

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

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

Кафедра «Прикладная математика и информатика»
(наименование)

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

Бизнес-информатика
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Информационная система инвентаризации и учета программного и аппаратного обеспечения на предприятии»

Студент

С.В. Закамсков
(И.О. Фамилия)

(личная подпись)

Руководитель

канд. пед. наук, доцент, О.М. Гущина
(ученая степень, звание, И.О. Фамилия)

Тольятти, 2021

Аннотация

Выпускная работа посвящена разработке проекта информационной системы инвентаризации и учета программного и аппаратного обеспечения для АО «ЛИПСИЯ».

В первой главе ВКР дано описание функционирования АО «ЛИПСИЯ», которое включает технико-экономический анализ, определение основных проблем существующей модели.

Вторая глава содержит описание модели «Как есть», выполненной с использованием языка графического описания UML. Также представлены Use Case Diagram, Class Diagram. Моделирование предметной области завершено постановкой цели и задач проектирования с формированием требований к информационной системе. Также здесь представлено описание выбора и обоснование методов проектирования информационной системы.

В третьей главе приводится описание процесса создания функциональной модели компании «Как должно быть». В ней дано описание программного, аппаратного, сетевого и технологического обеспечения автоматизированной информационной системы. Завершается работа оценкой экономической эффективности реализации проекта.

В заключении приводятся выводы о проделанной работе.

Работа представляет собой законченный проект, который рассматривается на предмет внедрения в деятельность компании.

Оглавление

Введение.....	4
Глава 1 Анализ предметной области.....	6
1.1 Техничко-экономическая характеристика предметной области	6
1.1.1 Характеристика предприятия и его деятельность	6
1.1.2 Краткая характеристика подразделения и его видов деятельности.....	7
1.1.3 Сущность задачи автоматизации	8
1.2 Концептуальное моделирование предметной области	9
1.3 Постановка задачи	11
1.4 Анализ существующих разработок и обоснование выбора технологии проектирования	13
Глава 2 Разработка и реализация проектных решений	16
2.1 Логическое моделирование предметной области	16
2.2 Физическое моделирование АИС	26
Глава 3 Оценка и обоснование экономической эффективности проекта	35
3.1 Выбор и обоснование методики расчета экономической эффективности	35
3.2 Расчет показателей экономической эффективности проекта	38
Заключение	40
Список используемой литературы	42

Введение

В условиях современной экономики, и особенно в период экономического кризиса, все большее значение приобретает учет оборудования и более эффективное управление им на предприятии. Контроль состояния оборудования дает информацию о степени изношенности, позволяет принять решения по замене оборудования, ремонту, либо продлению его срока службы, переиспользованию.

Очень активно развивается концепция управления ИТ-активами (ИТАМ, IT Asset Management [26]) – область деятельности по планированию, учету и отслеживанию состояния ИТ-активов, обеспечивающих соответствующую ценность для основной деятельности организации, представляющая контроль финансовых потоков на протяжении жизненного цикла ИТ-активов, с учетом всех сопутствующих признаков (артефактов).

Острой проблемой является то, что у отдела ИТ и бухгалтерии разные цели и задачи по учету оборудования в организации: для отдела ИТ учет начинается уже на стадии планировании закупки оборудования, важен детальный учет (информация о комплектующих), важно расположение оборудования. Для бухгалтерии это все не важно.

Объект исследования: процесс учета программного и аппаратного обеспечения на предприятии.

Предмет исследования: оптимизация выбранного процесса на предприятии.

Цель работы: разработать систему учета программного и аппаратного обеспечения.

Задачи работы:

- смоделировать процесс «Как есть»,
- выявить недостатки существующей модели,
- смоделировать процесс «Как должно быть»,

- разработать логическую и физическую модели АИС,
- обосновать экономическую выгоду от внедрения АИС.

Метод исследования: методология объектно-ориентированного анализа и проектирования.

В главе 1 описаны характеристики компании и ее деятельности. На примере отдела ИТ рассматривается существующее состояние процесса («Как есть»), используемое программное обеспечение; выявляются проблемы и недостатки. Обосновываются предложения по устранению выявленных недостатков и внедрению новых технологий («Как должно быть»), выбирается архитектура, цели и задачи системы. Рассматриваются готовые решения.

Глава 2 посвящена разработке автоматизированной системы управления объектом исследования. Разрабатывается логическая и физическая модель данных, обозначаются документы с входной оперативной информацией, интерфейс ввода данных и вывода результатной информации. Прорабатываются функциональная схема проекта, сценарии взаимодействия с системой. Описывается структура проекта, выделяются программные модули.

В главе 3 выбирается и обосновывается методика расчета экономической эффективности. На основании описанной методики производится расчет показателей экономической эффективности. Принимается решение о дальнейшей разработке проекта.

Глава 1 Анализ предметной области

1.1 Техничко-экономическая характеристика предметной области

1.1.1 Характеристика предприятия и его деятельность

АО «ЛИПСИЯ» занимается проектированием и производством конвейерного и холодильного оборудования с 2001 года.

Сферой деятельности АО «ЛИПСИЯ» является проектирование и производство конвейерного и холодильного оборудования. Структура предприятия приведена на рисунке 1.

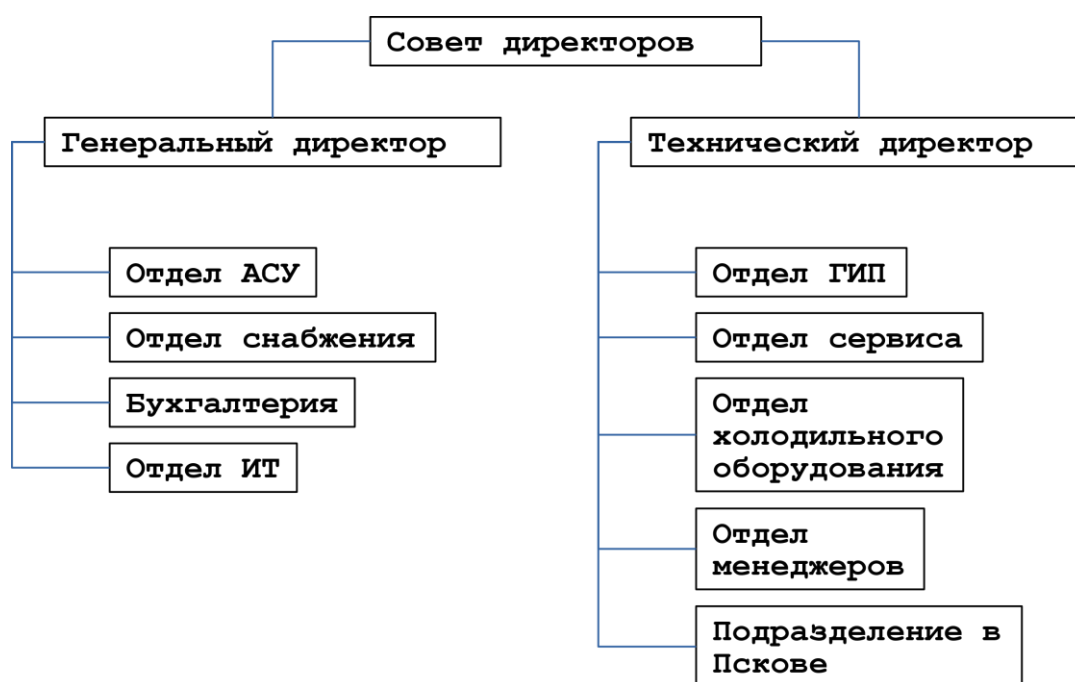


Рисунок 1 – Структура предприятия

Основные направления деятельности:

- проектирование, изготовление и монтаж конвейерного оборудования;

- автоматизация технологических процессов и производств;
- поставка запасных частей и компонентов для конвейерного оборудования.

АО «ЛИПСИЯ», являясь российским отделением немецкой инжиниринговой компании LIPSIA Fordertechnik GmbH, оказывает услуги в России и странах СНГ.

Компания нацелена на освоение новых рынков в сфере пищевой промышленности и охватывает полный цикл производства – проектирование, изготовление, монтаж, сопровождение – что обеспечивает выполнение заказов под «ключ».

1.1.2 Краткая характеристика подразделения и его видов деятельности

Более детально рассмотрим деятельность отдела ИТ.

Отдел состоит из одного сотрудника – администратора информационной безопасности вычислительной сети.

Обязанности регламентируются должностной инструкцией. Непосредственным руководителем является генеральный директор.

В обязанности отдела ИТ входят:

- обеспечение исправного состояния информационно-вычислительной системы;
- оперативное устранение неисправностей по заявкам от сотрудников;
- закупка оборудования (подбор оборудования, передача счёта в оплату, контроль своевременной оплаты, приемка заказа);
- учет программного и аппаратного обеспечения;
- контроль сроков лицензий на программное обеспечение;
- консультация сотрудников по работе с вычислительной техникой компании.

С быстрым увеличением количества обслуживаемого оборудования процессу учета необходимо уделить особое внимание.

Объектом управления выбирается процесс учета программного и аппаратного обеспечения на предприятии.

Целью объекта управления является планирование, учет и отслеживание состояния программного и аппаратного обеспечения.

Результатом деятельности объекта управления является:

- более эффективное управление ИТ-инфраструктурой;
- сохранение трудовых ресурсов;
- экономия денежных средств на неоправданных закупках программного и аппаратного обеспечения.

Метод исследования – методология объектно-ориентированного анализа и проектирования.

1.1.3 Сущность задачи автоматизации

Выбранный процесс является частью более обширной концепции управления ИТ-активами (ИТМА, IT Asset Management).

ИТ-актив – любой ИТ-ресурс, представляющий ценность для основной деятельности организации.

ИТАМ – «область деятельности по планированию, учету и отслеживанию состояния ИТ-активов, обеспечивающих соответствующую ценность для основной деятельности организации, представляющая контроль финансовых потоков на протяжении жизненного цикла ИТ-активов, с учетом всех сопутствующих признаков (артефактов)» [27][30].

Выбранный процесс (объект управления) включает в себя два процесса:

- Hardware Asset Management – управление аппаратными активами;
- Software Asset Management – управление программными активами.

Мировая практика использования ИТАМ показывает, что:

- экономия на платежах за лицензии и поддержку – от 7 до 25 %;
- обеспечение соответствия закупаемых ИТ-активов потребностям бизнеса – от 7 до 15 %;
- оптимизация использования ИТ-активов – от 4 до 10 %;
- сокращение операционных затрат на ввод и сопровождение рабочих мест – оптимизация деятельности ИТ-персонала;
- обеспечение непрерывности бизнеса.

Система управления ИТАМ интегрируется с системами смежных подразделений предприятия (бухгалтерия, финансы, безопасность и т. д.).

Одной из основных обязанностей отдела ИТ является учет программного и аппаратного обеспечения.

В мировой практике активно развивается концепция управления ИТ-активами, которая охватывает весь жизненный цикл ИТ-актива.

1.2 Концептуальное моделирование предметной области

Для описания концептуальной модели процесса учета программного и аппаратного обеспечения используется диаграмма прецедентов UML [4][5][32][35].

Программой учета в модели «Как есть» используются электронные таблицы. На рабочей станции сотрудника отдела ИТ расположен файл электронной таблицы.

При поступлении оборудования согласно счёту или товарной накладной сотрудник заносит информацию об оборудовании в электронную таблицу, присваивает оборудованию инвентарный номер. Есть возможность дополнить существующую запись новыми данными; удалить запись; просмотреть список оборудования; применить фильтр и поиск.

Модель «Как есть» представлена на рисунке 2.

Необходимо отметить трудоемкость ввода исходной информации об

оборудовании, что усугубляется дублированием этой процедуры в бухгалтерии для системы бухгалтерского учета.



Рисунок 2 – Модель «Как есть». Диаграмма прецедентов UML

Недостатки существующей системы:

- отсутствует централизованный доступ к данным учета оборудования;
- не фиксируется место расположения оборудования и ответственное лицо;
- не ведется информация о состоянии оборудования в течение всего жизненного цикла с момента планирования до вывода из эксплуатации, например перемещение, неисправности и ремонты, модернизация и т. д.

Клиент-серверная архитектура решит проблему с централизованным доступом, позволит работать с системой нескольким пользователям.

Возможности электронных таблиц для анализа данных достаточно обширны, поэтому в модели «Как должно быть» необходимо предусмотреть поиск и фильтрацию записей.

Необходимо учесть то, что только лишь модернизация технических

решений не приведет к желаемым улучшениям выбранного процесса. Также необходимо проработать и ввести на предприятии положение об учете ИТ-активов.

Модель «Как должно быть» представлена на рисунке 3.

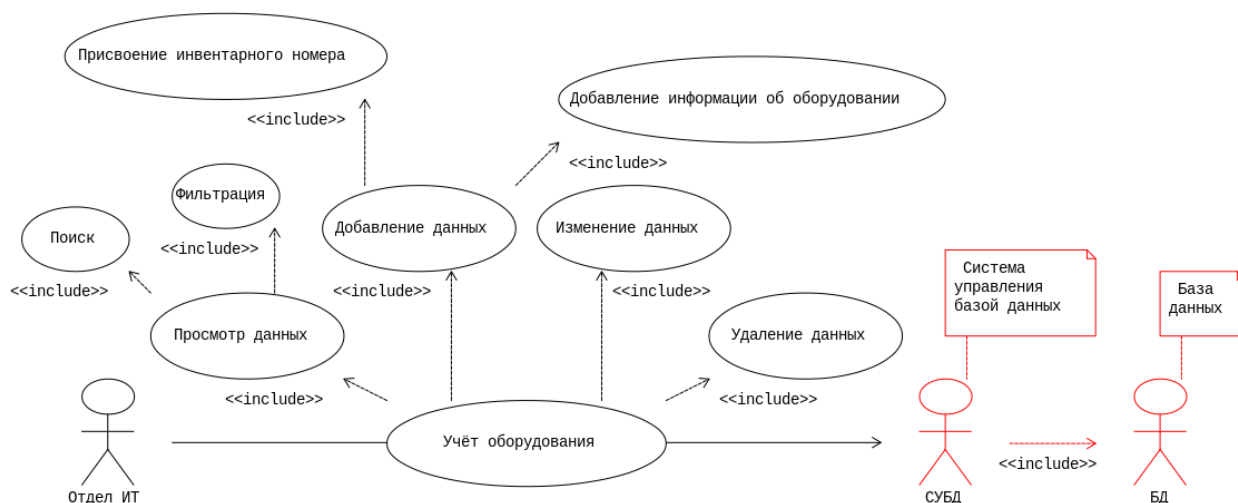


Рисунок 3 – Модель «Как должно быть». Диаграмма прецедентов UML

Согласно проведенному анализу, модель «Как есть» имеет ряд недостатков. Это неполная информация об оборудовании, хранение данных в файле электронной таблицы, расположение файла на рабочей станции сотрудника отдела ИТ.

Представленная модель «Как должно быть» позволяет решить выявленные недостатки.

1.3 Постановка задачи

Назначением реализации работы информационной системы по учету программного и аппаратного обеспечения может служить:

- учет состояния ИТ-активов предприятия: поступление, ввод в эксплуатацию, перемещение, модернизация, вывод из эксплуатации;

- хранение данных в реляционной базе данных;
- автоматизация ввода, контроля и загрузки данных первичных документов в базу данных с использованием экранных форм (например, в поле ввода «Производитель» можно выбрать производителя из выпадающего списка).

В рамках разрабатываемой системы отделу ИТ необходимо будет более полно отражать информацию об ИТ-активах, в первую очередь при внесении первичной информации. Первичными источниками данных являются счет или товарная накладная, для уточнения данных необходимо использовать интернет-ресурсы производителей оборудования.

Данные вносятся в систему посредством электронных форм веб-интерфейса.

Результатом работы системы является вывод информации через веб-интерфейс, например список оборудования содержит инвентарный номер, дату поступления и ввода в эксплуатацию, поля с характеристиками и т. д.

Для хранения данных используется реляционная база данных. Доступ к данным осуществляется через веб-интерфейс при вводе логина и пароля.

Описание входных показателей за год:

- платежи за лицензии и поддержку: 2 000 000 руб.,
- платежи на закупку оборудования: 2 000 000 руб.,
- затраты рабочего времени на внесение данных об оборудовании (с учетом дублирования в бухгалтерии): 247 часов.

Переведем часы в оплату труда: $50\,000 / 20 \times 8 = 312,5$ руб./час, $247 \times 312,5 = 77\,187,5$ рублей.

Описание результатных показателей:

- экономия на платежах за лицензии и поддержку: $2\,000\,000 \times 7\% = 140\,000$ руб.,
- экономия на платежах на закупку оборудования: $2\,000\,000 \times 7\% = 140\,000$ руб.,

- оптимизация затраты рабочего времени на внесение данных об оборудовании: 123,5 часа.

Это составит 38 593,75 рублей.

Итого экономия составит: 318 593,75 рублей в год.

Согласно «мировой практике после внедрения системы управления ИТАМ экономия затрат на платежах на программное и аппаратное обеспечение снижается минимум на 7%» [27]. Так же внедрение системы управления выбранным процессом поможет повысить производительность труда отдела ИТ и бухгалтерии.

1.4 Анализ существующих разработок и обоснование выбора технологии проектирования

В существующей («Как есть») модели процесса применяется электронная таблица Calc LibreOffice. Файл с данными расположен на рабочей станции сотрудника отдела ИТ.

На рынке программного обеспечения присутствует готовое решение на платформе 1С: Предприятие 8 – 1С: ТОИР Управление ремонтом и обслуживанием оборудования 2 [21].

Фирма «1С» – российская компания, специализирующаяся на дистрибуции, поддержке и разработке компьютерных программ и баз данных делового и домашнего назначения.

Возможности 1С: ТОИР:

- ведение списка оборудования;
- ведение документов принятия к учету и списания оборудования;
- учет перемещения оборудования;
- классификация оборудования;
- ведение технологических карт ремонтов;
- формирование и ведение базы нормативов ТОиР;

- ведение списка объектов регламентных мероприятий;
- хранение исторических данных о ремонтах оборудования;
- ведение базы данных текстовых и графических документов.

Система 1С: ТОИР предназначена для «специалистов по организации ремонтов и обслуживания промышленного оборудования, а также для всех подразделений, имеющих какое-либо отношение к управлению активами, ремонтам и обслуживанию» [23].

Для организации системы управления ремонтами и обслуживанием оборудования на предприятиях различных отраслей предназначен продукт 1С: ТОИР 2 КОРП. Решение относится к классу ЕАМ-систем (Enterprise Asset Management - управление основными фондами и активами предприятия).

Расчет стоимости системы 1С: ТОИР приведен в таблице 1.

Таблица 1 – Расчет стоимости системы 1С: ТОИР

Наименование	Стоимость
1С: Предприятие 8. ТОИР Управление ремонтами и обслуживанием оборудования 2 КОРП	330 000 руб.
1С: ТОИР Управление ремонтами и обслуживанием оборудования 2 КОРП. Клиентская лицензия на 5 рабочих мест	54 000 руб.
Итого:	384 000 руб.

Решение от компании 1С включает в себя очень богатый функционал, который покрывает требования к системе учета программного и аппаратного обеспечения. Плюсом данного решения является возможность совместной работы с программой 1С: Бухгалтерия на базе 1С: Предприятие 8.

Но для решения задачи на данном этапе является избыточным, основной функционал направлен на организацию технического обслуживания и ремонтных работ, стоимость данного решения для малого бизнеса достаточно высока.

На рынке программного обеспечения присутствуют готовые решения, которые решают задачи по учету оборудования, но также обладают избыточным функционалом.

Выводы по главе 1

В этой главе дана характеристика предприятия и его деятельности, рассматривается отдел ИТ и должностные обязанности сотрудников этого отдела. Обозначен объект управления – процесс учета программного и аппаратного обеспечения на предприятии. Указаны цели и результат деятельности объекта управления.

Выбирается метод исследования – методология объектно-ориентированного анализа и проектирования.

Проведя анализ объекта управления, были выявлены недостатки существующей модели. К ним относятся: отсутствие централизованного доступа к данным, однопользовательский режим, недостаточно полное перечня данных об оборудовании и его движении в течение всего срока эксплуатации. Модель «Как должно быть» призвана решить существующие проблемы.

Для бизнеса внедрение системы управления позволит снизить затраты на закупку программного и аппаратного обеспечения, высвободить трудовые ресурсы, более эффективно управлять ИТ-активами.

Наличие на рынке программного обеспечения готовых решения позволит сделать выбор между покупкой готовой системы и разработкой собственного решения.

Глава 2 Разработка и реализация проектных решений

2.1 Логическое моделирование предметной области

На основании логической модели «Как должно быть», приведенной на рисунке 3 (глава 1), и специфики решаемой задачи необходимо разработать диаграмму классов предметной области.

Диаграмма классов (англ. Class Diagram) – «статическая структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании» [22].

Так как перечень оборудования, подлежащего учету довольно обширен, удобнее распределить классы по модулям:

- inventory – содержит классы с атрибутами, необходимыми для инвентаризации;
- workstation – содержит классы по рабочим станциям;
- hardware – содержит классы, описывающие комплектующие для рабочих станций;
- periphery – содержит классы, описывающие прочее оборудование.

Классы с постфиксом Mixin, абстрактные, добавляют необходимые атрибуты и методы классам.

Базовый класс BaseHardware также является абстрактным, от него наследуются классы в модулях hardware и periphery.

Модуль workstation.

Класс Workstation представляет рабочую станцию (системный блок или ноутбук) и агрегирует в себе классы комплектующих (hardware).

Диаграмма классов модуля workstation приведена на рисунке 4.

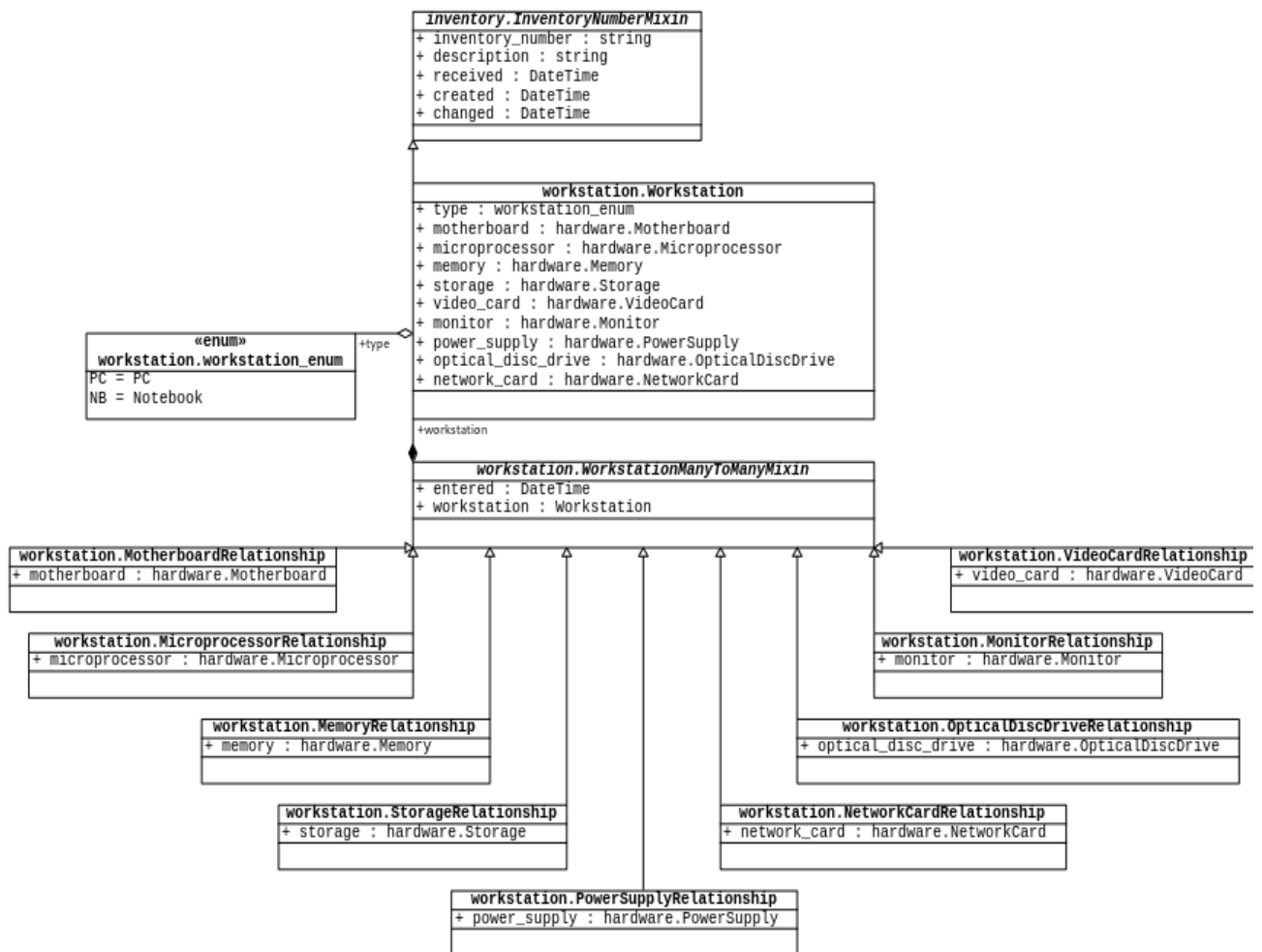


Рисунок 4 – Модуль workstation. Диаграмма классов UML

Для хранения состояния рабочей станции и комплектующих необходима связь многие-ко-многим, реализованная через промежуточные классы с атрибутом для хранения даты и времени.

Модуль hardware.

Модуль содержит классы комплектующих:

- Motherboard – материнские платы,
- Microprocessor – процессоры,
- Memory – оперативную память,
- Storage – накопители жестких дисков, твердотельные накопители,
- VideoCard – видеокарты,

- Monitor – мониторы,
- PowerSupply – блоки питания,
- OpticalDiscDrive – приводы оптических дисков,
- NetworkCard – сетевые карты.

Диаграмма классов модуля hardware приведена на рисунке 5.

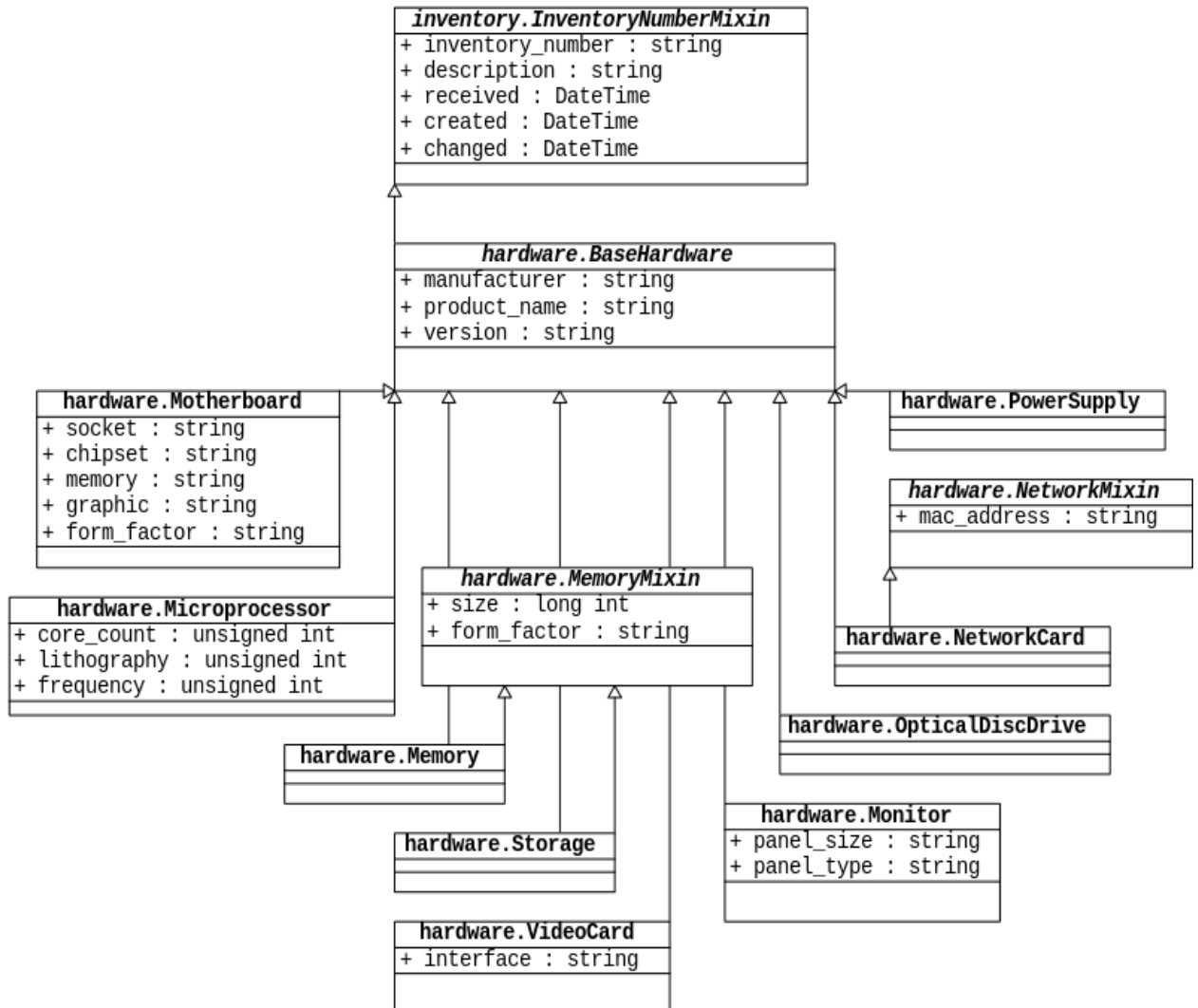


Рисунок 5 – Модуль hardware. Диаграмма классов UML

Здесь для добавления специфичных атрибутов также применяется концепция миксинов. Общие для всех классов атрибуты унаследованы от

BaseHardware.

Модуль `periphery`. Модуль содержит прочее оборудование.

Диаграмма классов модуля `periphery` приведена на рисунке 6.

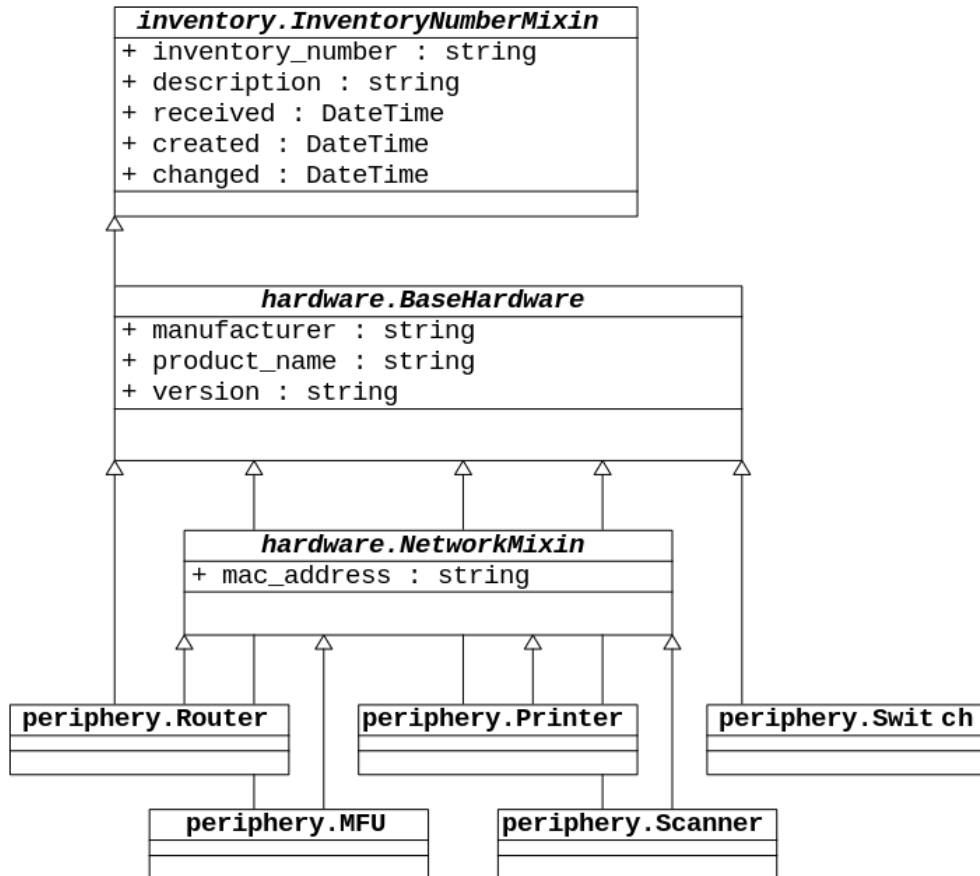


Рисунок 6 – Модуль `periphery`. Диаграмма классов UML

Классы, описывающие прочее оборудование:

- Router,
- Switch,
- MFU,
- Printer,
- Scanner.

Записи в базе данных однозначно идентифицируются по ID (идентификационный номер).

Также оборудование может содержать инвентарный номер.

В таблице 2 представлены классификаторы и система кодирования.

Таблица 2 – Классификаторы и система кодирования

Наименование кодируемого множества объектов	Система кодирования	Система классификации	Вид классификатора	Пример
ID	Порядковая	Отсутствует	Общесистемный	1, 2, 3 и т.д.
Инвентарный номер (inventory_number)	Порядковая	Отсутствует	Общесистемный	000001, 000002, 000003 и т.д.

ID и инвентарный номер присваивается оборудованию в порядке возрастания.

Входными документами являются счет или товарная накладная.

Первичные данными:

- наименование товара,
- количество,
- дата получения.

Объем данных зависит от количества позиций в документе – в среднем 3 позиции.

Частота возникновения документа – в среднем 1 документ в день.

В реляционной базе данных организация связей между объектами возможна следующими способами:

- один-к-одному,
- один-ко-многим,
- многие-ко-многим.

Связь один-к-одному достаточно специфична и применяется, когда необходимо дополнить одну таблицу полями из другой.

Один-ко-многим самая распространенная связь. Например, есть таблица с производителями оборудования и есть несколько таблиц по типу оборудования с полем с внешним ключом записи из таблицы производителя.

Многие-ко-многим – «это связь, при которой множественным записям из одной таблицы могут соответствовать множественные записи из другой. Например, запись в таблице рабочей станции ссылается на несколько записей в таблице оперативной памяти, причем запись в таблице оперативной памяти может ссылаться на несколько записей в таблице рабочей станции (вчера была установлена на одной рабочей станции, сегодня установили на другую)» [21].

Связь многие-ко-многим реализуется с помощью дополнительной таблицы, содержащей первичные ключи связываемых таблиц. В дополнительной таблице можно добавить поля для хранения нужной информации. К примеру, поле с датой установки оперативной памяти в рабочую станцию.

Разрабатываемая система позволяет пользователю взаимодействовать с базой данных с помощью полей форм веб-интерфейса; списков таблиц, записей; кнопок, элементов выбора.

Добавить 01 | Рабочая станция

Инвентарный номер:

Описание:

Поступило:

Запись создана: --

Запись изменена: --

Тип: PC Notebook

MOTHERBOARD RELATIONSHIPS

УСТАНОВЛЕНО	MOTHERBOARD	УДАЛИТЬ?
+ Добавить еще один Motherboard relationship		

MICROPROCESSOR RELATIONSHIPS

Рисунок 7 – Форма ввода данных рабочей станции

В форме выделяются обязательные поля для заполнения, происходит проверка введенных данных, отображаются сообщения об успешной или неуспешной операции записи.

Пример формы ввода данных рабочей станции приведен на рисунке 7.

Также отображается информация по связанным формам (связь много-ко-многим).

Пример формы ввода данных материнской платы приведен на рисунке 8.

Добавить 01 | Материнская плата

Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Инвентарный номер:

Описание:

Поступило: Сегодня 📅

Запись создана: --

Запись изменена: --

Производитель:

Наименование:

Версия:

Сокет:

Чипсет:

Тип памяти:

Графический адаптер:

Форм-фактор:

MOTHERBOARD RELATIONSHIPS		
УСТАНОВЛЕНО	WORKSTATION	УДАЛИТЬ?
+ Добавить еще один Motherboard relationship		

Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Рисунок 8 – Форма ввода данных материнской платы

Поле «Поступило» заполняется автоматически текущей датой, которую можно изменить вручную.

Результатная информация доступна в системе в виде таблицы. В полях таблицы отображается структурированная информация об объекте учета.

Пример вывода рабочих станций отображен на рисунках 9.

Выберите 01 | Рабочая станция для изменения

Действие: Выбрано 0 объектов из 3

<input type="checkbox"/>	ИНВЕНТАРНЫЙ НОМЕР	ОПИСАНИЕ	ПОСТУПИЛО	ЗАПИСЬ СОЗДАНА	ЗАПИСЬ ИЗМЕНЕНА	ТИП	МАТЕРИНСКИЕ ПЛАТЫ	ПРОЦЕССОРЫ	ОПЕРАТИВНАЯ ПАМЯТЬ	ЖЕСТКИЕ ДИСКИ	ВИДЕ
<input type="checkbox"/>	000001	Отдел ИТ	1 июня 2020 г.	10 июня 2020 г.	10 июня 2020 г.	PC	Материнская плата GIGABYTE GA-B250- F1nTech (rev. 1.0) 2020-06-09	Процессор Intel Core i5 Coffee Lake 2020-06-09	Оперативная память 8 GB Crucial CT8G4DFS824A 2020-06-09	Жесткий диск Seagate Barracuda 1 TB ST1000DM010 2020-06-09	GIGA GeFO 1660 2020-
<input type="checkbox"/>	000002	Бухгалтерия	9 июня 2020 г.	10 июня 2020 г.	10 июня 2020 г.	PC	Материнская плата MSI B450M-A PRO MAX 2020-06-09	Процессор AMD Ryzen 5 3600 2020-06-09	Оперативная память 8 GB 1 шт. Samsung DDR4 2666 DIMM 8GB 2020-06-09	Жесткий диск Western Digital WD10EZEX 2020-06-09	GIGA GeFO 1660 2020-
<input type="checkbox"/>	000001	Отдел ИТ	1 мая 2020 г.	10 июня 2020 г.	10 июня 2020 г.	PC	Материнская плата ASUS B360M-A Rev. 1.0 2020-06-09	Процессор Intel Core i5 Coffee Lake 2020-06-09	Оперативная память 8 GB Crucial CT8G4DFS824A 2020-06-09	Жесткий диск Seagate Barracuda 1 TB ST1000DM010 2020-06-09	GIGA GeFO 1660 2020-

3 01 | Рабочие станции

Рисунок 9 – Список рабочих станций

Основные поля списка рабочих станций – это инвентарный номер, где расположен, дата создания, редактирования, когда поступило, тип, связанные таблицы базы данных.

Пример вывода материнских плат отображен на рисунках 10.

Выберите 01 | Материнская плата для изменения

Действие: Выбрано 0 объектов из 4

<input type="checkbox"/>	ИНВЕНТАРНЫЙ НОМЕР	ОПИСАНИЕ	ПОСТУПИЛО	ЗАПИСЬ СОЗДАНА	ЗАПИСЬ ИЗМЕНЕНА	ПРОИЗВОДИТЕЛЬ	НАИМЕНОВАНИЕ	ВЕРСИЯ	СОКЕТ	ЧИПСЕТ	ТИП ПАМЯТИ	ГРАФИЧ
<input type="checkbox"/>	--	Материнская плата ASUS Z170I PRO GAMING	9 июня 2020 г.	9 июня 2020 г.	10 июня 2020 г.	ASUS	Z170I PRO GAMING	--	LGA1151	Intel Z170	2 слота DDR4 DIMM, 2133-3400 МГц	--
<input type="checkbox"/>	--	Материнская плата ASUSRock B450M Pro4-F	9 июня 2020 г.	9 июня 2020 г.	10 июня 2020 г.	ASRock	B450M Pro4-F	--	AM4	AMD B450	4 слота DDR4 DIMM, 2133-3200 МГц	--
<input type="checkbox"/>	--	Материнская плата GIGABYTE GA-B250-FnTech (rev. 1.0)	9 июня 2020 г.	9 июня 2020 г.	10 июня 2020 г.	GIGABYTE	GA-B250-FnTech	rev. 1.0	LGA1151	Intel B250	4 слота DDR4 DIMM, 2133-3400 МГц	--
<input type="checkbox"/>	--	Материнская плата MSI B450M-A PRO MAX	9 июня 2020 г.	9 июня 2020 г.	10 июня 2020 г.	MSI	B450M-A PRO MAX	--	AM4	AMD B450	DDR4 DIMM, 1866-4133 МГц	--

4 01 | Материнские платы

Добавить 01 | МАТЕРИНСКАЯ ПЛАТА

ФИЛЬТР

Производитель

- Все
- ASRock
- ASUS
- GIGABYTE
- MSI

Рисунок 10 – Список материнских плат

На страницах списков расположено поле ввода поиска, сбоку находится панель с фильтрами.

Если количество записей превышает определенное значение, то отображение списка делится на несколько страниц, добавляется панель пагинации.

В данном параграфе рассмотрены элементы логической модели системы, их структура и взаимосвязи.

Приведены примеры веб-интерфейса системы, формы для ввода данных в систему, а также представление информации об оборудовании в табличном виде.

2.2 Физическое моделирование АИС

Так как к разрабатываемой системе должен быть обеспечен централизованный доступ для нескольких сотрудников, выбирается трехзвенная клиент-серверная архитектура [3].

Клиент: веб-браузер (Chrome, Firefox, Edge).

Сервер приложения: прокси-сервер (Nginx [34]), wsgi интерфейс приложения (Gunicorn [25]), веб-фреймворк (Django [33]).

Система управления базой данных (PostgreSQL [31]).

Операционная система: Linux Debian 11 [23].

Здесь описывается состав функциональной структуры системы, выделяются подсистемы.

К основным относятся те функции, которые решают бизнес-задачи, возложенные на систему.

Служебные функции обеспечивают работу самой системы.

Иерархия функций управления и обработки данных приведена в таблице 3.

Сценарии диалога можно рассматривать как «последовательность переходов системы из одного состояния в другое» [12]. Важно, чтобы ни одно из состояний не было тупиковым, пользователь должен иметь

возможность перейти из любого текущего состояния диалога в требуемое (за один или несколько шагов). Для этого в ходе разработки интерфейса необходимо определить все возможные состояния диалога и пути перехода из одного состояния в другое.

Таблица 3 – Функции системы

Функции системы	
Основные	Служебные
Предоставление веб-интерфейса: – для просмотра списка оборудования, – для добавления оборудования, – для просмотра/редактирования данных выбранного оборудования, – для удаления оборудования.	Аутентификация и авторизация пользователя
Поиск по тексту в описании оборудования	Изменение пароля
Фильтры: – по инвентарному номеру, – по производителю, – по дате поступления, перемещения.	Добавление, удаление пользователей
Присвоение инвентарного номера оборудованию	Выход из системы

Целями разработки сценария диалога являются: выявление и устранение возможных тупиковых ситуаций ходе развития диалога; выбор рациональных путей перехода из одного состояния диалога в другое (из текущего в требуемое); выявление неоднозначных ситуаций, требующих оказания дополнительной помощи пользователю.

Сценарий диалога приведен в таблице 4.

Сценарий диалога позволяет описать процесс взаимодействия пользователя с приложением на уровне решаемой им прикладной задачи.

Система может находиться в двух состояниях:

- ожидания действия пользователя;
- выполнения процесса, реакции на действие пользователя.

Переход от первого состояния во второе осуществляется посредством выполнения запроса к базе данных, а обратно посредством перехода на другую страницу приложения.

Таблица 4 – Сценарии диалога

Состояние диалога	Для всех пользователей	
	Для аутентифицированных пользователей	Для администратора
Страница аутентификации	Форма ввода логина и пароля	
Главная страница	Разделы со списком типов оборудования	Раздел «Пользователи и группы»
Страница раздела со списком типов оборудования	Раздел со списком типов оборудования	-
Страница со списком оборудования	Таблица с записями информации об оборудовании	-
Страница конкретного оборудования	Поля формы оборудования, заполненные имеющимися данными	-
Страница добавления оборудования	Поля формы оборудования, заполненные по умолчанию, либо пустые	-
Страница изменения пароля	Поля формы для изменения пароля	-
Страница добавления/удаления пользователя	-	Поля формы для добавления пользователя, кнопка удаления пользователя

Орграф сценария диалога приведены на рисунке 11.

Сценарий диалога позволяет «однозначно определить все уровни системы, на которых пользователь принимает решение относительно следующего действия, и исключить возможность возникновения тупиковых сценариев» [21].

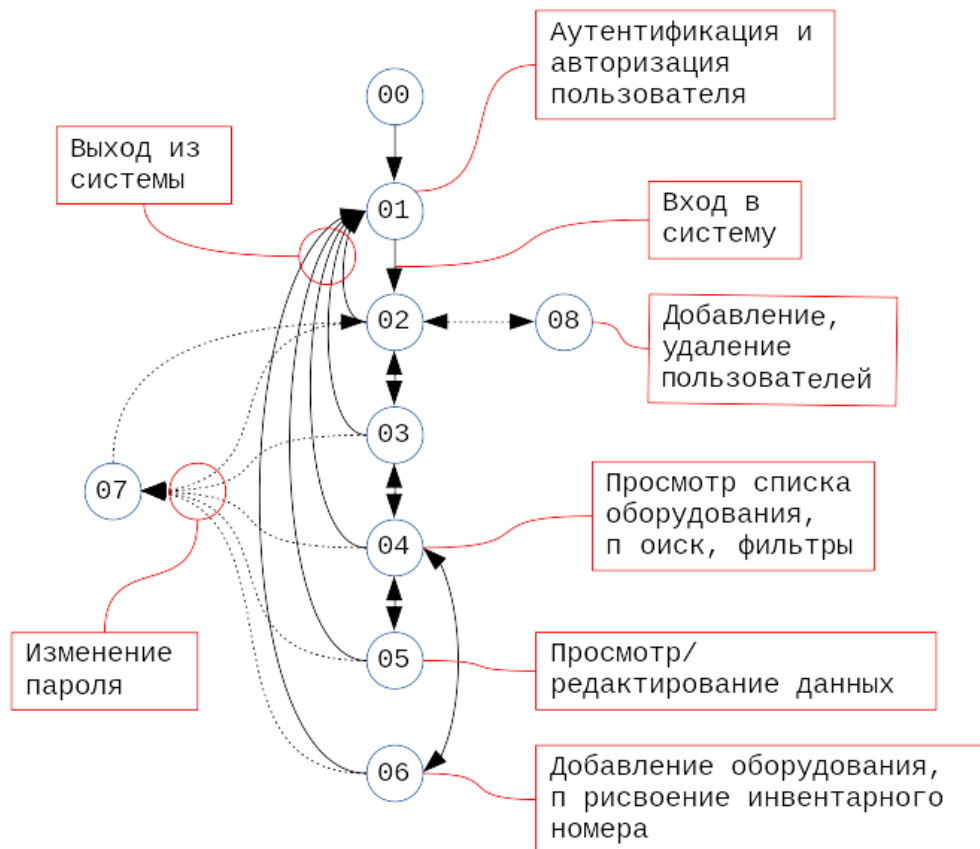


Рисунок 11 – Орграф сценария диалога системы

Рассмотрим структуру проекта (рисунок 12).

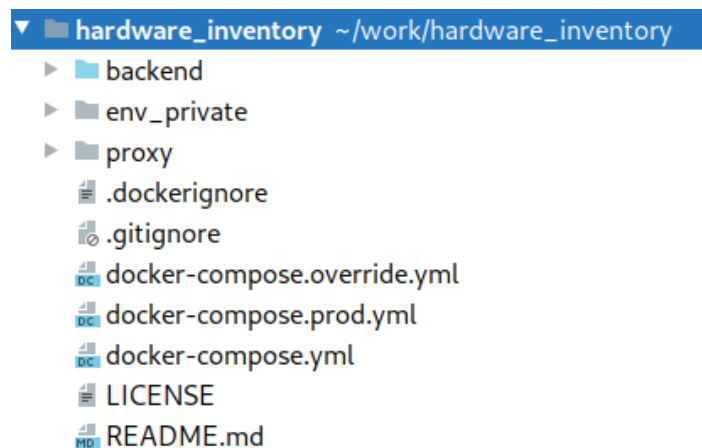


Рисунок 12 – Структура корня проекта

В корне проекта содержатся файлы системы версионирования Git [1], системы контейнеризации Docker [24], лицензия и инструкция к проекту.

Папка `env_private` содержит приватные переменные окружения проекта.

Папка `proxu` содержит настройки прокси-сервера.

Папка `backend` является корнем приложения Django.

Фреймворк Django придерживается концепции Model-View-Controller.

Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») – схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо [28].

Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.

Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Папка приложения содержит файлы по сборке контейнера, настройки для тестов, файл со списком пакетов зависимостей (рисунок 13).

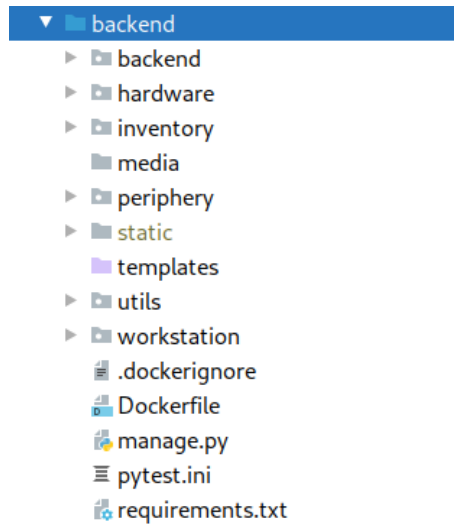


Рисунок 13 – Структура корня приложения

Файл `manage.py` – интерфейс для взаимодействия с приложением на Django.

Одноименная папка `backend` – пакет приложения, содержит файлы конфигурации.

Папки `media`, `static`, `templates` предназначены для статических файлов (изображения, видео, `html`, `css`, `js` и т. д.).

Пакет `utils` содержит код (функции, декораторы, классы), используемый в разных модулях.

Система содержит четыре основных модуля: `inventory`, `workstation`, `hardware` и `periphery`.

Модуль `inventory` описывает структуру классов модели базы данных общих для всего оборудования (рисунок 14).

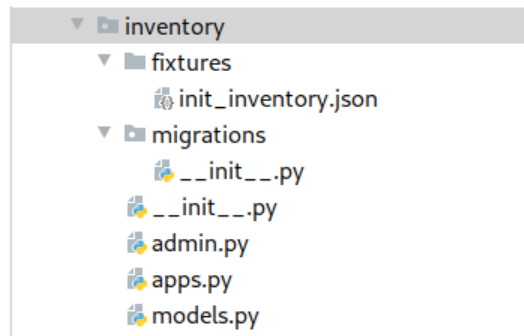


Рисунок 14 – Структура модуля inventory

Папка fixtures содержит данные для инициализации базы данных начальными данными.

Слой модель (Model) расположен в файлах models.py модулей.

Django предоставляет готовое решение для администрирования приложения через веб-интерфейс.

Слой представления (View) в данном приложении реализуется встроенным модулем Django – django.contrib.admin.

Слой контроллеров (Controller) необходимо реализовать в файлах admin.py модулей.

Модуль workstation описывает классы, отражающие параметры рабочих станций (рисунок 15).

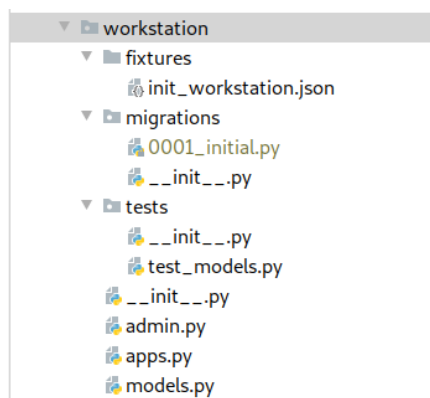


Рисунок 15 – Структура модуля workstation

Модуль hardware содержит классы, описывающие комплектующие для рабочих станций (рисунок 16).

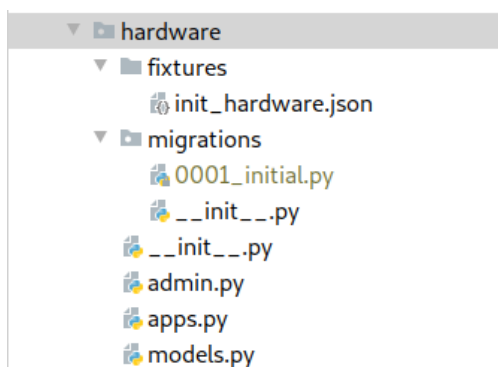


Рисунок 16 – Структура модуля hardware

В папке migrations хранится история изменений структуры базы данных.

Модуль periphery – содержит классы, описывающие прочее оборудование (рисунок 17).

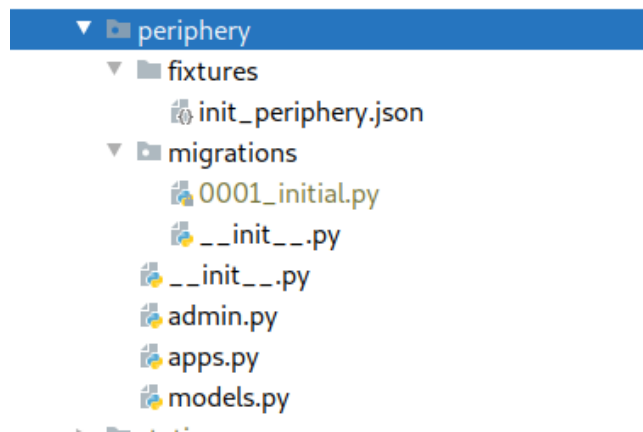


Рисунок 17 – Структура модуля periphery

Модуль – функционально законченный фрагмент программы.

Модульная структура позволяет «инкапсулировать, сделать код компактнее, что позволяет ускорить процесс разработки и сопровождения проекта» [30].

В данном параграфе выбрана архитектура информационной системы; проработана функциональная структура проекта, иерархия функций управления и обработки данных, с помощью орграфа отображены сценарии диалога с пользователем, переходы между состояниями. Приведена структурная схема проекта и организация модулей проекта.

Выводы по главе 2

В данной главе проводится анализ и разработка системы управления учетом аппаратного и программного обеспечения. С помощью диаграммы классов UML описана логическая модель предметной области, являющаяся прототипом базы данных; определен порядок классификации сущностей в базе данных; описаны требования к входной и выходной информации.

Выбрана физическая архитектура системы и технологии для разработки. Перечислены функции системы и сценарии диалога, обозначена структурная схема проекта, описаны программные модули.

Глава 3 Оценка и обоснование экономической эффективности проекта

3.1 Выбор и обоснование методики расчета экономической эффективности

Экономическая эффективность складывается из:

- сокращения затрат на проведение вычислительных операций;
- совершенствования производственной и хозяйственной деятельности.

Ключевыми инструментами оценки экономической эффективности являются расчет и сравнение трудовых и стоимостных показателей, а также срока окупаемости проекта.

Для обоснования экономической эффективности системы автоматизации выбирается расчет прямой эффективности от внедрения информационной системы.

Для расчета прямого эффекта от внедрения разработанного программного продукта необходимо рассмотреть показатели трудовых и стоимостных затрат.

К трудовым показателям относятся следующие:

1) Абсолютное снижение трудовых затрат, рассчитываемое по формуле:

$$\Delta T = T_0 - T_1, \quad (1)$$

где T_0 – время, затрачиваемое на выполнение автоматизируемых операций в базовом варианте,

T_1 – время, затрачиваемое на выполнение автоматизируемых операций в проектном варианте.

2) Коэффициент относительного снижения трудовых затрат k_T (в процентах), для расчета которого используется следующая формула:

$$k_T = (\Delta T_0/T_1) \cdot 100\%, \quad (2)$$

3) Индекс снижения трудовых затрат, рассчитываемый следующим образом:

$$Y_T = T_0/T_1, \quad (3)$$

К стоимостным показателям относятся следующие:

4) Абсолютное снижение стоимостных затрат:

$$\Delta C = C_0 - C_1, \quad (4)$$

где C_0 – стоимостные затраты на обработку информации по базовому варианту,

C_1 – стоимостные затраты на обработку информации по предлагаемому варианту.

5) Коэффициент относительного снижения стоимостных затрат k_C (в процентах), определяемый по следующей формуле:

$$k_C = (\Delta C_0/C_1) \cdot 100\%, \quad (5)$$

6) Индекс снижения стоимостных затрат, рассчитываемый по формуле:

$$Y_C = C_0/C_1, \quad (6)$$

Коэффициенты k_C и Y_C характеризуют рост производительности труда

за счет внедрения более экономичного варианта проектного решения.

Помимо рассмотренных показателей целесообразно также рассчитать срок окупаемости затрат на внедрение проекта (T_{OK}):

$$T_{OK} = k_{\Pi} / \Delta C, \quad (7)$$

где k_{Π} – капитальные затраты на создание проекта.

$$k_{\Pi} = C + ЗП + M_{\Pi} + Н, \quad (8)$$

где C – первоначальная стоимость программного продукта,

$ЗП$ – заработная плата разработчика на всех этапах проектирования и внедрения,

M_{Π} – затраты на использование ЭВМ на этапе проектирования и внедрения,

$Н$ – накладные расходы на этапе проектирования и внедрения.

$$M_{\Pi} = (C_{\Pi} \cdot t_{\Pi}) + (C_{Д} \cdot t_{Д}), \quad (9)$$

где C_{Π} и $C_{Д}$ – стоимости одного часа процессорного и дисплейного времени,

t_{Π} и $t_{Д}$ – процессорное и дисплейное время, необходимое для решения задачи (час).

Для обоснования экономической эффективности проекта выбирается расчет прямой эффективности от внедрения информационной системы

Для оценки экономической эффективности проекта рассчитываются и сравниваются трудовые и стоимостные показатели, а также срок окупаемости проекта.

3.2 Расчет показателей экономической эффективности проекта

На основании описанной методики необходимо выполнить расчет показателей экономической эффективности (таблица 5) и сделать вывод об экономической обоснованности внедрения автоматизированной системы.

Таблица 5 – Расчет показателей экономической эффективности проекта

Показатель	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
Трудоемкость	T_0 , час	T_1 , час	(1)	(2)	(3)
	247	123,5	123,5	100	2
Стоимость	C_0 , руб.	C_1 , руб.	(4)	(5)	(6)
	2 000 000	1 860 000	140 000	8	1,08

Рассчитаем капитальные затраты на разработку данной системы.

Примем заработную плату разработчика 48 000 руб./месяц или 2 400 руб./день.

Разработка системы займет 20 дней.

Следовательно заработная плата разработчика составит 48 000 руб.

$$k_{\Pi} = C + 3\Pi + M_{\Pi} + H$$

Рассчитаем затраты на использование ЭВМ.

Так как программа разработана на современных быстродействующих компьютерах, то в дополнительном процессорном времени необходимости нет, т.е. $C_{\Pi} = 0$ и $t_{\Pi} = 0$.

$C_{\text{д}}$ примем по тарифной ставке 40 руб/час.

$$M_{\Pi} = C_{\text{д}} \cdot t_{\text{д}} = 40 \cdot 20 \cdot 8 = 6400 \text{руб.}$$

Первоначальную стоимость программного продукта C примем за 0 руб., т. к. в разработке используется свободное программное обеспечение.

Накладные расходы H принимаются за 3 % от общей суммы затрат.

Рассчитаем капитальные затраты на этапе разработки:

$$k_{\Pi} = C + 3\Pi + M_{\Pi} + H = (0 + 48000 + 6400) \cdot 1.03 = 56032 \text{руб.}$$

Рассчитаем срок окупаемости затрат на внедрение проекта

$$T_{OK} = k_{\Pi} / \Delta C = 56\,032 / 140\,000 = 0,4 \text{ года или 5 месяцев.}$$

По результатам расчетов экономической эффективности проекта наблюдается сокращение трудоемкости в два раза и понижение расходов на 8 %. Срок окупаемости составит 5 месяцев.

Выводы по главе 3

Результаты оценки экономической эффективности проекта являются ключевым фактором при принятии решения о дальнейшей разработке и внедрении. На основании результатов расчета экономической эффективности принимается решение о дальнейшей разработке проекта.

Затраты на разработку проекта обоснованы, срок окупаемости приемлем для бизнеса.

Результаты успешного использования системы позволят продолжить дальнейшее развитие проекта.

Заключение

В динамично развивающейся компании отделу ИТ необходимо рационально подойти к процессу учета программного и аппаратного обеспечения. Оперативный ввод первичных данных, актуализация информации, проведение инвентаризации – базовые функции системы учета оборудования.

Разработка информационной системы – сложный процесс, включающий в себя анализ предметной области, выделение основных требований к проектируемой системе, определение целей и задач системы, создание модели базы данных, описание интерфейса взаимодействия пользователя с системой.

В процессе разработки были применены современные методики и технологии проектирования сложных информационных систем:

- фреймворк Django, написанный на высокоуровневом языке программирования Python, обладает низким порогом вхождения, имеет богатую библиотеку модулей, обширное сообщество, что позволяет быстро создать приложение;
- модульная архитектура позволит добавлять необходимый функционал при дальнейшем развитии приложения;
- система управления версиями позволит ускорить разработку приложения, добавление нового функционала, исправление ошибок;
- система контейнеризации позволит быстро развернуть приложение в рабочей среде.

В результате проведенной работы была разработана система учета программного и аппаратного обеспечения на предприятии.

Разработанная информационная система помогла решить ряд задач по оптимизации процесса учета оборудования.

Были выполнены задачи:

- построение модели процесса «Как есть», выявлены недостатки существующей модели, смоделирован процесс «Как должно быть»;
- разработаны логическая и физическая модели АИС;
- обоснована экономическая выгода от внедрения АИС.

Согласно концепции управления ИТ-активами разработанную систему можно развивать по следующим направлениям:

- интеграция с внешними отделами (бухгалтерия, финансовый отдел),
- идентификация оборудования с помощью устройств считывания штрих-кода,
- контроль использования контрактных обязательств (гарантия, сервис),
- планирование бюджета на основании накопленной информации.

Дальнейшие улучшения непосредственно затрагивают внешние для отдела ИТ процессы. Поэтому необходимо заручиться поддержкой высшего руководства и донести до руководителей других отделов как система поможет в их работе и в работе всей компании.

Для малого и среднего бизнеса практика управления ИТ-активами может в меньшей мере охватывать другие отделы компании и влиять на их деятельность. Однако ИТ-активы также необходимо отслеживать, управлять рисками, выполнять необходимые финансовые расчеты по использованию ресурсов и распределению бюджета, учитывать информацию о местонахождении ИТ-активов.

Список используемой литературы

1. 1С:ТОИР Управление ремонтами и обслуживанием оборудования 2 КОРП [Электронный ресурс]. – URL: <https://solutions.c.ru/catalog/eam2/features>.
2. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (ИСО 5807–85) [Текст]. Введен 1992–01–01. – М.: Изд-во стандартов, 1992. – 14 с. – (Единая система программной документации).
3. ГОСТ 2.105-95. Общие требования к текстовым документам [Текст]. – М.: Изд-во стандартов, 1996. – 29 с. – (Единая система конструкторской документации).
4. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения.
5. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
6. ГОСТ 34.320-96. Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы.
7. ГОСТ 34.601-90. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
8. ГОСТ 34.602-89. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
9. ГОСТ 34.603-92. Информационные технологии. Виды испытаний автоматизированных систем.

10. ГОСТ 6.61.1-87. Единая система классификации и кодирования технико-экономической информации. Основные положения [Текст]. – М.: Изд-во стандартов, 1994.

11. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

12. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления.

13. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов

14. ГОСТ Р ИСО/МЭК 12207-99. Информационная технология. Процессы жизненного цикла программных средств.

15. ГОСТ Р ИСО/МЭК 14764-2002. Информационная технология. Сопровождение программных средств.

16. ГОСТ-34 РД 50-680-88. Руководящий документ по стандартизации. Методические указания. Автоматизированные системы. Основные положения.

17. Долженко А. И. Технологии командной разработки программного обеспечения информационных систем [Электронный ре-сурс] : [курс лекций] / А. И. Долженко. - 2-е изд., испр. - Москва : ИНТУИТ , 2016. - 300 с.

18. Котляров В. П. Основы тестирования программного обеспечения [Электронный ресурс] : учеб. пособие / В. П. Котляров. - 2-е изд., испр. - Москва : ИНТУИТ, 2016. - 335 с.

19. Привалов И. М. Основы аппаратного и программного обеспечения [Электронный ресурс] : учеб.-метод. пособие / И. М. Привалов ; Сев.-Кавказ. федерал. ун-т. - Ставрополь : СКФУ, 2015. - 145 с.

20. Рейнжиниринг бизнес-процессов [Электронный ресурс] : учеб. пособие / А. О. Блинов [и др.] ; под ред. А. О. Блинова. - Москва : ЮНИТИ-ДАНА, 2015. – 343 с.

21. Тельнов Ю. Ф. Инжиниринг предприятия и управление бизнес-процессами [Электронный ресурс] : методология и технология : учеб. пособие для студентов магистратуры, обучающихся по направлению «Прикладная информатика» / Ю. Ф. Тельнов, И. Г. Фёдоров. – Москва : ЮНИТИ-ДАНА, 2015. - 207 с.

22. Class diagram [Электронный ресурс]. – URL: https://en.wikipedia.org/wiki/Class_diagram.

23. Debian operating system [Электронный ресурс]. – URL: <https://www.debian.org/>.

24. Get Started with Docker [Электронный ресурс]. – URL: <https://www.docker.com/>.

25. Gunicorn is a Python WSGI HTTP Server for UNIX [Электронный ресурс]. – URL: <https://gunicorn.org/>.

26. IAITAM [Электронный ресурс]. – URL: <https://iaitam.org/>.

27. Martin Thompson Practical ITAM: The essential guide for IT Asset Managers: Getting started and making a difference in the field of IT Asset Management 1st Edition / Martin Thompson. – United States : CreateSpace Independent Publishing Platform, 2017 – P. 174.

28. Model-View-Controller [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Model-View-Controller>.

29. Peter Morville Information Architecture for the World Wide Web: Designing Large-Scale Web Sites, 3rd Edition 3rd Edition / Peter Morville, Louis Rosenfeld. – United States : O'Reilly Media, 2006 – P. 528.

30. Phara Estime McLachlan The Guide to Successful IT Asset Management: Get to grips with the fundamentals of IT Asset Management, Software Asset Management, and Software License Compliance Audits with this guide / Phara Estime McLachlan. – United Kingdom : Packt Publishing, 2018 – P. 248.

31. PostgreSQL Home page [Электронный ресурс]. – URL: <https://www.postgresql.org/>.

32. Russ Miles Learning Uml 2.0 1st Edition / Russ Miles, Kim Hamilton. – United States : O'Reilly Media, 2006 – P. 288.

33. Web framework Django [Электронный ресурс]. – URL: <https://www.djangoproject.com/>.

34. Web server Nginx [Электронный ресурс]. – URL: <https://nginx.org/en/>.

35. What is UML [Электронный ресурс]. – URL: <https://www.uml.org/>.