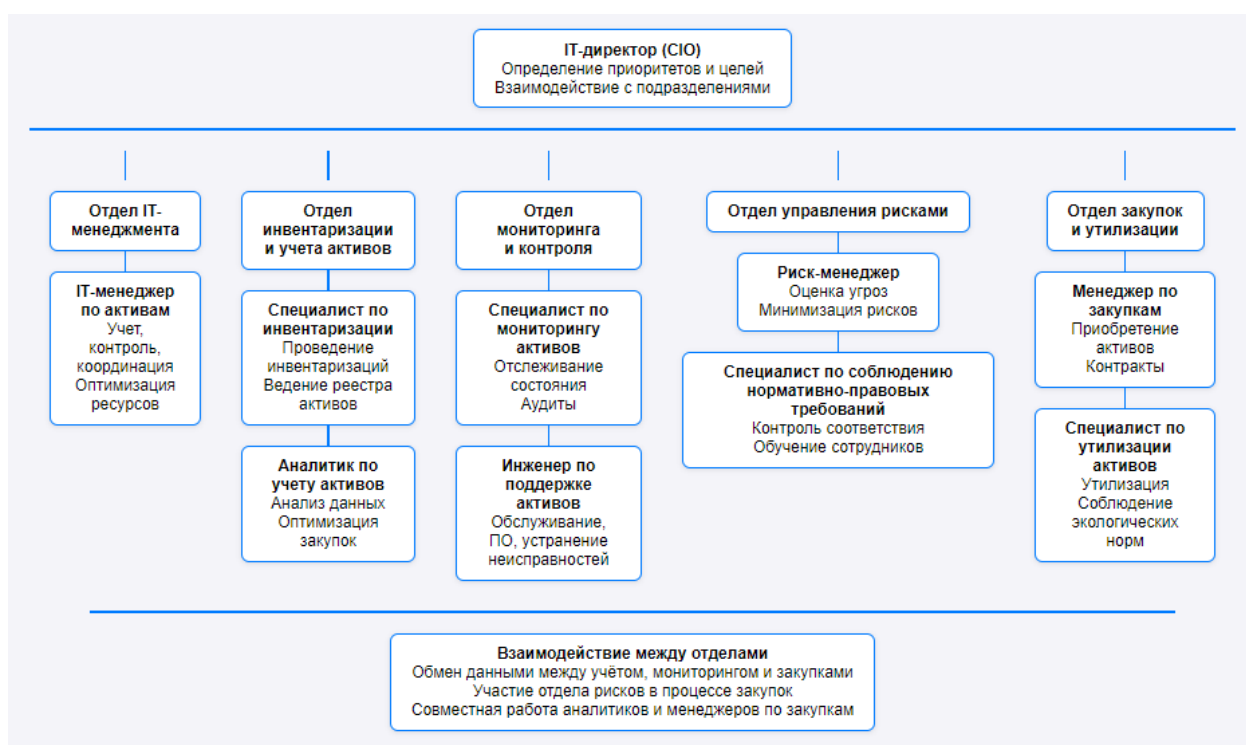


Рисунок 2.1 Схема организации предприятия по контролю и учёту ИТ-активов.

Согласно представленной схеме, центральным звеном в структуре является директор по информационным технологиям (CIO), который устанавливает приоритеты в работе с ИТ-активами и обеспечивает коммуникацию между ИТ-департаментом и другими отделами компании, формируя единое информационное пространство.

Менеджер ИТ-активов руководит учетом и контролем ресурсов, координирует работу команды и отвечает за оптимизацию активов.

Отдел инвентаризации и учета активов занимается идентификацией и классификацией активов, ведением реестра, анализом информации, отчетностью и участвует в планировании закупок и списания.

Отдел мониторинга и контроля отвечает за состояние инфраструктуры. Функциями отдела являются проведение регулярных проверок, отслеживание работоспособности активов.

Отдел управления рисками нужен для выявления угроз безопасности активов, минимизации потенциального ущерба от этих угроз.

В структуре управления ИТ-активами есть также отделы, занимающиеся *закупками и утилизацией*. Менеджер по закупкам отвечает за приобретение новых ресурсов и контроль договоров, а специалист по утилизации организует безопасное списание устаревших активов, соблюдая экологические нормы.

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

Эффективное управление ИТ-активами требует совместной работы всех отделов. Например, отдел инвентаризации должен делиться данными с мониторингом, чтобы быстро реагировать на проблемы. Недостаток четких процедур контроля может привести к рискам для активов, поэтому важно регулярно проверять и интегрировать системы, обучать сотрудников и обеспечить взаимодействие между подразделениями.

2.2 Моделирование бизнес-процессов учета и контроля ИТ-активов

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

Контекстная диаграмма для системы «Учет и контроль ИТ-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.

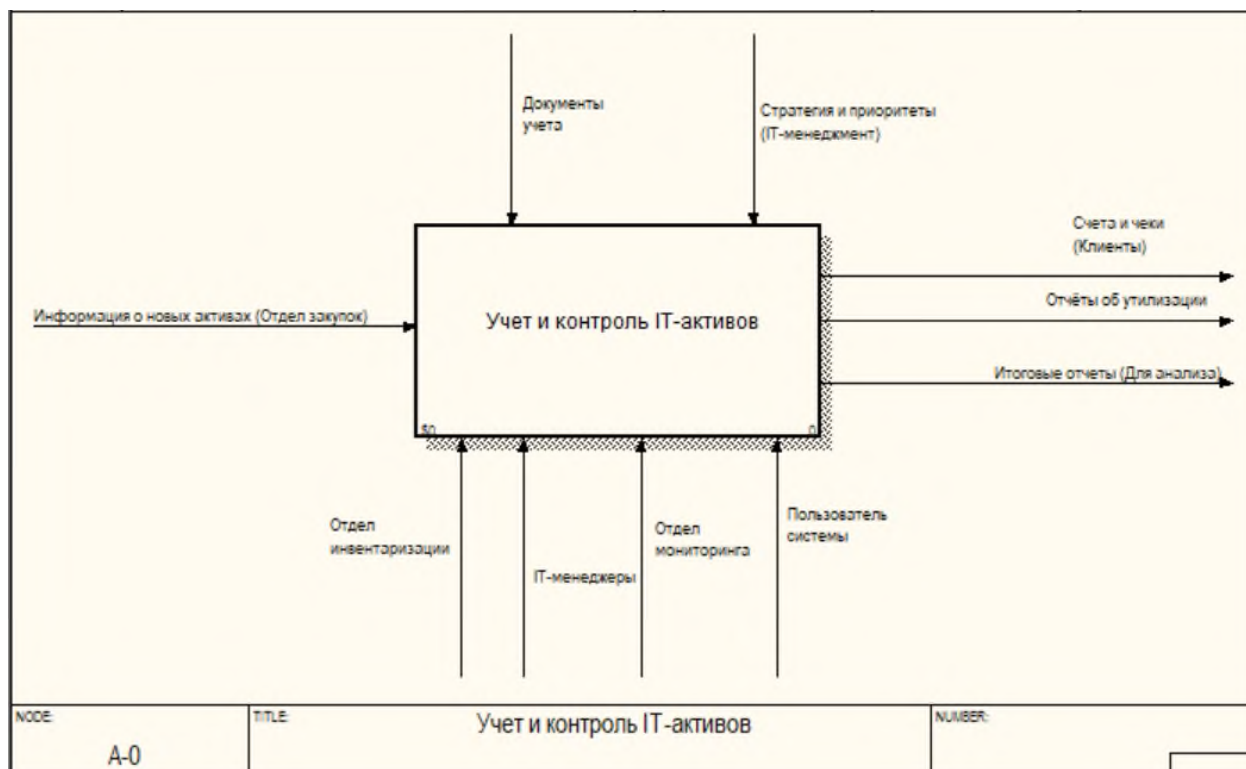


Рисунок 2.2 Контекстная диаграмма для системы «Учет и контроль IT-активов»

На **вход** подаётся информация о новых активах. Этим занимается отдел закупок, предоставляя все необходимые документы.

Управление на диаграмме представлено через документы учёта, а также стратегию и приоритеты предприятия, чем занимается IT-менеджмент.

Механизмами взаимодействия являются отдел инвентаризации, мониторинга, сами пользователи системы и IT-менеджеры.

Отдел инвентаризации ответственен за регистрацию и мониторинг активов.

Отдел мониторинга обеспечивает отчётность, следит за состоянием активов, их производительностью и наличием модификаций.

На **выходе** получаем счета, чеки, отчёты об утилизации, итоговые отчёты о всём жизненном цикле актива.

На основе этих документов будут приниматься решения о продлении работы или вывода из эксплуатации актива.

Для системы процесс декомпозируется на четыре этапа жизненного цикла. На каждом этапе описаны подзадачи. Входные, выходные данные, управления и механизмы взаимодействия остаются аналогичными контекстной диаграмме.

Диаграмма представлена на рисунке 2.3.

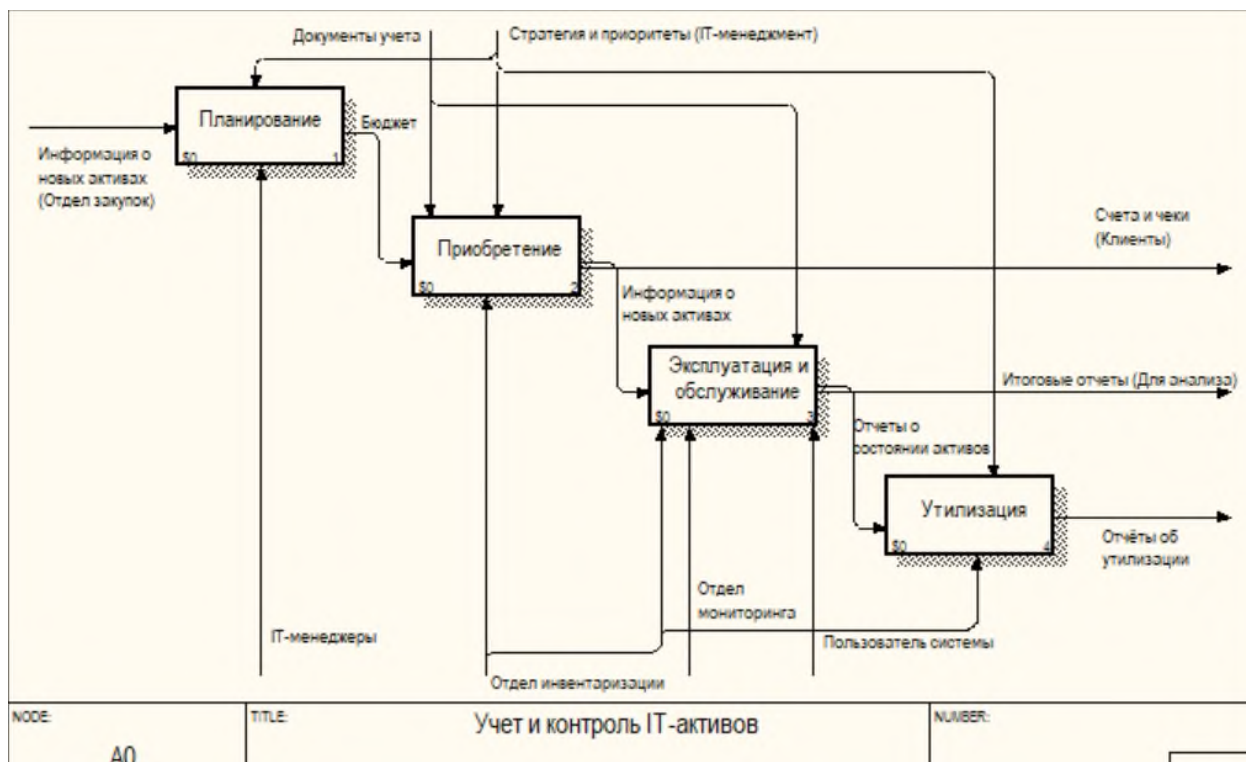


Рисунок 2.3 Диаграмма декомпозиции A0 «Управление жизненным циклом IT-активов

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

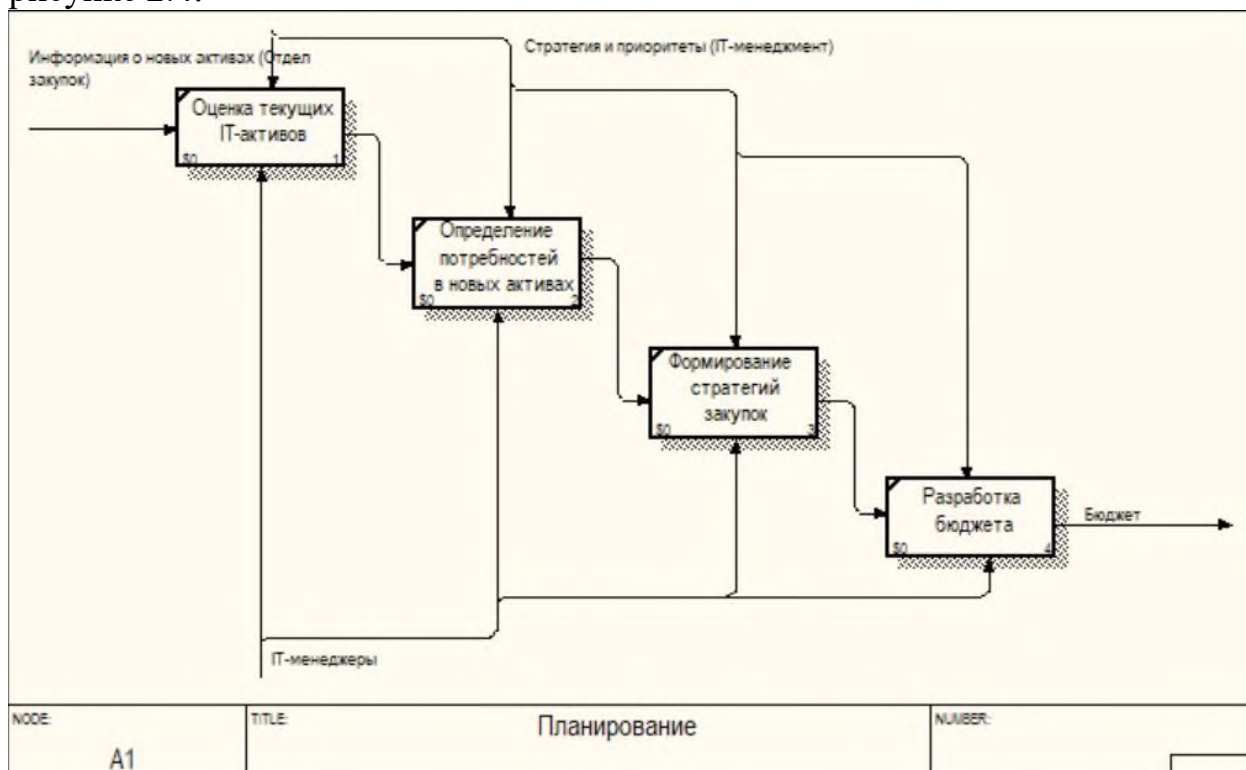


Рисунок 2.4 Диаграмма декомпозиции A1: Этап «Планирование»

Первым этапом является *оценка текущих ИТ-активов*. Происходит анализ всех возможных активов компании.

После анализа следует определить требования к нашим активам. Это происходит на этапе *определения потребностей в новых активах*.

Основываясь на полученной информации, формируются *стратегии закупок*.

На последнем этапе для установления финансовых рамок каждой категории ИТ-активов происходит *разработка бюджета*.

Теперь рассмотрим связующие элементы процесса:

На *вход* подаются данные об инвентаризации и отчётности. Этим занимается отдел инвентаризации.

Управление на диаграмме представлено через также стратегию и приоритеты предприятия, чем занимается ИТ-менеджмент.

Механизмами взаимодействия являются сотрудники отдела планирования и ИТ-менеджеры.

На *выходе* получаем разработанный бюджет, план закупок и рекомендации по модификациям.

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

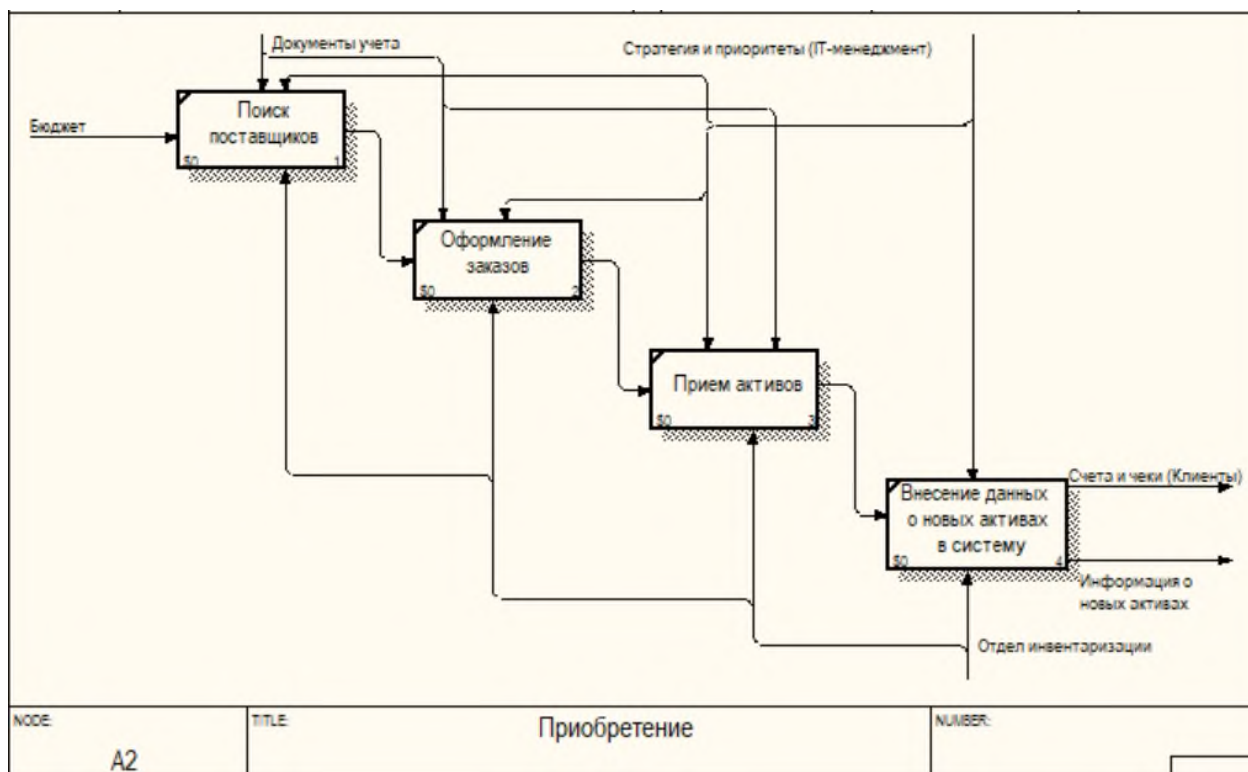


Рисунок 2.5 Диаграмма декомпозиции A2: Этап «Приобретение»

Начальным этапом является **поиск поставщиков**. Делается анализ репутации и сбор информации о поставщиках.

После того, как был выбран поставщик, происходит **оформление заказов** на покупку, где указываются цены, количество и условия поставки. Заказы обязательно нужно согласовать с руководством и внутренними стандартами.

Далее происходит **приём активов**, то есть проверяются все поступившие товары на соответствие условиям заказа.

На последнем этапе активам присваиваются серийные номера, вводится вся информация в систему. Иными словами, происходит **внедрение данных в систему**.

Теперь рассмотрим связующие элементы процесса:

На **входе** имеем утверждённый бюджет и план закупок.

Управление на диаграмме представлено через документацию учёта активов.

Механизмами взаимодействия являются отдел закупок и пользователи системы.

На **выходе** получаем новые активы с внесённой в систему информацией.

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

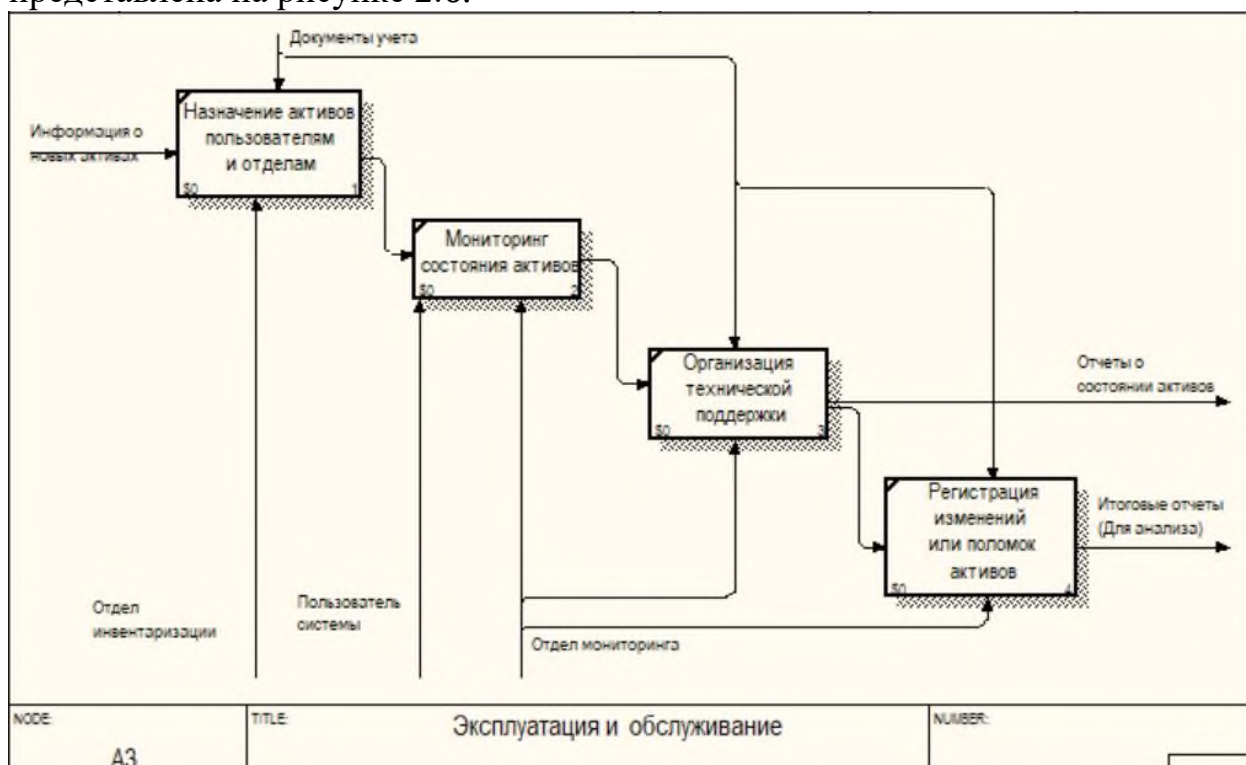


Рисунок 2.6 Диаграмма декомпозиции АЗ: Этап «Эксплуатация и обслуживание»

Первым этапом является **оценка состояния активов на списание**. Те активы, которые предлагаются для утилизации, проходят тщательные проверки, где проверяется их техническое состояние и экономическая целесообразность их поддержки.

Далее идёт **утверждение утилизации**. С руководством согласовывается утилизация актива, выбирается способ утилизации.

Далее происходит **организация процесса утилизации**, здесь выбираются кампании, занимающиеся утилизацией, и реализуется процесс.

На последнем этапе происходит **обновление данных в системе после утилизации**. После утилизации в базе данных системы должны быть внесены соответствующие изменения.

Теперь рассмотрим связующие элементы процесса:

На **входе** у нас есть информация о состоянии активов.

Управлением является политика утилизации кампании.

Механизмами взаимодействия являются отдел инвентаризации и лицензированные кампании, занимающиеся утилизацией.

На **выходе** получаем отчёт об утилизации активов.

Таким образом, имеется одна контекстная диаграмма и пять диаграмм декомпозиции, соответствующие требованиям IDEF0.

2.3 Разработка и проектирование информационной модели системы

При разработке структуры базы данных для разрабатываемой системы необходимо сформировать общую схему данных, обеспечивающую ясное отображение ключевых информационных компонентов и их взаимодействий.

Для формирования указанной схемы будем использовать модель «сущность-связь» (ER), она дает возможность определить основные объекты предметной области, их характеристики и отношения между ними.

Проведем анализ предметной области базы данных, выделим основные сущности и их атрибуты:

1. Счет (Bill)

- Идентификационный номер.
- Дата принятия актива.
- Суммарная цена (до вычета налогов).
- Стоимость оказанных услуг.
- Окончательная цена (после вычета налогов).
- Скидка.
- Идентификационный номер пользователя.

2. Позиция счета (Bill Item)
 - Идентификационный номер.
 - Идентификатор счета.
 - Идентификатор IT-актива.
 - Количество активов.
 - Итоговая цена актива.
3. Бренд (Brand)
 - Идентификационный номер.
 - Название бренда актива.
 - Статус бренда (в наличии и не в наличии).
4. Программное обеспечение (Software)
 - Идентификационный номер.
 - Название ПО.
 - Тип программного обеспечения.
 - Дата установки ПО.
5. Носители (Media)
 - Идентификационный номер.
 - Тип носителя.
 - Описание носителя.
 - Вместимость носителя.
6. Оборудование (Hardware)
 - Идентификационный номер.
 - Название оборудования.
 - Тип оборудования.
 - Производитель оборудования.
 - День приобретения оборудования.
 - Конец гарантии.
7. Лицензии (License)
 - Идентификационный номер.
 - Ключ активации лицензии.
 - Дата активации лицензии.
 - Срок действия лицензии.
8. Контракты (Contract)
 - Идентификационный номер.
 - Регистрационный номер контракта.
 - Начало действия.
 - Окончание действия.
 - Описание контракта.

- Поставщик.
- 9. Категория (Category)
 - Идентификационный номер.
 - Название категории.
 - Статус категории.
 - Идентификатор носителя.
 - Идентификатор оборудования.
 - Идентификатор лицензии.
 - Идентификатор контракта.
- 10. Склад (Instock)
 - Идентификационный номер.
 - Идентификатор IT-актива.
 - Цена закупки актива.
 - Дата поступления актива.
- 11. Активы (IT–Assets)
 - Идентификационный номер.
 - Название актива.
 - SKU актива.
 - Цена актива.
 - Количество.
 - Описание актива.
 - Идентификатор бренда.
 - Идентификатор категории.
 - Статус актива.
 - Идентификатор отдела.
- 12. Пользователь (User)
 - Идентификационный номер.
 - Имя пользователя.
 - Пароль.
 - Роль пользователя.
 - Телефон пользователя.
 - Идентификатор отдела.
- 13. Отдел (Department)
 - Идентификационный номер.
 - Название отдела.
 - Идентификатор руководителя.
 - Количество сотрудников в отделе.

Теперь определим связи между полученными сущностями:

1. Счета (Bill) привязаны к Клиентам (Customer) через поле `customer_id`. Один клиент может обладать несколькими счетами, но каждый конкретный счет относится только к одному клиенту. Это отношение типа «один ко многим» между сущностями Bill и Customer.

2. Счета (Bill) ассоциированы с Пользователями (User) по полю `user_id`. На одного пользователя может приходиться множество созданных счетов, однако каждый отдельный счет закреплен только за одним пользователем. Связь между Bill и User также является «один ко многим».

3. Состав счета (Bill Item) определяется по полю `bill_id`, связывающему элементы с общим счетом. Один счет может включать несколько позиций (элементов), но каждая позиция принадлежит строго одному счету. Отношение между Bill и Bill Item – «один ко многим».

4. Позиции счета (Bill Item) ссылаются на конкретные IT-активы (IT-Assets) через поле `asset_id`. Один IT-актив может фигурировать в разных позициях счетов, но каждая позиция указывает на единственный актив. Связь Bill Item и IT-Assets – «один ко многим».

5. IT-активы (IT-Assets) связаны с Брендами (Brand) по полю `asset_brand`. Один бренд может выпускать множество активов, в то время как каждый актив принадлежит только одному бренду. Отношение IT-Assets к Brand – «один ко многим».

6. Принадлежность IT-активов (IT-Assets) к Категориям (Category) задается полем `asset_category`. Одна категория объединяет множество активов, но каждый актив классифицируется в единственную категорию. Связь IT-Assets и Category – «один ко многим».

7. Записи о наличии на складе (Instock) относятся к IT-активам (IT-Assets) по полю `asset_id`. Один IT-актив может иметь несколько записей о количестве на складе (разные экземпляры), но каждая запись Instock соответствует только одному активу. Отношение Instock к IT-Assets – «один ко многим».

8. Цифровые активы (Digital Assets) классифицируются по Категориям (Category) через поле `category_id`. Одна категория может включать множество цифровых активов, тогда как каждый цифровой актив относится к одной категории. Связь Digital Assets и Category – «один ко многим».

9. Категории (Category) и активы ПО (Software) связаны через поле `software_id`. Каждая категория может содержать ровно один актив ПО, и каждый актив ПО привязан к единственной категории. Это отношение «один к одному» между Category и Software.

10. Категории (Category) и носители (Media) объединены полем media_id. Категория включает не более одного носителя, и каждый носитель принадлежит только одной категории. Связь Category и Media – «один к одному».

11. Категории (Category) и оборудование (Hardware) связаны по полю hardware_id. Каждой категории соответствует только одно оборудование, и каждое оборудование входит в одну категорию. Отношение Category и Hardware – «один к одному».

12. Категории (Category) и лицензии (License) имеют связь через поле license_id. Категория содержит максимум одну лицензию, а каждая лицензия закреплена за одной категорией. Связь Category и License – «один к одному».

13. Категории (Category) и контракты (Contract) сопоставлены по полю contract_id. Одна категория может включать только один контракт, и каждый контракт относится к одной категории. Отношение Category и Contract – «один к одному».

14. Пользователи (User) распределены по Отделам (Department). Множество пользователей может входить в состав одного отдела.

15. IT-активы (IT-Assets) могут быть закреплены за Отделами (Department). Активы распределяются по различным подразделениям организации.

Выделим группы пользователей базы данных:

1. Администраторы системы имеют полный доступ к данным, что позволяет им управлять информацией, настраивать параметры и контролировать безопасность.

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

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.

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 является целочисленным. bill_id – внешний ключ, связывающий элемент со счетом.

7. asset_id – обязательный идентификатор IT-актива, а asset_qty_need – необходимое количество актива, заданное INT.

На рисунке Б1.1 приложения Б представлена концептуальная модель, в которой каждая сущность взята в отдельный прямоугольник.

Концептуальную модель базы данных представлена в виде ER-диаграммы на рисунке Б1.2 приложения Б.

Основная задача логического уровня проектирования заключается в том, чтобы полученную инфологическую модель преобразовать в схему, где сущности, атрибуты и связи, которые мы ранее определили, оформляются в соответствии с правилами моделирования.

Именно на логическом уровне формируется структура будущей базы данных. На рисунке 2.8 представлена логическая модель БД «Учёт и контроль IT-активов».

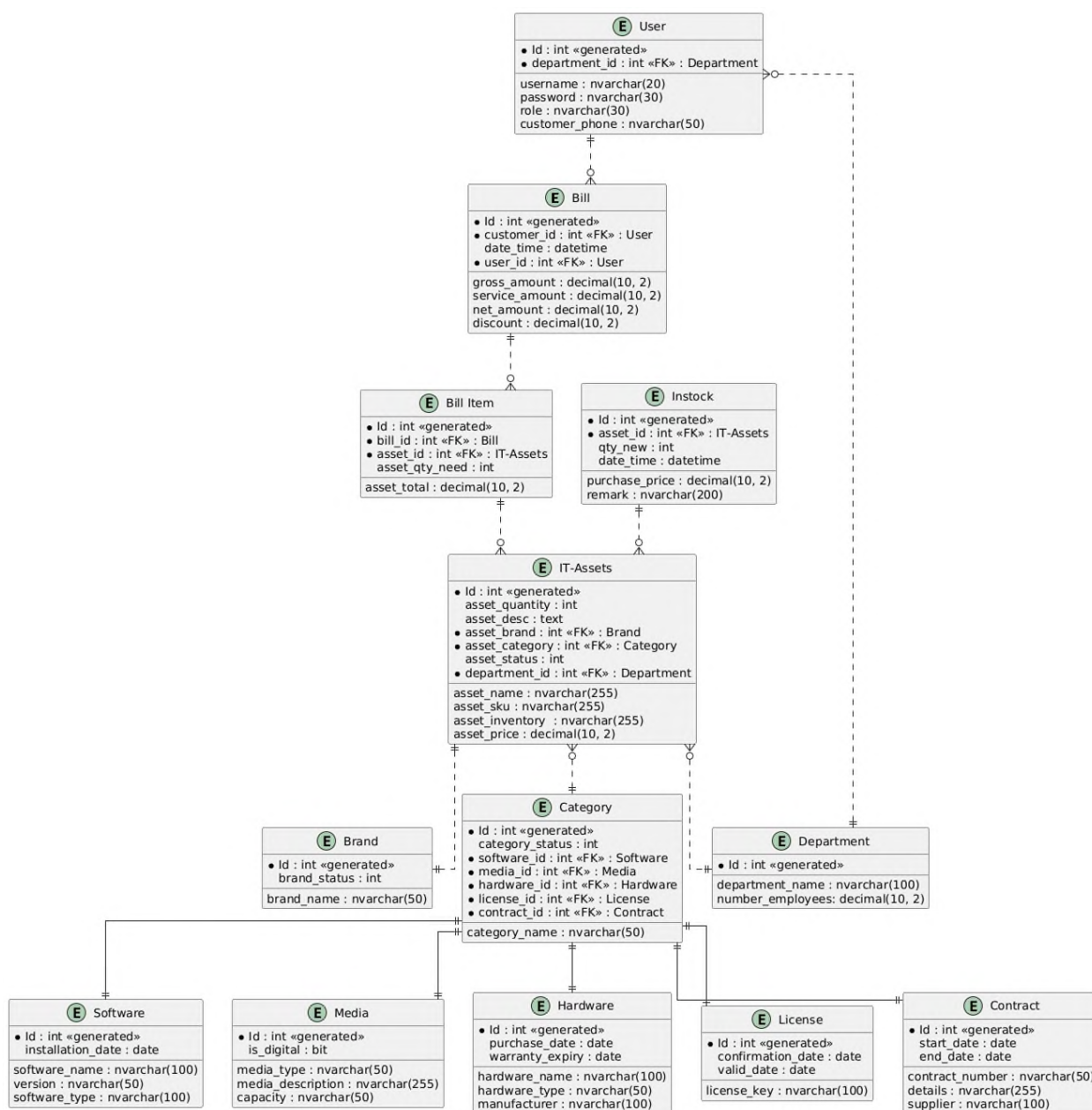


Рисунок 2.8 Логическая модель БД «Учёт и контроль ИТ-активов»

Реляционные таблицы базы данных представлены в таблицах Б2.1-Б2.13 в Приложении Б, они содержат подробную информацию о сущностях и их атрибутах.

Физическое проектирование БД реализовано при помощи «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. SSMS включает в себя

графический интерфейс и позволяет работать с T-SQL, упрощая выполнение сложных SQL-команд» [6].

Следует начать с создания базы данных, необходимой для управления ИТ-активами. Мы создаем базу данных с именем IT_Asset_Management и переключаемся на нее с помощью команды USE. В листинге B1 представлен код создания базы данных и таблиц на языке Transact-SQL с обеспечением ссылочной целостности данных.

После создания структуры база данных наполняется тестовыми данными для проверки работоспособности системы. Для этого используются команды INSERT INTO, которые добавляют записи в каждую из таблиц. Пример заполнения таблиц можно увидеть в Приложении B2.

Для эффективного управления ИТ-активами разработан набор SQL-запросов, позволяющих:

1. Фильтровать активы по категориям, брендам, статусам, датам и стоимости.
2. Анализировать востребованность активов, динамику закупок, распределение по отделам.
3. Контролировать сроки действия лицензий, гарантии, контрактов.

В листинге Приложения B3 приведены запросы, включающие поиск активов с истекающей гарантией, анализ распределения активов по отделам, выявление дублирующихся активов и отслеживание лицензий с истекающим сроком.

2.4 Выводы

Во второй главе были получены следующие результаты:

1. Представлен многоуровневый процесс управления ИТ-активами, включающий взаимодействие ключевых отделов, в результате которого обозначены ключевые роли и функции системы.
2. Проведена визуализация процессов учета и контроля ИТ-активов с использованием методологии IDEF0.
3. Разработана концептуальная модель БД, выделяющая ключевые сущности и атрибуты ИТ-активов, определены основные требования к атрибутам БД и связи между сущностями.
4. Проведено логическое и физическое проектирование БД на платформе Microsoft SQL Server, а также разработаны запросы по поиску, анализу ИТ-активов, контролю и управлению ролями пользователей.

ГЛАВА 3

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ УЧЕТА И КОНТРОЛЯ ИТ-АКТИВОВ

3.1 Выбор технологического стека для программной реализации системы учета и контроля ИТ- активов

Основу разработки веб-приложения составила среда Microsoft Visual Studio .NET, выбранная благодаря глубокой интеграции с платформой .NET и поддержке многоязыковой разработки. Встроенные инструменты, такие как отладчик и профилировщик, существенно упростили написание кода, его тестирование и оптимизацию производительности [4].

Веб-интерфейс построен на ASP.NET, активно использовавшем компоненты для работы с данными. GridView применялся для автоматического отображения табличных данных с пагинацией, а Repeater — для гибкой настройки шаблонов вывода, включая обработку вложенных списков. Для декларативной привязки к Microsoft SQL Server и сокращения кода CRUD-операций использовался SqlDataSource.

Серверная логика на C# реализована с использованием паттерна MVC, что обеспечило четкое разделение ответственности между компонентами приложения. Строгая типизация C# способствовала раннему выявлению ошибок, например, при несоответствии типов данных в методах взаимодействия с API поставщиков [8].

Клиентская часть создана с использованием HTML5 и CSS3. Адаптивная верстка и стандартные компоненты (меню, модальные окна) реализованы с помощью Bootstrap. Для создания кастомных анимаций и уникального визуального стиля применялся W3.CSS. Медиа-запросы гарантируют корректное отображение на экранах от 320 пикселей. Динамическая подгрузка контента (например, фильтрация товаров) реализована через AJAX-запросы с использованием jQuery.

Структура базы данных нормализована: данные инвентаря, заказов и пользователей хранятся в отдельных таблицах, связанных внешними ключами. Оптимизация SQL-запросов включала индексацию часто используемых полей (артикулы товаров) и применение хранимых процедур для формирования отчетов. Для работы с большими объемами данных на клиенте (10 000+ строк) использовался плагин Datatables.net с серверной обработкой и постраничной загрузкой [1].

Интерактивные элементы интерфейса (валидация форм в реальном времени, drag-and-drop загрузка изображений товаров) реализованы на JavaScript. Библиотека jQuery UI использовалась для интеграции календаря выбора дат в отчетах. Печатные формы счетов формируются с использованием CSS-медиазапросов типа print для скрытия ненужных элементов интерфейса.

Визуализация данных выполнена с помощью Google Charts, используются столбчатые и круговые диаграммы. Для повышения производительности данные диаграмм кэшируются на сервере. Функциональность реального времени (например, отслеживание новых заказов) реализована с использованием SignalR, обеспечивающего обновление данных на графиках без перезагрузки страницы [3].

3.2 Проектирование архитектуры приложения

Функционал системы управления активами, заявками, пользователями и отчетностью представлен в виде диаграммы вариантов использования (рисунок 3.1). Основными акторами являются Пользователь, Администратор и IT-менеджер, каждый из которых взаимодействует с системой через специфичные для их ролей варианты использования.

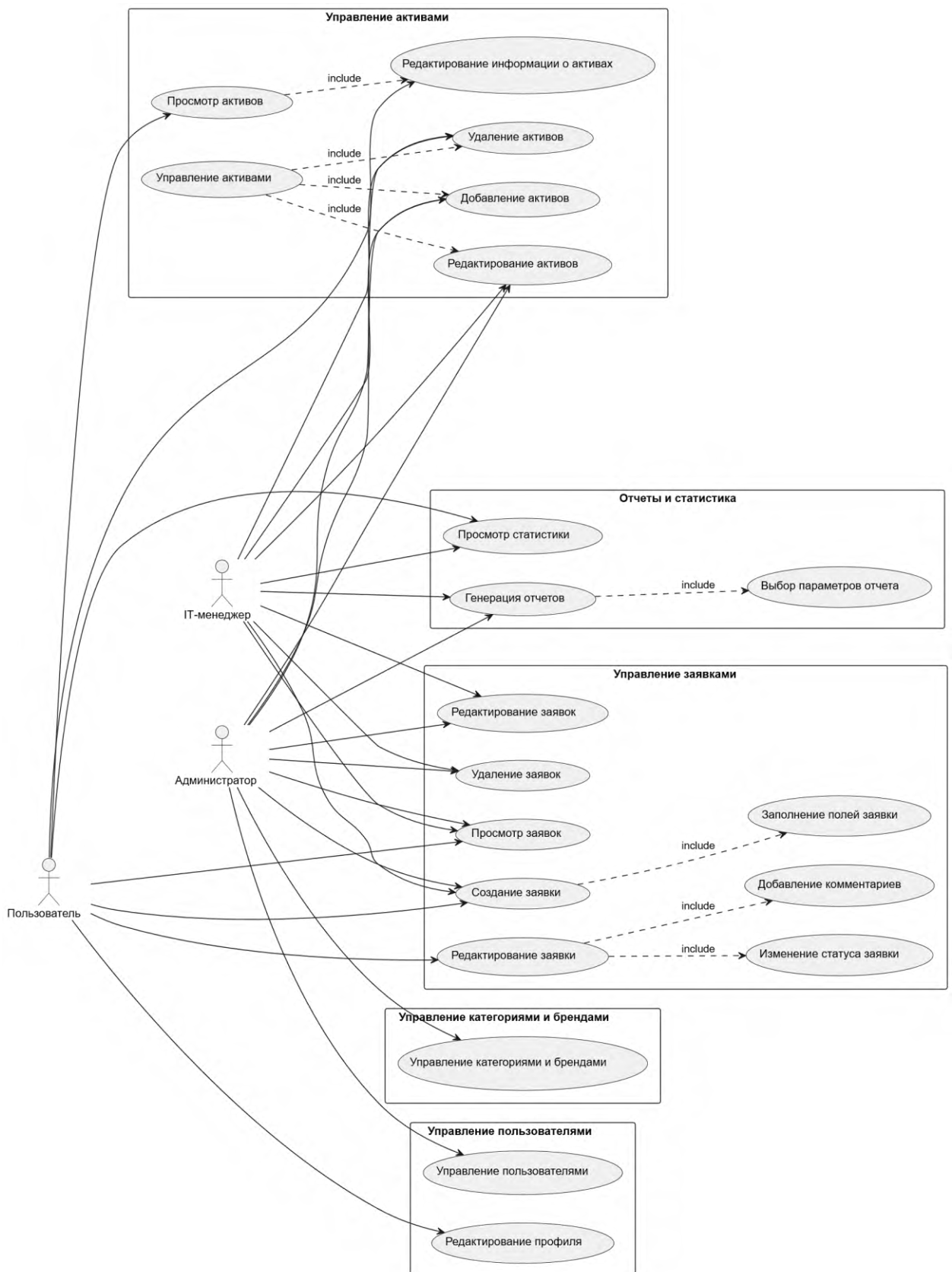


Рисунок 3.1 Диаграмма вариантов использования

Структура системы управления активами, включая ключевые модули, такие как веб-приложение (ASP.NET), база данных (SQL Server), а также

компоненты для управления активами, заявками, пользователями, отчетами и аутентификацией, отображается на диаграмме компонентов (рисунок 3.2). Основные взаимодействия между компонентами, такие как обмен данными через HTTP и SQL, показаны стрелками, что демонстрирует интеграцию между интерфейсом, бизнес-логикой и хранилищем данных. Диаграмма также подчеркивает использование API для связи между модулями, например, между управлением активами и категориями.

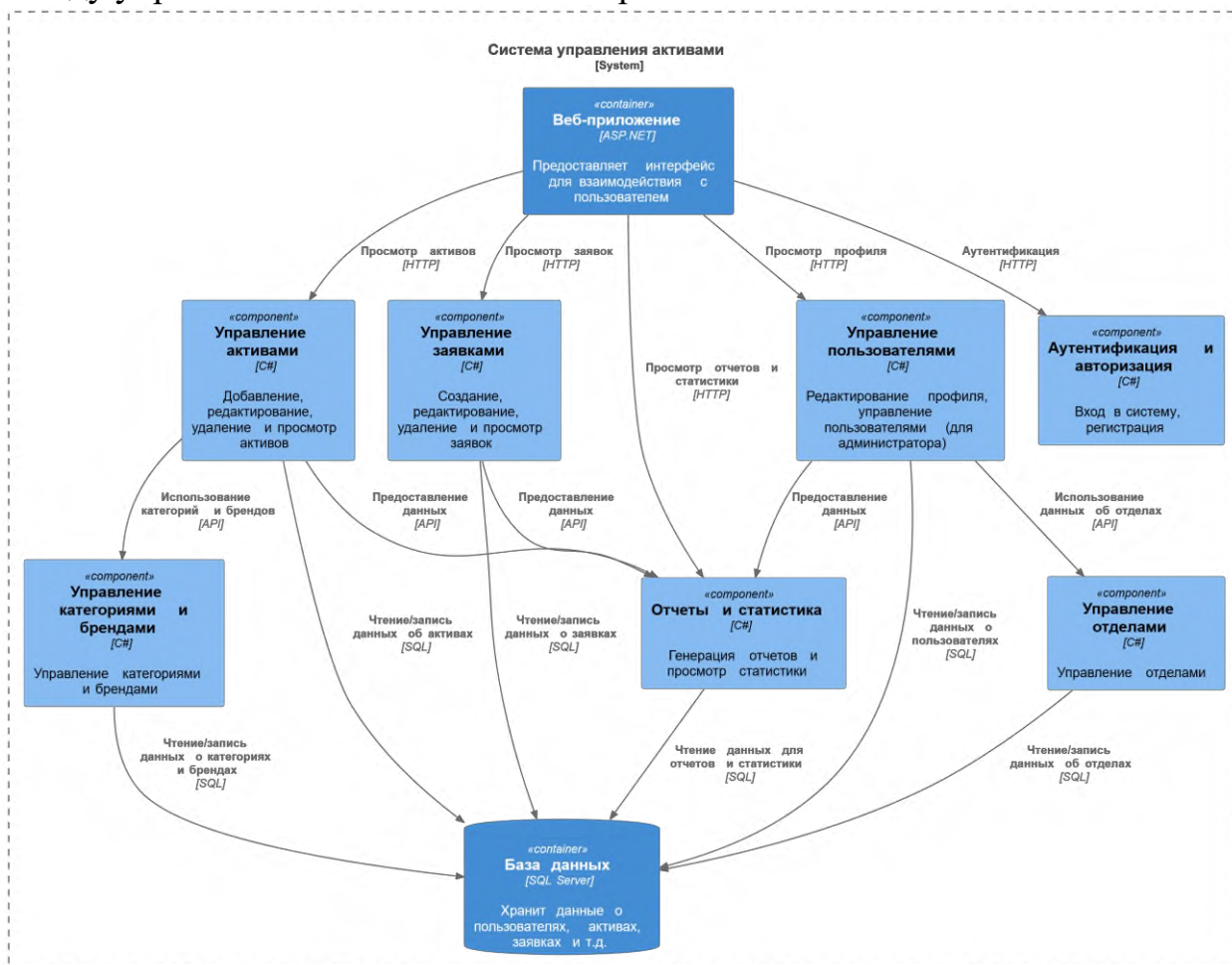


Рисунок 3.2 Диаграмма компонентов

Физическая архитектура системы, включая взаимодействие между клиентским браузером, веб-сервером на ASP.NET с модулями и сервером базы данных, представлена на диаграмме развёртывания (рисунок 3.3). На диаграмме показаны ключевые компоненты, такие как AssetService, Departments, Assets, Categories и Users, а также их связь через веб-интерфейсы и API. Особое внимание уделено аутентификации через ASP.NET Web Forms, что подчеркивает безопасность системы [2].

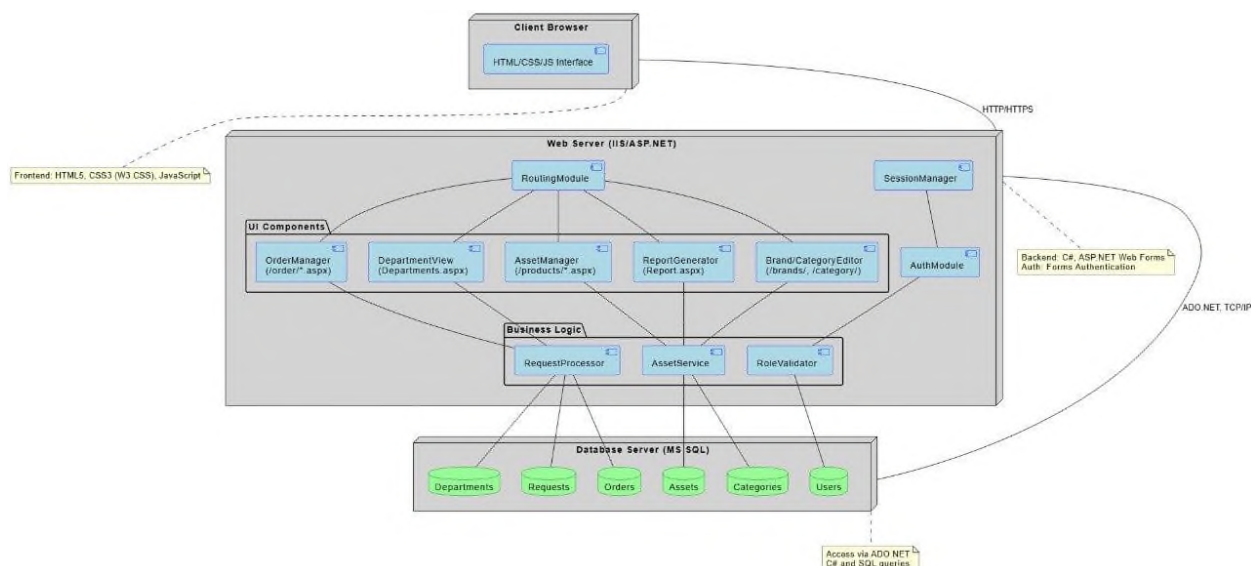


Рисунок 3.3 Диаграмма развёртывания

Взаимодействие между пользователями (User, Admin, IT-manager), интерфейсом, бэкендом и базой данных в рамках ключевых сценариев: просмотр активов, добавление актива и управление заявками демонстрируется на диаграмме последовательности (рисунок 3.4). На диаграмме показаны этапы обработки запросов, включая проверку ролей, валидацию сессии, выполнение SQL-запросов (SELECT, INSERT, UPDATE) и отправку уведомлений, что отражает логику работы системы и взаимодействие её компонентов.



Рисунок 3.4 Диаграмма последовательности

На основании архитектуры, представленной на разработанных диаграммах, система управления ИТ-активами структурирована в виде взаимосвязанных подсистем. В таблице 3.1 представлены ключевые подсистемы, их структура:

Таблица 3.1 – Функциональная структура системы

№	Подсистема	Модули
1.	Управления активами	Управление жизненным циклом активов Инвентаризация активов Управление физическими и виртуальными активами Управление связями между активами
2.	Управления заявками	Создание и обработка заявок Назначение ответственных Контроль выполнения Уведомления
3.	Администрирование	Управление профилями Управление отделами Настройку ролей и прав доступа
4.	Отчетность и аналитика	Генератора отчетов Аналитических панелей Настроек экспорта данных
5.	Справочники	Управление категориями Управление брендами Управление типами активов

Назначение функциональных составляющих системы:

Подсистема управления активами предназначена для автоматизации процессов учета и контроля ИТ-активов, добавления, редактирования и списания активов, поиска и фильтрации активов, управления зависимостями между активами.

Подсистема управления заявками предназначена для организации процесса обработки запросов пользователей, технической поддержки, контроля сроков выполнения, автоматического оповещения участников.

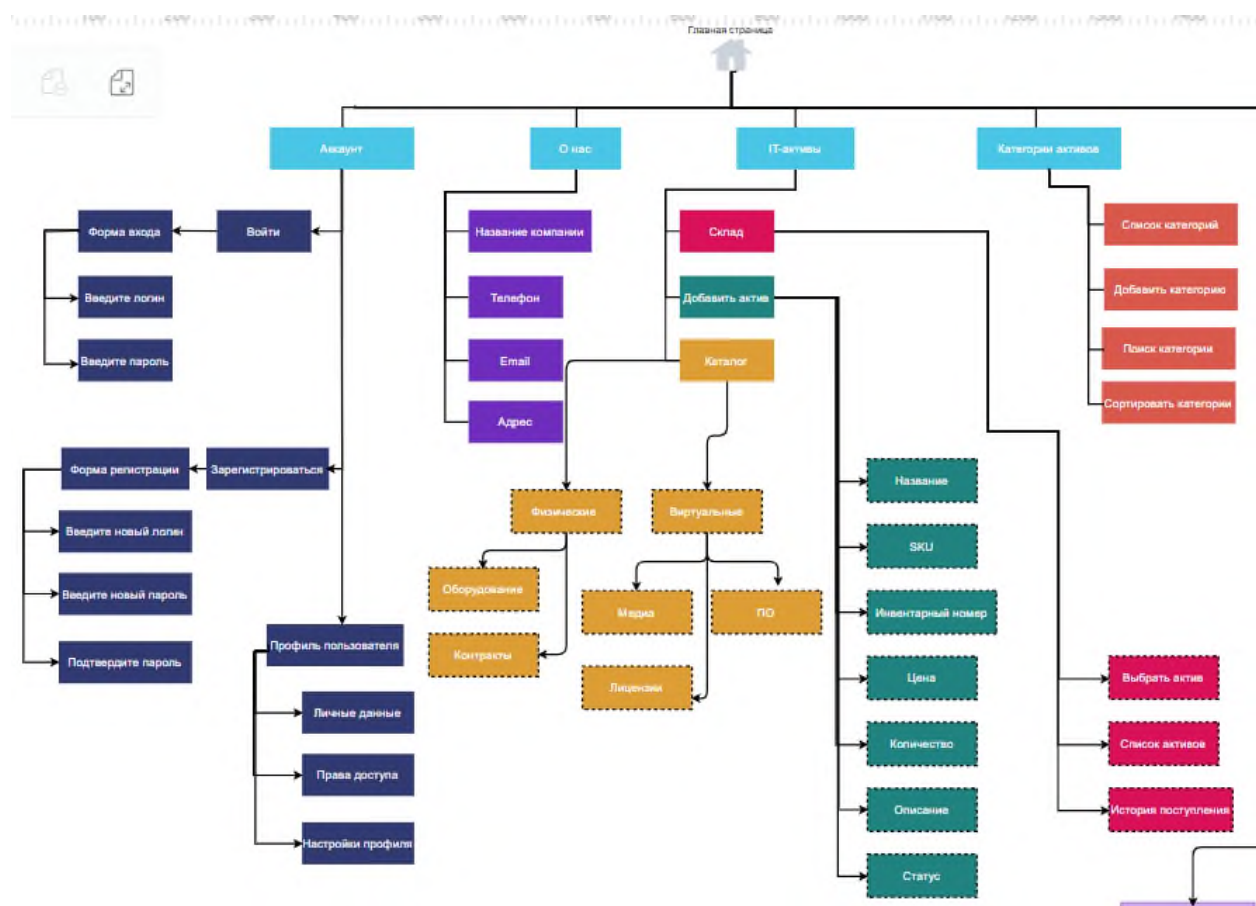
Подсистема администрирования предназначена для обеспечения контроля доступа к системе, разграничения прав пользователей, управления организационной структурой.

Подсистема отчетности и аналитики предназначена для предоставления стандартных и настраиваемых отчетов, визуализации данных об активах, инструментов для анализа эффективности.

Подсистема справочников предназначена для классификации активов, предоставления нормативно-справочной информации, базовой структуры данных системы.

3.3 Проектирование и разработка пользовательского интерфейса и элементов веб-приложения

Структура веб-приложения, включающая основные разделы представлена на карте сайта на рисунке 3.5. Элементы организованы иерархически, демонстрируя взаимосвязь между пользовательскими функциями и администрированием системы.



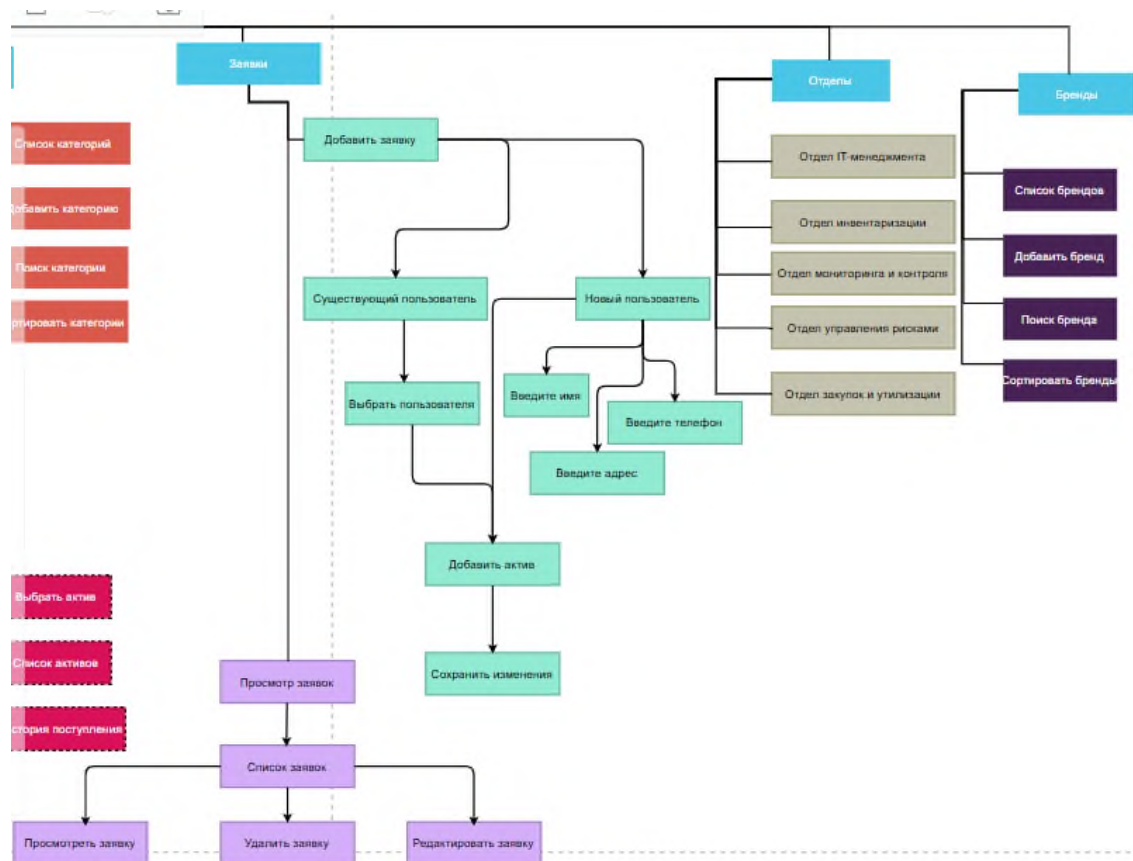


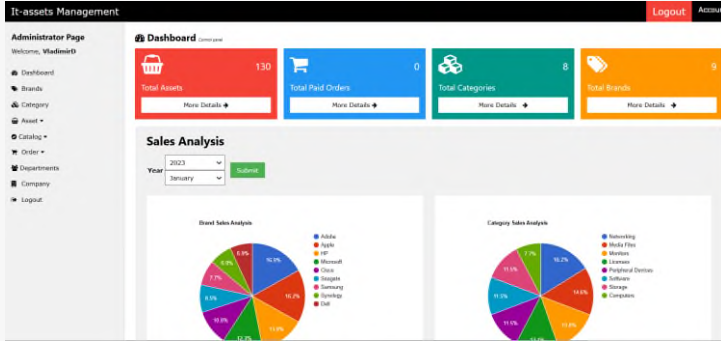
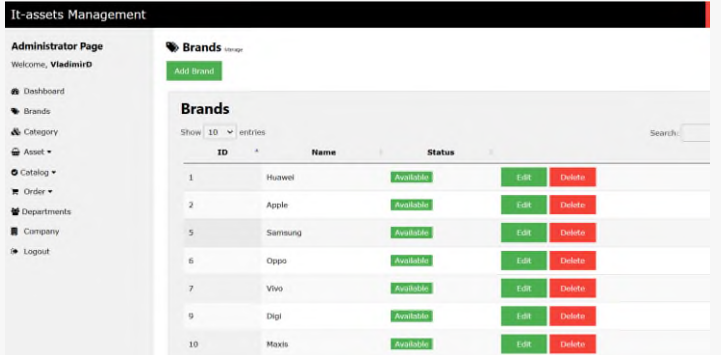
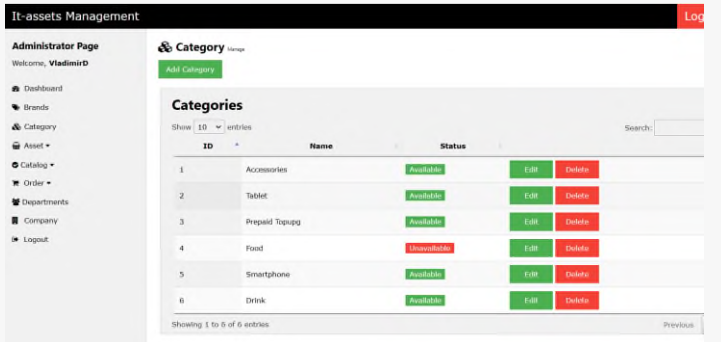
Рисунок 3.5 Карта сайта

Классификация ИТ-оборудования и услуг изображена на схеме каталога, представленной на рисунке Г.1.1 приложения Г. Схема демонстрирует логическую организацию оборудования и сопутствующих услуг для удобства навигации и управления ИТ-активами [5].

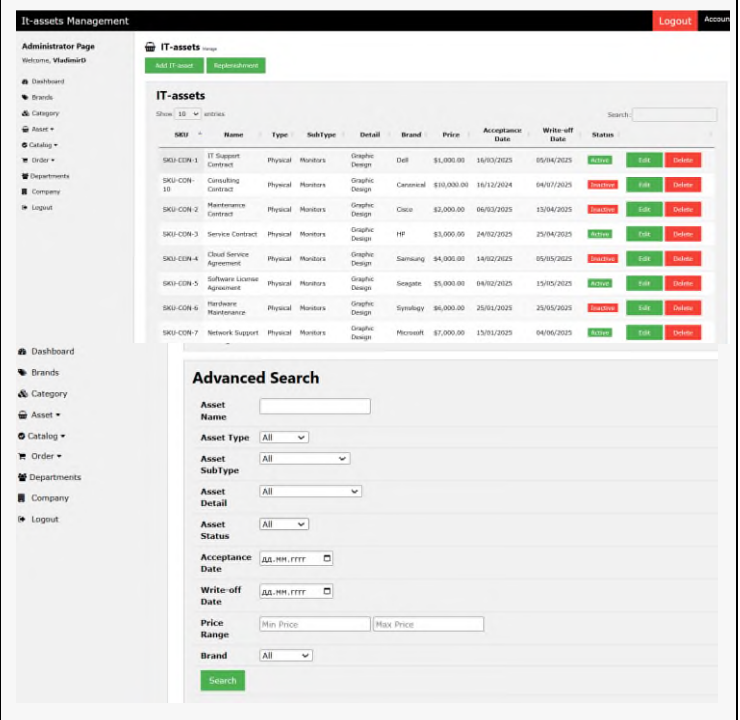
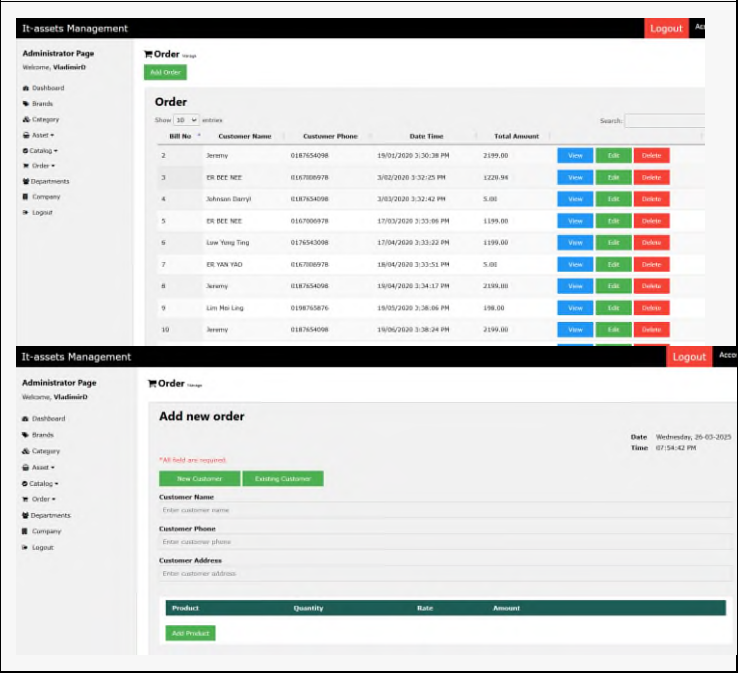
На основании структурной и логической схемы системы были разработаны интерфейсные формы, охватывающие все сценарии работы пользователей: *главная страница; форма авторизации; форма регистрации; форма управления брендами (Brands); форма управления категориями (Category); каталог активов (Catalog); форма управления заявками (Orders); форма управления отделами (Departments); профиль пользователя (User Profile); страница компании (Company).*

Разработанные формы позволяют выполнять основные функции системы. Для получения полного представления об интерфейсе и работе каждого раздела системы эти формы представлены в таблице 3.2.

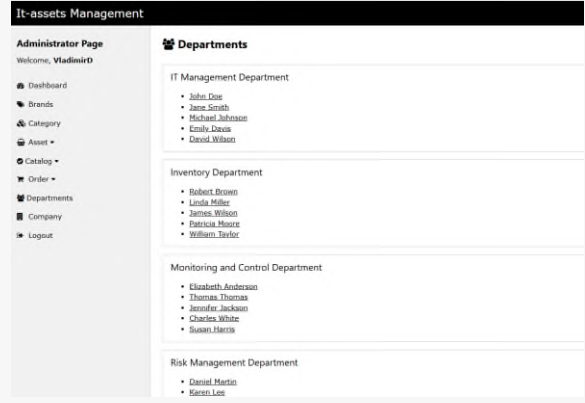
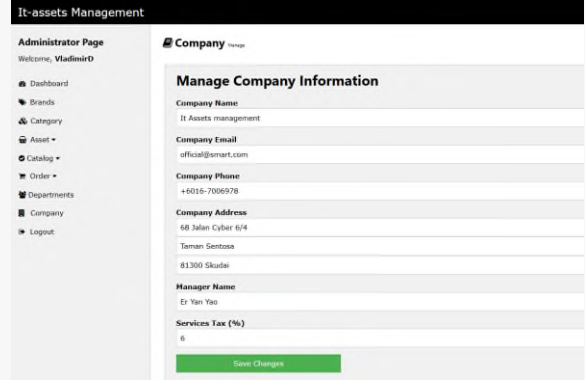
Таблица 3.2 – Элементы интерфейса веб-приложения

Раздел	Функции	Основной экран
Главная страница	Просмотр актуальной информации по задачам и событиям системы. Графики и сводка по активам, заявкам, статусам.	
Бренды	Управление брендами оборудования. Список брендов с возможностью добавления/редактирования.	
Виды активов	Управление категориями активов (физические, виртуальные). Древоподобная структура категорий (оборудование, ПО, лицензии и т.д.).	

Продолжение таблицы 3.2

Каталог активов	Просмотр всех активов с фильтрацией по типам и подкатегориям. Детализированный каталог (компьютеры, сетевые устройства, ПО, медиафайлы и т.п.).	
Заявки	Создание, просмотр и редактирование заявок (поддержка, оборудование). Таблица заявок с фильтрами по статусу, приоритету, дате.	

Продолжение таблицы 3.2

<i>Отделы</i>	Управление отделами и пользователями (для администратора). Список отделов с привязкой пользователей и их активов.	
<i>Профиль пользователя</i>	Редактирование персональных данных пользователя. Форма с полями: имя, <u>email</u>, телефон, отдел.	
<i>Информация о компании</i>	Просмотр информации о компании	

Распишем подробно каждую из разработанных форм.

Формы авторизации и регистрации пользователей веб-приложения предназначены для контроля доступа и управления учётными данными. Форма регистрации позволяет новым пользователям создать свою уникальную учётную запись. Форма авторизации используется для подтверждения личности уже зарегистрированным пользователям. Таким образом, эти формы разграничивают права доступа.

Взаимодействие с базой данных происходит с помощью SqlConnection. Стоит отметить, что строка подключения хранится не напрямую в коде, а извлекается из конфигурационного файла Web.config с помощью класса

ConfigurationManager. Подключение происходит с помощью представленного ниже кода:

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

При нажатии на кнопку входа выполняется следующий SQL-запрос:

```
SELECT COUNT(*) FROM Users WHERE Username = @username AND PasswordHash  
= @passwordHash)
```

Введённый пароль хешируется с использованием SHA-256, затем сравнивается с хешем из базы данных. Происходит также предотвращение SQL-инъекций. Если пользователь существует, то создаётся сессия, где сохраняются необходимые данные для пользователя и пользователь перенаправляется на главную страницу. В противном случае выводится сообщение об ошибке и закрывается соединение с базой данных.

При регистрации сначала проверяется корректность введённых данных, затем аналогично хешируется введённый пароль. С помощью представленного ниже запроса в базу данных передаются логин, полученный хеш пароля, email и роль пользователя:

```
INSERT INTO Users (Username, PasswordHash, Email, Role) VALUES (@username,  
@passwordHash, @email, @role)
```

При успешной регистрации осуществляется переход на форму ввода, в противном случае получаем сообщение об ошибке.

На рисунке 3.6 представлена форма авторизации пользователя. Форму регистрации можно увидеть на рисунке Г1.2 приложения Г.

IT-assets Management

VladimirD

...

User

Login

Create New Account

© 2024-2025 Vladimir Demin

Рисунок 3.6 Форма авторизации

Листинги кода аутентификации и регистрации пользователя приведены в подразделах Г2.1 и Г2.2 приложения Г.

Главная страница веб-приложения позволяет видеть пользователю отчётность по имеющимся активам и перемещаться по другим страницам, на этой странице описаны ключевые элементы класса `im_order_Default`, который реализует основную логику для обработки операций, связанных с графиками и заявками.

При каждой загрузке страницы, вызывается метод `Page_Load`, определяющий тип запроса и инициализирующий метрики:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        InitializeDashboardMetrics();
        PopulateSelectionDropdowns();
    }
    LoadTimeSensitiveData();
}
```

Главная страница веб-сайта представлена на рисунке 3.7.

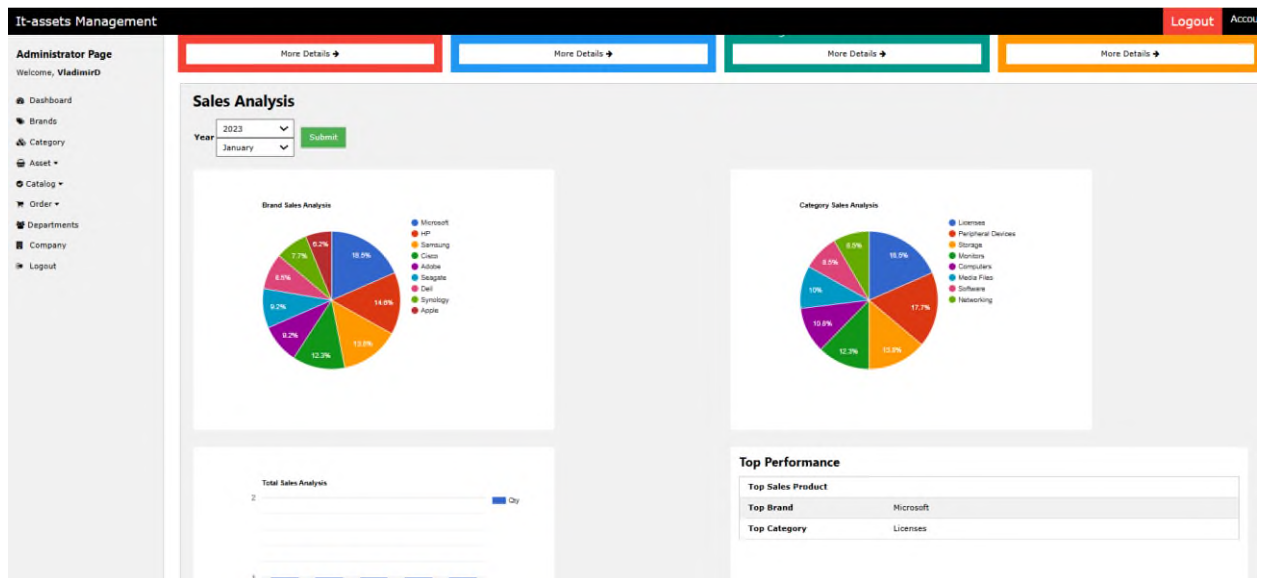


Рисунок 3.7 Главная страница

Основные функции главной страницы:

- 1) Происходит загрузка и обновление отчётов по текущим активам через метод `updateCount`, который выполняет SQL-запросы к базе данных.
- 2) Проверяются и устанавливаются новые значения выпадающих списков.
- 3) Регулярное обновление и загрузка новых данных.

Метод `btnDateSubmit_Click` используется для смены года и месяца, статистику которого в данный момент смотрит пользователь. После нажатия страница обновляется, и пользователь видит данные за новый отрезок времени.

```
protected void btnDateSubmit_Click(object sender, EventArgs e)
{
    int selectedYear = int.Parse(ddlYears.SelectedValue);
    int selectedMonth = int.Parse(ddlMonths.SelectedValue);

    Response.Redirect($"im_order_Default.aspx?year={selectedYear}&month={selectedMonth}");
}
```

Метод `btnReplenish_Command` отправляет пользователя на страницу складирования активов.

```
protected void btnReplenish_Command(object sender, CommandEventArgs e)
{
    int productId = Convert.ToInt32(e.CommandArgument);
    Session["ReplenishProductID"] = productId;
    Response.Redirect("InventoryManagement.aspx?action=restock");
}
```

Форма добавления актива предоставляет пользователю возможность вводить информацию о новом продукте. В неё включаются такие данные, как название, артикул (SKU), цена, количество на складе, описание, принадлежность к бренду, категория и текущий статус.

При сохранении введённых данных осуществляется запись новой записи в соответствующую таблицу базы данных. Значения, указанные пользователем в полях формы, подставляются в параметры этого запроса. После успешного добавления актива система автоматически перенаправляет пользователя на страницу со списком всех активов. Этот список представлен в табличном формате и содержит основные сведения: название, артикул, цену и статус каждого продукта.

Форма редактирования использует тот же принцип работы, что и форма добавления, но предназначена для изменения существующих записей. При её открытии система проверяет наличие идентификатора актива, чтобы загрузить текущие данные для корректировки. Сохранение изменений в этом случае приводит к обновлению информации в базе данных.

Как при добавлении, так и при редактировании реализован механизм обработки возможных ошибок. В случае возникновения проблем в работе системы пользователю отображается подробное сообщение об ошибке, помогающее понять причину сбоя.

Представления форм добавления и списка активов представлены на рисунках 3.8, 3.9.

ID	Name	SKU	Asset Tag	Price	Quantity	Description	Status	Brand	Category	Department
1	Laptop	SKU001	INV001	1000.00	10	High-quality laptop	Available	Dell	hardware	IT
2	Desktop Computer	SKU002	INV002	800.00	5	Gaming desktop computer	Unavailable	HP	hardware	IT
3	Printer	SKU003	INV003	150.00	20	Laser printer	Available	Canon	hardware	Finance
4	Monitor	SKU004	INV004	200.00	15	27-inch monitor	Available	LG	hardware	IT
5	Keyboard	SKU005	INV005	25.00	30	Mechanical keyboard	Available	Logitech	hardware	IT
6	Mouse	SKU006	INV006	15.00	50	Wireless mouse	Available	Logitech	hardware	IT
7	Router	SKU007	INV007	60.00	25	Wi-Fi router	Unavailable	Netgear	hardware	IT
8	Office 365 License	SKU008	INV008	120.00	100	Office productivity suite	Available	Microsoft	licenses	IT
9	Adobe Creative Cloud	SKU009	INV009	60.00	75	Creative software suite	Available	Adobe	licenses	Design
10	Project Management Software	SKU010	INV010	300.00	20	Team collaboration tool	Available	Atlassian	software	Project Management

Рисунок 3.8 Список IT-активов

Add new It-asset

*All field are required.

Asset Name

SKU

Price

Quantity

Description

Brands

Category

Status

Back
Save Changes

Рисунок 3.9 Добавление IT-активов

Форма каталога активов является главным интерфейсом для управления IT-активами. Она включает в себя три основные функции:

1. Просмотр существующих активов.
2. Добавление новых активов.
3. Модификацию текущих активов.

Список активов представлен в табличном виде с помощью соответствующих запросов с объединением таблиц:

```
SELECT a.id, a.name, b.title AS brand, c.name AS category
FROM assets a
JOIN brands b ON a.brand_id = b.id
JOIN categories c ON a.category_id = c.id
ORDER BY a.id DESC
```

При добавлении актива сервер формирует модальное окно с валидируемой формой. При подтверждении данные передаются через AJAX-запрос с помощью метода POST. Затем уже серверная часть выполняет запрос, вводя новый актив:

```
INSERT INTO assets (name, brand_id, category_id, status)
VALUES (@name, @brandId, @categoryId, 'active')
```

При редактировании активов значения подгружаются через отдельный GET-запрос. После внесения правок отправляется PATCH-запрос с изменениями. Сервер выполняет обновление только модифицированных полей:

```
UPDATE assets SET name = @newName WHERE id = @assetId
```

Представления форм каталога активов с расширенным поиском представлены на рисунках 3.10, 3.11.

SKU	Name	Type	SubType	Detail	Brand	Price	Acceptance Date	Write-off Date	Status	Actions
SKU-CON-1	IT Support Contract	Physical	Monitors	Graphic Design	Dell	\$1,000.00	30/03/2025	19/04/2025	Active	Edit, Delete
SKU-CON-10	Consulting Contract	Physical	Monitors	Graphic Design	Canonical	\$10,000.00	30/12/2024	18/07/2025	Inactive	Edit, Delete
SKU-CON-2	Maintenance Contract	Physical	Monitors	Graphic Design	Cisco	\$2,000.00	20/03/2025	29/04/2025	Inactive	Edit, Delete
SKU-CON-3	Service Contract	Physical	Monitors	Graphic Design	HP	\$3,000.00	10/03/2025	09/05/2025	Active	Edit, Delete

Рисунок 3.10 Форма каталога активов

Рисунок 3.11 Расширенный поиск активов в каталоге

Форма учёта категорий предназначена для управления категориями в системе. Этот интерфейс позволяет пользователям регистрировать новые

категории, обновлять существующие и проверять их на уникальность, что является важной частью управления структурой данных приложения.

Функция регистрации новой категории активируется пользователем и подготавливает форму для ввода данных. Процесс сохранения включает проверку на отсутствие дублирования имени категории в базе данных перед непосредственной записью новой позиции.

При обновлении информации о категории система идентифицирует запись для изменения по сохранённому идентификатору и применяет внесённые правки. Перечень категорий ИТ-активов представлен в таблицах Г3.1-Г3.5.

В приложении Г расположены таблицы Г3.1-Г3.5 с категориями ИТ-активов.

Форма учёта брендов 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

Эти функция аналогичны функциям учёта категорий и инициируют режим добавления нового бренда при нажатии кнопки "Добавить".

Редактирование бренда выполняется аналогично редактированию категории.

Форма учёта брендов представлена на рисунке 3.12.

Brands				
Show	10	▼	entries	
ID	Name	Status		
1	Huawei	Available	Edit	Delete
2	Apple	Available	Edit	Delete
5	Samsung	Available	Edit	Delete
6	Oppo	Available	Edit	Delete
7	Vivo	Available	Edit	Delete
9	Digi	Available	Edit	Delete
10	Maxis	Available	Edit	Delete

Рисунок 3.12 Список производителей (брендов активов)

Форма управления заявками – страница, позволяющая пользователям контролировать собственные заявки и следить за их статусом.

Пользователи могут включать необходимые активы в заявки, система при этом будет предотвращать дублирования. Также пользователям добавлена возможность менять количество активов внутри заявки. Если какой-нибудь из активов больше не нужен, реализована возможность исключить его из заявки.

Система самостоятельно вычисляет сумму стоимостей активов в заявке. Эта сумма также будет автоматически обновляться при каждом изменении внутри заявки. Код вычисления итоговой суммы представлен ниже:

```
private decimal CalculateOrderTotal()
{
    decimal subtotal = orderItems.Sum(i => i.UnitPrice * i.Quantity);
    decimal serviceFee = subtotal * 0.05m;
    decimal tax = (subtotal + serviceFee) * 0.2m;

    decimal total = subtotal + serviceFee + tax;

    ViewState["OrderTotal"] = total;
    lblTotal.Text = total.ToString("C");
    return total;
}
```

При загрузке страницы форма очищает все данные от предыдущих сеансов пользователя.

Сразу после оформления заявки система уменьшает количество активов на складе на то число, которое было указано в заявке.

Вид формы управления заявками представлен на рисунке 3.13, на рисунке 3.14 представлена также форма, с помощью которой пользователь может сделать новую заявку.

Order						
Show 10 entries		Search:				
Bill No	Customer Name	Customer Phone	Date Time	Total Amount	Status	
2	Jeremy	0187654098	19/01/2020 3:30:38 PM	2199.00	In Progress	View Edit Delete
3	ER BEE NEE	0167006978	3/02/2020 3:32:25 PM	1220.94	Completed	View Edit Delete
4	Johnson Darryl	0187654098	3/03/2020 3:32:42 PM	5.00	Cancelled	View Edit Delete

Рисунок 3.13 Форма управления заявками

It-assets Management
Logout

Administrator Page
Welcome, VladimirD

Dashboard
Brands
Category
Asset
Order
Report
Company
Logout

Order

Add new order

Date: четверг, 05-12-2024
Time: 04:10:24

*All field are required.

New Customer Existing Customer

Customer Name
Enter customer name

Customer Phone
Enter customer phone

Customer Address
Enter customer address

Product	Quantity	Rate	Amount
Активация Windows			

Add Product

Рисунок 3.14 Добавление новой заявки

Форма отделов отображает иерархическую структуру предприятия, предоставляя возможность просмотра сотрудников в рамках каждого подразделения. Данные загружаются из базы данных путём объединения информации из таблиц отделов и сотрудников, с упорядочиванием по названиям подразделений и фамилиям работников.

Для визуализации структуры используется компонент Repeater, позволяющий отображать отделы с возможностью раскрытия списка

сотрудников внутри каждого из них. Это обеспечивает наглядное представление организационной схемы.

Вид формы отделов представлен на рисунке 3.15.

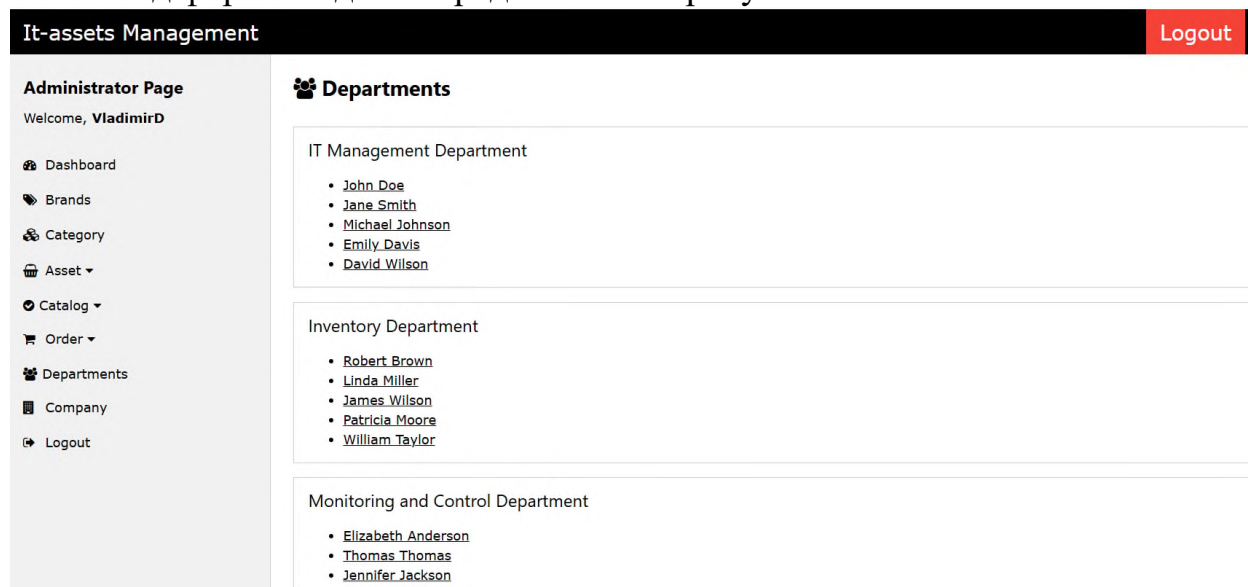


Рисунок 3.15 Форма отделов

Форма профиля предоставляет пользователям возможность просматривать и редактировать свои персональные данные. При загрузке страницы данные извлекаются из базы данных с использованием параметризованного запроса.

Форма включает следующие функциональные элементы:

- поля для редактирования личной информации;
- список последних заявок пользователя;
- возможность смены пароля.

Процесс обновления личных данных реализован с использованием механизма транзакций для обеспечения надёжности и согласованности информации.

Вид формы профиля представлен на рисунке 3.16.

Administrator Page
Welcome, VladimirD

- Dashboard
- Brands
- Category
- Asset
- Catalog
- Order
- Departments
- Company
- Logout

User Profile

User Information

Username: VladimirD

Role: User

Department:

Full Name: Vladimir Demin

Email: john.doe@example.com

Phone: 123-456-7890

Department: IT Management Department

Save

User Orders

Order ID	Customer Name	Customer Phone	Date Time	Total Amount
1	VladimirD	+375291218555	9/04/2025 8:42:53 PM	100.00
2	VladimirD	+375291218555	9/04/2025 8:42:53 PM	200.00

Рисунок 3.16 Форма профиля

Форма компании позволяет администраторам управлять основной информацией о предприятии. Данные загружаются из таблицы company при инициализации страницы.

Особенности реализации:

- все поля формы привязаны к данным из базы;
- подтверждение изменений через модальное окно;
- валидация вводимых данных на стороне сервера.

Вид формы компании представлен на рисунке 3.17.

Administrator Page
Welcome, VladimirD

- Dashboard
- Brands
- Category
- Asset
- Catalog
- Order
- Departments
- Company
- Logout

Company

Manage Company Information

Company Name: IT-Assets Management

Company Email: official@smart.com

Company Phone: +6016-7006978

Company Address: 68 Jalan Cyber 6/4, Taman Sentosa, 81300 Skudai

Manager Name: Er Yan Yao

Services Tax (%): 6

Save Changes

Рисунок 3.17 Форма информации о компании

Все формы используют единый стиль оформления и реализуют паттерн MVC для разделения логики представления и обработки данных.

3.4 Описание действий в системе учета и контроля IT-активов в соответствии с ролями пользователей

Данное описание предназначено для пользователей и администраторов веб-приложения управления IT-активами. Оно описывает функциональные возможности системы, роли пользователей и их права доступа.

Таблица 3.3 – Описание ролей пользователей системы

Роль	Основные функции	Ограничения
Пользователь (User)	Просмотр списка своих активов Редактирование информации о своих активах Просмотр статистики по своим активам Просмотр и редактирование своего профиля.	Не может добавлять или удалять активы Не имеет доступа к данным других пользователей.
Администратор (Admin)	Управление всеми активами (добавление, редактирование, удаление) Просмотр и управление данными всех пользователей Генерация отчетов по активам Управление категориями, брендами и отделами Настройка системы	Нет ограничений на доступ к данным.
IT-менеджер (IT-manager)	Управление активами (добавление, редактирование, удаление) Просмотр статистики по всем активам Генерация отчетов Управление категориями и брендами	Не может редактировать данные пользователей.

Описание действий пользователя:

1. Вход

Для входа в систему следует открыть страницу авторизации, ввести логин и пароль, затем выбрать роль «User», в конце нажимаем «Login».

2. Просмотр своих активов

После успешного входа для того, чтобы увидеть свои закреплённые активы, следует открыть раздел «IT-assets».

3. Создание заявки

Чтобы создать новую заявку, нужно перейти в раздел «Заявки» и нажать «Создать заявку». Далее заполнить все обязательные поля, выбрать актив. Далее завершить процесс, нажав «Отправить».

4. Просмотр созданной заявки

Все заявки с подробностями доступны в разделе «Заявки». Для просмотра полной информации следует нажать на название нужной заявки. Если текущий статус заявки позволяет вносить правки, можно воспользоваться кнопкой «Редактировать».

5. Поиск заявок

В разделе «Заявки» для удобного поиска можно применять фильтры по статусу заявки, её приоритету, дате создания или типу. Быстро найти заявку, по ключевым словам, поможет строка поиска.

6. Личные данные

Обновить личные данные можно в разделе «User Profile». Для этого следует внести изменения в своё имя, поле электронной почты, телефон и отдел, затем сохраните их.

7. Просмотр статистики

Статистику по собственным активам можно увидеть на главной странице.

Описание действий администратора:

1. Вход

Для входа в систему следует открыть страницу авторизации, ввести логин и пароль, затем выбрать роль «Admin», в конце нажимаем «Login».

2. Управление IT-активами

Управление IT-активами осуществляется в разделе «IT-assets». Для добавления нового актива нажимаем кнопку «Add IT-asset», для изменения существующего существует кнопка «Edit», где будет осуществлён переход на форму, в которой можно изменить абсолютно все атрибуты выбранного актива, там же есть кнопка «Delete» для удаления актива.

3. Создание заявки

Чтобы создать новую заявку, нужно перейти в раздел «Заявки» и нажать «Создать заявку». Далее заполнить все обязательные поля, выбрать актив, выбрать приоритет. При необходимости можно выбрать пользователя, от чьего имени будет создаваться заявка. Далее завершить процесс, нажав «Отправить».

4. Просмотр созданной заявки

В отличие от пользователя, можно помимо просмотра, менять статусы заявок: «В обработке», «Принята», «Отклонена». Можно назначить ответственного или изменить приоритет заявки.

5. Удаление заявок

Нужно перейти в раздел «Заявки», выбрать необходимую заявку и нажать «Удалить», подтвердив действие.

6. Поиск и фильтрация заявок

Для поиска и фильтрации используются фильтры и строка поиска.

7. Управление отделами

Управление отделами осуществляется в разделе «Departments», где представлен список отделов и пользователей. Для того, чтобы редактировать данные, нужно нажать и перейти на нужного пользователя.

8. Работа с типами и брендами активов

Работа с категориями и брендами активов ведётся в разделах «Category» и «Brands». Там представлены кнопки для добавления, редактирование и удаление типов или брендов ИТ-активов.

9. Уведомления о заявках

Администратор будет регулярно получать уведомления на адрес своей электронной почты при создании, изменении статуса и написании комментария к заявке.

Описание действий ИТ-менеджера:

1. Вход

Для того, чтобы войти в систему, требуется открыть страницу авторизации, ввести свой логин и пароль, выбрать роль «ИТ-manager», в конце нажимаем «Login».

2. Управление активами

Управление аналогично тому, что представлено в руководстве администратора.

3. Создание заявки

Управление аналогично тому, что представлено в руководстве администратора за исключением того, что можно выбирать других пользователей.

4. Просмотр и редактирование заявок

Есть возможность просматривать и управлять всеми заявками в разделе «Заявки». При клике, будут открыты все детали заявки. Кнопка «Редактировать» позволяет изменить статус заявки, назначить ответственного человека к этой заявке, добавить комментарий или поменять приоритет.

5. Удаление заявок

Для удаления заявки в разделе «Заявки» нужно найти её и нажать «Удалить», подтвердив действие.

6. Фильтрация и поиск заявок

Искать нужные для мониторинга и редактирования заявки можно искать с помощью строки поиска и фильтров.

7. Просмотр статистики

Общая статистика по всем доступна на главной странице и странице «Report».

8. Уведомления о заявках

Происходит аналогично тому, что представлено в руководстве администратора.

3.5 Выводы

В третьей главе дипломной работы были получены следующие результаты:

1. Определен технологический стек для разработки системы учета IT-активов, включающий серверные (ASP.NET, C#, MS SQL Server) и клиентские (HTML5, CSS3, JavaScript, Bootstrap) технологии.

2. Спроектирована архитектура системы на основе диаграмм UML (вариантов использования, компонентов, развертывания и последовательностей), что позволило определить функциональные модули и их взаимодействие.

3. Разработан пользовательский интерфейс, включающий формы для управления активами, заявками, пользователями и отчетностью, с реализацией CRUD-операций, валидации данных и адаптивного дизайна.

4. Реализована система ролевого доступа, определяющая права и действия пользователей (User, Admin, IT-manager).

ЗАКЛЮЧЕНИЕ

В процессе выполнения дипломной работы были получены следующие результаты:

1. Проведен обзор и анализ современных систем управления ИТ-активами предприятий, в результате которого выявлены их преимущества и недостатки существующих на рынке ИТАМ и сформированы требования к системе.

2. Выполнено функциональное моделирование бизнес-процессов учета и контроля ИТ-активов предприятия, что позволило разработать проект технологического процесса автоматизированного управления активами.

3. Разработана БД «Учёт и контроль ИТ-активов» системы управления ИТ-активами предприятия и выполнена её программная реализация на платформе Microsoft SQL Server, а также разработаны запросы по поиску, анализу ИТ-активов, контролю и управлению ролями пользователей.

4. Спроектирован пользовательский интерфейс к БД с использованием технологий Microsoft Visual Studio .NET и ASP.NET.

5. Реализованы функциональные подсистемы, включая управление активами, категориями, брендами и обработку заявок. В ходе тестирования подтверждена работоспособность системы, что соответствует результату, полученному на основе теоретических исследований и испытаний созданного экспериментального образца технологии.

6. Составлено описание действий пользователей системы для обеспечения корректного использования и дальнейшего сопровождения, а также разработаны методические рекомендации действий пользователей.

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

Разработанная ИТАМ может быть применена в малом и среднем бизнесе, управляющем ИТ-инфраструктурой: в ИТ-отделах для учета оборудования и лицензий, сервисных центрах для автоматизации обработки заявок, отделах закупок для контроля остатков, а также в аналитических подразделениях для генерации отчетов по эффективности использования ресурсов.

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>.
11. ГОСТ Р ИСО/МЭК 1-1-2021 «Информационные технологии. Управление ИТ-активами. Часть1. Системы управления ИТ-активами. Требования».
12. Система учёта и контроля ИТ-активов предприятия / Демин В. О. // Новые математические методы и компьютерные технологии в проектировании, производстве и научных исследованиях : тез. докл. XXVIII Республиканской науч. конф., 17–19 марта. 2025 г., г. Гомель : в 2 ч. / ГГУ имени Франциска Скорины. - Гомель, 2025. Ч.1.-С.161-162.

ПРИЛОЖЕНИЯ

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 настраиваемость	Высокая	Высокая	Ограниченная	Высокая	Ограниченная
Методы добавления активов	Вручную	Автоматически	Ручное добавление/ загрузка через CSV/ WMI	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
Отчеты	Настраиваемые, детализированные	Полные отчеты по лицензиям и использованию ПО	Простой модуль отчетности, автоматизация отчетов	Настраиваемые шаблоны отчетов	Встроенная отчетность

Продолжение таблицы А1

GPS-отслеживание	Да	Нет	Нет	Нет	Нет
Интеграция с другими системами	Да(но через API)	Да	Да	Да	Да
Клиентская поддержка	Высокое качество, оперативная реакция	Доступная	Высокое качество, оперативная реакция	Высокое качество, оперативная реакция	Доступная
Управление лицензиями	Нет автоматического управления, ручное управление	Полное управление лицензиями и ПО, автоматическое отслеживание	Нет	Нет	Полное управление лицензиями и ПО, автоматическое отслеживание
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 фиксированной стоимостью
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

Б1– Концептуальная модель и ER-диаграмма к базе данных

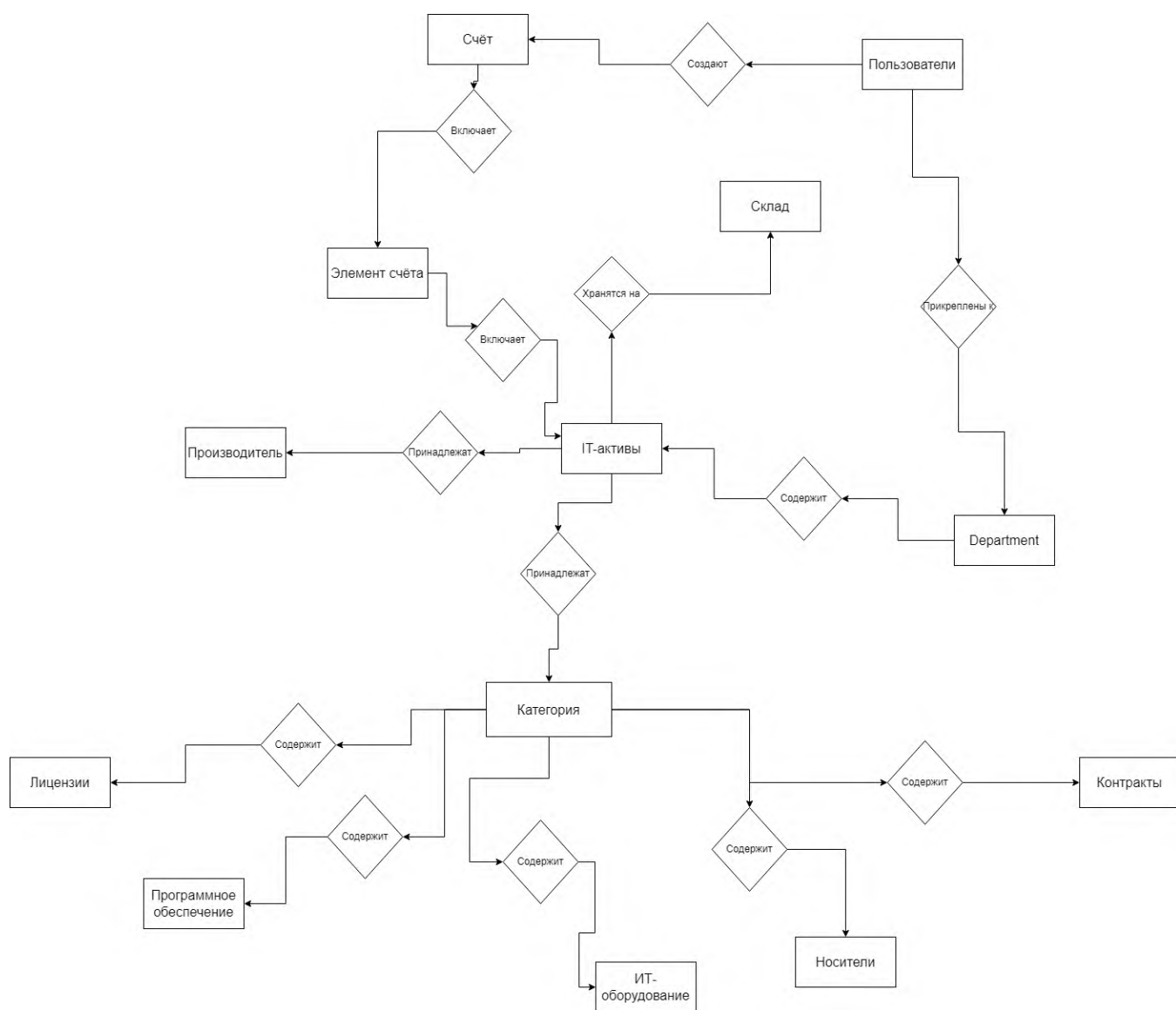


Рисунок Б1.1 Концептуальная модель

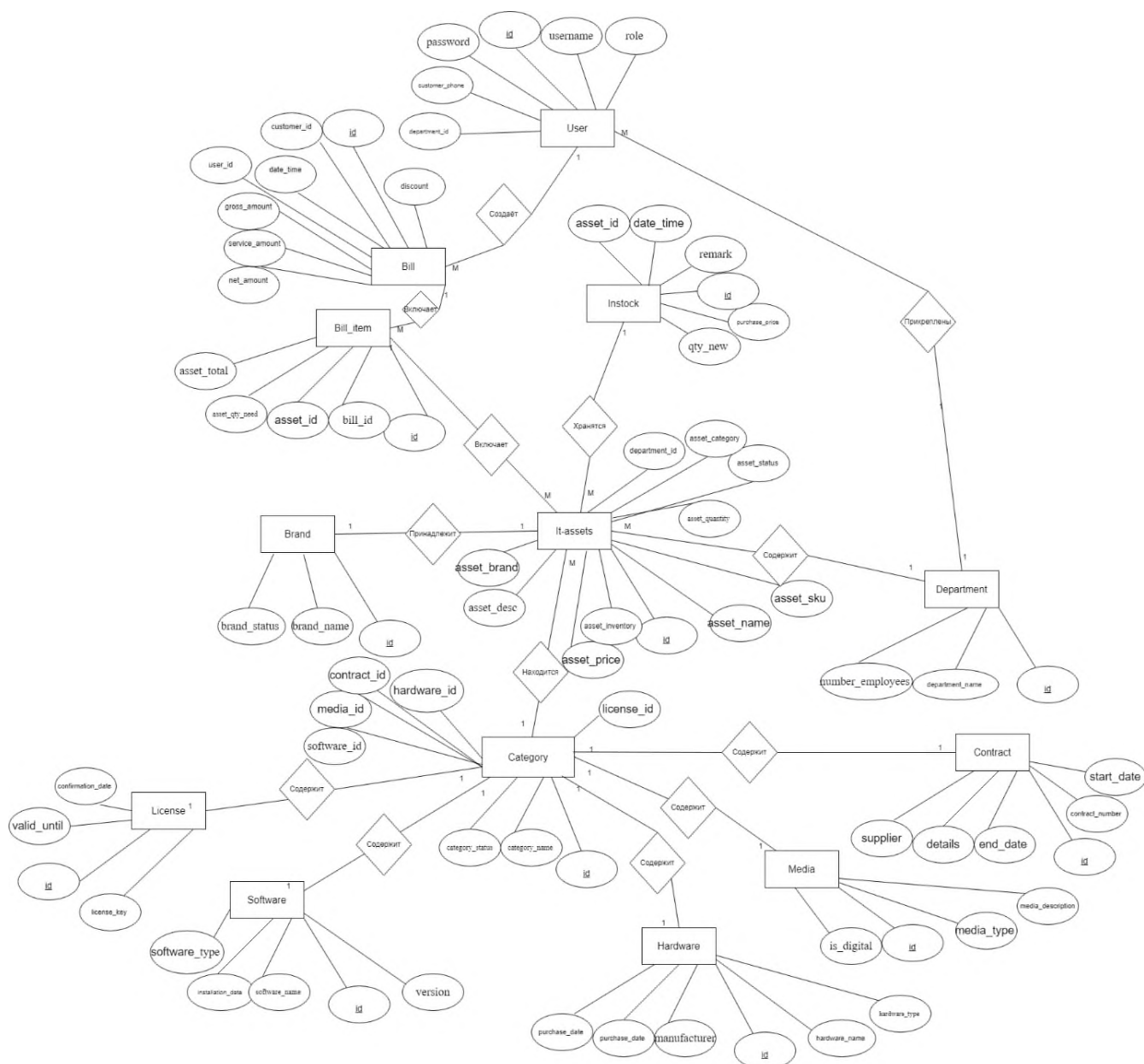


Рисунок Б1.2 Диаграмма в нотации Чена (ER-диаграмма)

Б2– Структура реляционных отношений

Таблица Б2.1 – 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

Продолжение таблицы Б2.1

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 Б2.2 – 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 ИТ-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 Б2.3 – 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 Б2.4 – 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

Продолжение таблицы Б2.4

Дата установки	SW_INSTALL_DATE	date(3)	обязательное поле
----------------	-----------------	---------	-------------------

Таблица Б2.5 – 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 Б2.6 – 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 Б2.7 – Схема отношения ЛИЦЕНЗИИ (License)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	L_ID	int(4)	первичный ключ
Ключ лицензии	L_KEY	nvarchar(100)	обязательное поле
Дата подтверждения	L_CONFIRM_DATE	date(3)	
Срок действия	L_VALID_UNTIL	date(3)	

Таблица Б2.8 – 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 Б2.9 – 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 Б2.10 – Схема отношения НА СКЛАДЕ (Instock)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	IS_ID	int(4)	первичный ключ
Идентификатор актива ИТ-	IS_ASSET_ID	int(4)	внешний ключ (к IT-Assets)

Продолжение таблицы Б2.10

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 Б2.11 – Схема отношения ИТ-АКТИВЫ (IT-Assets)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	IA_ID	int(4)	первичный ключ
Название ИТ-актива	IA_NAME	nvarchar(255)	обязательное поле
SKU ИТ-актива	IA_SKU	nvarchar(255)	обязательное поле
Инвентарный номер	IA_INVENTORY	nvarchar(255)	
Цена ИТ-актива	IA_PRICE	decimal(10, 2)	обязательное поле
Количество ИТ-активов	IA_QUANTITY	int(4)	обязательное поле
Описание ИТ-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 ИТ-актива	IA_STATUS	int(4)	обязательное поле
Идентификатор отдела	IA_DEP_ID	int(4)	внешний ключ (к Department)

Таблица Б2.12 – Схема отношения ПОЛЬЗОВАТЕЛИ (User)

Содержание поля	Имя поля	Тип, длина	Примечания
Идентификатор	U_ID	int(4)	первичный ключ
Имя пользователя	U_USERNAME	nvarchar(20)	обязательное поле
Пароль	U_PASSWORD	nvarchar(30)	обязательное поле

Продолжение таблицы Б2.12

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 Б2.13 – 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

В1 - Листинг создания базы данных

```
-- 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)
);

```

B2 - Листинг заполнения базы данных

```

-- 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')
```

В3 - Запросы поиска, анализа и контроля IT-активов

1. Запрос для поиска активных IT-активов определенного типа и бренда:

```

SELECT * FROM [IT-Assets]
WHERE asset_status = 1
AND asset_type = 'Physical'
AND asset_brand IN (SELECT Id FROM Brand WHERE brand_name IN ('Dell', 'HP'))
```

2. Поиск IT-активов с истекающей гарантией (в ближайшие 30 дней):

```

SELECT a.*, h.warranty_expiry
FROM [IT-Assets] a
JOIN Hardware h ON a.Id = h.hardware_id
WHERE h.warranty_expiry BETWEEN GETDATE() AND DATEADD(day, 30, GETDATE())
```

3. Запрос для поиска программного обеспечения с истекающими лицензиями:

```

SELECT s.*, l.valid_until
FROM Software s
```

```
JOIN License l ON s.Id = l.software_id  
WHERE l.valid_until BETWEEN GETDATE() AND DATEADD(month, 3, GETDATE())
```

4. Поиск самых дорогих IT-активов по категориям:

```
SELECT c.category_name, MAX(a.asset_price) as max_price  
FROM [IT-Assets] a  
JOIN Category c ON a.asset_category = c.Id  
GROUP BY c.category_name  
ORDER BY max_price DESC
```

5. Запрос для анализа распределения IT-активов по отделам:

```
SELECT d.department_name, COUNT(a.Id) as asset_count, SUM(a.asset_price) as  
total_value  
FROM [IT-Assets] a  
JOIN Department d ON a.department_id = d.Id  
GROUP BY d.department_name  
ORDER BY asset_count DESC
```

6. Поиск дублирующихся IT-активов по SKU:

```
SELECT asset_sku, COUNT(*) as duplicate_count  
FROM [IT-Assets]  
GROUP BY asset_sku  
HAVING COUNT(*) > 1
```

7. Запрос для поиска неиспользуемых IT-активов (не связанных с контрактами):

```
SELECT a.*  
FROM [IT-Assets] a  
LEFT JOIN Contract c ON a.Id = c.asset_id  
WHERE c.Id IS NULL AND a.asset_status = 1
```

8. Анализ динамики закупок IT-активов по месяцам:

```
SELECT YEAR(i.date_time) as year, MONTH(i.date_time) as month,  
COUNT(*) as purchase_count, SUM(i.purchase_price) as total_spent  
FROM Instock i  
GROUP BY YEAR(i.date_time), MONTH(i.date_time)  
ORDER BY year, month
```

9. Поиск IT-активов с низким остатком на складе:

```
SELECT a.*, i.qty_new as stock_quantity  
FROM [IT-Assets] a  
JOIN Instock i ON a.Id = i.asset_id  
WHERE i.qty_new < 5
```

10. Запрос для анализа контрактов с истекающим сроком:

```
SELECT c.*, DATEDIFF(day, GETDATE(), c.end_date) as days_remaining  
FROM Contract c  
WHERE c.end_date BETWEEN GETDATE() AND DATEADD(month, 3, GETDATE())  
ORDER BY days_remaining
```

11. Поиск IT-активов по сложному критерию (тип, подтип, цена):

```
SELECT * FROM [IT-Assets]
WHERE asset_type = 'Physical'
AND asset_subtype IN ('Computers', 'Networking')
AND asset_price BETWEEN 500 AND 2000
AND asset_status = 1
```

12. Анализ лицензий по типу программного обеспечения:

```
SELECT s.software_type, COUNT(l.Id) as license_count,
      MIN(l.valid_until) as earliest_expiry
FROM Software s
JOIN License l ON s.Id = l.software_id
GROUP BY s.software_type
```

13. Поиск IT-активов, принятых на учет в определенный период:

```
SELECT * FROM [IT-Assets]
WHERE acceptance_date BETWEEN '2023-01-01' AND '2023-12-31'
ORDER BY acceptance_date DESC
```

14. Поиск IT-активов с наибольшим количеством на складе:

```
SELECT a.*, i.qty_new
FROM [IT-Assets] a
JOIN Instock i ON a.Id = i.asset_id
ORDER BY i.qty_new DESC
```

15. Анализ медиа-носителей по типу и объему:

```
SELECT media_type, capacity, COUNT(*) as count
FROM Media
GROUP BY media_type, capacity
ORDER BY media_type, count DESC
```

16. Поиск IT-активов без привязанных лицензий:

```
SELECT a.*
FROM [IT-Assets] a
LEFT JOIN License l ON a.Id = l.asset_id
WHERE l.Id IS NULL AND a.asset_type = 'Virtual'
```

17. Запрос для анализа стоимости IT-активов по отделам:

```
SELECT d.department_name, SUM(a.asset_price) as total_value
FROM [IT-Assets] a
JOIN Department d ON a.department_id = d.Id
GROUP BY d.department_name
ORDER BY total_value DESC
```

18. Поиск устаревшего оборудования (старше 3 лет):

```
SELECT a.*, h.purchase_date
FROM [IT-Assets] a
JOIN Hardware h ON a.Id = h.hardware_id
WHERE h.purchase_date < DATEADD(year, -3, GETDATE())
```

19. Анализ контрактов по поставщикам:

```

SELECT supplier, COUNT(*) as contract_count,
        AVG(DATEDIFF(day, start_date, end_date)) as avg_duration
FROM Contract
GROUP BY supplier
ORDER BY contract_count DESC

```

20. Поиск IT-активов с определенными характеристиками (комплексный):

```

SELECT a.*
FROM [IT-Assets] a
JOIN Brand b ON a.asset_brand = b.Id
JOIN Category c ON a.asset_category = c.Id
WHERE b.brand_name = 'Dell'
AND c.category_name = 'Hardware'
AND a.asset_price > 1000
AND a.asset_status = 1

```

21. Запрос для анализа использования цифровых носителей:

```

SELECT m.media_type, COUNT(*) as count,
        SUM(CASE WHEN m.is_digital = 1 THEN 1 ELSE 0 END) as
        digital_count
FROM Media m
GROUP BY m.media_type

```

22. Поиск IT-активов с истекшим сроком списания:

```

SELECT * FROM [IT-Assets]
WHERE write_off_date < GETDATE()
AND asset_status = 1

```

23. Анализ закупок по месяцам и категориям:

```

SELECT c.category_name,
        YEAR(i.date_time) as year,
        MONTH(i.date_time) as month,
        COUNT(*) as purchase_count,
        SUM(i.purchase_price) as total_spent
FROM Instock i
JOIN [IT-Assets] a ON i.asset_id = a.Id
JOIN Category c ON a.asset_category = c.Id
GROUP BY c.category_name, YEAR(i.date_time), MONTH(i.date_time)
ORDER BY year, month, category_name

```

24. Поиск дублирующихся лицензий:

```

SELECT license_key, COUNT(*) as duplicate_count
FROM License
GROUP BY license_key
HAVING COUNT(*) > 1

```

25. Запрос для анализа IT-активов по статусу:

```

SELECT
        CASE asset_status
        WHEN 1 THEN 'Active'

```

```

        ELSE 'Inactive'
    END as status,
    COUNT(*) as count,
    SUM(asset_price) as total_value
FROM [IT-Assets]
GROUP BY asset_status

```

26. Поиск IT-активов с наибольшим сроком эксплуатации:

```

SELECT a.*, h.purchase_date,
       DATEDIFF(month, h.purchase_date, GETDATE()) as months_in_use
FROM [IT-Assets] a
JOIN Hardware h ON a.Id = h.hardware_id
ORDER BY months_in_use DESC

```

27. Анализ лицензий по сроку действия:

```

SELECT
    CASE
        WHEN DATEDIFF(day, GETDATE(), valid_until) < 0 THEN 'Expired'
        WHEN DATEDIFF(day, GETDATE(), valid_until) < 30 THEN 'Expiring soon'
        ELSE 'Active'
    END as license_status,
    COUNT(*) as count
FROM License
GROUP BY
    CASE
        WHEN DATEDIFF(day, GETDATE(), valid_until) < 0 THEN 'Expired'
        WHEN DATEDIFF(day, GETDATE(), valid_until) < 30 THEN 'Expiring soon'
        ELSE 'Active'
    END

```

28. Поиск IT-активов с определенным сочетанием характеристик:

```

SELECT a.*, b.brand_name, c.category_name
FROM [IT-Assets] a
JOIN Brand b ON a.asset_brand = b.Id
JOIN Category c ON a.asset_category = c.Id
WHERE a.asset_type = 'Physical'
AND a.asset_subtype = 'Computers'
AND b.brand_name IN ('HP', 'Lenovo')
AND a.asset_price BETWEEN 800 AND 1500
AND a.asset_status = 1

```

29. Запрос для поиска самых популярных брендов:

```

SELECT b.brand_name, COUNT(a.Id) as asset_count
FROM [IT-Assets] a
JOIN Brand b ON a.asset_brand = b.Id
GROUP BY b.brand_name
ORDER BY asset_count DESC

```

30. Комплексный анализ IT-активов по всем параметрам:

```
SELECT
    a.asset_type,
    a.asset_subtype,
    b.brand_name,
    c.category_name,
    COUNT(*) as asset_count,
    SUM(a.asset_price) as total_value,
    AVG(a.asset_price) as avg_price
FROM [IT-Assets] a
JOIN Brand b ON a.asset_brand = b.Id
JOIN Category c ON a.asset_category = c.Id
GROUP BY
    a.asset_type,
    a.asset_subtype,
    b.brand_name,
    c.category_name
ORDER BY
    a.asset_type,
    a.asset_subtype,
    total_value DESC
```

ПРИЛОЖЕНИЕ Г

Г1 – Пользовательский интерфейс и элементы веб-приложения

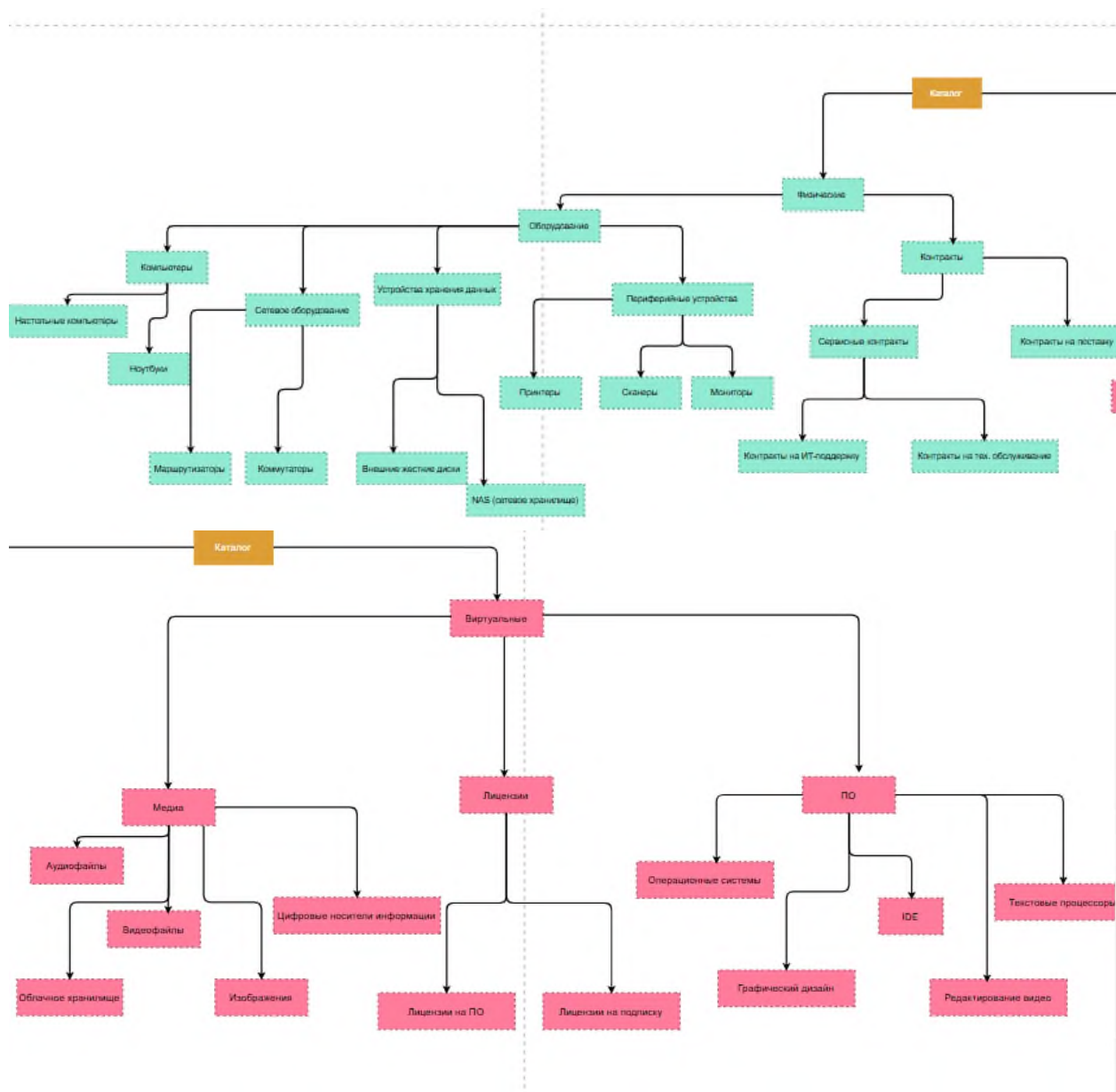
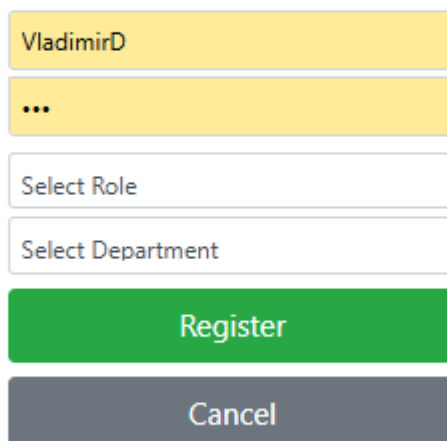


Рисунок Г1.1 Схема каталога

IT-assets Management



VladimirD

...

Select Role

Select Department

Register

Cancel

© 2024-2025 Vladimir Demin

Рисунок Г1.2 Форма регистрации

Г2 - Листинг авторизации и регистрации пользователя

Г2.1 - Листинг 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();
}

```

Г2.2 – Листинг регистрации 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();
}

```

Г3 – Заполненная база данных

ID	Software Name	Version	Software Type	Installation Date
1	Microsoft Office 365	2023	Productivity	2023-01-15
2	Adobe Photoshop	2022	Graphics Editing	2022-07-10
3	AutoCAD	2023	Design	2023-02-20
4	Zoom	2022	Communication	2022-05-12
5	Slack	2023	Collaboration	2023-03-01
6	Visual Studio	2022	Development	2022-04-24
7	Postman	2023	API Development	2023-01-30
8	SAS	2022	Analytics	2022-09-29
9	Notepad++	2023	Text Editing	2023-04-05
10	Git	Latest	Version Control	2023-05-01

Рисунок Г3.1 - Список доступного программного обеспечения

ID	Media Type	Description	Capacity	Is Digital
1	USB Drive	16GB USB Flash Drive	16GB	Yes
2	DVD	Backup Data Disk	4.7GB	No
3	External Hard Drive	1TB Storage Device	1TB	Yes
4	SSD	256GB SSD Drive	256GB	Yes
5	CD	Music Album	700MB	No
6	MicroSD Card	32GB MicroSD Card	32GB	Yes
7	Blu-Ray	Movie Disk	25GB	No
8	Flash Drive	8GB Flash Drive	8GB	Yes
9	Tape	Data Backup Tape	200GB	No
10	Network Attached Storage	NAS Device	4TB	Yes

Рисунок Г3.2 - Список доступных носителей

ID	Hardware Name	Hardware Type	Manufacturer	Purchase Date	Warranty Expiry
1	Dell XPS 15	Laptop	Dell	2021-12-01	2023-12-01
2	Cisco Router	Router	Cisco	2022-03-15	2024-03-15
3	HP LaserJet Pro M404n	Printer	HP	2022-06-10	2024-06-10
4	Apple iMac	Desktop	Apple	2021-11-20	2023-11-20
5	Samsung Galaxy Tab	Tablet	Samsung	2022-02-10	2024-02-10
6	Logitech MX Master 3	Mouse	Logitech	2023-04-01	2025-04-01
7	Samsung 27" Monitor	Monitor	Samsung	2023-05-15	2025-05-15
8	Lenovo ThinkPad X1	Laptop	Lenovo	2023-01-25	2025-01-25
9	Razer Gaming Keyboard	Keyboard	Razer	2022-08-05	2024-08-05
10	Dell UltraSharp 24"	Monitor	Dell	2022-09-15	2024-09-15

Рисунок Г3.3 - Список доступного оборудования

ID	License Key	Confirmation Date	Valid Until
1	ABCD-EFGH-IJKL-MNOP	2023-01-10	2024-01-10
2	QRST-UVWX-YZAB-CDEF	2022-06-22	2023-06-22
3	WXYZ-5678-QRST-1234	2023-02-15	2024-02-15
4	MNOP-1234-QRST-5678	2022-07-30	2023-07-30
5	ABCDE-12345-XYZ-7890	2023-03-01	2024-03-01
6	FGHIJ-67890-QRST-5432	2022-08-12	2023-08-12
7	KLMNO-09876-QRST-1234	2023-04-05	2024-04-05
8	QRST-13579-IJKL-24680	2022-09-18	2023-09-18
9	UVWXY-24680-ABCDE-13579	2023-05-10	2024-05-10
10	ZYXWV-13579-QWERT-24680	2022-10-20	2023-10-20

Рисунок Г3.4 - Список доступных лицензий

ID	Contract Number	Start Date	End Date	Details	Supplier
1	CON-2023-001	2023-01-01	2024-01-01	Annual IT Support	ABC Tech Ltd.
2	CON-2022-002	2022-04-01	2023-04-01	Software Licensing Agreement	XYZ Software Inc.
3	CON-2023-003	2023-02-01	2024-02-01	Cloud Services Agreement	CloudTech LLC
4	CON-2022-004	2022-05-01	2023-05-01	Hardware Purchase Agreement	Hardware Co.
5	CON-2023-005	2023-03-01	2024-03-01	Consulting Services Agreement	Consulting Group
6	CON-2022-006	2022-08-01	2023-08-01	Annual Maintenance Contract	TechFix Inc.
7	CON-2023-007	2023-04-15	2024-04-15	Network Security Agreement	SecuriTech
8	CON-2022-008	2022-11-01	2023-11-01	Data Recovery Service	DataSafe Ltd.
9	CON-2023-009	2023-06-01	2024-06-01	Website Development Agreement	WebDev Inc.
10	CON-2023-010	2023-07-01	2024-07-01	Mobile App Development	AppCreators LLC

Рисунок Г3.5 - Список доступных контрактов