

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 Автоматизация учета материалов на примере предприятия ООО
«МДМ»

Студент

А.А. Поздняков

(И.О. Фамилия)

(личная подпись)

Руководитель

Е.А. Ерофеева

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

АННОТАЦИЯ

Бакалаврская работа посвящена актуальной тематике автоматизации учета материалов на предприятии.

В результате выполнения работы разработан модуль учета материальных ценностей в условиях предприятия.

В ходе анализа предметной области изучены особенности учета материальных ценностей и деятельности предприятия, разработаны бизнес-модели «AS-IS».

При рассмотрении аналогов проектируемого программного средства рассмотрен их функционал и проведено сравнительное исследование, показавшее, что основную задачу учета материалов рассмотренные программы решают. Но не все удовлетворяют требованиям организации. Таким как невысокая цена, простота обучения и работы с системой, учет специфики функционирования организации. В связи с этим принято решение в проектировании и разработки современной информационной системы, которая бы позволила автоматизировать проведение учета материалов. Также были определены инструменты и технологии проектирования и разработки веб-ориентированной информационной системы для учета материальных ценностей.

Созданная информационная система представляет собой веб-приложение. Приложение позволит ООО «МДМ» автоматизировать процессы учета материальных ценностей, и таким образом повысить эффективность и надежность процессов учета, от чего напрямую зависит эффективность деятельности предприятия.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
Глава 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	6
1.1 Анализ деятельности предприятия, особенности учета материалов	6
1.1.1 Анализ деятельности предприятия ООО «МДМ».....	6
1.1.2 Особенности учета материалов	13
1.2 Организация учета материалов.....	15
1.3 Функциональное моделирование предметной области	17
1.4 Обзор аналогичных программных средств	22
Глава 2 ОБЗОР И ВЫБОР СРЕДСТВ РАЗРАБОТКИ СИСТЕМЫ УЧЕТА МАТЕРИАЛОВ.....	29
2.1 Обзор технологий разработки веб-ориентированных информационных систем	29
2.1.1 Статические и динамические веб-приложения.....	29
2.1.2 Анализ технологий, которые используются при создании веб-ориентированных информационных систем.....	30
2.1.3 Анализ технических и программных средств веб-разработки.....	31
2.1.4 Язык гипертекстовой разметки – HTML	37
2.1.5 Каскадные таблицы стилей – CSS.....	39
2.2 Выбор средств разработки системы учета материалов	43
2.2.1 СУБД для системы учета материальных ценностей	43
2.2.2 Язык программирования для создания системы	48
2.2.3 Среда разработки для создания системы учета материалов.....	54
Глава 3 РЕАЛИЗАЦИЯ СИСТЕМЫ УЧЕТА МАТЕРИАЛОВ. ПРОВЕРКА РАБОТОСПОСОБНОСТИ СОЗДАННОЙ СИСТЕМЫ	57
3.1 Проектирование базы данных.....	57
3.2 Проектирование структуры и функционала модуля учета материальных ценностей	64
3.3 Реализация программного модуля.....	66
3.4 Тестирование работоспособности созданного модуля	82
3.5 Контрольный пример функционирования модуля	87
3.6 Оценка и обоснование экономической эффективности проекта	93
ЗАКЛЮЧЕНИЕ.....	104
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	106

ВВЕДЕНИЕ

Материальные запасы являются одной из составных частей имущества организации, необходимых для стабильного функционирования хозяйственного процесса любого предприятия. Материальные ценности предприятия приходится на производственную направленность и являются его материальной базой для непрерывного функционирования всех рабочих процессов организации.

Правильное определения потребности в материально-производственных запасах влияет на эффективность деятельности организации. Обеспечение оптимальным объемом материальных запасов приводит к снижению затрат, повышению финансовых результатов, а также к ритмичной и слаженной работы организации. Занижение количества необходимых материалов может привести к перебоям в работе, к несвоевременному выполнению организацией своих обязательств. Завышение материальных запасов ведет к замораживанию и омертвлению ресурсов. И в том, и в другом случае следствием является неустойчивое финансовое состояние, нерациональное использование ресурсов.

Выполнение задачи поддерживать оптимальное количество производственных и торговых запасов, не привлекая при этом излишних финансовых ресурсов предприятия, напрямую зависит от организации эффективного контроля за запасами.

Одним из вариантов повышения скорости и качества хранения информации о запасах является внедрение информационной системы, назначением которой является учет материальных ценностей.

Существует множество инструментов и способов разработки веб-приложений. Для разработки веб-приложения для автоматизации учета материалов на предприятии будет использован язык программирования и СУБД: PHP, MySQL; архитектура информационной системы: клиент-серверная;

архитектурный шаблон: MVC; PHP-фреймворк: Yii2; CSS-фреймворк: Bootstrap 4.

Целью исследования является разработка веб-приложения для автоматизации учета материалов на примере предприятия ООО «МДМ».

Объект исследования: система учета материалов в «МДМ».

Предмет исследования: модуль автоматизации системы учета материалов в «МДМ»

Задачи дипломного проекта:

- исследовать предметную область;
- проанализировать деятельность предприятия, особенности учета материалов;
- сделать обзор аналогичных программных средств;
- спроектировать программное средство - веб-приложение для автоматизации системы учета материалов;
- сконструировать веб-приложение для автоматизации системы учета материалов;
- протестировать работоспособность созданного веб-приложения.

Методы исследования: анализ научной и учебной литературы, синтез, сопоставительный, сравнительный и системный анализ, обобщение, изучение практических разработок и документации, моделирование, методы проектирования баз данных, методы и приемы защиты веб-приложений и СУБД, методы проектирования и разработки приложений, тестирование, описательный метод.

Работа состоит из введения, трех частей и заключения.

Глава 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Анализ деятельности предприятия, особенности учета материалов

1.1.1 Анализ деятельности предприятия ООО «МДМ»

ООО «МДМ» - торгово-производственная компания, обладающая мощным производственным и техническим потенциалом, с безупречной деловой репутацией, занимающая особое положение на внутреннем рынке страны.

Основными направлениями деятельности Компании являются производство и поставка продукции для нефтегазового сектора.

Компания поставляет большой спектр продукции для нефтегазовой, нефтехимической и нефтегазоперерабатывающей отраслей.

Отгрузка продукции осуществляется железнодорожным и автомобильным транспортом, а также напрямую с заводов-изготовителей. Доставка осуществляется собственным автотранспортом предприятия, ж/д транспортом и на условиях самовывоза.

Основной вид деятельности компании 46.72 - Торговля оптовая металлами и металлическими рудами.

В процессе работы ООО «МДМ» возникает необходимость учета материально-товарных ценностей, которые идут на производственные или управленческие нужды. Материально-производственные запасы являются весомой частью активов. Независимо от вида деятельности – предприятие обязано вести учет материально производственных запасов (МПЗ). Бухгалтерский учет материальных запасов регулируется 402–ФЗ от 06.12.2011 «О бухгалтерском учете» [5].

Ведение бухгалтерского учета осуществляется бухгалтерской службой, возглавляемой Главным бухгалтером Общества. Структура бухгалтерской службы представлена на рис. 1.

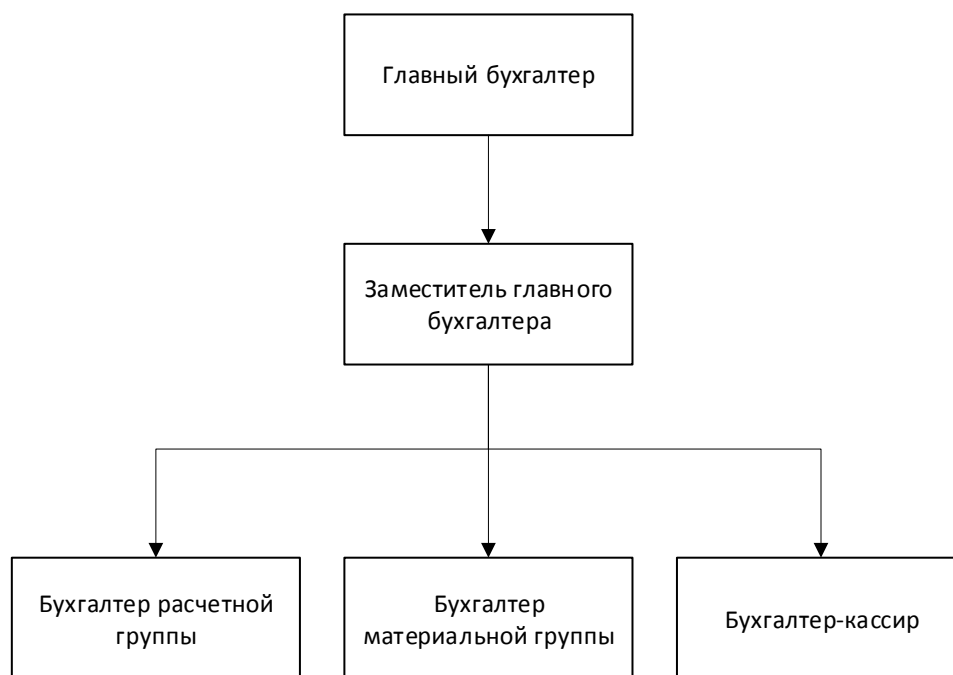


Рисунок 1 – Структура бухгалтерской службы

Главный бухгалтер назначается на должность и освобождается от должности Генеральным директором, непосредственно подчиняется ему и несет ответственность за формирование учетной политики, ведение бухгалтерского учета, своевременное представление полной и достоверной бухгалтерской отчетности.

Главный бухгалтер обеспечивает соответствие осуществляемых хозяйственных операций законодательству Российской Федерации, а также контроль за движением имущества и выполнением обязательств.

Хозяйственные операции в бухгалтерском учете оформляются типовыми первичными документами, которые утверждены законодательно, формами, разработанными предприятием самостоятельно, с учетом требований 402–ФЗ от

06.12.2011 «О бухгалтерском учете». Документы, которыми оформляются факты хозяйственной жизни с денежными средствами, принимаются к отражению в бухгалтерском учете при наличии на документе подписей Генерального директора и главного бухгалтера или уполномоченных ими на то лиц. Порядок организации и ведение делопроизводства осуществляются в соответствии с утвержденной номенклатурой дел [42].

Учет материальных запасов ведется материально ответственными лицами. Договор об индивидуальной ответственности за недостачу имущества заключается с лицом старше 18 лет, непосредственно отвечающим и работающим с материалами, за которые отвечает (ст. 244 ТК РФ). МОЛ предприятия ведет учет всех материальных запасов, имеющихся на складе. Последний должен быть оборудован тарой для возможных перемещений продукции.

Выдача МПЗ производится по соответствующим документам, подписанным руководителем отделения. Поступление на склад проводится на основании договоров. При этом происходит проверка поступившего товара.

После проверки товара на складе происходит его оформление, и данные отправляют в бухгалтерию, где уже и проходит оплата полученных материалов. Каждый месяц заведующий складом отчитывается перед бухгалтерией, представляя соответствующие документы.

Получение товара оформляется по накладным, если имеются какие-то различия между количеством на накладную и количеством по факту пишется акт приемки материалов. Что касается внутреннего перемещения, то здесь оформляется накладная – требования, причем все документы в двух экземплярах (один оформляется для склада, второй – для бухгалтерии).

Поступающие на склад материалы регистрируются материально ответственным лицом (МОЛ) в книге учета соответствующих материалов (п. 54 приказа Минфина РФ «Об утверждении Методических указаний по

бухгалтерскому учету материально-производственных запасов» от 28.12.2001 № 119н). Также можно производить учет в карточке учета материалов. При этом применение книг либо карточек учета материалов равнозначно (п. 274 приказа № 119н).

Особенность книг складского учета – необходимость нумерации их листов и прошнуровывания. Форма книги складского учета материалов, имевшая номер М–17 (как и карточка складского учета, введенная в применение постановлением Госкомстата России от 30.10.1997 № 71а), была утверждена приказом Минфина РФ от 30.12.1999 № 107н, утратившим свою силу с 01.10.2005 (приказ Минфина РФ от 29.09.2004 № 87н).

В обязательном порядке ведется учет материальных ценностей, хранящихся на складах, документируются следующие данные:

- заведующий складом;
- наименование подразделения ответственного за размещение ТМЦ;
- персональные данные ответственного лица (ФИО, персональный код);
- уникальный номер склада, штрихкод ячейки согласно плану размещения ТМЦ;
- количество и единицы измерения;
- нормы стратегического запаса с указанием технических сведений и цены;
- классификационные характеристики (код, наименование);
- номер по порядку, дата поставки;
- реквизиты первичного документа (дата и номер товарно-транспортной накладной);
- информация о поставщике ТМЦ;
- сведения о поступлении и отгрузке (приход/расход).

В структуре документа также присутствует блок, отражающий факт проверки сведений в книге. В нем указывается дата проведения контроля, его результаты, должность проверяющего лица. Каждая запись в данном блоке заверяется подписью проверяющего.

На последней странице книги фиксируется количество ее пронумерованных страниц. Документ заверен подписью главного бухгалтера, которая проставляется им вместе с датой начала ведения книги – также на ее последней странице.

Для эффективного налаживания процессов на предприятии, внедрения и соблюдения внутренних контролей, а также недопущения недостат и ошибок в учетных данных на предприятии разработаны внутренние акты учета МПЗ. Способы бухгалтерского учета МПЗ описывается в учетной политике предприятия. Опишем основные пункты, которые раскрыты в регламенте учета МПЗ при его поступлении на склад:

- оформление заказа/заявки на закупку;
- проверка заказа на соответствие бюджету, плану или др. нормативам;
- утверждение заказа;
- приемка МПЗ на склад кладовщиком, в том числе описание процесса проверки входящих документов от контрагентов, самих купленных МПЗ;
- отражение кладовщиком данных в системе складского учета;
- передача первичных документов в бухгалтерию;
- сверка складского и бухгалтерского учета;
- выявление недостат, неотфактурованных поставок и пр.

С целью организации своевременного учета ТМЦ, на предприятии приняты к обращению документы внешнего и внутреннего происхождения.

Документы внешнего происхождения оформляются сторонними организациями, как правило — это сопроводительные документы. Они

поступают при приемке продукции на склад: товарная накладная и счет-фактура, товарно-транспортная накладная.

Документы внутреннего происхождения оформляются службами предприятия при перемещении материальных ценностей между подразделениями предприятия.

Контроль заполнения документа осуществляется бухгалтерской службой с определенной периодичностью, о чем делается запись в соответствующей форме о факте проверки, ее результаты, указываются сведения о проверяющем и ставится его подпись. Фиксируется дата начала заполнения и окончания с указанием количества страниц, информация подтверждается подписью главного бухгалтера [5].

В зависимости от предназначения, способа получения сведения о приходе, перемещении материальных ценностей, принадлежащих организации, МПЗ регистрируются на счетах:

- 41 – Товары;
- 40 – Произведенная продукция;
- 10 – Изделия, используемые во всех видах деятельности в качестве сырья, инструментов и обеспечивающие хозяйственные нужды.

Склад в логистических процессах организации является местом преобразования материальных потоков. Склад представляет собой часть логистической цепочки «поставка-производство-сбыт». Складская логистика имеет большое значение в системе логистической системе предприятия, влияя на способность конкурировать с другими предприятиями в современных экономических условиях.

Одним из способов совершенствования деятельности склада, является автоматизация его процессов. Правильно организованная работа логистического и складского комплекса является залогом непрерывной и

бесперебойной работы отдела обеспечения. Последовательность процессов в складской логистике показана на рис. 2.

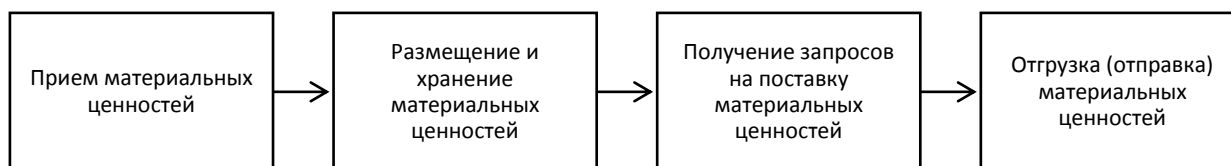


Рисунок 2 - Последовательность процессов в складской логистике

Материальные ценности поставляются;

- от поставщиков по счетам;
- от учредителей;
- безвозмездное поступление.

Разгрузка ТМЦ производится на складе в отведенной зоне по количеству поставленных тарных мест в соответствии с сопроводительными документами. Сотрудники склада проводят визуальную оценку целостности упаковки, комплектность и соответствие веса принятого материала заявленному.

При приемке материалов на склад оформляется приходный ордер или акт о приемке материалов в случае неотфактурованных поставок. Перемещение материалов внутри предприятия осуществляется с оформлением лимитно-заборной карты (отпуск со склада в производство), или требованием-накладной (накладной) на отпуск материалов, в том числе на сторону при отгрузке давальческого сырья. В случае возврата остатков на склад оформляется требование-накладная.

После приемки всем материально-материальным ценностям назначается инвентарный номер, для обеспечения удобного учета, контроля передвижения и выдачи. В случае выявления несоответствий сотрудник склада оформляет акт, информация доводится до руководства и поставщика [42].

1.1.2 Особенности учета материалов

В процессе работы организации часто сталкиваются с необходимостью учета материально-товарных ценностей, которые идут на производственные или управленческие нужды. Материально-производственные запасы являются весомой частью активов. Независимо от вида деятельности – предприятие обязано вести учет материально производственных запасов (МПЗ).

Деятельность подразделений предприятия и сотрудников регламентируются рабочими и должностными инструкциями. На предприятии применяются процедуры, разработанные в соответствии с законодательством Российской Федерации. Хранение материальных ценностей на складах организовано по группам в соответствии с назначением и по соответствующим субсчетам (табл. 1).

Таблица 1 - Классификация материальных запасов

Обозначение субсчета	Наименование субсчета
10.01	«Сырье и материалы»
10.02	«Комплекующие изделия, покупные полуфабрикаты»
10.03	«Топливо»
10.04	«Тара и тарные материалы»
10.05	«Запчасти»
10.06	«Прочие материалы»
10.07	«Материалы, переданные в переработку на сторону»
10.08	«Строительные материалы»
10.09	«Инвентарь»
10.10	«Специальная оснастка и спецодежда на складе»
10.11	«Специальная оснастка и спецодежда в использовании»

Контроль наличия и движения материалов производится с использованием как унифицированных форм (постановление Госкомстата РФ от 30.07.1997 № 71), так и внутренних документов, разработанных в

соответствии с требованиями закона «О бухгалтерском учете» от 06.12.2011 № 402-ФЗ.

К унифицированным формам относятся:

- требование-накладная;
- накладная на отпуск материалов на сторону;
- приходный ордер;
- лимитно-заборная карта.

Каждая группа материалов идентифицируется номенклатурным номером (единицей бухгалтерского учета), который представляет собой короткое числовое обозначение, содержащее информацию о характеристиках материалов.

Выбор единицы бухгалтерского учета материалов осуществляется в зависимости от характера МПЗ, порядка их приобретения и использования.

Единица бухгалтерского учета материально-производственных запасов выбирается предприятием самостоятельно таким образом, чтобы обеспечить формирование полной и достоверной информации об этих запасах, а также надлежащий контроль за их наличием и движением [42].

Принятие материальных запасов к учету производится по их фактической себестоимости [30]. Фактическая стоимость материальных запасов, которые были приобретены за плату, формируется из размера платы поставщику за материальные запасы. Учет НДС производится в зависимости от операции, на которой производится переработка материала, облагается она НДС или нет.

Если закупка материала осуществляется в иностранной валюте, то перевод в российский эквивалент осуществляется по курсу Центрального банка Российской Федерации на дату приемки продукции на склад, при оплате авансом, на дату перечисления денежных средств [5].

Грамотное списание материалов необходимо в целях получения достоверных данных о себестоимости продукции/услуг или величине расходов

компании за период. В соответствии с п. 16 ПБУ 5/01 выделяется 3 способа списания МЗ:

1. По себестоимости каждой отдельной единицы.
2. По себестоимости средней.
3. По себестоимости первых по дате покупки МПЗ (метод ФИФО).

Оптимальный способ списания предприятие выбирает самостоятельно с обязательным закреплением нормы в учетной политике. Запрещается изменять применяемый способ по каждой группе МПЗ в течение отчетного года (п. 21 ПБУ 5/01), за исключением тех случаев, когда требования законодательства отменены властями РФ. Каждый из упомянутых методов имеет свои особенности:

Таким образом прием материальных ценностей осуществляется с учетом цен приобретения. Те материальные ценности, которые получены другим, не по договору купли-продажи, способом оцениваются в соответствии со способом получения ценностей. Налогообложение отпуска материальных ценностей происходит по образцу бухгалтерского учета и обязательно оформляются документально [26].

1.2 Организация учета материалов

Ответственность за организацию отчетности в организации несет начальник склада. Учет материальных запасов ведется материально ответственными лицами. Договор об индивидуальной ответственности за недостачу имущества заключается с лицом старше 18 лет, непосредственно отвечающим и работающим с материалами, за которые отвечает (ст. 244 ТК РФ). МОЛ предприятия ведет учет всех материальных запасов, имеющих на складе. Последний должен быть оборудован тарой для возможных перемещений продукции.

Выдача МПЗ производится по соответствующим документам, подписанным руководителем отделения. Поступление на склад проводится на основании договоров. При этом происходит проверка поступившего товара.

После проверки товара на складе происходит его оформление, и данные отправляют в бухгалтерию, где уже и проходит оплата полученных материалов. Каждый месяц заведующий складом отчитывается перед бухгалтерией, представляя соответствующие документы.

Получение товара оформляется по накладным, если имеются какие-то различия между количеством на накладную и количеством по факту пишется акт приемки материалов. Что касается внутреннего перемещения, то здесь оформляется накладная – требования, причем все документы в двух экземплярах (один оформляется для склада, второй – для бухгалтерии).

Поступающие на склад материалы регистрируются материально ответственным лицом (МОЛ) в книге учета соответствующих материалов (п. 54 приказа Минфина РФ «Об утверждении Методических указаний по бухгалтерскому учету материально-производственных запасов» от 28.12.2001 № 119н). Также можно производить учет в карточке учета материалов. При этом применение книг либо карточек учета материалов равнозначно (п. 274 приказа № 119н).

Особенность книг складского учета – необходимость нумерации их листов и прошнуровывания. Форма книги складского учета материалов, имевшая номер М–17 (как и карточка складского учета, введенная в применение постановлением Госкомстата России от 30.10.1997 № 71а), была утверждена приказом Минфина РФ от 30.12.1999 № 107н, утратившим свою силу с 01.10.2005 (приказ Минфина РФ от 29.09.2004 № 87н) [26].

Склад в логистических процессах организации является местом преобразования материальных потоков. Склад представляет собой часть логистической цепочки «поставка-производство-сбыт». Складская логистика

имеет большое значение в системе логистической системе предприятия, влияя на способность конкурировать с другими предприятиями в современных экономических условиях.

Одним из способов совершенствования деятельности склада, является автоматизация его процессов. Правильно организованная работа логистического и складского комплекса является залогом непрерывной и бесперебойной работы отдела обеспечения. Последовательность процессов в складской логистике показана на рис. 3.

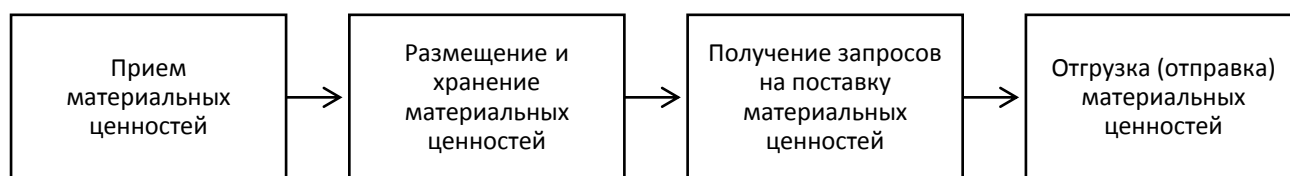


Рисунок 3 - Последовательность процессов в складской логистике

После приемки всем материально-материальных ценностям назначается инвентарный номер, для обеспечения удобного учета, контроля передвижения и выдачи.

1.3 Функциональное моделирование предметной области

Более подробно функции предприятия можно рассмотреть на бизнес-модели AS-IS («как есть»), созданной на базе методологии IDEF0. Модели в нотации IDEF0 необходимы с целью высокоуровневого описания бизнеса фирмы в функциональном плане. Нотация DFD дает возможность отражения последовательности работ, которые выполняются по ходу процесса, и потоков сведений, циркулирующих между данными работами [9].

На рис. 4 – 5 представлены модели процесса учета имущества и материальных ценностей в ООО «МДМ».

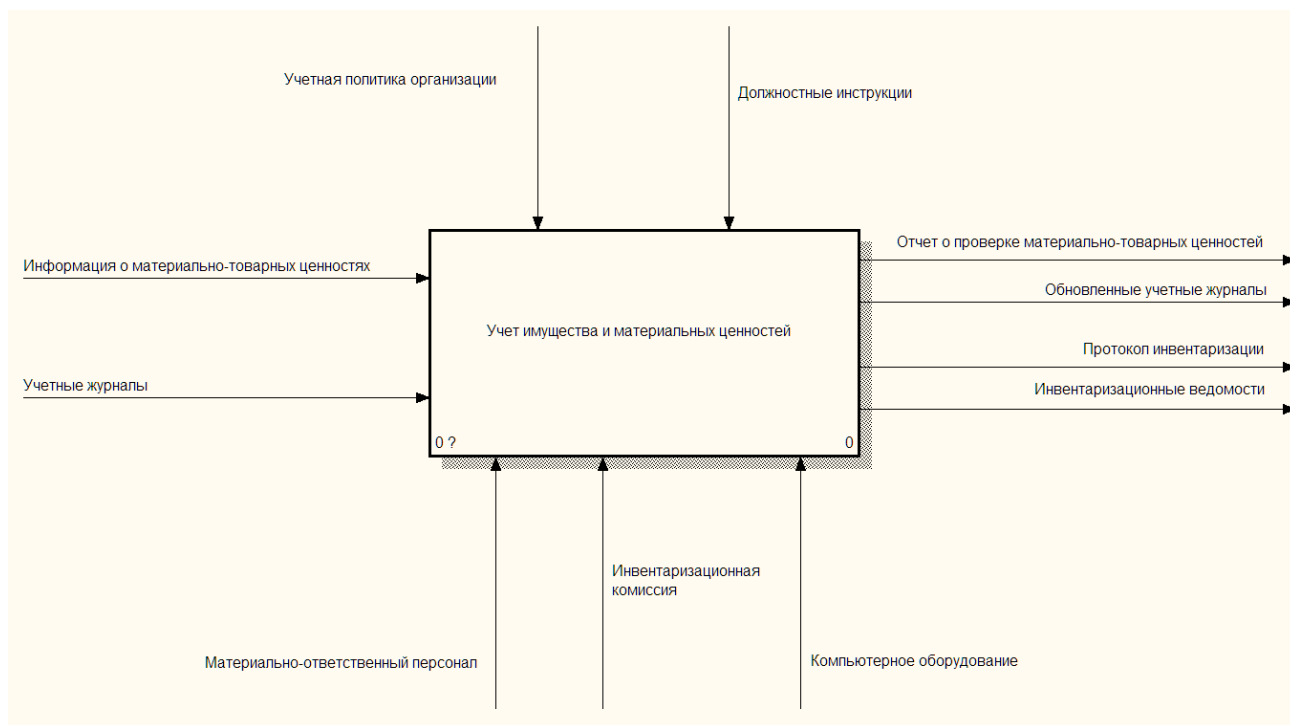


Рисунок 4 - Контекстная диаграмма А-0 «Учет материальных ценностей»

Входящими потоками процесса являются учетные журналы и информация о материально-товарных ценностях, на выходе находится обновленный журнал учета и отчет о проведенной проверке. В качестве механизмов выступает материально-ответственный персонал и компьютерное оборудование. Управляют процессом – учетная политика организации и должностные инструкции персонала.

Как видно из рисунка, учет имущества и материальных ценностей начинается с поступления информации о новых материально-товарных ценностях, а также о материально-товарных ценностях, которые описаны в учетных журналах. После этого учитывая политику организации и в соответствии с должностными инструкциями материально-ответственный

персонал проводит поиск объектов для проверки, находя их в журнале учета и определив физическое присутствие на складе путем сверки инвентарных номеров в книге учета и нанесенном на материальную ценность. В случае если материальная ценность только поступила, осуществляется присвоение ей инвентарного номера, запись ее в журнал и нанесение номера на материальную ценность. После этого формируется отчет о проверке материально-товарных ценностей и обновляются данные в учетных журналах [32].

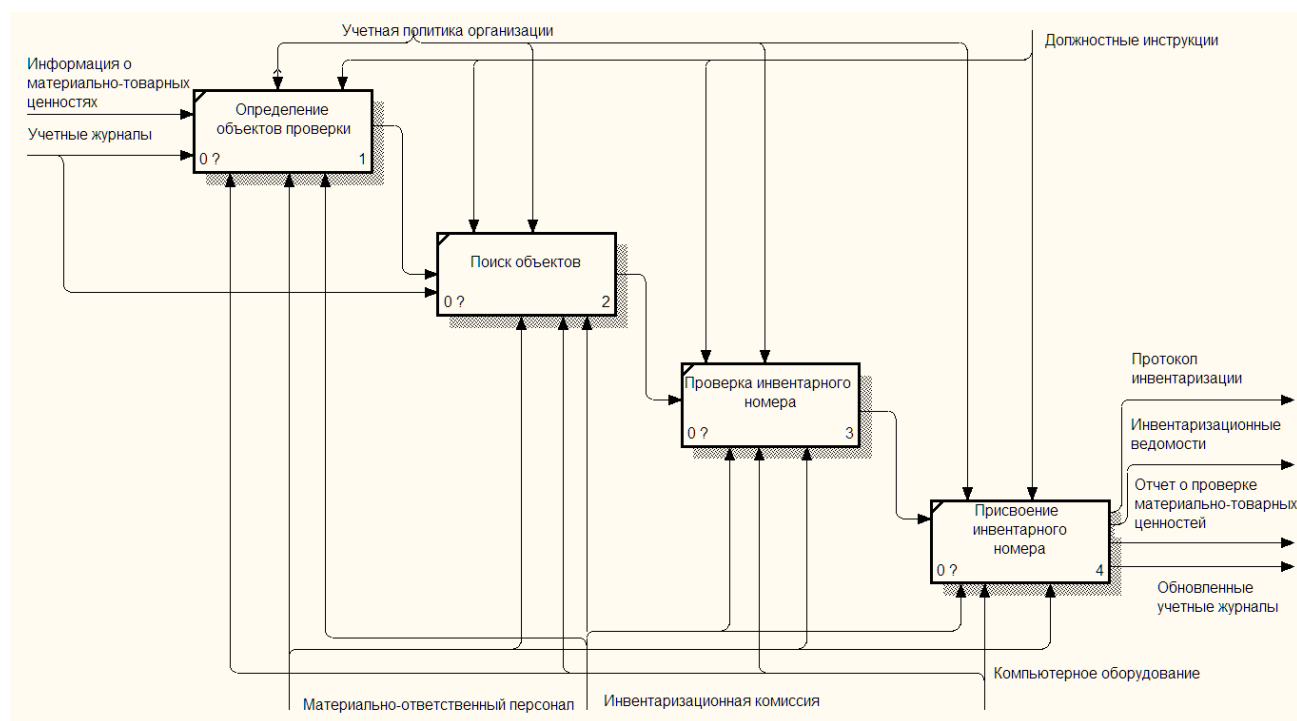


Рисунок 5 - Диаграмма уровня А0 «Учет материальных ценностей»

Таким образом, в результате анализа бизнес-процессов учета имущества и материальных ценностей, можно увидеть, что процессы генерации инвентарных номеров и присвоения инвентарных номеров не автоматизированы, также есть риск порчи инвентарных номеров, которые нанесены на материальные ценности, что может усложнить процесс учета. В связи с этим отделению необходима информационная система, которая обеспечит автоматизацию и

снизит риски порчи инвентарных номеров, которые нанесены на материальные ценности. Требования к системе: обеспечение целостности данных, защита доступа к информации, сохранение и редактирование записей о учетно-материальных ценностях, возможность работы через веб-интерфейс.

Инвентаризация проводится в конце отчетного периода с целью установления достоверной информации о наличии материалов на складах, периодичность устанавливается на предприятии с учетом специфики его деятельности.

Инвентаризация проводится на основании приказа руководителя предприятия, где указываются причины, объекты, подлежащие учету, состав комиссии и сроки выполнения работ. Инвентаризации подлежат ТМЦ, производственные запасы, продукция и прочие запасы.

В инвентаризационную комиссию включаются административные должностные лица и представитель бухгалтерии. Материально ответственные лица не могут быть включены в комиссию.

До начала инвентаризации материально ответственные лица осуществляют подготовку: вывоз остатков материала на склад, подготовку и передачу первичных документов в бухгалтерию. Своей подписью в соответствующих документах материально ответственные лица подтверждают, что ТМЦ, входящие в их сферу ответственности оприходованы, а вышедшие списаны или перемещены на склад.

Проверка ТМЦ проводится в присутствии членов комиссии и материально ответственного лица. В ходе инвентаризации проверяется наличие ТМЦ, указанных в ведомостях по месту хранения или применения, проводится выборочное или полное измерение веса каждой позиции. По окончании работы комиссии в помещении, принимаются меры ограничивающие доступ к помещению до окончания инвентаризации. При необходимости вход в

проверенное помещение может быть санкционирован руководителем в письменном виде.

Полученные данные фиксируются в инвентаризационной описи на бумажном или электронном носителе с соблюдением требований по заполнению. В случае использования бумажных носителей записи должны быть читаемы, не допускаются исправления и замазывание, ошибочно внесенная информация вычеркивается, рядом ставится актуальная запись и подпись вносившего исправление. Исправления вносятся в присутствии членов комиссии и материально ответственного лица до завершения процедуры инвентаризации. Полученные сведения передаются в бухгалтерию для дальнейшей обработки.

По завершении пересчета ТМЦ и материалов проводится сверка полученных результатов и данными бухгалтерского учета, оформляются сличительные ведомости по факту расхождений, если таковые выявлены. Недостатки и излишки, выявленные в процессе инвентаризации оцениваются в случае недостачи по приходным документа, излишки соответственно рыночной стоимости. Отдельно фиксируются материалы, находящиеся в производстве.

Сличительные ведомости оформляются в двух экземплярах, один остается в бухгалтерии, второй передается материально ответственному лицу.

Результаты инвентаризации анализируются членами комиссии, итоговые выводы и решения фиксируются в протоколе заседания комиссии. Ответственное подразделение разрабатывает мероприятия, направленные на устранение причин появления излишком или недостачи.

По итогам инвентаризации издается приказ, фиксирующий полученные результаты, в котором содержатся указания о дальнейших действиях, направленных на приведение в порядок документов, закрепляется ответственность за выполнение положений приказа.

Таким образом, в результате анализа бизнес-процессов учета имущества и материальных ценностей, можно увидеть, что процессы генерации инвентарных номеров и присвоения инвентарных номеров не автоматизированы, также есть риск порчи инвентарных номеров, которые нанесены на материальные ценности, что может усложнить процесс учета. В связи с этим отделению необходима информационная система, которая обеспечит автоматизацию и снизит риски порчи инвентарных номеров, которые нанесены на материальные ценности.

1.4 Обзор аналогичных программных средств

Автоматизация учета поступающего сырья, материалов и ТМЦ, движения материалов в процессе производства и учета готовой продукции актуальна в современной действительности. Соответственно на рынке имеются различные предложения для удовлетворения потребностей большого и малого бизнеса: АйТи; АиТСофт; АСК; Бизнес Сервис-Софт; Бэст; Омега; Риккон; Трансфер Эквипмент Восток; 1С складской учет; Эксперт; INFIN; Oracle; Renaissance; Учет ТМЦ; Большая птица; MoBill-Warehouse и другие.

Особое внимание привлекают следующие программные продукты: 1С (1С: Управление складом) и «Большая птица» (БОСС-Кадровик), «Учет ТМЦ». Внедрение автоматизации процессов учета значительно ускоряет работу с документами, делает ее прозрачной, обеспечивает прослеживаемость. Указанные продукты содержат справочники по основным направлениям, набор унифицированных форм документов, осуществляют учет по нескольким складам.

MoBill-Warehouse предназначена для обеспечения автоматизации предприятия в области организации учета товарно-материальных ценностей на складе предприятия и оформления всех необходимых отчетных документов.

Система автоматизирует работу складов по приемке, хранению, перемещению и отпуску ТМЦ, а также распределение ТМЦ по материально-ответственным лицам [32].

Внедрение системы способствует организации деятельности организаций по своевременному и ритмичному выполнению планов снабжения или распределения, позволяет организовать продвижение материалов с наименьшими издержками от поставщиков к потребителям. Возможности MoBill-Warehouse.

Ведение единой клиентской базы:

- работа с организациями и физическими лицами;
- учет связей между организациями и физическими лицами;
- доступ к единой базе данных по клиентам для всех служб и отделов;
- хранение всей истории взаимоотношений с клиентом;
- интеграция с КЛАДР;
- карточка контактного лица;
- карточка организации;
- учет сотрудников организации.

Организация учета ТМЦ:

- ведение справочника номенклатуры ТМЦ;
- учет инвентарных и серийных номеров;
- широкий набор операций движения ТМЦ;
- учет мест хранения ТМЦ, в том числе и по адресам (КЛАДР);
- учет хранения ТМЦ на складах, выдачи и возврата от материально-ответственных лиц, подразделений предприятия, сторонних организаций;
- описание структуры склада;

- выполнение комплектации и разуконплектации ТМЦ (полной или частичной);
- учет кабельной продукции;
- загрузка и прикрепление любых файлов (фотографий, инструкций, электронных документов) к карточкам ТМЦ;
- ведение учета ТМЦ по нескольким собственникам в одной системе;
- ведение складских журналов;
- формирование и выгрузка различных отчетов в системе;
- распечатка документов;
- возможность интеграции с бухгалтерскими программами.

Фирма 1С предлагает различные программы для автоматизации деятельности предприятия, в том числе складского учета. «1С: Торговля и склад» обладает гибкостью, имеется возможность настроить ее функционал под нужды любого предприятия. Она способна выполнять функции учета, от ведения справочников и ввода первичных документов, до формирования регламентированной отчетности, создания аналитических отчетов. Позволяет автоматизировать складской учет и торговлю (компонента "Оперативный учет"), осуществлять учет наличия и движения материальных и денежных средств, может использоваться автономно и совместно с другими компонентами.

Типовая конфигурация позволяет:

- вести отдельный управленческий и финансовый учет;
- вести учет от имени нескольких юридических лиц;
- вести партионный учет товарного запаса с возможностью выбора метода списания себестоимости (FIFO, LIFO, по средней);
- вести отдельный учет собственных товаров и товаров, взятых на реализацию;

- оформлять закупку и продажу товаров;
- производить автоматическое начальное заполнение документов на основе ранее введенных данных;
- вести учет взаиморасчетов с покупателями и поставщиками, детализировать взаиморасчеты по отдельным договорам;
- формировать необходимые первичные документы;
- оформлять счета-фактуры, автоматически строить книгу продаж и книгу покупок, вести количественный учет в разрезе номеров ГТД;
- выполнять резервирование товаров и контроль оплаты;
- вести учет денежных средств на расчетных счетах и в кассе;
- вести учет товарных кредитов и контроль их погашения;
- вести учет переданных на реализацию товаров, их возврат и оплату;
- вести учет разных типов ТМЦ: материалов, товаров, запчастей, продукции и прочего;
- поддерживать учет на нескольких удаленных друг от друга складах;
- вести учет одного товара в нескольких единицах измерения.
- проводить разбивку учета продукции по партиям, сортам и прочим атрибутам;
- проводить адресное хранение товаров;
- осуществлять расчет себестоимости товара, исходя из цены товара в отбираемой партии;
- распределять ответственности за каждую партию за конкретным кладовщиком;
- осуществлять отдельный учет собственных и принятых на хранение или реализацию товаров;

- поддерживать оформления всех операций по складу с выпиской утвержденных форм документов: внутреннего перемещения, прихода, списания, расхода, инвентаризации и прочих;
- выбирать статьи затрат и доходов при складских операциях;
- вести учет в себестоимости дополнительных расходов при оприходовании товаров;
- редактировать список комплекта во время его сборки;
- осуществлять формирование отчетов, оборотных ведомостей с широким выбором фильтров.

Складской учет "Склад+" – программа для формирования приходных и расходных документов. Позволяет печатать накладные, счета-фактуры, счета и приходные ордера.

Программа "Складской учет и торговля" предназначена для автоматизации складских и торговых операций в организациях оптовой и розничной торговли (рис. 6). Преимуществами программы являются простота, удобство и высокая скорость работы, низкие требования к аппаратному обеспечению [42].

Среди функциональных возможностей программы можно выделить следующие:

- автоматизация складского учета - программа позволяет автоматизировать все учетные операции по приходу и отпуску товаров;
- автоматизация торговых операций - реализована простая, но эффективная технология контроля над реализацией товаров;
- все операции отражаются в соответствующих документах, которые хранятся в журналах неограниченно долго;
- многопользовательский режим работы - с одной базой данных может одновременно работать несколько пользователей;

- встроенный генератор отчетов - пользователь может изменять вид документов и отчетов.

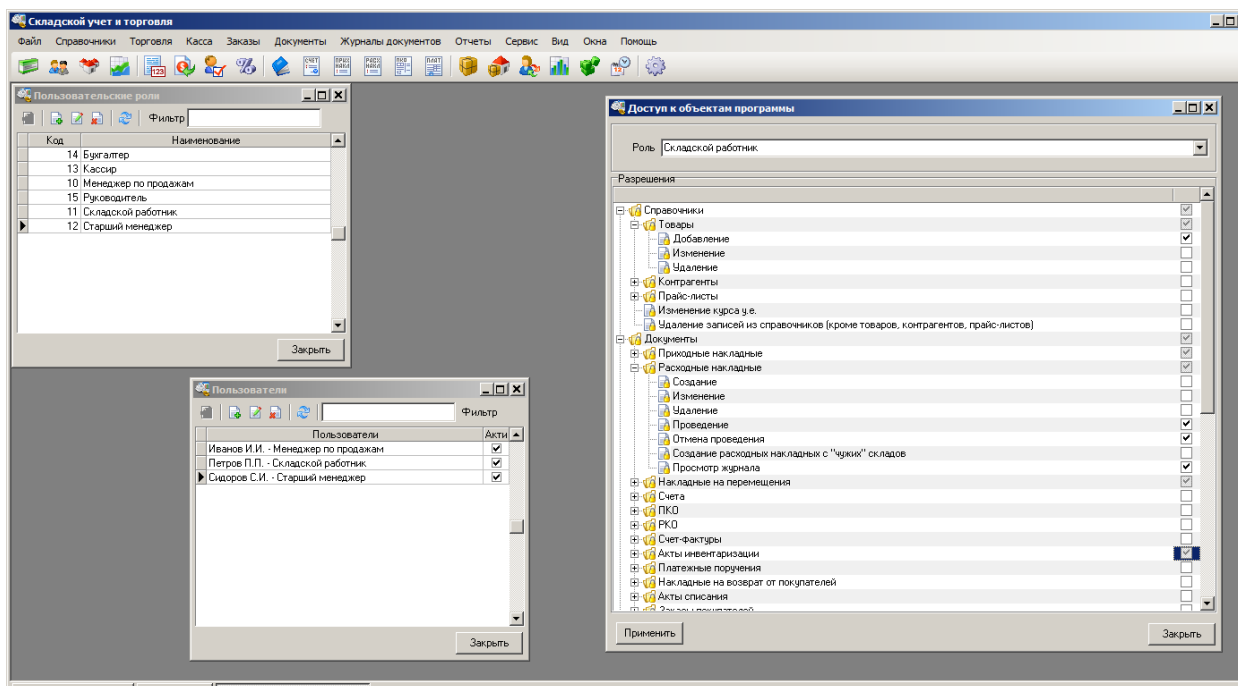


Рисунок 6 - Рабочее окно программы «Складской учет и торговля»

Отличительной особенностью программы "Складской учет и торговля" является ее ориентация на быстроту и удобство выполнения торговых операций. Программа включает средства для быстрого подбора товаров по заказу покупателей, предоставления скидок, формирования необходимого пакета документов "одним щелчком" мыши. В любой момент доступна полная информация о наличии товара на складе, зарезервированном товаре, ожидаемом приходе (транзите).

Складской работник получает возможность полностью контролировать складские остатки на одном или нескольких складах, проводить инвентаризации, списания и оприходования товара. Руководителю организации доступны отчеты по выручке, эффективности работы менеджеров, взаиморасчетам с контрагентами [32].

Таким образом, рассмотрев программы, можно сделать вывод, что основную задачу учета, они решают. Но не все удовлетворяют требованиям организации, таким как невысокая цена, простота обучения и работы с системой, учет специфики функционирования организации. В связи с этим принято решение в проектировании и разработки информационной системы, которая бы позволила автоматизировать систему инвентаризации на предприятии и учитывала особенности предметной области и предприятия.

Глава 2 ОБЗОР И ВЫБОР СРЕДСТВ РАЗРАБОТКИ СИСТЕМЫ УЧЕТА МАТЕРИАЛОВ

2.1 Обзор технологий разработки веб-ориентированных информационных систем

2.1.1 Статические и динамические веб-приложения

Все веб-приложения в сети могут быть двух типов: статические и динамические [14]. Статические приложения основаны на связанной структуре данных и хранят предоставляемую информацию.

Динамические веб-приложения отличны от статических. Динамический веб-сайт позволяет:

1. Уменьшать значение администратора через автоматизацию создания контента сайта и выдачи информации.
2. Информация каталогизируется с использованием баз данных или хранилищ.
3. Возможные функции: принятие данных от пользователя, обработка и сохранение их на сервере.
4. Размещение заказов, сложные арифметические действия и работа с базой данных.
5. Идентификация пользователя и отображение интерфейса.
6. Отображение инвентарных списков, заказов и сведений о товарах.

Для создания статических сайтов не надо особых знаний, при этом, такие сайты трудоемки и дороги в сопровождении. Динамические сайты сложны в разработке, но обеспечивают веб-приложения широкими возможностями формирования и управления контентом, что в перспективе экономит время администраторов сайтов. В связи с этим в качестве основания для разработки веб-ориентированной информационной системы будет применяться динамическое веб-приложение с базой данных.

2.1.2 Анализ технологий, которые используются при создании веб-ориентированных информационных систем

Кроме динамических и статических приложений, в современном информационном мире стали популярными и другие решения создания сайтов. Так, одним из наиболее популярных является использование возможностей систем управления контентом, которые помогают организовывать и управлять контентом веб-страниц. Системы управления контентом - системы, которые управляют не только информацией на сайте, но и всеми разрозненными и разнообразными фрагментами корпоративной информации. Контент хранится в базе данных, а на специальных языках пишутся программы, создающие HTML-странички, которые и показываются пользователю (Рис 7).

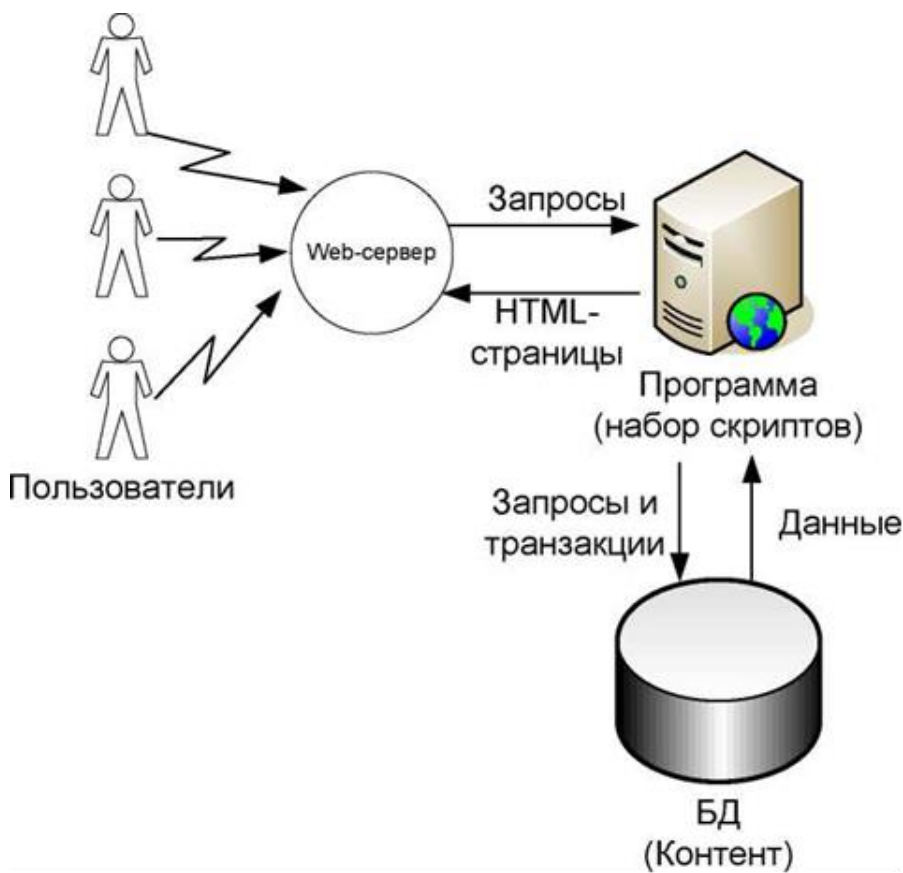


Рисунок 7 - Структура динамического веб-сайта



Рисунок 8 - Система управления контентом

Основные составляющие систем управления контентом: способ представления информационного наполнения сайта; способы описания и механизмы управления дизайном; механизмы авторизации и поддержки сообществ пользователей; механизмы управления контентом (управление структурой сайта); механизмы поддержки бизнес-процессов.

2.1.3 Анализ технических и программных средств веб-разработки

Для создания и хранения гипертекстовых документов предназначено большое количество как программных, так и аппаратных средств. Создаются гипертекстовые документы с помощью текстовых редакторов. В этих целях можно использовать среды разработки: PHPStorm, Netbeans, Eclipse, Lobster, и визуальные редакторы: Adobe Dreamweaver, Adobe Muse, Macaw.

Atom — это бесплатный текстовый редактор с открытым исходным кодом для OS X, Linux, Windows с поддержкой плагинов. Сайт программы - <https://atom.io>. Программа для редактирования текста и программного кода с

максимально привычным для разработчиков интерфейсом (рис. 9). "Обзорщик" позволяет переключаться между файлами и папками, перемещать текстовые документы по одному и массово, просматривать структуру директорий древовидным списком [26].

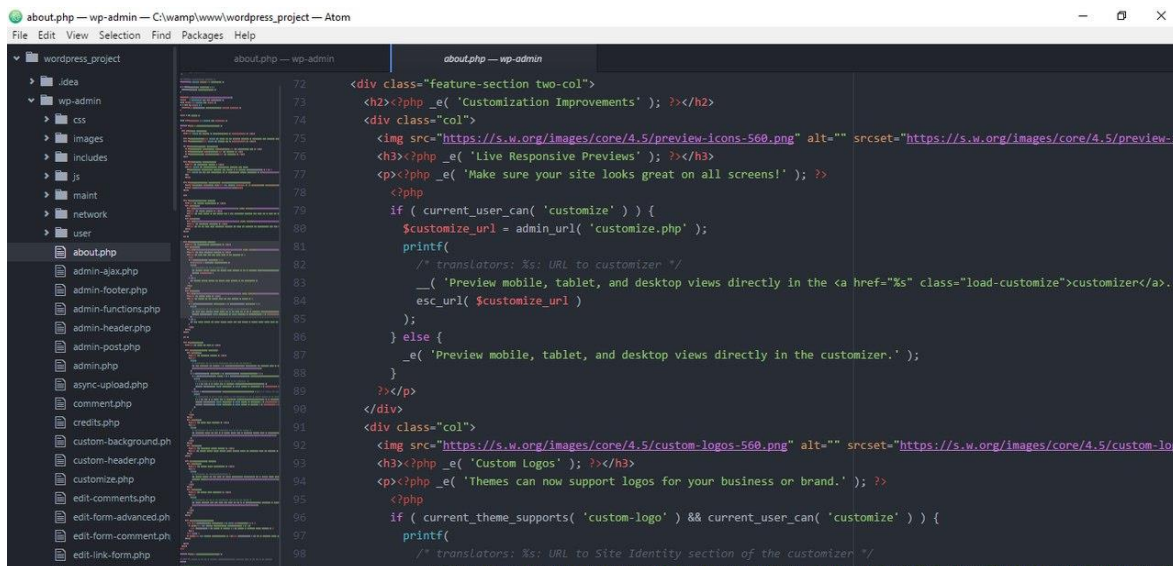


Рисунок 9 - Рабочее окно программы Atom

Ключевые особенности и функции редактора Atom [26]:

- Включает в себя удобный файловый менеджер с возможностью просмотра древовидным списком.
- Содержит функцию подсветки синтаксиса.
- Поддерживает массовое выделение и работу с несколькими курсорами.
- Работает со всеми существующими кодировками и форматами текстовых файлов.
- Является совершенно бесплатным.
- Возможна установка множества плагинов, которые значительно расширяют функционал редактора.

Notepad++ -текстовый редактор, предназначенный для программистов (рис. 10).

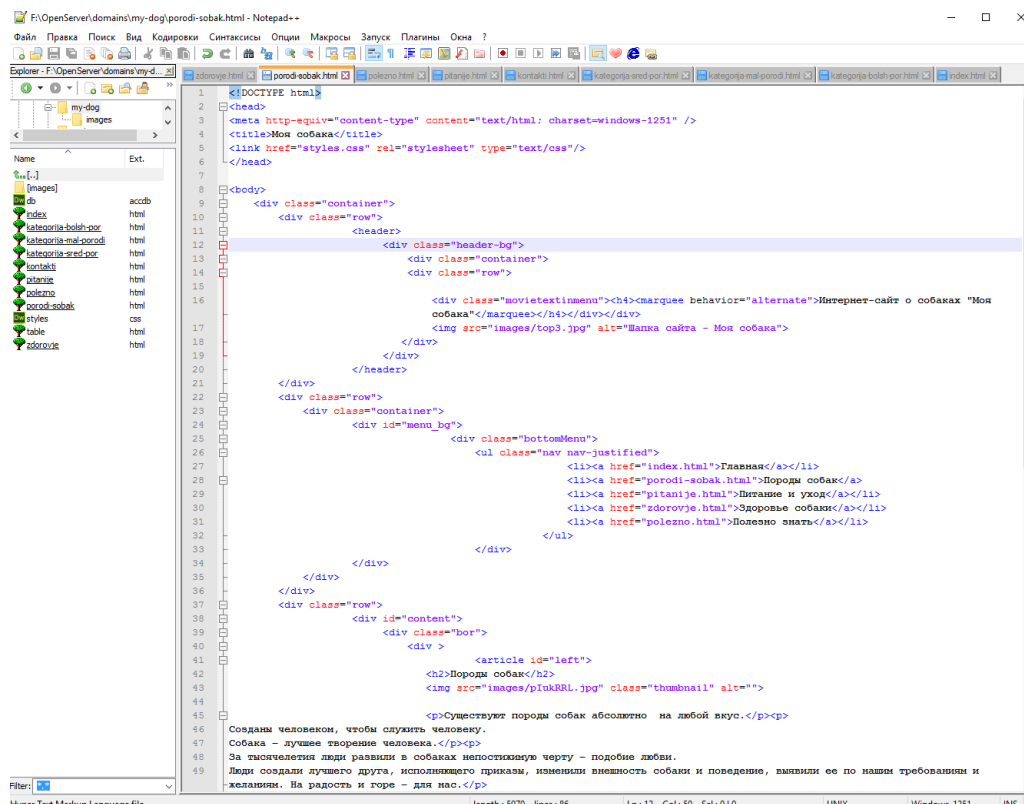


Рисунок 10 - Окно программы Notepad++

Основные особенности программы: - подсветка текста и возможность сворачивания блоков; - поддержка большого количества языков; - WYSIWYG (печатаешь и получаешь то, что видишь на экране); - настраиваемый пользователем режим подсветки синтаксиса; - авто-завершение набираемого слова; - одновременная работа с множеством документов; - одновременный просмотр нескольких документов; - поддержка регулярных выражений поиска/замены; - полная поддержка перетягивания фрагментов текста; - динамическое изменение окон просмотра; - автоматическое определение состояния файла; - увеличение и уменьшение; - заметки; - выделение скобок при редактировании текста; - запись макроса и его выполнение [27].

Sublime Text 2 (рис. 11) — это платный текстовый редактор, написанный на C++, который:

- Работает в Linux, OS X и Windows. - Обладает приличной скоростью работы. - Приятным интерфейсом (включая всевозможные анимации). - Гибко настраиваем (правда, не в GUI, а в json-конфигах).
- Имеет множество плагинов, число которых растёт как на дрожжах.
- Поддерживает VIM-режим. - Использует fuzzy-поиск.

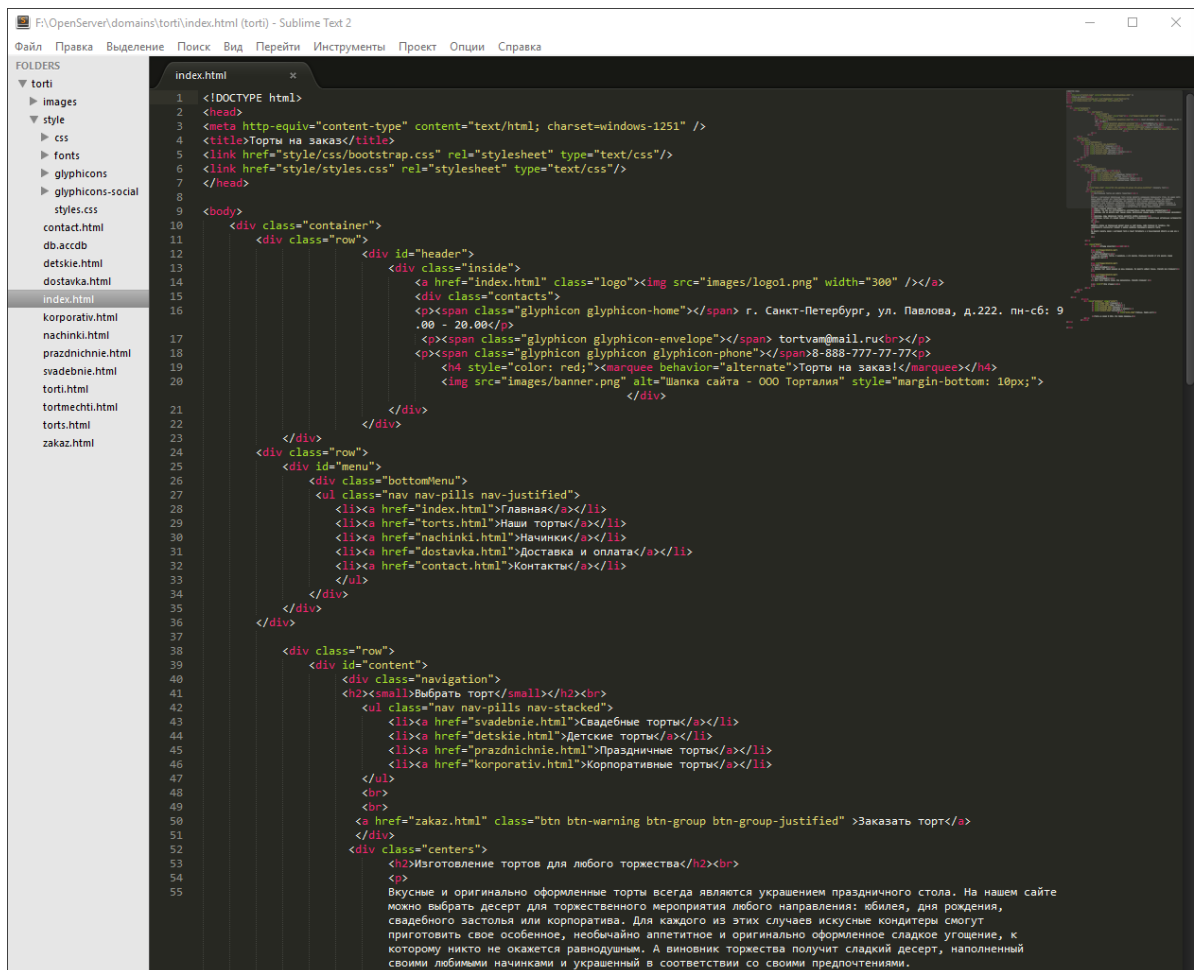


Рисунок 11 - Рабочее окно программы Sublime Text 2

RELEASES

Neon Packages
Oxygen Packages
Mars Packages
Luna Packages
Kepler Packages
Juno Packages
Indigo Packages
Helios Packages
Galileo Packages
Ganymede Packages
All Releases



Eclipse for PHP Developers

Package Description

The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

This package includes:

- Eclipse Git Team Provider
- JavaScript Development Tools
- Mylyn Task List
- PHP Development Tools (PDT)
- Eclipse XML Editors and Tools

Download Links

Windows 32-bit
Windows 64-bit
Mac OS X (Cocoa) 64-bit
Linux 32-bit
Linux 64-bit

Downloaded 0 Times

► Checksums...

Рисунок 12 – Страница с описанием Eclipse

Компания модернизировала интегрированную среду разработки (IDE) на JavaScript для платформы Eclipse. На рис. 13 показано рабочее окно программы, с файлами проекта.

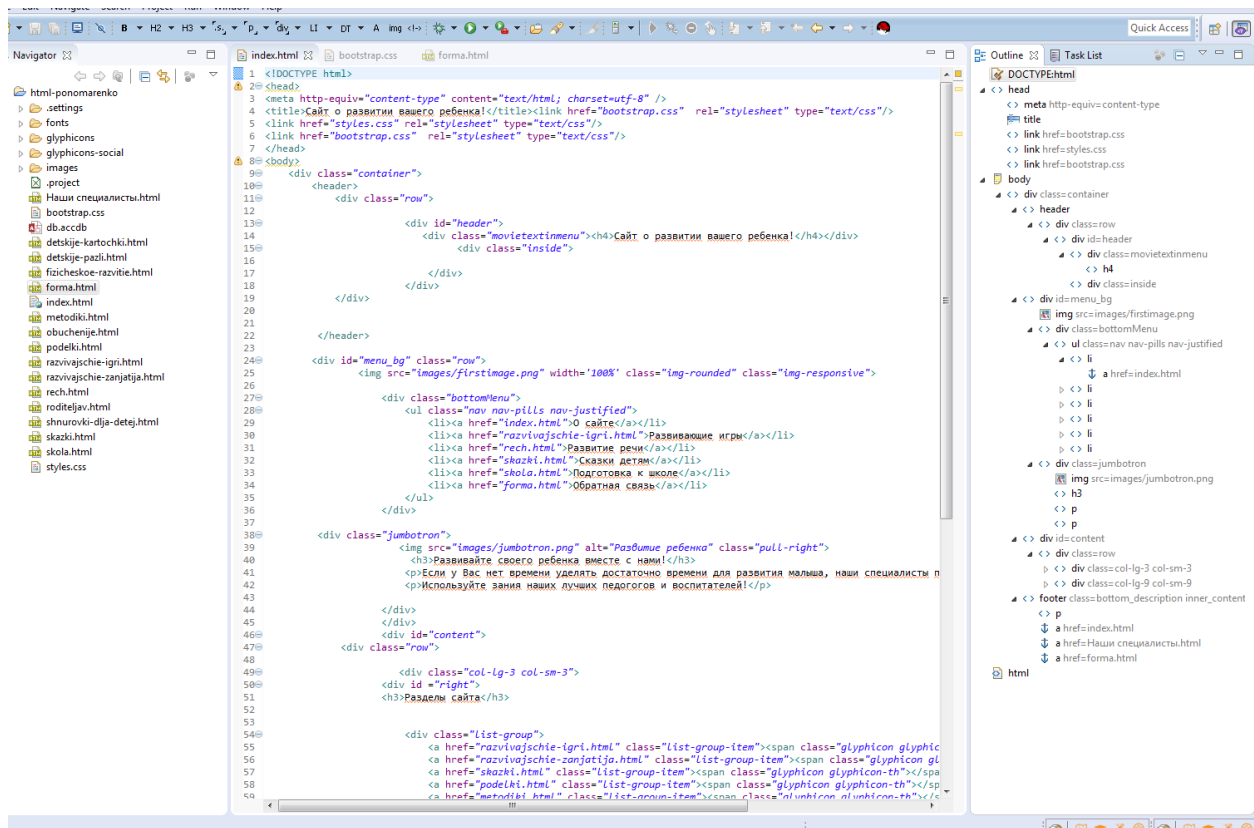


Рисунок 13 – Рабочее окно программы Eclipse Neon

Программный комплекс NetBeans (рис. 8). Продукт является интегрированной средой разработки (Integrated Development Environment, IDE) и платформой. NetBeans имеет встроенную поддержку следующих языков программирования: Java, C, C++, PHP, HTML и JavaScript.

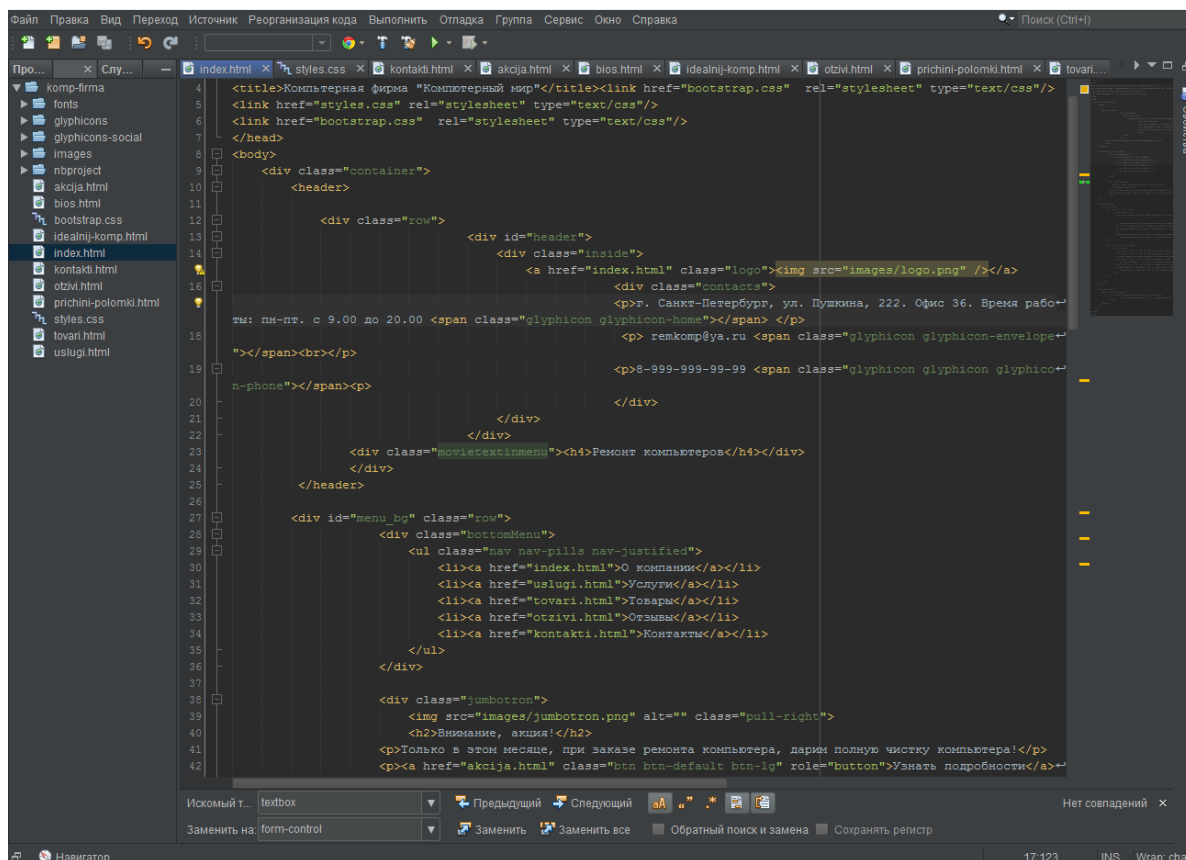


Рисунок 14 – Рабочее окно NetBeans

Визуальный редактор для создания сайтов - Adobe Dreamweaver (рис. 15). Dreamweaver упрощает и ускоряет разработку, написание кода и публикацию веб-сайтов и веб-приложений, которые хорошо выглядят на любом экране.

Adobe Dreamweaver поддерживает HTML, CSS, JavaScript и другие языки и технологии, — где бы вы ни находились. Специальные подсказки по коду позволяют быстро получать информацию о HTML, CSS и других стандартах веб-разработки и редактировать код, а визуальные средства помогают

уменьшить количество ошибок и ускорить разработку сайтов. Сайты созданные в Adobe Dreamweaver адаптируются к любому размеру экрана.

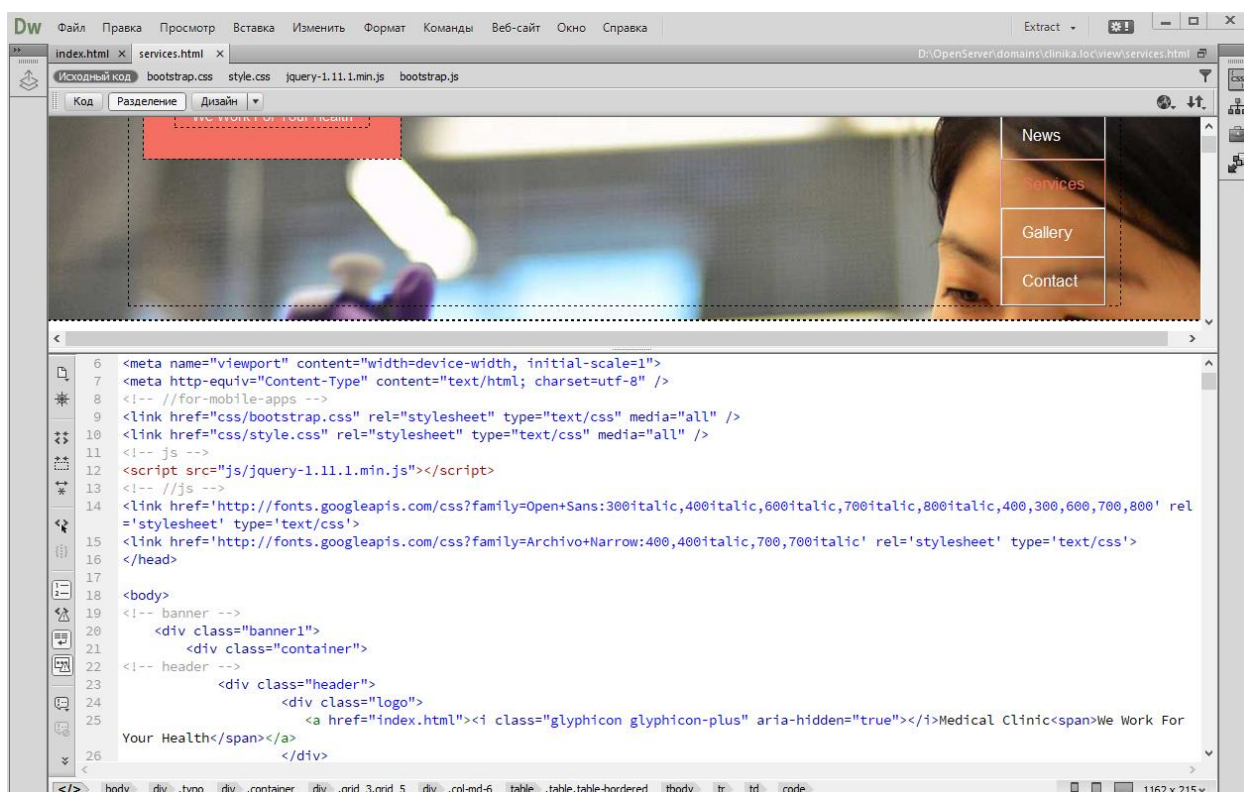


Рисунок 15 - Окно программы Dreamweaver

Новая версия редактора поддерживает передовые платформы (PHP, HTML5, jQuery и jQuery Mobile) и системы управления контентом (WordPress, Joomla! и Drupal) [18].

2.1.4 Язык гипертекстовой разметки – HTML

Язык разметки гипертекста используются в следующих целях:

- для гипертекстовых новостей, почты, сопровождающей информации и сопутствующей гиперсреды;
- меню с функциями;

- результатов запросов к БД;
- простых структурированных документов со встроенной графикой;
- гипертекстовых обзоров имеющейся информации.

Имеются теги, действие которых начинается и заканчивается одним открывающим тегом: `` - тег для вывода изображения `image.gif` в окно браузера; `<input type="text" value="">` - тег для ввода значений в окне браузера [17].

При создании HTML-документов рекомендуется соблюдать следующую структуру: `html> <head> <!-- Раздел заголовка --> </head> <body> <!-- Тело документа --> </body> </html>` Здесь `<html>` - это тег, обозначающий начало HTML-документа. Браузер, встретивший данный тег, будет «знать», что представленную ниже информацию следует интерпретировать как HTML-страницу. Тег `<head>` - раздел заголовка, который обычно включает (рис. 16):

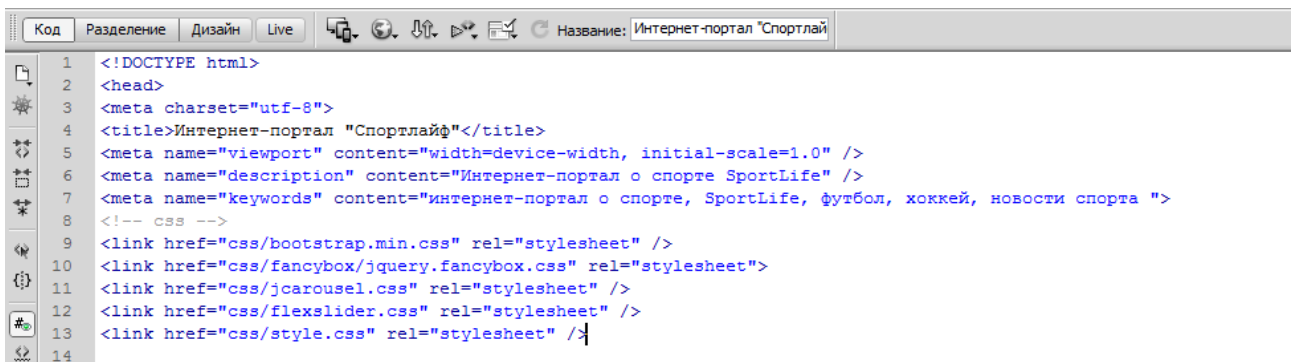


Рисунок 16 – Структура блока `head`

В конце тела страницы также нужно размещать подключаемые внешние javascript файлы (скрипты) (рис. 17), это ускоряет загрузку страницы, в сравнении, если эти скрипты подключаются в середине блока `<head></head>`.



```

337 <script src="js/jquery.js"></script>
338 <script src="js/jquery.easing.1.3.js"></script>
339 <script src="js/bootstrap.min.js"></script>
340 <script src="js/jquery.fancybox.pack.js"></script>
341 <script src="js/jquery.fancybox-media.js"></script>
342 <script src="js/google-code-prettify/prettify.js"></script>
343 <script src="js/portfolio/jquery.quicksand.js"></script>
344 <script src="js/portfolio/setting.js"></script>
345 <script src="js/jquery.flexslider.js"></script>
346 <script src="js/animate.js"></script>
347 <script src="js/custom.js"></script>
348 </body>
349 </html>

```

Рисунок 17 – Подключение скриптов в конце блока body

Текст HTML-документа представляет собой текст и ссылки, ведущие либо на отдельные фрагменты того же документа, либо на другие документы [10].

Программы, интерпретирующие данную ёданную разметку и выводящие информацию на экран, стали называться браузерами: Firefox, Chrome, Opera, Internet Explorer и др. [17].

2.1.5 Каскадные таблицы стилей – CSS

Для того, чтобы придать веб-приложению какой-либо конкретный вид применялись каскадные таблицы стилей. Существуют простые селекторы, которые можно применять для указанного элемента (можно посмотреть следующий пример – к любому заголовку h1, h2, h3: h1 {font-family:sans-serif} h2 {font-family:sans-serif} h3 {font-family:sans-serif}).

Селекторы класса:

*.firstclass{ color: green } - все элементы, имеющие class=firstclass или

.firstclass{ color: green } - все элементы, имеющие class=firstclass а также

H1.firstclass{ color: green } - только элементы H1, имеющие class=firstclass

Селекторы идентификатора ID:

- `h1#chapter1 { font-family: sans-serif }` - для `<h1 id="chapter1">...</h1>`
- `#chapter1 { font-family: sans-serif }` - для любого элемента с `id="chapter1"`

Селекторы атрибутов:

- `h1[class] { font-family: sans-serif }` - элемент имеет class
- `h1[class="fancy"] { font-family: sans-serif }` - элемент имеет class="fancy"
- `*[title] { font-family: sans-serif }` - любой элемент, имеющий заголовок

Селектор потомков (устанавливает иерархию применения):

`tr h1 { font-family: sans-serif }` - `<tr><td><h1>...</h1></td></tr>`

Псевдоклассы (особый вид динамических атрибутов, которые изменяются в зависимости от определенных действий):

- `a:link` - ссылки, которые не были посещены
- `a:visited` - посещенные ссылки
- `a:hover` - выделенная в данный момент ссылка
- `a:active` - активные ссылки.

При оформлении страницы доступны следующие семейства (family) шрифтов:

- **Serif** — шрифт с засечками. Обычно используется при бумажной печати. Наиболее используемый шрифт — Times;
- **Sans-serif** — шрифт без засечек. Подходит для заголовков. Наиболее часто применяемые шрифты этого семейства — Arial, Helvetica, Verdana;
- **Monospace** — шрифт, который обеспечивает равную ширину символов. Служит для вывода примеров кода. Наиболее распространен шрифт Courier;

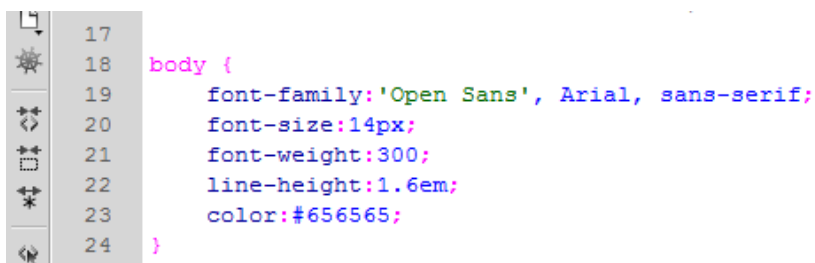
- Fantasy, Cursive — декоративные и курсивные шрифты. Не рекомендуются, поскольку шрифты этой группы необязательно присутствуют в компьютере.

Выбор шрифта осуществляется свойством `font-family`. Пример использования семейств стилей: `<p style="font-family: serif">Serif: Образец текста</p>`; `<p style="font-family: sans-serif">Sans-serif: Образец текста</p>`; `<p style="font-family: cursive">Cursive: Образец текста</p>`; `<p style="font-family: fantasy">Fantasy: Образец текста</p>`; `<p style="font-family: monospace">Monospace: Образец текста</p>`; Свойство `font-family` может содержать перечисление шрифтов; `p{font-family:"Times New Roman", Times, serif;}`

В этом случае браузер последовательно будет пытаться найти соответствующий шрифт в системе. Если конкретный шрифт не будет найден (в примере "Times New Roman", Times), то будет применен шрифт, назначенный для serif-семейства по умолчанию [25].

Толщина шрифта регулируется с помощью свойства `font-weight`: `font-weight: normal | bold | bolder | lighter | 100, 200.. 900`. Здесь `normal` — обычная толщина; `bold` — жирный шрифт; `bolder` — предельно жирный шрифт; `lighter` — тонкий шрифт; `100` — тонкий шрифт; `400` — соответствует нормальному; `700` — жирному.

Свойство `color` задает цвет шрифта (рис. 18, 19). Полный перечень допустимых значений свойства `color` приведен в спецификации.

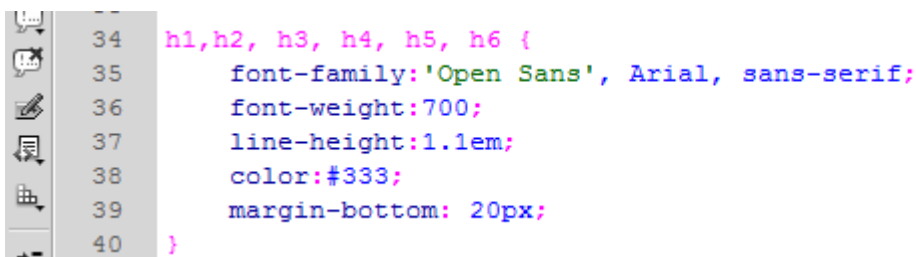


```

17
18 body {
19     font-family:'Open Sans', Arial, sans-serif;
20     font-size:14px;
21     font-weight:300;
22     line-height:1.6em;
23     color:#656565;
24 }

```

Рисунок 18 – Определение стилей для текста в блоке `body`



```

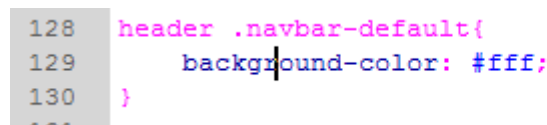
34 h1,h2, h3, h4, h5, h6 {
35     font-family:'Open Sans', Arial, sans-serif;
36     font-weight:700;
37     line-height:1.1em;
38     color:#333;
39     margin-bottom: 20px;
40 }

```

Рисунок 19 – Определение стилей для заголовков сайта

Фон элементов может быть задан однородным цветом, одиночным или мозаично расположенным [7].

Используются следующие свойства фона: background-color — однородный цвет константой или кодом в одной из допустимых систем цветности): header .navbar-default { background-color: #fff;}



```

128 header .navbar-default{
129     background-color: #fff;
130 }
131

```

Рисунок 20 – Определение фонового цвета для навигационного меню верхней части сайта

По спецификации может быть подвергнут преобразованиям при отображении таким как:

- изменение регистра букв (capitalize | uppercase | lowercase);
- изменение пробелов (collapse | preserve | preserve-breaks);
- ограничение длины строки;
- формирование переносов слов;
- форматирование текста;
- выравнивание и разреживание;
- отступы;
- декорирование.

Наличие внутреннего поля позволяет сформировать рамку на заданном расстоянии от содержимого элемента; наличие внешнего поля — установить отступ между рядом расположенными элементами [3].

Пример: `container { padding:0 20px 0 20px; }`

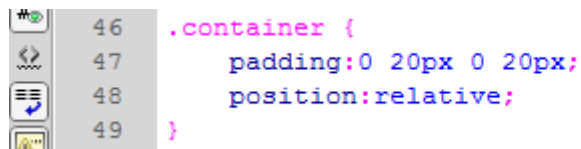


Рисунок 21 – Определение отступов для блока container

Она может быть задана единственным свойством `border` или отдельно для каждой стороны блока: `'border-top'`, `'border-right'`, `'border-bottom'`, `'border-left'`. Указываются толщина, тип начертания границы и цвет.

Внешнее прозрачное поле может быть задано либо единственным свойством `'margin'` с указанием одинакового размера границы для всех сторон, либо перечислением размеров по каждой из сторон, либо с использованием свойств `'margin-top'`, `'margin-right'`, `'margin-bottom'`, `'margin-left'` для каждой из сторон в отдельности.

Статическое позиционирование устанавливается для всех элементов по умолчанию и означает нормальное следование элементов. В явном виде спецификатор `static` применяется для перекрытия унаследованных стилей [20].

2.2 Выбор средств разработки системы учета материалов

2.2.1 СУБД для системы учета материальных ценностей

Это база данных, основывающаяся на реляционной модели. Реляционные системы по управлению БД используют для реляционных баз данных [40].

Каждая реляционная таблица, представляющая собой двумерный массив, должна обладать следующими свойствами [38]:

- каждый элемент таблицы представляет собой только один элемент данных;
- все ячейки в столбце таблицы являются однородными – все элементы столбца должны иметь одинаковый тип (INT, FLOAT, VARCHAR и т. д.);
- каждый столбец имеет уникальное имя, идентифицирующее данный столбец;
- в таблице не может быть одинаковых строк;
- порядок следования строк и столбцов произволен.

Реляционные БД построены на определенной теории этих баз данных. Эта теория основана как на теории множеств, так и на теории отношений. По сути, реляционная БД является неким количеством таблиц, связанных между собой. Записи - это строчки, а поля - элементы, из которых она состоит. Таблицы - это отношения в теории реляционных БД, записи называют кортежами, а поля именуются атрибутами [26].

Сама таблица реляционной базы данных не способна иметь в себе одинаковых записей (строк). Этот критерий унаследован из теории множеств. Ключ - это наименьшее число полей, которым и отличается запись от остальных записей. Все значения этого ключа обязаны быть уникальными в одной таблице. один ключ должен быть в каждой таблице, ведь таблица не имеет возможности содержать одинаковые записи. Атомарные, простые ключи - ключи таблицы, состоящие всего лишь из одного поля. Но они могут состоять и из более двух полей — это ключи составные. Таблица БД вполне способна содержать и один ключ, и несколько ключей. Один из этих ключей - первичный, а все оставшиеся - потенциальные (именно в теории реляционных БД) или альтернативные (в определенных воплощениях одних из баз данных) [29].

Ключи, которые включают в себя из поле (может и поля), не несущие данных из предметной области, но служащие некой заменой (суррогатом) обыденного (натурального) первичного ключа называются суррогатными первичными ключами. Их используют довольно часто. Но чаще всего суррогатные ключи - это счетчики (генераторы, последовательности) `autoincrement`, глобально-уникальные идентификаторы (GUID). Корректная структура БД обязана следовать спец. правилам, базирующихся на теории отношений и именованных нормальными формами [11].

Данное определение введено при создании реляционной модели баз данных Эдгаром Коддом.

Реляционная база данных – это таблицы, связанные между собой. Строки называют записями, а элементы записи — полями. В теории же реляционных баз таблицы - это отношения, записи именуют кортежами, а поля — атрибутами [26].

Таблица такой БД имеет способность включать в себя как единственный ключ, так и несколько ключей. Один ключ становится первичным , а все прочие - потенциальные (в теории реляционных БД) и альтернативные (в определенных воплощениях баз данных) [29].

Крайне часто используются ключи суррогатные, первичные. В виде суррогатных ключей часто используются счетчики (генераторы, последовательности) `autoincrement` или глобально-уникальные идентификаторы (GUID). Корректная структура БД просто обязана следовать специальным нормам, базирующимся на теории отношений, называемых нормальными формами [11].

Нормальная форма – это понятие ввел Эдгаром Коддом. Нормализация – это процесс преобразования базы данных, к виду, который отвечает нормальным формам и обеспечивает минимальную избыточность. Цель нормализации – защита базы данных от структурных и логических проблем,

являющимися аномалиями данных. К примеру, если есть несколько одинаковых записей в таблице, то в этом случае есть риск нарушения целостности данных при последующем обновлении таблицы. Таблица, прошедшая нормализацию, меньше подвергается этим проблемам, поскольку структура такой таблицы определяет связи между данными, таким образом исключается необходимость существования записей с информацией, которая повторяется. Избыточность можно устранить при помощи разбиения самих отношений (таблиц) так, чтобы в каждом из отношений всегда сохранялись первичные факты (факты, не выводящиеся из прочих фактов). Следуя из этого заключения, у нормализации нет цели снижения, повышения производительности работы или также уменьшения, или увеличения самого объёма БД. Уменьшение противоречий, хранимых в базах данных информации - вот конечная цель нормализации. Нормализация применима к таблице, которая представляет собой правильное отношение.

Работа с базами данных делается через системы управления базами данных (СУБД). Почти все управление реляционными БД обеспечивает слаженную работу с языком SQL, который определяет и модифицирует структуру данных, добавляет, изменяет и удаляет данные, а также делает разнообразные выборки данных [17].

Наиболее востребованы серверы реляционных баз данных, основанные на клиент-серверной архитектуре. Они обеспечивают работу с базами данных множества клиентов (десятки, сотни, тысячи и миллионы клиентов. Их число зависит только от оборудования и ПО) [41].

Реляционные СУБД делятся на системы с открытым кодом (OpenSource) и с закрытым. Согласно исследованиям компании IDC, компании из Европы чаще отдают предпочтение открытому программному обеспечению (ПО). Делают это они из-за повышения качества и гибкости данного ПО [20].

Система управления базами данных представляет собой программное обеспечение, с помощью которого пользователи могут создавать, определять и поддерживать базу данных. Особенность СУБД с открытым кодом является то, что они имеют при себе исходный код, который можно изменить, оптимизировать, ускорить за счет какого – либо алгоритма.

В Таблице 2 представлен мировой рейтинг топ-10 баз данных по версии GitHub TOPDB (Top Database Index) [30], что основан на количестве поисковых запросов.

Таблица 2 - Рейтинг СУБД [30]

Позиция в рейтинге	СУБД	Рейтинг	Тренд
1	Oracle	28.87 %	-0.3 %
2	MySQL	17.97 %	-0.6 %
3	SQL Server	14.12 %	+0.2 %
4	Microsoft Access	8.36 %	-0.4 %
5	PostgreSQL	4.43 %	+0.6 %
6	MongoDB	4.19 %	+0.5 %
7	Firebase	3.25 %	+0.4 %
8	Elasticsearch	2.69 %	+0.2 %
9	Splunk	2.63 %	+0.3 %
10	Redis	2.2 %	+0.1 %

Таблица 3 - Сравнение СУБД

Параметры	MySQL	PostgreSQL
Краткое описание	Широко используемая свободная реляционная система управления базами данных	Широко используемая свободная реляционная система управления базами данных
Основная модель хранения данных	Реляционная база данных	Реляционная база данных
Дополнительная модель хранения данных	База данных типа Key/Value, документно-ориентированная база данных	База данных типа Key/Value, документно-ориентированная база данных
Разработчик	Oracle	PostgreSQL Global Development Group
Лицензия	Открытое программное обеспечение	Открытое программное обеспечение
Язык реализации	C++, C	C
Поддерживаемые операционные системы сервера	FreeBSD, Linux, Solaris, OS X, Windows	FreeBSD, Linux, Solaris, OS X, Windows, NetBSD, OpenBSD, HP-UX, Unix
Схема данных	Да	Да
Типизация	Да	Да
Поддержка XML	Да	Да

Поддержка вторичных индексов	Да	Да
SQL	Да	Да
API и другие методы доступа	Проприетарное нативное API, ADO.NET, JDBC, ODBC	Нативная C библиотека, потоковое API для больших объектов, ADO.NET, JDBC, ODBC
Поддерживаемые языки программирования	Ada, C, C#, C++, D, Delphi, Eiffel, Erlang, Haskell, Java, JavaScript (Node.js), Objective-C, OCaml, Perl, PHP, Python, Ruby, Scheme, Tcl	.Net, C, C++, Delphi, Java, Perl, PHP, Python, Tcl
Язык написания скриптов на стороне сервера	Да	Функции, определенные пользователем
Триггеры	Да	Да
Методы разбиения	Горизонтальное разбиение, шардинг с MySQL Cluster или MySQL Fabric	декларативное разбиение (по диапазону или списку) начиная с PostgreSQL 10.0
Методы репликаций	Master-Master, Master-Slave	Master-Slave
MapReduce	Нет	Нет
Концепции согласования	Немедленное согласование	Немедленное согласование
Параллелизм	Да	Да
Возможность хранения только в памяти	Да	Нет
Контроль доступа пользователей	Концепт пользователей с детальной авторизацией	Детальные права доступа в соответствии с SQL стандартом

К преимуществам СУБД MySQL относится:

- упрощение ввода записей;
- упрощение поиска записей;
- гибкость поиска записей;
- гибкость вывода записей;
- доступ к записям многопользовательский;
- возможность передачи записей в электронном виде.

2.2.2 Язык программирования для создания системы

Имеется огромное количество языков программирования в среде разработки веб-приложений. Благодаря им имеем возможность создавать приложения сервера. Сравнение различных языков указано в таблице 4.

Таблица 4 - Сравнение серверных языков программирования

	PHP	Ruby	Python
Предназначение	Создание веб-приложений	Создание веб-приложений и компьютерных программ	Создание веб-приложений, компьютерных программ, анализ данных
Разработан под влиянием	C, PERL, JAVA, C++, TCL	ADA, C++, CLU, DYLAN, EIFFEL, LISP, PERL, PYTHON	ABC, ALGOL 68, C, C++, ICON, JAVA, LISP, PERL
ООП подход	Да	Да	Да
Доступные средства разработки	+	+-	+
Наличие сообществ поддержки	+	+-	+
Наличие библиотека программного кода	Множество библиотек разного назначения	Библиотек программного кода не много	Множество библиотек разного назначения
Рейтинг (TIOBE) [29]	2,4	1	9,7
Популярные веб-фреймворки	Laravel, Yii2, CodeIgniter, Symfony, CakePHP	Ruby on Rails, Sinatra, Roda, Ramaze	Django, TurboGears, web2py, CubicWeb, Flask
Поддержка платформ и веб-серверов	Linux, Windows, Mac	Linux, Windows, Mac	Linux, Windows, Mac

Язык PHP по параметрам более подходит для веб-разработки, что видно в таблице, именно поэтому для самой разработки программы мы будем пользоваться именно языком программирования PHP. Он создан в 1994 году программистом Расмусом Лерддорфом из Дании и с самого начала являлся набором скриптов на совершенно ином языке Perl. Чуть позднее данный набор переписали в интерпретатор на Си языке. Ведь как только возник PHP (это сокращение от PHP: Hypertext Preprocessor - PHP: Препроцессор гипертекста), он являлся комфортным набором инструментов необходимых именно для простого создания как веб-сайтов, так и веб-приложений.

Данный язык PHP осуществляет свою деятельность на сервере и является неким дополнением. Интерпретируется сервером и выдает HTML-код, любой другой вывод именно код PHP, отсылаемый браузеру пользователя [16].

PHP используют для разработки веб-сайтов всех направлений. Последовательность команд хранится в файле. Эти команды исполняет веб-сервер. PHP прочитывает файл по строке, сверху вниз и реализует команды, которые были записаны [33].

Одна из главных возможностей PHP - это конечно же работа с файлами, которая позволяет нам сохранять информацию между самими вызовами скриптов.

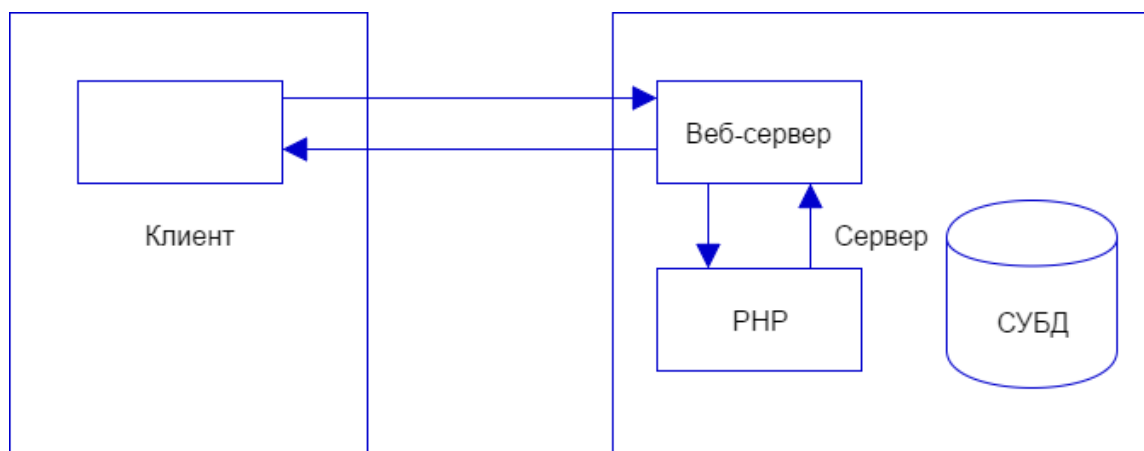


Рисунок 22 - Архитектура веб-приложений, реализуемых на языке PHP

В скрипте PHP доступны способы открытия доступа к данным, которые клиент уже предоставил по известному протоколу http. Для обращения к переменным, которые были переданы HTTP-запросами, мы используем определенный массив - `$_REQUEST`.

Исключая крайне глобальный массив `$_REQUEST` представлено и пара суперглобальных ассоциативных массивов, которые дают нам возможность обрабатывать данные, предоставляемые человеку:

- `$_GET[]` - содержащая абсолютно каждое значение, передающееся в сценарий используя метод формы GET;
- `$_POST[]` - содержащий значения, передающиеся в сценарий с помощью формы POST и ее метода;

– `$_SERVER[]`. Он содержит все значения, приходящие от сервера.

Сама архитектура данного приложения базируется на способе клиент-сервер, с помощью MVC шаблона.

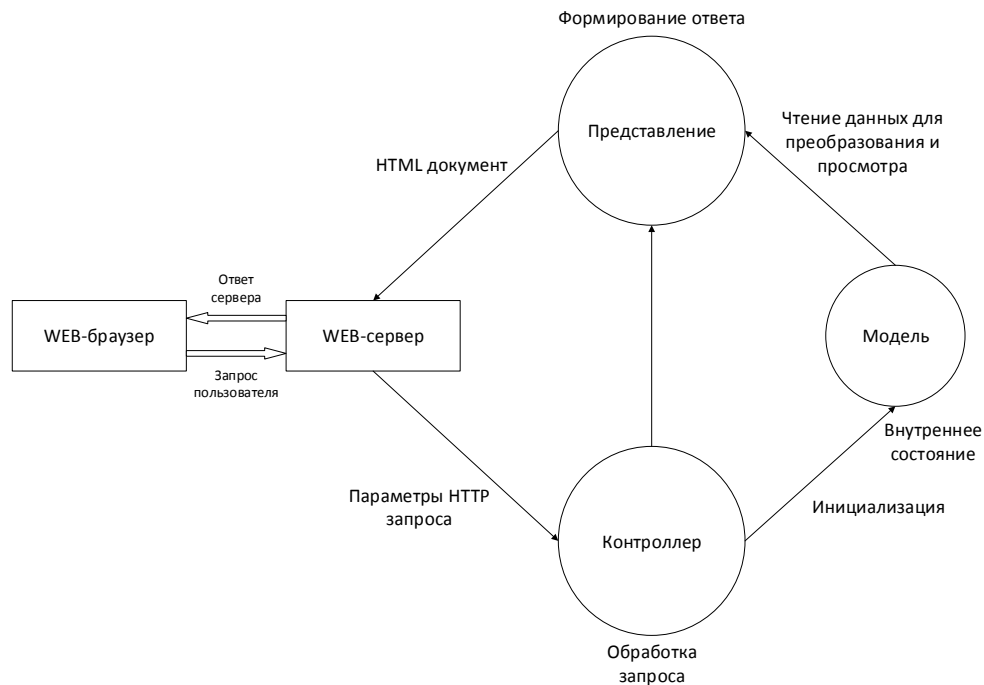


Рисунок 23 - Структура приложения

Сама суть шаблона - это раздел приложения на три отрасли: представления, модели, контроллеры.

Модели например, являются той частью приложения, что реализует за бизнес логическую последовательность. Модели ответственны за показ в нужном виде полученных данных.

Контроллеры же отвечают за правильную обработку всех запросов от пользователей, за появление ответа со слоями моделей и различных представлений.

Контроллер вполне способен являться менеджером, отвечающим за передачу всех ресурсов, необходимых для обработки запроса к соответствующим слоям приложения.

Разнообразные приложения способны иметь данные сущности за исключением MVC, Yii:

- входные скрипты: так называемые скрипты PHP, доступные только последнему пользователю приложения;
- масштабно имеющиеся в доступе объекты, осуществляющие правильную, слаженную работу элементов приложения, их ориентирование для обработки запроса - приложения;
- элементы приложения - объекты, которые были зарегистрированы в приложении. Они предоставляют самые разные возможности необходимые для обработки нынешнего запроса;
- модули - самостоятельные пакеты, имеющие в себе абсолютно все MVC средства. Приложение способно организовать и при помощи пары модулей;
- код, необходимый для выполнения и до, и после обработки контроллеров - это фильтр;
- виджеты - объекты, имеющие возможность состоять из представления. Способны включать в себя разные логические цепочки, а также виджеты используются и в разных представлениях.

Схему структуры приложения вы видите на рис. 24. Запросы, которые обрабатываются Yii приложением имеют похожий путь:

- Для начала создается запрос для входного скрипта `web/index.php`.
- Он загружает саму форму, создает экземпляр приложения необходимый для обработки запроса.
- Определяет запрашиваемый маршрут с компонентом `request` приложение.
- Оно же воспроизводит экземпляр контроллера.
- Контроллер в свою очередь, собирает экземпляр действия, реализует фильтры этого действия.

- действие не осуществляется при любом некорректном исполнении фильтров.
- действие воплощается только при удачном выполнении абсолютно всех из фильтров .
- Из базы данных действие имеет возможность загружать модель данных.
- Действие сразу рендерит представление, а также предоставляет ему саму модель информации.
- Итоги рендеринга отправляются в элемент приложения response.
- Он передает уже готовые данные пользователю.

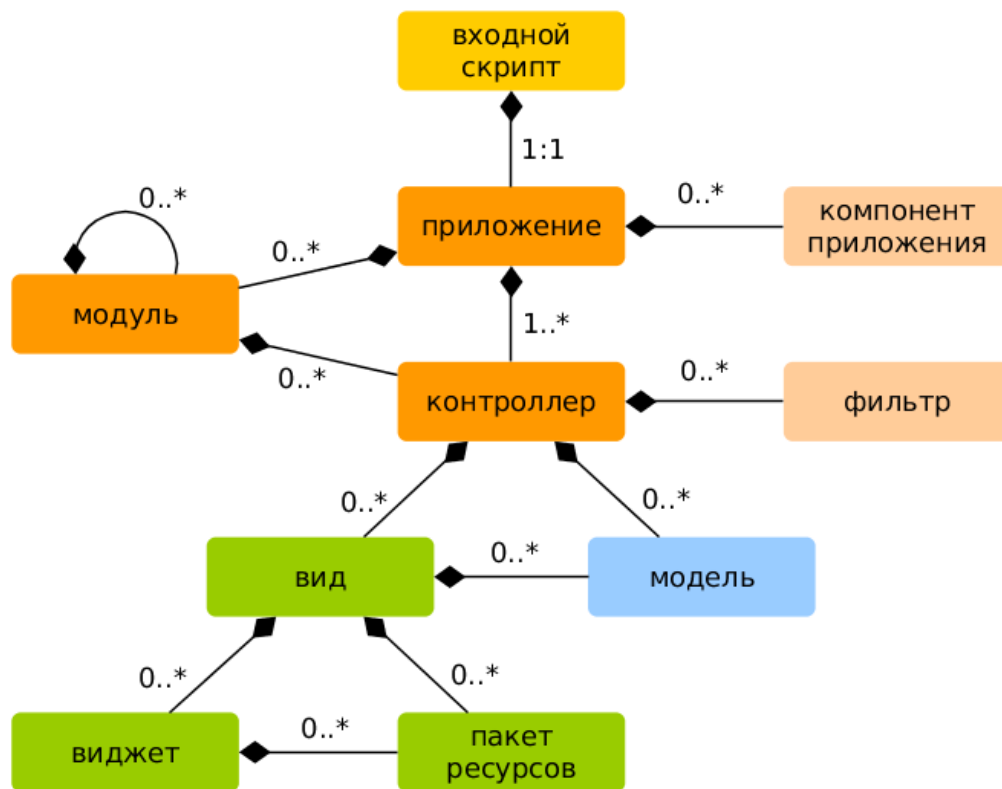


Рисунок 24 - Схема приложения

На рис. 25 представлена диаграмма обработки запроса приложением.

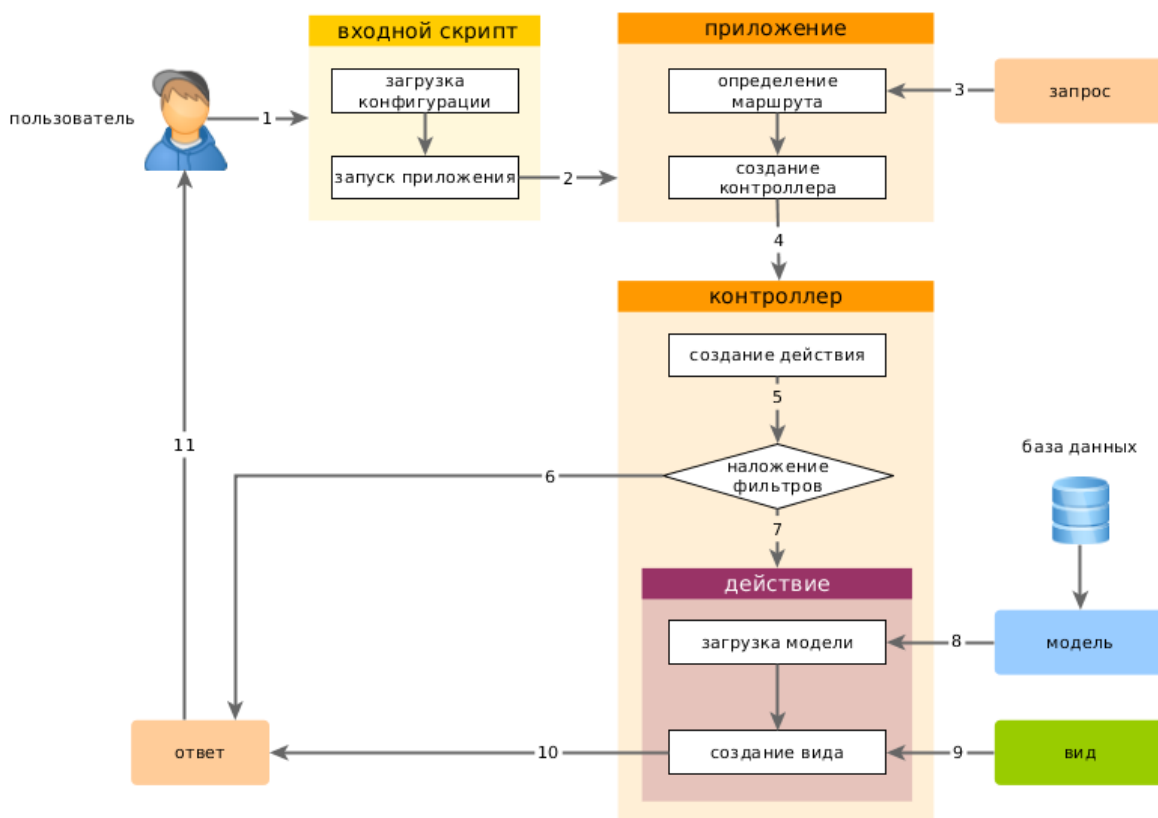


Рисунок 25 - Схема обработки запроса

Представленные выше данные показывают программное обеспечение, которое будет использовано при построении системы учета материалов ООО «МДМ».

2.2.3 Среда разработки для создания системы учета материалов

Есть большое количество сред, которые способны поддерживать язык PHP. Сравнение распространенных сред представлено в таблице 5.

Таблица 5 - Сравнение сред разработки

Критерии оценки	Visual Studio	PHPStorm	NetBeans	Komodo
-----------------	---------------	----------	----------	--------

Наличие встроенной СУБД	+	-	-	-
Стоимость лицензии				
бесплатная	+		+	+
платная		+		+
Платформа, на которой способна работать среда разработки				
Windows	+	+	+	+
Unix/Linux		+	+	
Windows+Linux		+	+	
Требования к техническим характеристикам ПК				
высокие	+	+		+
не высокие			+	
Сложность настройки и администрирования				
никаких сложностей		+	+	
Скорость работы				
быстро		+	+	
медленно	+			+
Функционал				
Интеграция с БД (в том числе и MySQL)	-	+	+	-
Автодополнение PHP-кода	+	+	+	+
Поддержка PHP	Дополнительный модуль	Встроенная	Встроенная	Встроенная
Поддержка PHP-фреймворков	-	+	-	-

Ключевые возможности PHPStorm:

- Редактор интеллекта кода PHP, имеющий подсветку синтаксиса, автодополнение кода, настройки расширения кода для форматирования, а также оперативное предотвращение ошибок
- Он работает и со всеми синтаксическими улучшениями PHP 7.0, 5.6, 5.5, 5.4 и 5.3, генераторами, и конечно же с сопрограммами
- Рефакторинги PHP, code (re)arranger и детекторы кода дубликации
- Поддерживает как Vagrant, Composer, встроенного REST-клиента, так и Command Line Tools, консоль SSH
- Фреймворки (MVC view для Symfony2, Yii), спец. плагины для главных фреймворков PHP (Symfony, Magento, Drupal, Yii, CakePHP и многие другие) он также поддерживает

- Отладчик визуала приложений PHP, валидация формы отладчика, PHPUnit, имеющий кодовое покрытие (поддержание PHPUnit 5), и разумеется интеграция профилировщика
 - Редакторы HTML, CSS, JavaScript. А также отладка и тест модулей JS. HTML5, CSS, Sass, SCSS, Less, Stylus, Compass, CoffeeScript, TypeScript, ECMAScript Harmony, Emmet, прочие технологии веб-разработки этот код поддерживает
 - Инструменты фронтенд-разработки
 - Он поддерживает стили кода и PSR1/PSR2, Symfony2, Zend, Drupal встроенные стили, и конечно же другие
 - Внедрение вместе с системой управления различными версиями, что бесспорно включает в себя и интерфейс унификации
 - Развертывание приложений удаленно, самозапускающаяся синхронизация с применением FTP, SFTP, FTPS и др.
 - Изменения, которые мы имеем возможность сразу же увидеть в браузере и без всякой перезагрузки страницы - это Live Edit.
 - PHP UML
 - Баг-трекеры вместе с интеграцией
 - SQL редактор, инструменты по работе с Б. Д.
- Кроссплатформенность (Windows, Mac OS X, Linux).

Глава 3 РЕАЛИЗАЦИЯ СИСТЕМЫ УЧЕТА МАТЕРИАЛОВ. ПРОВЕРКА РАБОТОСПОСОБНОСТИ СОЗДАННОЙ СИСТЕМЫ

3.1 Проектирование базы данных

Процесс проектирования баз данных имеет в себе важные этапы, одни из которых - логическое и физическое проектирование. Преобразование критериев к данным в самой структуре этой информации - это логическое проектирование. Получается в итоге структура, которая ориентирована на СУБД, то есть структура спецификации программ и Б. Д.. Именно на данном шаге способны образовываться Б. Д., по отношению к самым разным СУБД и осуществляться анализ различных моделей для сравнения. Как правило, логическая модель базы данных представляется в виде ER-модели данных, которая предназначена для описания объектов и отношений в базе данных. Сущность, идентификатор, атрибут, а также сама связь - это основные понятия модели ER.

Сущность (entity) является объектом, который выделяется (идентифицируется) человеком конкретно в предметной области. Это то, для чего пользователь хочет видеть, сохранять итоги своих наблюдений (информацию).

Списком имен в блоке сущности ассоциации изображаются атрибуты. При этом атрибуты должны вмещаться в отдельную строку. На диаграмме атрибуты, составляющие первичный ключ, располагаются в верхней части прямоугольника, и отделяющая от прочих (не входящих в первичных ключ) горизонтальной линией. Атрибут — любая характеристика сущности, значимая для рассматриваемой предметной области. Он предназначен для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности.

В результате моделирования базы данных была создана ER-диаграмма базы данных (рисунок 26).

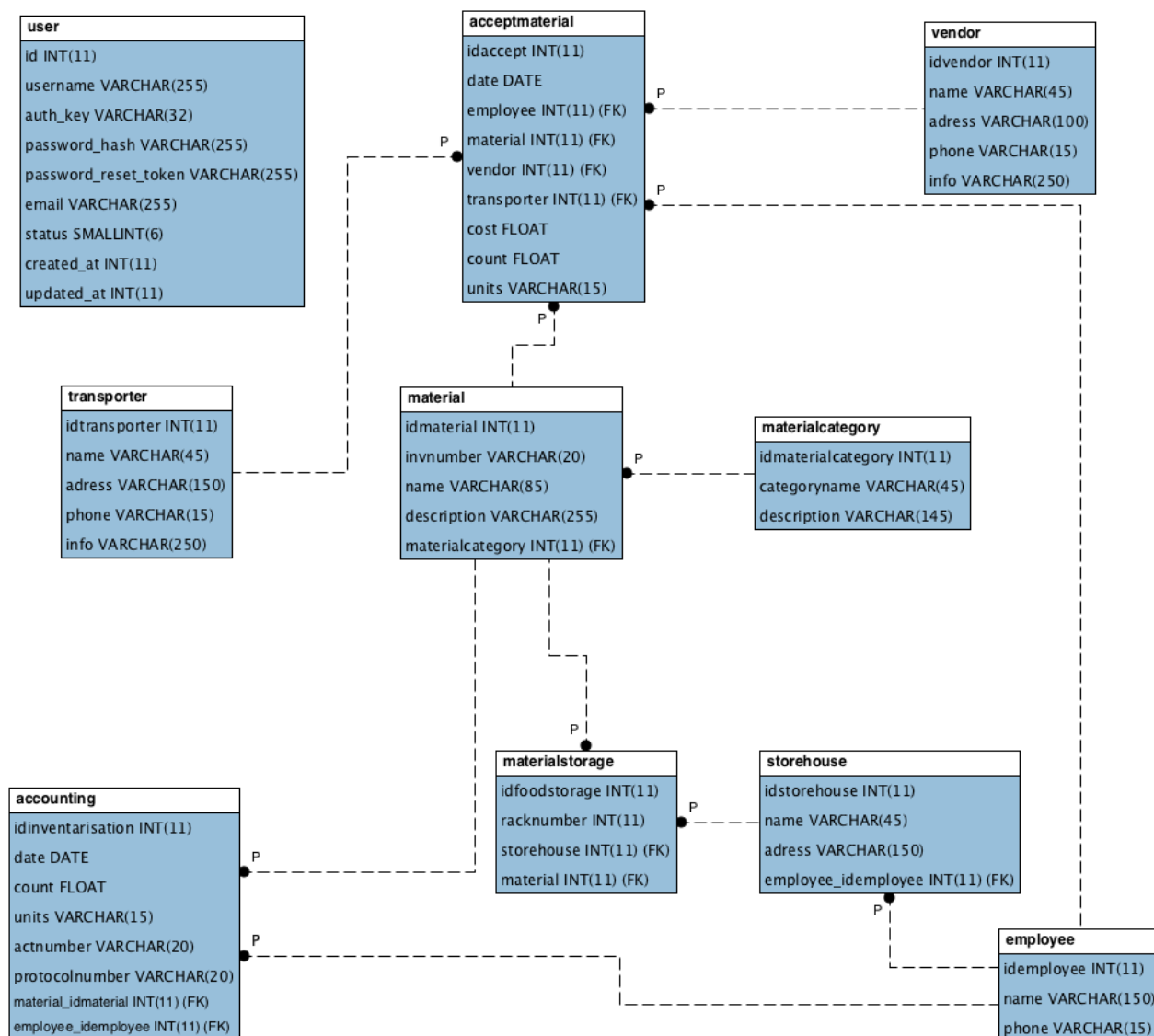


Рисунок 26 - Логическая модель базы данных

Перечень и описание сущностей представлен в таблице 6.

Определение особенностей характеристик по сохранению информации, методов к доступу является физическим проектированием, выполнение SQL-команд на сервере MySQL. Среда MySQL Workbench - это один из наилучших вариантов для физического проектирования Б. Д.. В результате моделирования базы данных был создан проект логической модели базы данных (рисунок 27).

Таблица 6 – Описание сущностей

Сущность	Описание
Приемка	Данные о приемках материалов
Категории МТО (materialcategory)	Категории материально-товарного обеспечения
МТО (material)	Список материально-товарных опеспечения
Места хранения (materialstorage)	Описание мест хранения МТО
Учет материалов (accounting)	Данные учета материалов
Склады (storehouse)	Информация о складах
Поставщики (vendor)	Список и информация о поставщиках
Перевозчики (transporter)	Список и информация о перевозчиках
Сотрудники (employee)	Список и информация о сотрудника
Пользователи (user)	Пользователи системы

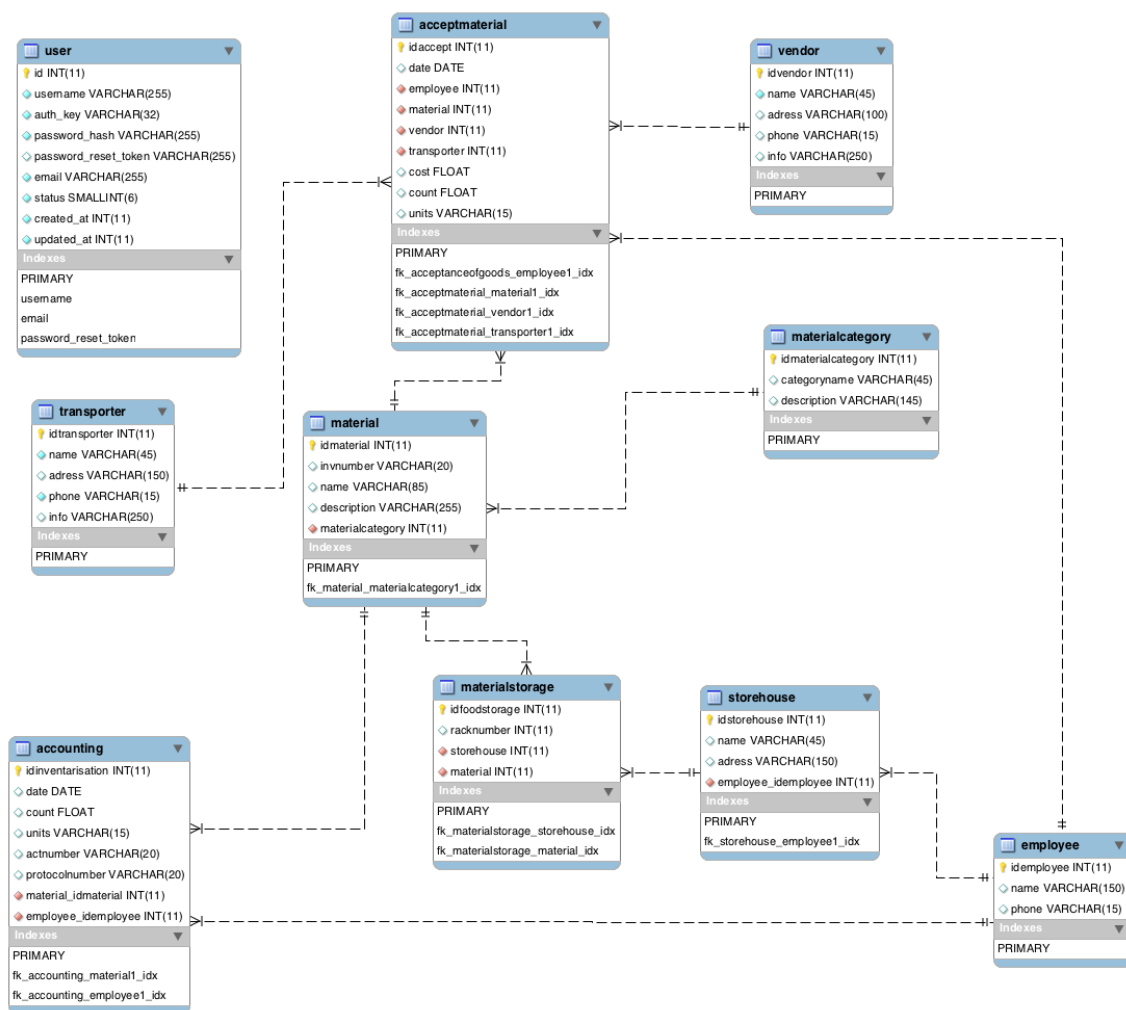


Рисунок 27 - Модель базы данных в среде MySQL Workbench

На основании представленной модели был создан SQL-скрипт для реализации физической модели базы данных на сервере MySQL:

```
CREATE TABLE IF NOT EXISTS accounting.employee (
    idemployee INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(150) NULL DEFAULT NULL,
    phone VARCHAR(15) NULL DEFAULT NULL,
    PRIMARY KEY (idemployee))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.materialcategory (
    idmaterialcategory INT(11) NOT NULL AUTO_INCREMENT,
    categoryname VARCHAR(45) NULL DEFAULT NULL,
    description VARCHAR(145) NULL DEFAULT NULL,
    PRIMARY KEY (idmaterialcategory))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.material (
    idmaterial INT(11) NOT NULL AUTO_INCREMENT,
    invnumber VARCHAR(20) NULL DEFAULT NULL,
    name VARCHAR(85) NULL DEFAULT NULL,
    description VARCHAR(255) NULL DEFAULT NULL,
    materialcategory INT(11) NOT NULL,
    PRIMARY KEY (idmaterial),
    INDEX fk_material_materialcategory1_idx (materialcategory
ASC),
    CONSTRAINT fk_material_materialcategory1
        FOREIGN KEY (materialcategory)
        REFERENCES accounting.materialcategory
(idmaterialcategory)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.transporter (
    idtransporter INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(45) NOT NULL,
    adress VARCHAR(150) NULL DEFAULT NULL,
    phone VARCHAR(15) NOT NULL,
    info VARCHAR(250) NULL DEFAULT NULL,
    PRIMARY KEY (idtransporter))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.vendor (
    idvendor INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(45) NOT NULL,
    adress VARCHAR(100) NULL DEFAULT NULL,
    phone VARCHAR(15) NULL DEFAULT NULL,
    info VARCHAR(250) NULL DEFAULT NULL,
```

```

    PRIMARY KEY (idvendor))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.acceptmaterial (
    idaccept INT(11) NOT NULL AUTO_INCREMENT,
    date DATE NULL DEFAULT NULL,
    employee INT(11) NOT NULL,
    material INT(11) NOT NULL,
    vendor INT(11) NOT NULL,
    transporter INT(11) NOT NULL,
    cost FLOAT NULL DEFAULT NULL,
    count FLOAT NULL DEFAULT NULL,
    units VARCHAR(15) NULL DEFAULT NULL,
    PRIMARY KEY (idaccept),
    INDEX fk_acceptanceofgoods_employee1_idx (employee ASC),
    INDEX fk_acceptmaterial_material1_idx (material ASC),
    INDEX fk_acceptmaterial_vendor1_idx (vendor ASC),
    INDEX fk_acceptmaterial_transporter1_idx (transporter ASC),
    CONSTRAINT fk_acceptanceofgoods_employee1
        FOREIGN KEY (employee)
        REFERENCES accounting.employee (idemployee)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_acceptmaterial_material1
        FOREIGN KEY (material)
        REFERENCES accounting.material (idmaterial)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_acceptmaterial_transporter1
        FOREIGN KEY (transporter)
        REFERENCES accounting.transporter (idtransporter)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT fk_acceptmaterial_vendor1
        FOREIGN KEY (vendor)
        REFERENCES accounting.vendor (idvendor)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.accounting (
    idinventarisation INT(11) NOT NULL AUTO_INCREMENT,
    date DATE NULL DEFAULT NULL,
    count FLOAT NULL DEFAULT NULL,
    units VARCHAR(15) NULL DEFAULT NULL,
    actnumber VARCHAR(20) NULL DEFAULT NULL,
    protocolnumber VARCHAR(20) NULL DEFAULT NULL,
    material_idmaterial INT(11) NOT NULL,
    employee_idemployee INT(11) NOT NULL,

```

```

PRIMARY KEY (idinventarisatation),
INDEX fk_accounting_material1_idx (material_idmaterial ASC),
INDEX fk_accounting_employee1_idx (employee_idemployee ASC),
CONSTRAINT fk_accounting_material1
    FOREIGN KEY (material_idmaterial)
    REFERENCES accounting.material (idmaterial)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT fk_accounting_employee1
    FOREIGN KEY (employee_idemployee)
    REFERENCES accounting.employee (idemployee)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.storehouse (
    idstorehouse INT(11) NOT NULL AUTO_INCREMENT,
    name VARCHAR(45) NULL DEFAULT NULL,
    adress VARCHAR(150) NULL DEFAULT NULL,
    employee_idemployee INT(11) NOT NULL,
    PRIMARY KEY (idstorehouse),
    INDEX fk_storehouse_employee1_idx (employee_idemployee ASC),
    CONSTRAINT fk_storehouse_employee1
        FOREIGN KEY (employee_idemployee)
        REFERENCES accounting.employee (idemployee)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.materialstorage (
    idfoodstorage INT(11) NOT NULL AUTO_INCREMENT,
    racknumber INT(11) NULL DEFAULT NULL,
    storehouse INT(11) NOT NULL,
    material INT(11) NOT NULL,
    PRIMARY KEY (idfoodstorage),
    INDEX fk_materialstorage_storehouse_idx (storehouse ASC),
    INDEX fk_materialstorage_material_idx (material ASC),
    CONSTRAINT fk_materialstorage_material
        FOREIGN KEY (material)
        REFERENCES accounting.material (idmaterial),
    CONSTRAINT fk_materialstorage_storehouse
        FOREIGN KEY (storehouse)
        REFERENCES accounting.storehouse (idstorehouse))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
CREATE TABLE IF NOT EXISTS accounting.user (
    id INT(11) NOT NULL AUTO_INCREMENT,
    username VARCHAR(255) CHARACTER SET 'utf8mb4' COLLATE
'utf8mb4_unicode_ci' NOT NULL,

```

```

        auth_key VARCHAR(32) CHARACTER SET 'utf8mb4' COLLATE
'utf8mb4_unicode_ci' NOT NULL,
        password_hash VARCHAR(255) CHARACTER SET 'utf8mb4' COLLATE
'utf8mb4_unicode_ci' NOT NULL,
        password_reset_token VARCHAR(255) CHARACTER SET 'utf8mb4'
COLLATE 'utf8mb4_unicode_ci' NULL DEFAULT NULL,
        email VARCHAR(255) CHARACTER SET 'utf8mb4' COLLATE
'utf8mb4_unicode_ci' NOT NULL,
        status SMALLINT(6) NOT NULL DEFAULT '10',
        created_at INT(11) NOT NULL,
        updated_at INT(11) NOT NULL,
        PRIMARY KEY (id),
        UNIQUE INDEX username (username ASC),
        UNIQUE INDEX email (email ASC),
        UNIQUE INDEX password_reset_token (password_reset_token
ASC) )
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_unicode_ci;

```

После создания SQL-скрипта на создание базы данных он был выполнен в консоли PHPMyAdmin (рис. 28).

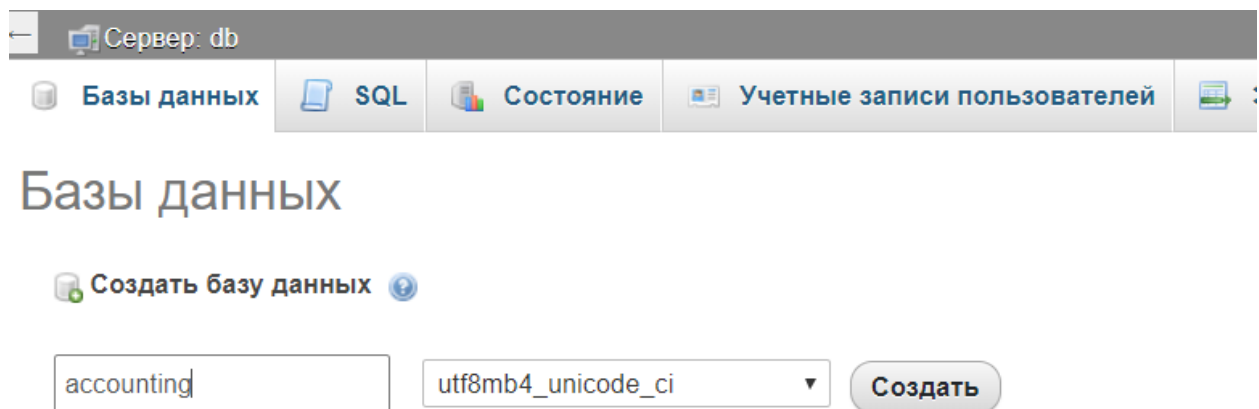


Рисунок 28 - Создание базы данных

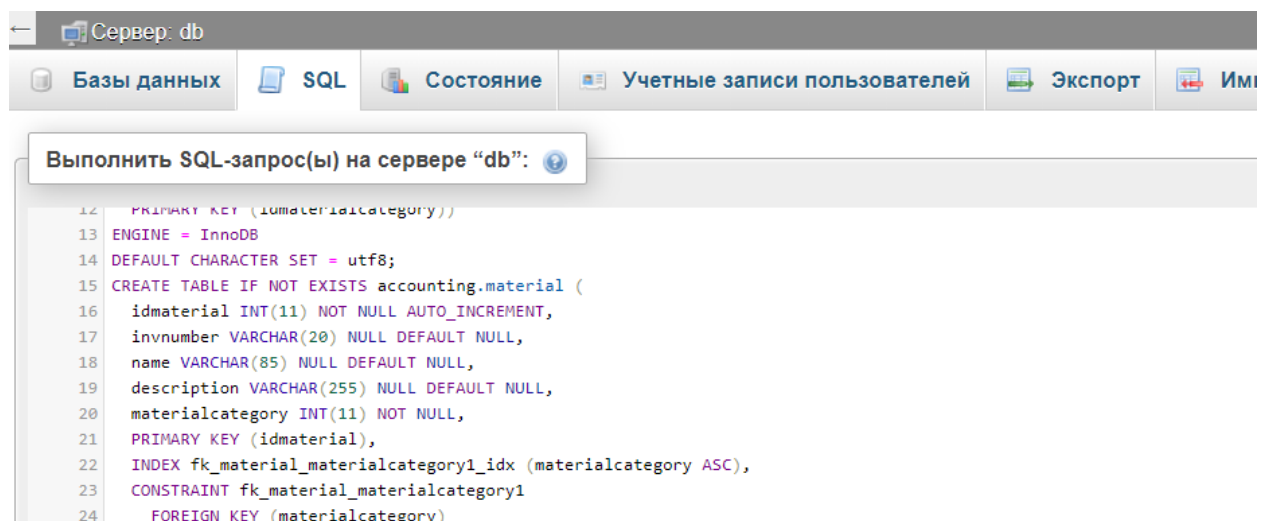


Рисунок 29 - SQL-консоль. Выполнение SQL-скрипта

3.2 Проектирование структуры и функционала модуля учета материальных ценностей

В разрабатываемой системе будет использована иерархическая структура (рисунок 31). Такое представление информации хорошо подходит для проработанной, систематизированной и организованной информации. Перемещение по уровням такой структуры осуществляется в направлении от общего к частному.

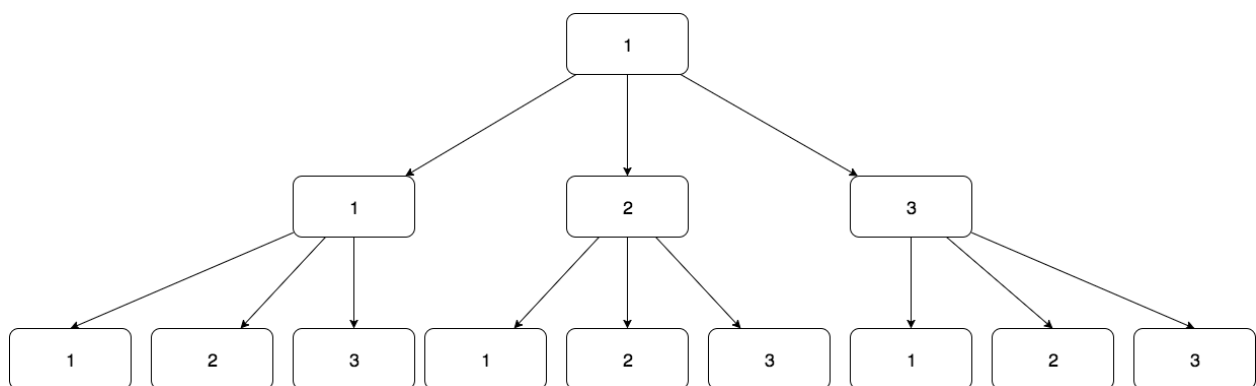


Рисунок 31 - Иерархическая структура приложения

Такая информационная архитектура подходит для системы учета материалов и позволит легко воспринимать пользователям системы, информацию, которая представлена на нем, а также осуществлять операции по управлению информацией системы (рисунок 32).

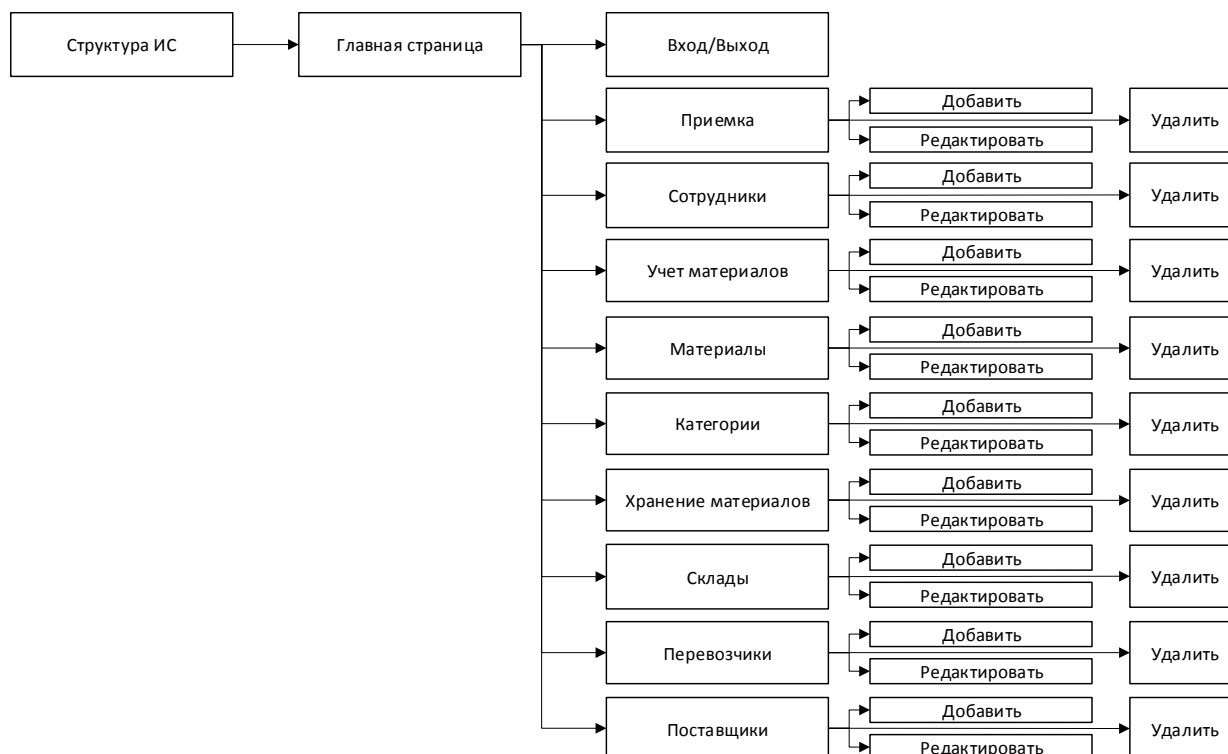


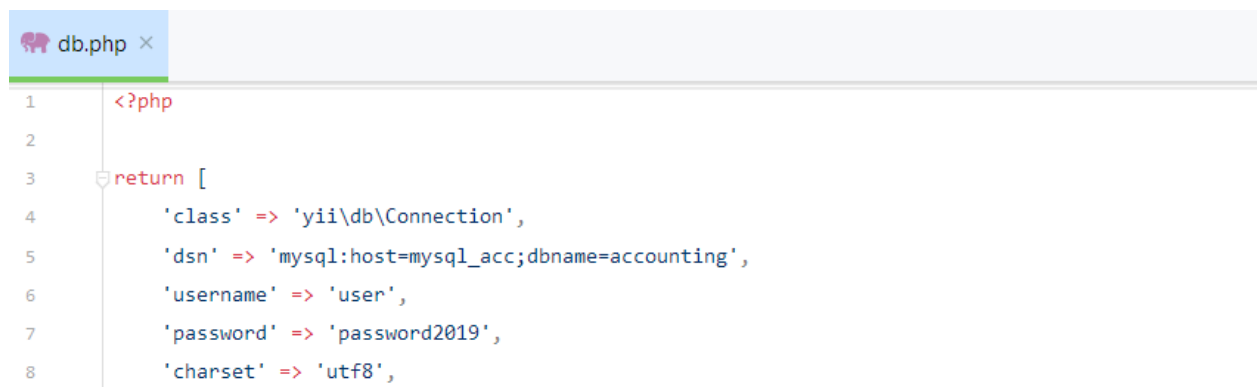
Рисунок 32 - Схематическое изображение структуры ИС

Таким образом, разработанная структура приложения отображает иерархию структурных элементов системы и будет использована в дальнейших этапах разработки проекта информационной системы автоматизации системы учета материалов.

3.3 Реализация программного модуля

После процесса моделирования были разработаны рабочие окна для отображения данных, а также страницы с формами ввода и редактирования данных приложения. Процесс реализации модуля показан на рис. 33 - 38.

Для подключения к базе данных необходимо сохранить параметры подключения к базе данных (рис. 33).



```
1  <?php
2
3  return [
4      'class' => 'yii\db\Connection',
5      'dsn' => 'mysql:host=mysql_acc;dbname=accounting',
6      'username' => 'user',
7      'password' => 'password2019',
8      'charset' => 'utf8',
```

Рисунок 33 - Определение параметров подключения к базе данных

Для отображения ошибок в исходном коде необходимо включить панель отладки и разрешить подключение с разных IP (только в режиме разработки) (рис. 34).

```

73 if (YII_ENV_DEV) {
74     // configuration adjustments for 'dev' environment
75     $config['bootstrap'][] = 'debug';
76     $config['modules']['debug'] = [
77         'class' => 'yii\debug\Module',
78         // uncomment the following to add your IP if you are not connecting from localhost.
79         //'allowedIPs' => ['127.0.0.1', '::1'],
80         'allowedIPs' => ['*']
81     ];
82
83
84     $config['bootstrap'][] = 'gii';
85     $config['modules']['gii'] = [
86         'class' => 'yii\gii\Module',
87         // uncomment the following to add your IP if you are not connecting from localhost.
88         //'allowedIPs' => ['127.0.0.1', '::1'],
89         'allowedIPs' => ['*']
90     ];
91 }

```

Рисунок 34 - Настройка окружения разработки

После подключения к базе данных и настройки окружения разработки был осуществлен процесс разработки моделей, контроллеров и представлений для ввода, вывода и редактирования информации.

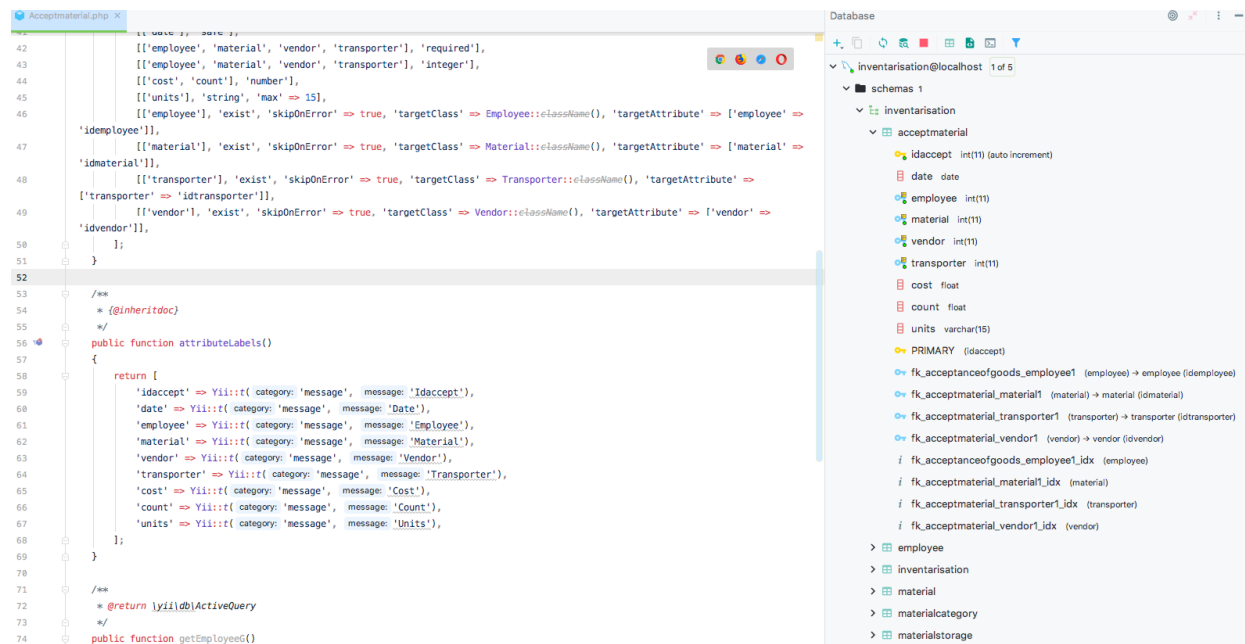


Рисунок 35 - Работа над моделью «Приемка»

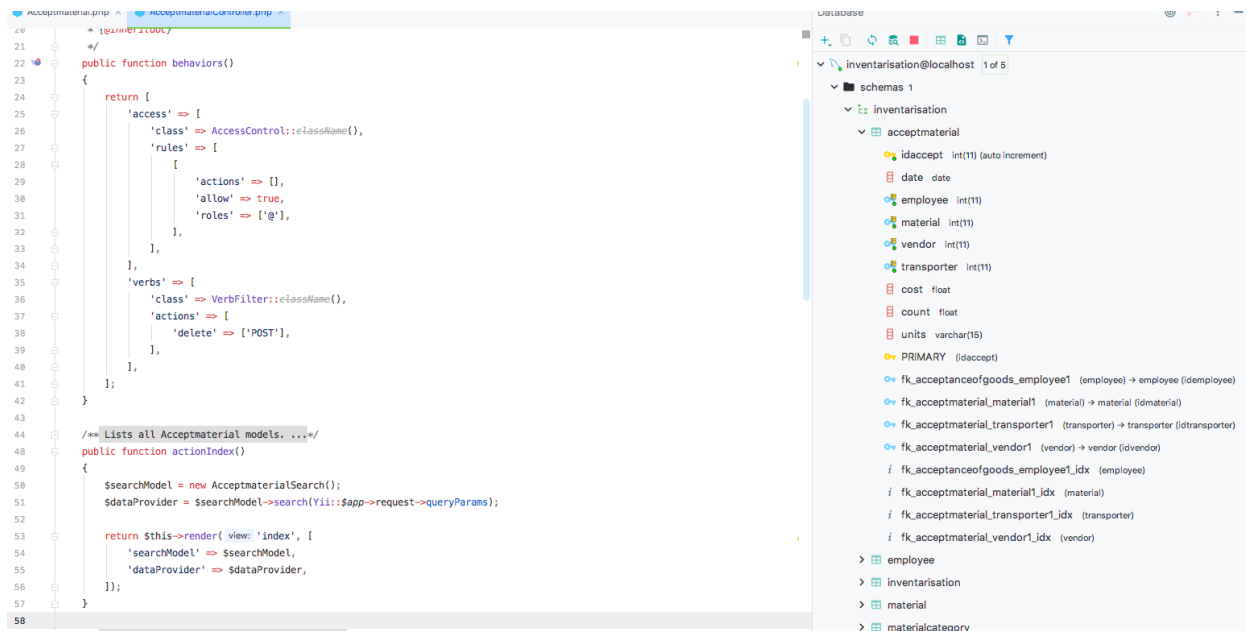


Рисунок 36 - Работа над контроллером «Приемка»

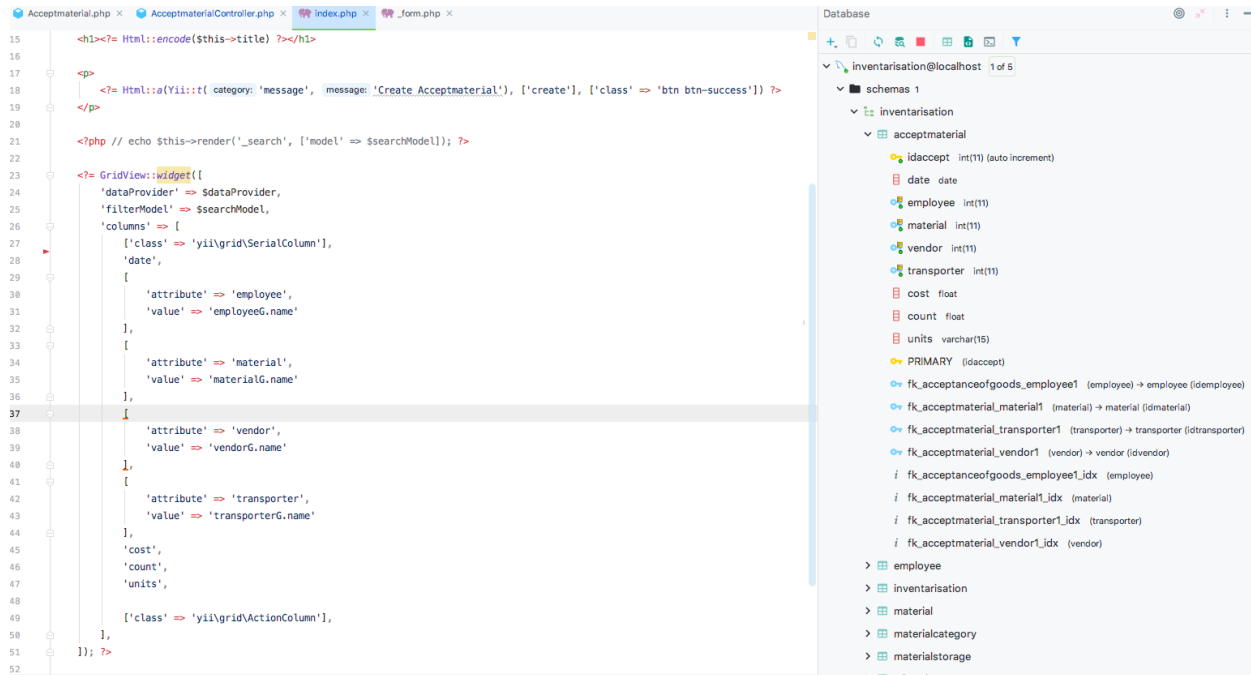


Рисунок 37 - Работа над представлением «Приемка»

```

16 <div class="acceptmaterial-form">
17 <?php $form = ActiveForm::begin(); ?>
18 <?php
19 echo $form->field($model, attribute: 'date')->widget(DateTimePicker::className(), [
20     'options' => ['placeholder' => 'Выберите дату ...'],
21     'pluginOptions' => [
22         'autoclose' => true
23     ]
24 ]);
25 ?>
26 <?=$form->field($model, attribute: 'employee')->dropDownList(ArrayHelper::map(Employee::find()->all(), from: 'idemployee', to: 'name')) ?>
27 <?=$form->field($model, attribute: 'material')->dropDownList(ArrayHelper::map(Material::find()->all(), from: 'idmaterial', to: 'invnumber')) ?>
28 <?=$form->field($model, attribute: 'vendor')->dropDownList(ArrayHelper::map(Vendor::find()->all(), from: 'idvendor', to: 'name')) ?>
29 <?=$form->field($model, attribute: 'transporter')->dropDownList(ArrayHelper::map(Transporter::find()->all(), from: 'idtransporter', to: 'name')) ?>
30 <?=$form->field($model, attribute: 'cost')->textInput() ?>
31 <?=$form->field($model, attribute: 'count')->textInput() ?>
32 <?php
33 $items = [
34     'Литры' => 'Литры',
35     'Килограммы' => 'Килограммы',
36     'Тонны' => 'Тонны',
37     'Грамы' => 'Грамы'
38 ];
39 $params = [
40     'prompt' => 'Выберите единицу измерения...'
41 ];
42 echo $form->field($model, attribute: 'units')->dropDownList($items, $params);
43 ?>
44 <div class="form-group">
45 <?=$form->submitButton(Yii::t('category: message', message: 'Save'), ['class' => 'btn btn-success']) ?>
46 </div>
47 <?php ActiveForm::end(); ?>
48 </div>

```

Рисунок 38 - Форма добавления записи «Приемка»

Следующим этапом было создание подмодуля «Сотрудники» (рис. 39 – 41)

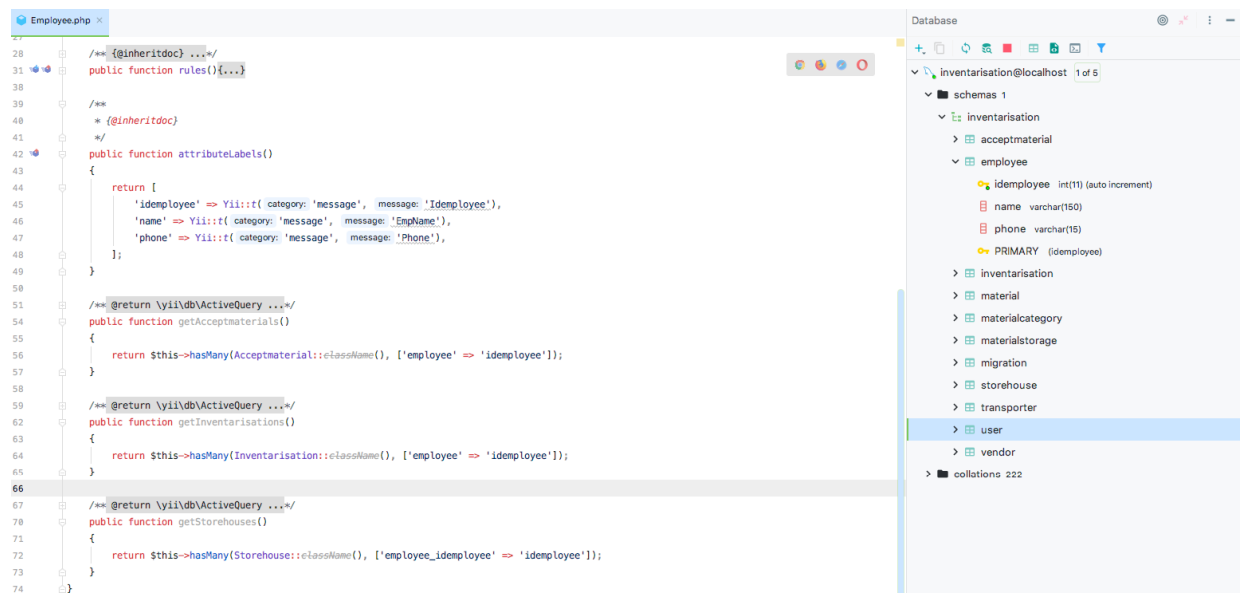


Рисунок 39 - Работа над моделью «Сотрудники»



Рисунок 40 - Работа над index-представлением «Сотрудники»



Рисунок 41- Работа над формой добавления сотрудника

Далее было проведено процесс создания подмодуля «Учет материалов» (рис. 42 – 44)

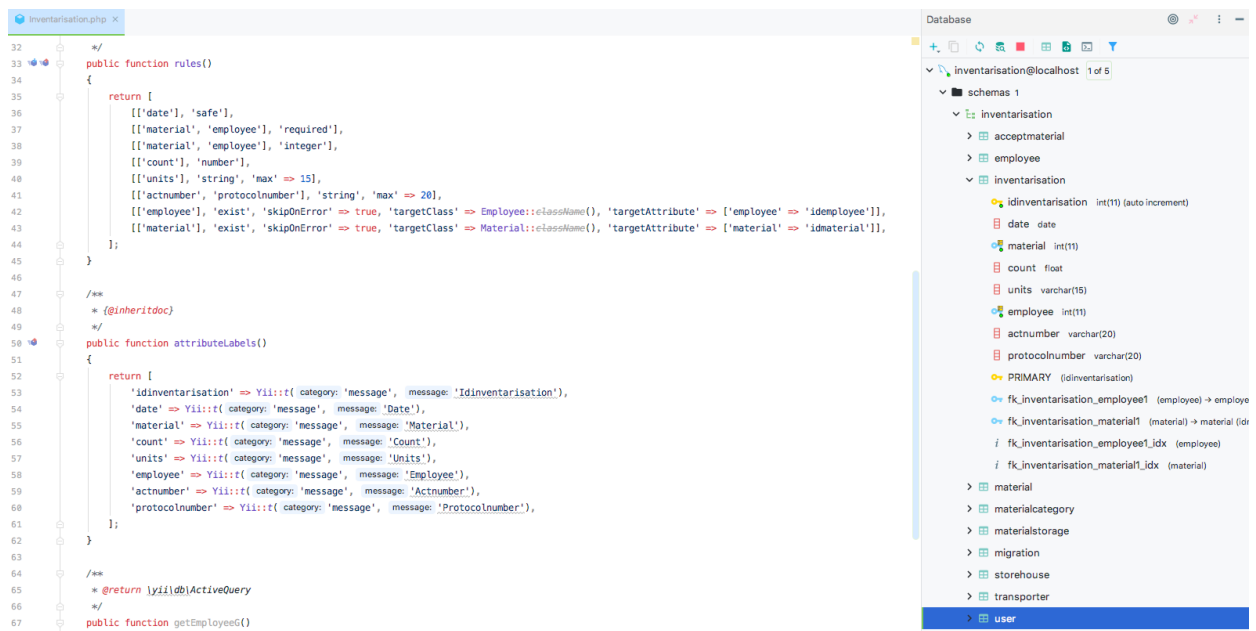


Рисунок 42 - Работа над моделью «Учет материалов»



Рисунок 43 - Работа над index-представлением «Учет материалов»

```

15 <?php $form = ActiveForm::begin(); ?>
16 <?php
17 echo $form->field($model, attribute: 'date')->widget(DateTimePicker::className(), [
18     'options' => ['placeholder' => 'Выберите дату ...'],
19     'pluginOptions' => [
20         'autoclose' => true
21     ]
22 ]);
23 ?>
24 <?= $form->field($model, attribute: 'material')->dropDownList(ArrayHelper::map(\app\models\Material::find()->all(), from: 'idmaterial', to: 'invnumber')) ?>
25 <?= $form->field($model, attribute: 'count')->textInput() ?>
26 <?php
27 $items = [...];
28 $params = [
29     'prompt' => 'Выберите единицу измерения...'
30 ];
31 echo $form->field($model, attribute: 'units')->dropDownList($items, $params);
32 ?>
33 <?= $form->field($model, attribute: 'employee')->dropDownList(ArrayHelper::map(\app\models\Employee::find()->all(), from: 'idemployee', to: 'name')) ?>
34 <?= $form->field($model, attribute: 'actnumber')->textInput(['maxlength' => true]) ?>
35 <?= $form->field($model, attribute: 'protocolnumber')->textInput(['maxlength' => true]) ?>
36 <div class="form-group">
37     <?= Html::submitButton(Yii::t('category: 'message', message: 'Save'), ['class' => 'btn btn-success']) ?>
38 </div>
39 <?php ActiveForm::end(); ?>

```

Рисунок 44 - Работа над формой добавления записи в подмодуль «Учет материалов»

Создание подмодуля «Материалы» (МТО) показано на рис. 45 – 47.

The image shows a development environment with two main components:

- Left Panel (Code Editor):** Displays the PHP code for the `Material` model. The code includes:
 - A `rules()` method defining validation rules for `materialcategory`, `invnumber`, `name`, `description`, and `materialcategory`.
 - An `attributeLabels()` method returning labels for the same attributes.
 - Comments indicating the model is part of the `Material` module.
- Right Panel (Database Diagram):** Shows a database schema for the `inventarisat` database. The `material` table is highlighted, showing its structure:
 - `idmaterial`: `int(11)` (auto increment), PRIMARY key.
 - `invnumber`: `varchar(20)`.
 - `name`: `varchar(85)`.
 - `description`: `varchar(255)`.
 - `materialcategory`: `int(11)`, FOREIGN KEY to `materialcategory` table.

Рисунок 45 - Работа над моделью «Материалы»

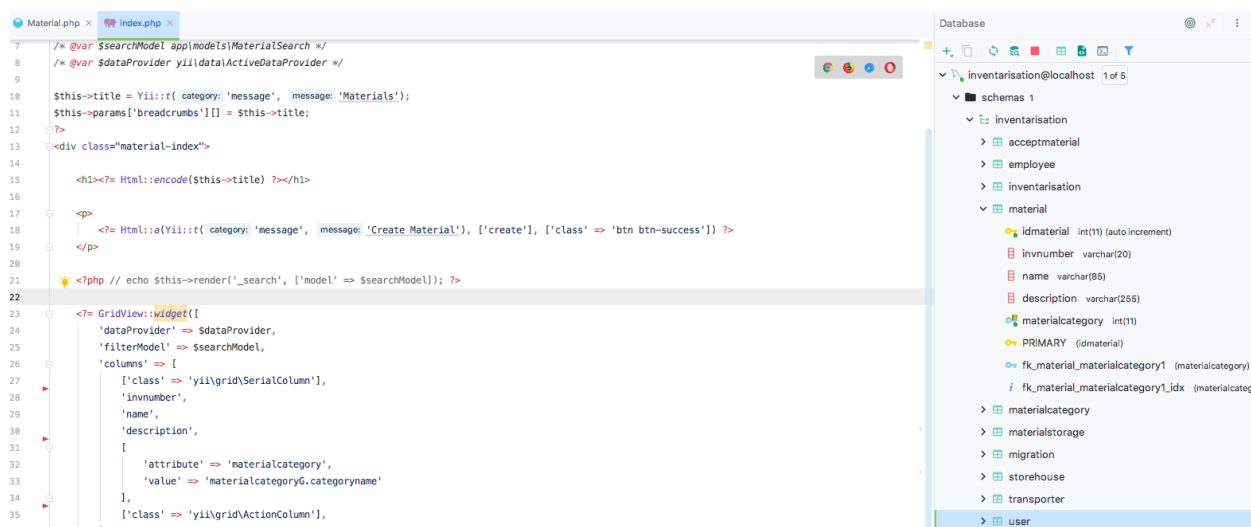


Рисунок 46 - Работа над index-представлением «Материалы»



Рисунок 47 - Работа над формой добавления материалов

Процесс разработки подмодуля управлением категориями материалов показан 48.



Рисунок 48 - Работа над моделью «Категории материалов»

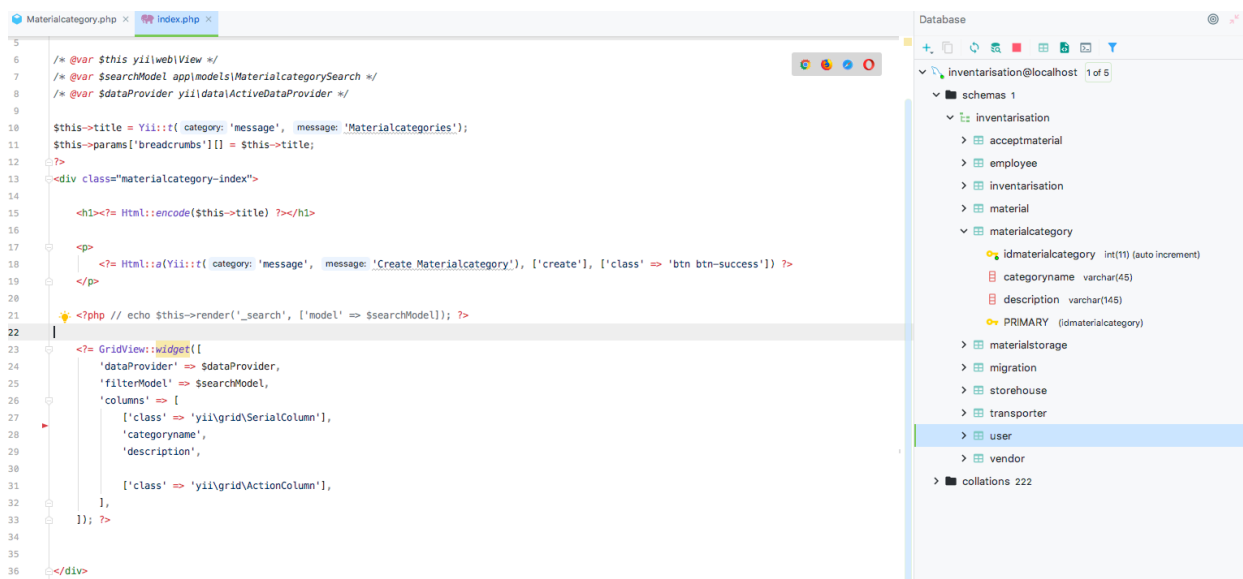


Рисунок 49 - Работа над index-представлением «Категории материалов»

```

13     <?php $form = ActiveForm::begin(); ?>
14
15     <?= $form->field($model, attribute: 'categoryname')->textInput(['maxlength' => true]) ?>
16
17     <?= $form->field($model, attribute: 'description')->textInput(['maxlength' => true]) ?>
18
19     <div class="form-group">
20         <?= Html::submitButton(Yii::t('category: 'message', 'message: 'Save'), ['class' => 'btn btn-success']) ?>
21     </div>
22
23     <?php ActiveForm::end(); ?>

```

Рисунок 50 - Работа над формой добавления категорий материалов

Создание подмодуля «Хранение материалов» показано на рис. 51 – 53.

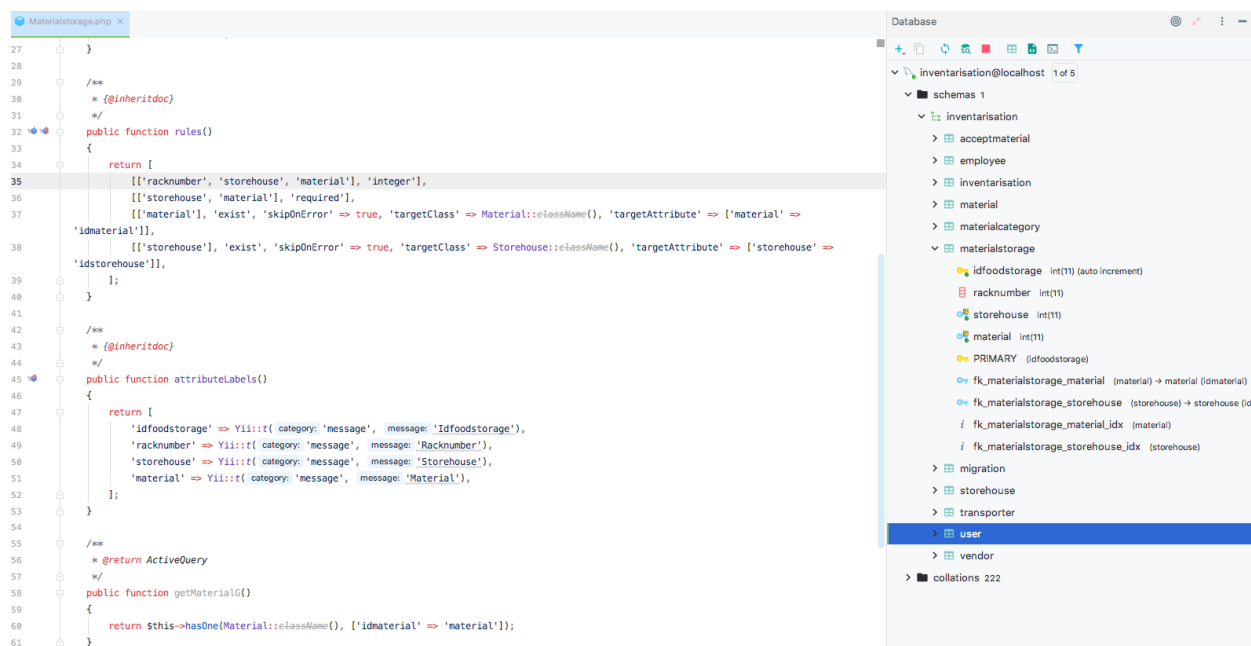


Рисунок 51 - Работа над моделью «Хранение материалов»

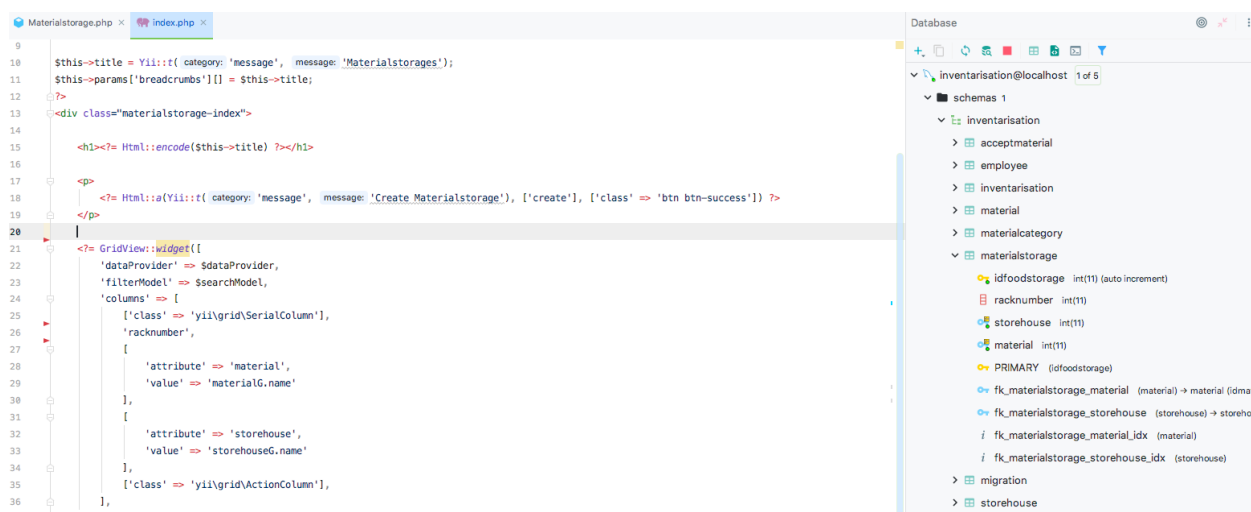


Рисунок 52 - Работа над index-представлением «Хранение материалов»



Рисунок 53 - Работа над формой добавления записи о хранении материала

После этого был создан подмодуль «Склады», который предназначен для управления списком складов (рис. 54 –56).

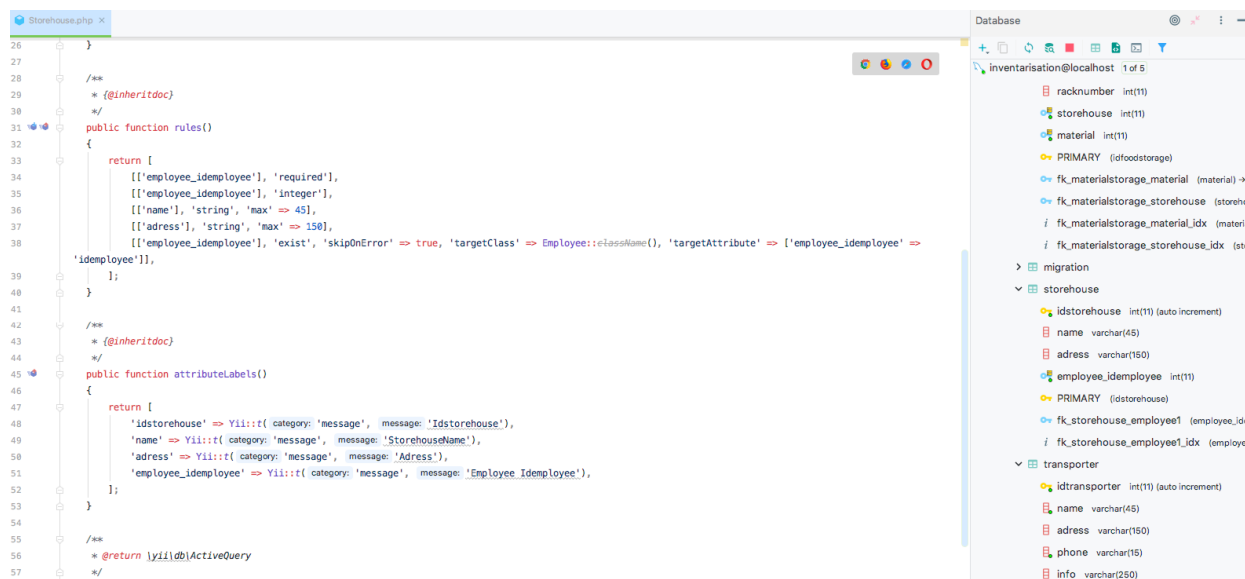


Рисунок 54 - Работа над моделью «Склады»

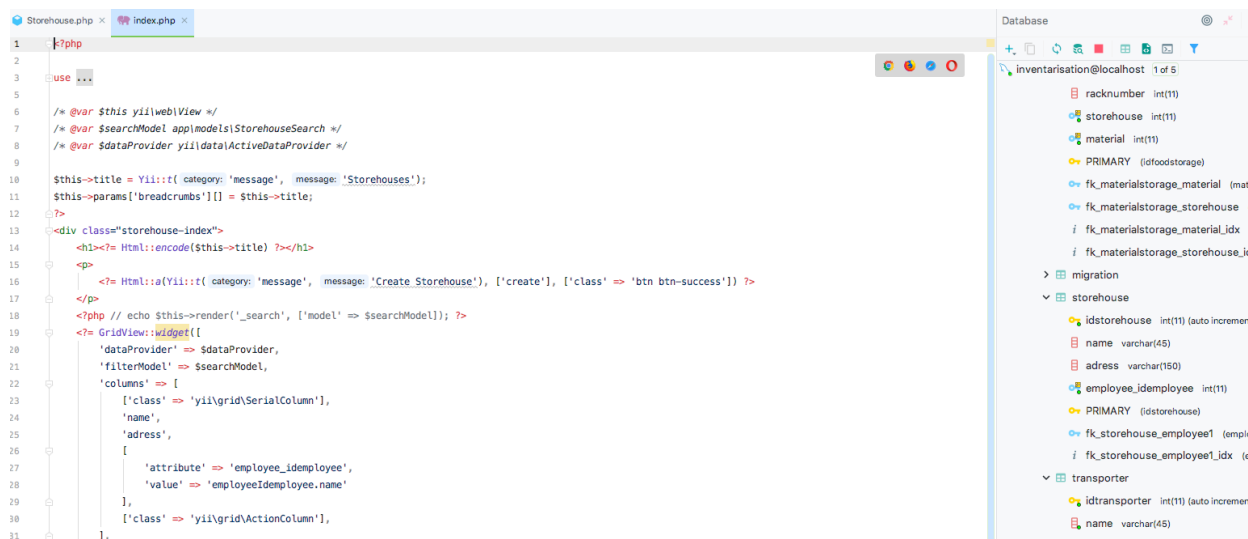


Рисунок 55 - Работа над index-представлением «Склады»

```

14 <?php $form = ActiveForm::begin(); ?>
15
16 <?= $form->field($model, attribute: 'name')->textInput(['maxlength' => true]) ?>
17
18 <?= $form->field($model, attribute: 'adress')->textInput(['maxlength' => true]) ?>
19
20 <?= $form->field($model, attribute: 'employee_idemployee')->dropDownList(ArrayHelper::map($app\models\Employee::find()->all(), from: 'idemployee', to: 'name')) ?>
21
22
23 <div class="form-group">
24     <?= Html::submitButton(Yii::t('category: 'message', message: 'Save'), ['class' => 'btn btn-success']) ?>
25 </div>
26
27 <?php ActiveForm::end(); ?>

```

Рисунок 56 - Работа над формой добавления складов

Процесс создания подмодуля «Перевозчики» показан на рис. 57 – 59.

The image shows a code editor with the file `Transporter.php` open. The code defines a model for the 'transporter' entity, including rules and attribute labels. The database schema on the right shows the 'storehouse' and 'transporter' tables, with 'transporter' having a primary key 'idtransporter' and a foreign key 'fk_storehouse_employee1'.

```

26 }
27
28 /**
29  * {@inheritdoc}
30  */
31 public function rules()
32 {
33     return [
34         [['name', 'phone'], 'required'],
35         [['name'], 'string', 'max' => 45],
36         [['address'], 'string', 'max' => 150],
37         [['phone'], 'string', 'max' => 15],
38         [['info'], 'string', 'max' => 250],
39     ];
40 }
41
42 /**
43  * {@inheritdoc}
44  */
45 public function attributeLabels()
46 {
47     return [
48         'idtransporter' => Yii::t('category: 'message', message: 'Idtransporter'),
49         'name' => Yii::t('category: 'message', message: 'Name'),
50         'address' => Yii::t('category: 'message', message: 'Address'),
51         'phone' => Yii::t('category: 'message', message: 'Phone'),
52         'info' => Yii::t('category: 'message', message: 'Info'),
53     ];
54 }
55
56 /**
57  * @return \yii\db\ActiveQuery
58  */
59 public function getAcceptmaterials()
60 {
61     return $this->hasMany(Acceptmaterial::class, ['transporter' => 'idtransporter']);
62 }

```

Database schema details:

- storehouse**
 - idstorehouse: int(11) (auto increment)
 - name: varchar(45)
 - address: varchar(150)
 - employee_idemployee: int(11)
 - PRIMARY (idstorehouse)
 - fk_storehouse_employee1 (employee_idemployee)
- transporter**
 - idtransporter: int(11) (auto increment)
 - name: varchar(45)
 - address: varchar(150)
 - phone: varchar(15)
 - info: varchar(250)
 - PRIMARY (idtransporter)
- user**
 - id: int(11) (auto increment)
 - username: varchar(255)
 - auth_key: varchar(32)
 - password_hash: varchar(255)
 - password_reset_token: varchar(255)
 - email: varchar(255)
 - status: smallint(6) = 10
 - created_at: int(11)
 - updated_at: int(11)
 - PRIMARY (id)

Рисунок 57 - Работа над моделью «Перевозчики»

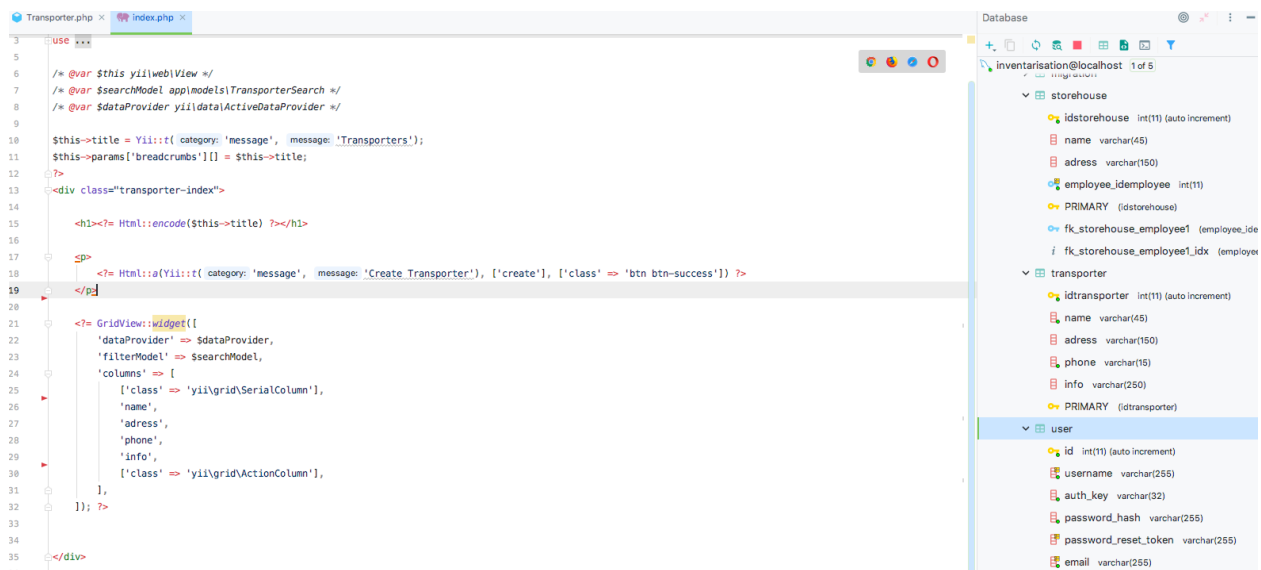


Рисунок 58 - Работа над index-представлением «Перевозчики»

```

13      <?php $form = ActiveForm::begin(); ?>
14
15      <?= $form->field($model, attribute: 'name')->textInput(['maxlength' => true]) ?>
16
17      <?= $form->field($model, attribute: 'adress')->textInput(['maxlength' => true]) ?>
18
19      <?= $form->field($model, attribute: 'phone')->textInput(['maxlength' => true]) ?>
20
21      <?= $form->field($model, attribute: 'info')->textInput(['maxlength' => true]) ?>
22
23      <div class="form-group">
24          <?= Html::submitButton(Yii::t('category: 'message', message: 'Save'), ['class' => 'btn btn-success']) ?>
25      </div>
26
27      <?php ActiveForm::end(); ?>

```

Рисунок 59 - Работа над формой добавления перевозчиков

Последним шагом в создании системы было создание подмодуля «Поставщики», который показан на рис. 60 – 62.

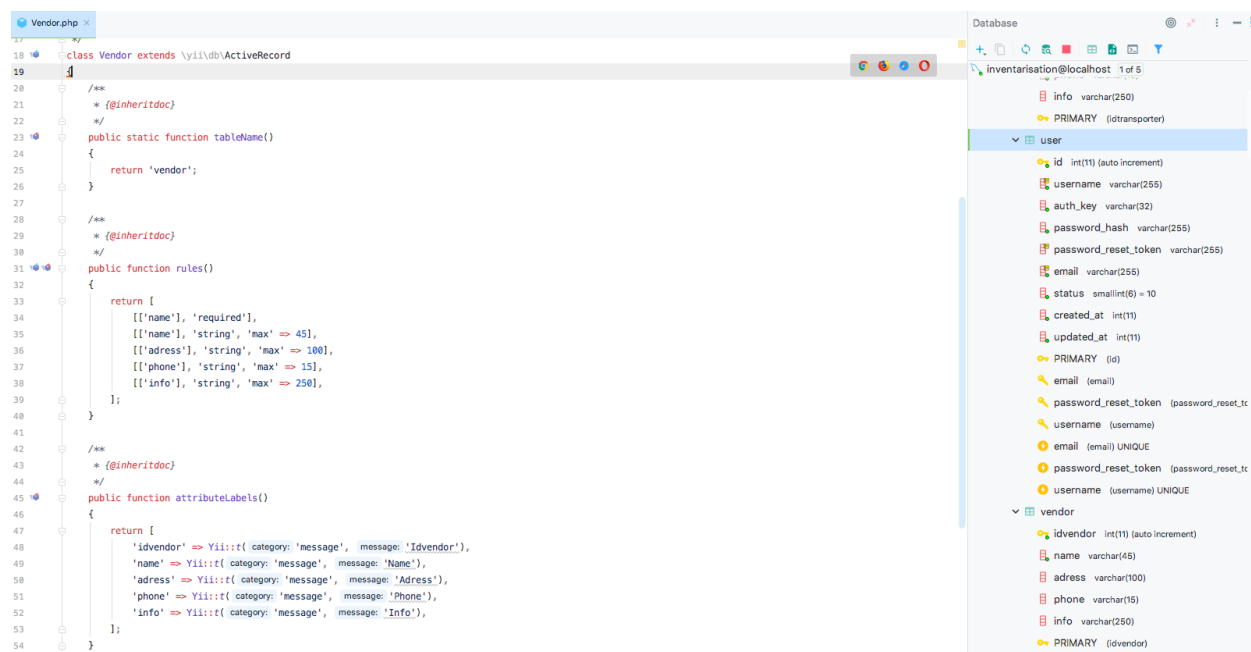


Рисунок 60 - Работа над моделью «Поставщики»

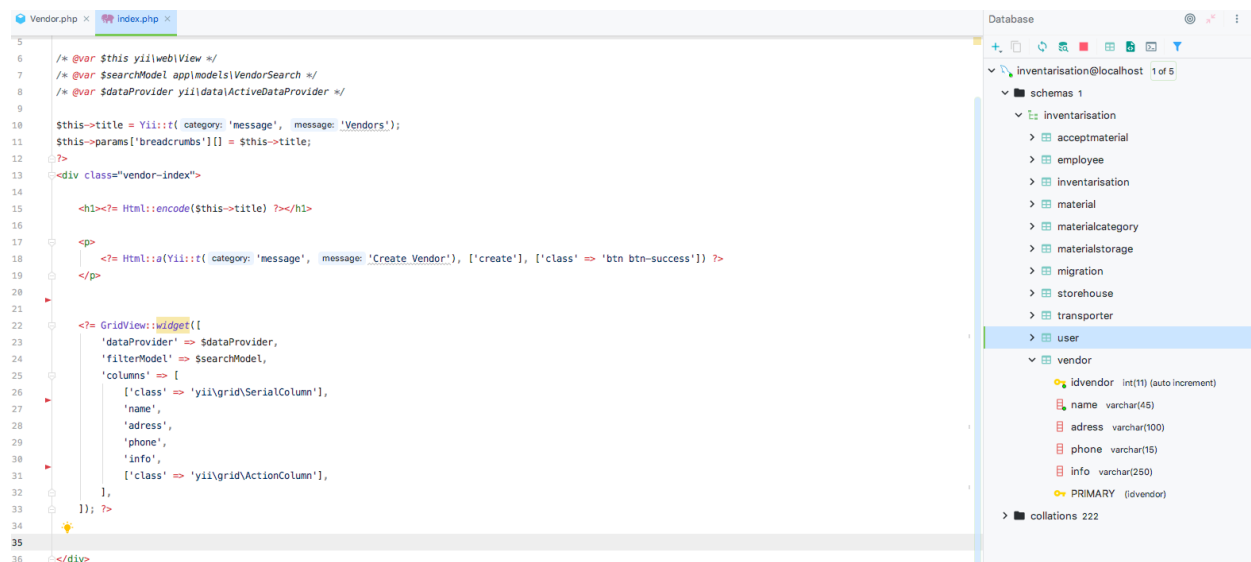


Рисунок 61 - Работа над index-представлением «Поставщики»

```

13 <?php $form = ActiveForm::begin(); ?>
14
15 <?= $form->field($model, attribute: 'name')->textInput(['maxlength' => true]) ?>
16
17 <?= $form->field($model, attribute: 'adress')->textInput(['maxlength' => true]) ?>
18
19 <?= $form->field($model, attribute: 'phone')->textInput(['maxlength' => true]) ?>
20
21 <?= $form->field($model, attribute: 'info')->textInput(['maxlength' => true]) ?>
22
23 <div class="form-group">
24     <?= Html::submitButton(Yii::t('category: 'message', message: 'Save!'), ['class' => 'btn btn-success']) ?>
25 </div>
26
27 <?php ActiveForm::end(); ?>

```

Рисунок 62 - Работа над формой добавления поставщиков

После этого была создана миграция для создания таблицы с пользователями (рис. 63).

```

m200101_234826_create_user_table.php
1 <?php
2
3 use yii\db\Migration;
4
5 /**
6  * Handles the creation of table '{{%user}}'.
7  */
8 class m200101_234826_create_user_table extends Migration
9 {
10     /**
11      * {@inheritdoc}
12      */
13     public function safeUp()
14     {
15         $this->createTable( table: '{{%user}}', [
16             'id' => $this->primaryKey(),
17             'username' => $this->string()->notNull()->unique(),
18             'auth_key' => $this->string( length: 32)->notNull(),
19             'password_hash' => $this->string()->notNull(),
20             'password_reset_token' => $this->string()->unique(),
21             'email' => $this->string()->notNull()->unique(),
22             'status' => $this->smallInteger()->notNull()->defaultValue( default: 10),
23             'created_at' => $this->integer()->notNull(),
24             'updated_at' => $this->integer()->notNull()
25         ]);
26     }
27
28     /**
29      * {@inheritdoc}
30      */
31     public function safeDown()
32     {
33         $this->dropTable( table: '{{%user}}' );
34     }
35 }

```

Рисунок 63 - Миграция для создания таблицы «User»

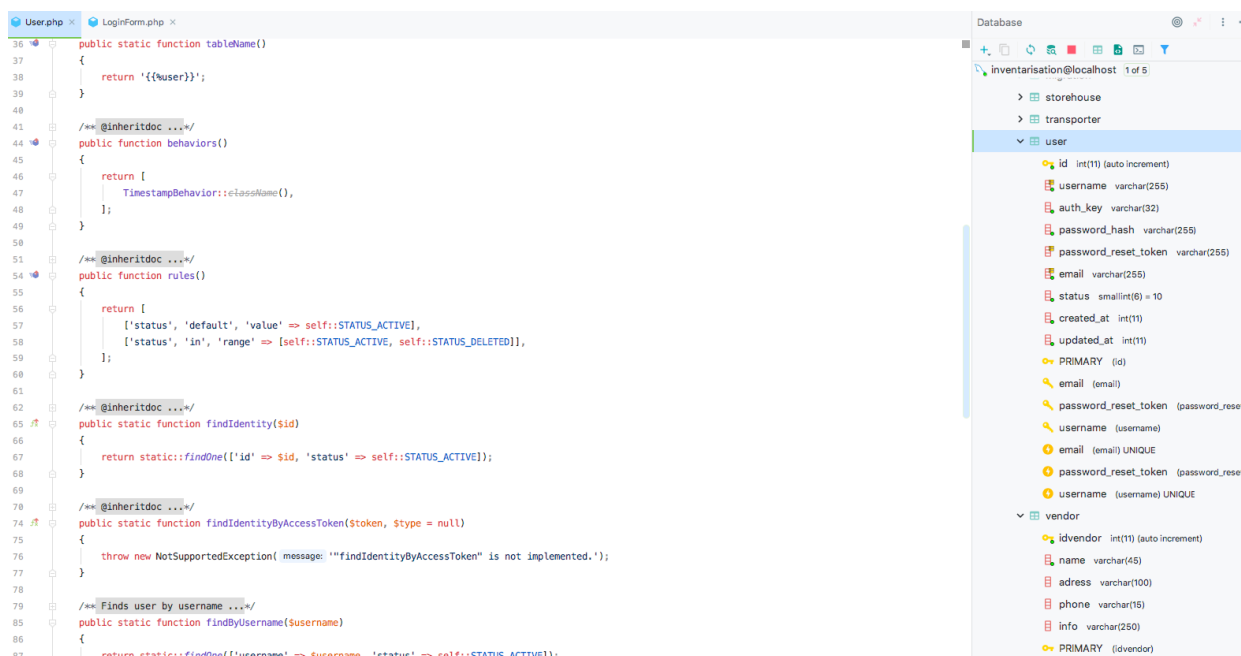


Рисунок 64 - Работа над моделью «User»

После этого была создана модель для авторизации в системе (рис. 65).

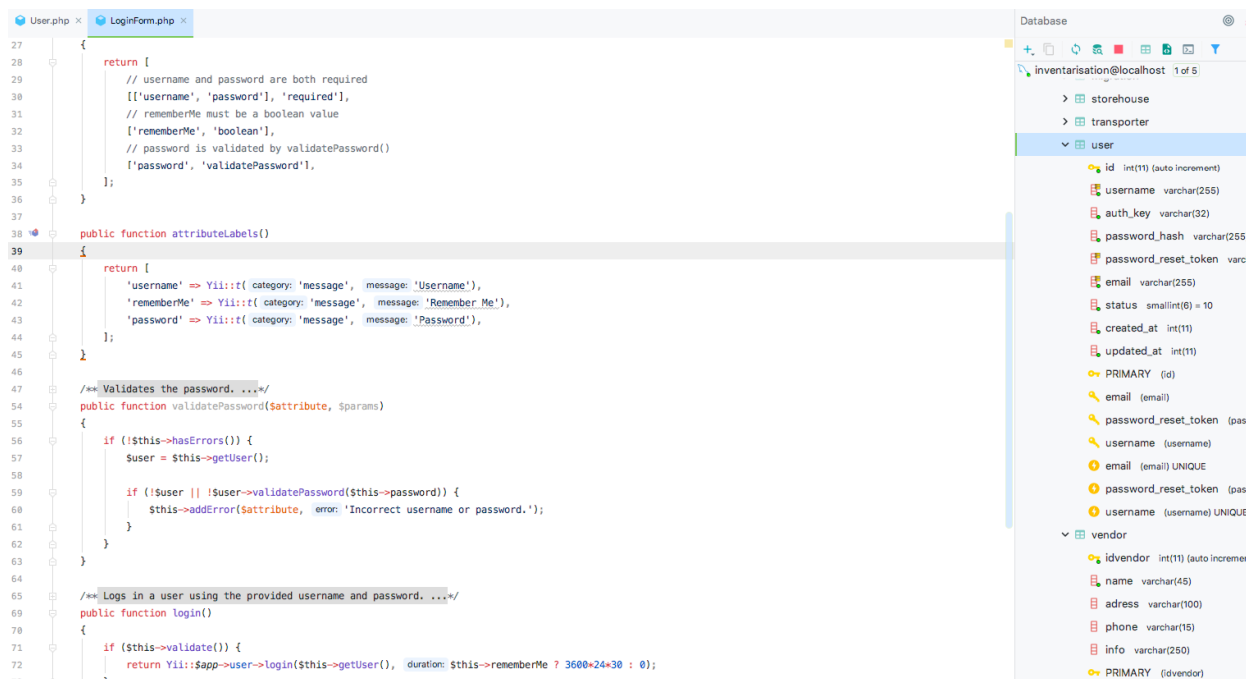


Рисунок 65 - Работа над моделью «LoginForm»

После этого было настроено отображение главной страницы (рис. 66).

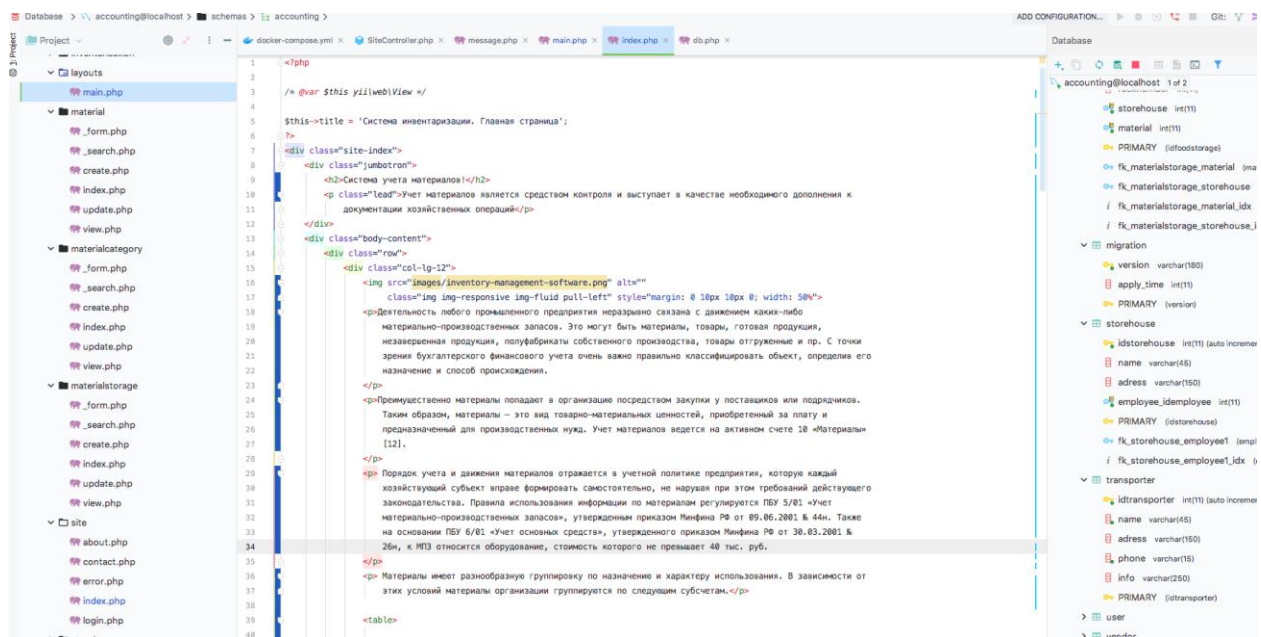


Рисунок 66 - Работа над главной страницей приложения

3.4 Тестирование работоспособности созданного модуля

Качество ПО (Software quality) — это характеристики ПО, относящихся к его способности удовлетворять потребности.

Тестирование ПО (Software testing) — проверка поведения программы, осуществляемая на конце тестов.

Этапы тестирования (рис. 67):

- планирование работ (Test Management);
- проектирование тестов ручной разработкой или автоматической генерацией (Test Design);
- выполнение тестирования с получением результатов (Test Execution);
- анализ результатов выполнения с целью оценки качества ПО (Test Analysis).

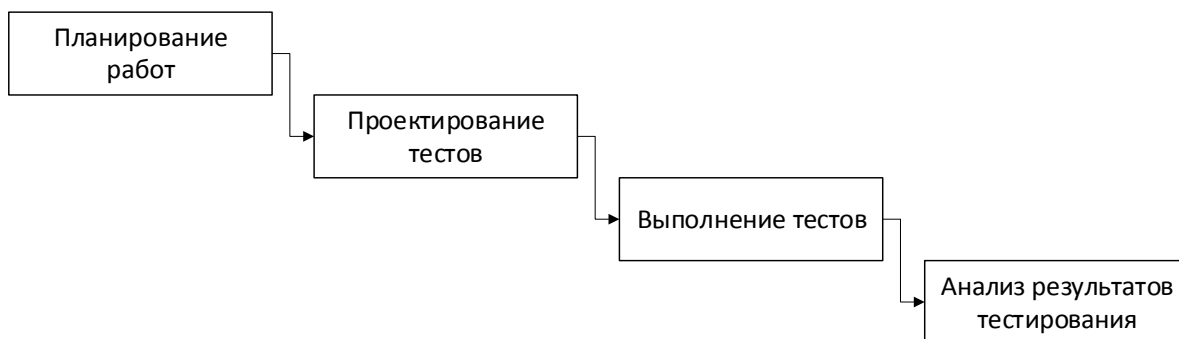


Рисунок 67 - Последовательность тестирования ПО

Ошибка может быть: в коде, тесте или в коде и тесте одновременно. Объект может давать данные, требующие анализа.

На каждом этапе жизни системы должны выполняться: 1. Верификация (Verification) — оценка системы или её компонентов для оценки текущего этапа начальным условиям. 2. Валидация (Validation) — определение соответствия ПО потребностям пользователя к системе [29].

Цели тестирования: 1. ПО работает правильно. 2. Все соответствует описанным требованиям. 3. Состояние продукта.

Выполняется запуск тестовых наборов. Итерация тестирования запускается при изменении тестируемого объекта.

Этапы итерации:

- обновление тестовых наборов, что требуют изменений наборов: добавление новых тестов, работа с текущими тестами;
- приемочные испытания, где бывают тесты, проверяющие выполнение предназначения. При неудаче объект дорабатывается, а итерация завершается;
- запуск основного набора тестов, что приводит к временным затратам на тесты. Потом формируется ведомость с тестами, завершившихся с ошибкой;

- анализ результатов тестирования. Выполняется классификация ошибок, оценка объекта. Потом принимается решение о доработке объекта и запуске следующей итерации или завершении.

Так как этап тестирования является итеративным, итерации ограничены сроками сдачи и стандартами качества. Завершают после выполнения итерации без ошибок, при нужном числе тестов или по истечении времени. При ошибках формируется список [14].

Выявить ошибки сложно. Ошибка (Software error) может не давать сбой, а некорректную работу. Сбой (Software failure) происходит при дефектах (Software defect) в компоненте или системе.

Тестовый случай (TestCase) – документ, описывающий шаги, условия и параметры для анализа реализации тестируемой функции.

Каждый тест содержит три базовые части (рис. 68):

- предусловия (PreConditions) — шаги, переводящие систему в состояние, пригодное для проведения проверки;
- описание теста (Description) — шаги, переводящие из состояния в состояние. Потом делается вывод о реализации требований;
- постусловия (PostConditions) — шаги, переводящие систему в изначальное положение.

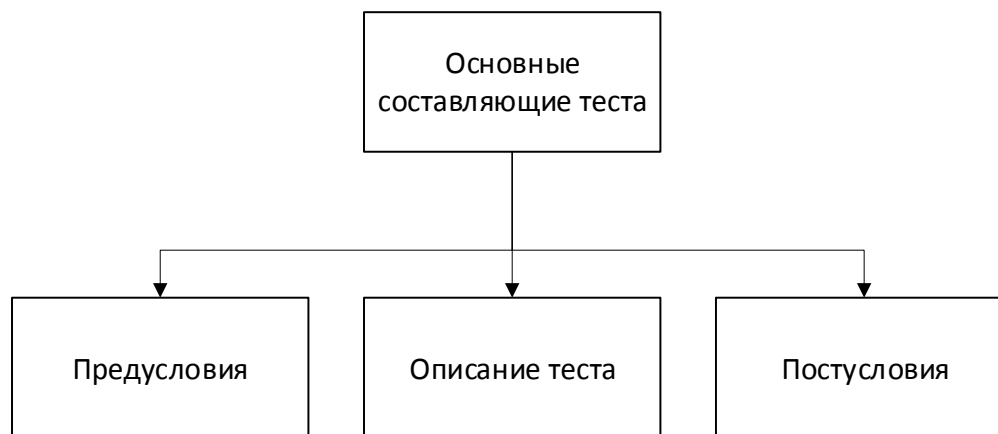


Рисунок 69 - Базовые части теста

В процессе тестирования были проведены тест сборки и функциональные тесты. Тест сборки является копией «дымового» тестирования, что выполняется для проверки работы приложения. Если программа не проходит тесты, дальше нет смысла, так как она не работает.

Примеры того, что программа не прошла «smoke testing»: она не запустилась, также тест не проходит по ошибке подключения к базе данных. В результате «smoke testing» программа запустилась, и соединение с базой данных установлено. Прохождение теста указывает на возможность проведения функционального и конфигурационного тестирования.

Функциональное тестирование представляет собой проверку того, выполняет ли программа функции, предусмотренные техническим заданием, а также степень точности выполнения проверяемых функций.

Таблица 7 - Тест главной страницы

Название:	Тест главной страницы		
Функция:	Отображение главной страницы и меню приложения		
Действие		Ожидаемый результат	Результат теста:
Предусловие:			
Ввод адреса веб-приложения в адресную строку браузера		Открытое главное окно приложения	пройден
Шаги теста:			
Выберите пункты из предоставленного меню		Пункт меню подсвечена	пройден
Нажмите кнопку перехода на другую страницу приложения		Переход на выбранную веб-страницу	пройден

Таблица 8 - Тест добавления материала

Название:	Тест добавления материала в систему		
Функция:	Добавление материала		
Действие		Ожидаемый результат	Результат теста:
Предусловие:			
Откройте программу		Открытое главное окно программы	пройден
Шаги теста:			
Выбрать пункт «Материалы»		Подсвечен пункт меню	пройден

	«Материалы»	
Нажать кнопку «Материалы»	Открыта форма для ввода данных	пройден
Ввод данных в форму	Данные введены	пройден
Нажать на кнопку «Сохранить»	Данные нового материала сохранены	пройден

Таблица 9 - Тест учета материалов

Название:	Тест учета материалов	
Функция:	Добавление записи	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Откройте программу	Открытое главное окно программы	пройден
Шаги теста:		
Выбрать пункт «Учет материалов»	Страница «Учет материалов» открыта	пройден
Нажать кнопку «Добавить запись»	Открыта форма для ввода данных	пройден
Ввод данных в форму	Данные введены	пройден
Нажать на кнопку «Сохранить»	Данные, введенные в форму сохранены	пройден

При конфигурационном тестировании определено, что веб-приложение функционирует в разных браузерах (табл. 10).

Таблица 10 - Работа веб-приложения в разных браузерах

Параметр	Google Chrome	Mozilla Firefox	Opera
Открытие главного модуля программы	+	+	+
Отображение меню программы	+	+	+
Выбор модуля программы	+	+	+
Отображение списка элементов из выбранного модуля	+	+	+
Добавление записей в выбранном модуле	+	+	+
Редактирование записей в выбранном модуле	+	+	+
Удаление записей в выбранном модуле	+	+	+

3.5 Контрольный пример функционирования модуля

Контрольный пример функционирования модуля и внешний вид созданных страниц и форм приложения представлен на рис. 69 – 88.

Главная / Приемки									
Приемки									
Добавить приемку									
Показаны записи 1-1 из 1.									
#	Дата	Сотрудник	Материал	Поставщик	Перевозчик	Стоимость	Количество	Единицы измерения	
1	2020-01-01	Иван	Материал	ООО "ЕУ"	фв	111	5	Тонна	

Рисунок 69 - Отображение данных «Приемка»

Главная / Приемки / Добавить приемку

Добавить приемку

Дата

Выберите дату ...

Сотрудник

Иванов Иван

Материал

898989

Поставщик

ООО "ЕУ"

Перевозчик

фв

Стоимость

Количество

Единицы измерения

Выберите единицу измерения...

Сохранить

Рисунок 70 - Ввод данных о приемке материалов

Сотрудники

Добавить сотрудника

Показаны записи 1-2 из 2.





#	ФИО	Номер телефона	
	<input type="text"/>	<input type="text"/>	
1	Петров Олег	+7897008881	 
2	Иванов Иван	+7881898988	 

Рисунок 71 - Отображение данных «Сотрудники»

Добавить сотрудника

ФИО

Номер телефона

Сохранить

Рисунок 72 - Ввод данных о сотруднике

Учет материалов

Добавить запись

Показаны записи 1-1 из 1.



#	Дата	Материал	Сотрудник	Количество	Единицы измерения	Номер акта	Номер протокола	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	2020-01-03	Цемент	Иванов Иван	7	Литры	2	2	 

Рисунок 73 - Отображение данных «Учет материалов»

Главная / Учет материалов / Добавить запись

Добавить запись

Дата

×

2020-01-26 15:05

←

Январь 2020

→

Пн	Вт	Ср	Чт	Пт	Сб	Вс
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Петров Олег

Номер акта

Номер протокола

Рисунок 74 - Ввод данных в список учета материалов

Главная / Материалы

Материалы

Добавить материал

Показаны записи 1-1 из 1.




#	Инвентарный номер	Наименование материала	Описание	Категория материала	
1	ЯЯЯЯЯ	Цемент	Цемент 200	Категория 1	  

Рисунок 75 - Отображение данных «Материалы»

Главная / Материалы / Добавить материал

Добавить материал

Инвентарный номер

Наименование материала

Описание

Категория материала

Категория 1

↓

Сохранить

Рисунок 76 - Ввод данных о материале

Категории материалов

Добавить категорию материала

Показаны записи 1-1 из 1.




#	Название категории	Описание	
	<input type="text"/>	<input type="text"/>	
1	Категория 1	Категория 1	  

Рисунок 77 - Страница отображения категорий материалов

Добавить категорию материала

Название категории

Описание

Сохранить

Рисунок 78 - Форма добавления категорий материалов

Места хранения материалов

Добавить место хранения материала

Показаны записи 1-1 из 1.




#	Номер стеллажа	Материал	Склад	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	444	Цемент	storehouse1	  

Рисунок 79 - Отображение данных «Хранение материалов»

Главная / Места хранения материалов / Добавить место хранения материала

Добавить место хранения материала

Материал

Цемент

Склад

storehouse1

Номер стеллажа

Сохранить

Рисунок 80 - Ввод данных о хранении материала

Главная / Склады

Склады

Добавить склад

Показаны записи 1-2 из 2.







#	Наименование склада	Адрес	Сотрудник	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	storehouse1	Ленина, 22	Петров Олег	  
2	storehouse2	Пушкина, 16	Иванов Иван	  

Рисунок 81 - Страница отображения складов

Главная / Склады / Добавить склад

Добавить склад

Наименование склада

Адрес

Сотрудник

Петров Олег

Сохранить

Рисунок 82 - Страница с формой добавления складов

Перевозчики

Добавить перевозчика

Показаны записи 1-1 из 1.




#	Наименование	Адрес	Номер телефона	Информация	
1	ООО "ТранспортСервис"	г. Москва, ул. Матросова, 22	+78758778877	Перевозка грузов	  

Рисунок 83 - Отображение данных «Перевозчики»

Добавить перевозчика

Наименование

Адрес

Номер телефона

Информация

Сохранить

Рисунок 84 - Ввод данных о перевозчиках

Поставщики

Добавить поставщика

Показаны записи 1-1 из 1.




#	Наименование	Адрес	Номер телефона	Информация	
1	ООО "ЕУСтройМат"	г. Москва, ул. Строителей, 223	+7876775677	Производство строительных материалов	  

Рисунок 85 - Отображение данных «Поставщики»

Главная / Поставщики / Добавить поставщика

Добавить поставщика

Наименование

Адрес

Номер телефона

Информация

Сохранить

Рисунок 86 - Ввод данных о поставщиках

Форма авторизации показан на рис. 87.

Главная / Вход в систему

Вход в систему

Для входа в систему необходимо заполнить поля ниже:

Имя пользователя

Пароль

☒ Запомнить меня

Войти

Рисунок 88 - Форма авторизации в системе

3.6 Оценка и обоснование экономической эффективности проекта

Под инвестициями в программные продукты понимаются финансовые вложения предприятия на разработку (или приобретение) программного продукта. В связи с этим определим стоимость программного продукта.

Стоимость программного продукта определяется по формуле:

$$C_{\text{пп}} = C_{\text{тз}} + C_{\text{м}} + C_{\text{э}} + C_{\text{а}} + C_{\text{пр}} + C_{\text{кр}} + H_{\text{инн}}, \text{ руб. (3.1)}$$

где:

$C_{тз}$ - трудозатраты (стоимость рабочего времени, затраченного на создание программного продукта), руб.;

C_m - стоимость израсходованных на создание программного продукта материалов, руб.;

$C_э$ - стоимость затраченной электроэнергии, руб.;

C_a - амортизационные отчисления, руб.;

$C_{пр}$ - прочие расходы, руб.

$C_{кр}$ - коммерческие расходы, руб.;

$H_{инн}$ - инновационная надбавка (информационная рента), руб.

Рассмотрим способы расчета каждого показателя.

Расчет трудозатрат

Трудозатраты (стоимость рабочего времени, затраченного на создание программного продукта) рассчитываются по формуле:

$$C_{тз} = t_{пп} C_{зп} K_d K_{пр} K_{сн} K_p, \text{ руб.} \quad (3.2)$$

где:

$t_{пп}$ - трудоемкость создания программного продукта, мес. Трудоемкость создания программного продукта принимается примерно равной запланированному количеству месяцев на разработку в соответствии с планом.

$C_{зп}$ - ставка заработной платы программиста начального уровня подготовки, руб./мес.

Данная величина рассчитывается по формуле:

$$C_{зп} = C_{зп}^{min} K_8, \text{ руб./мес.} \quad (3.3)$$

где

$C_{зп}^{min}$ - минимальная часовая тарифная ставка по Единой тарифной сетке

(ЕТС) для работников бюджетных организаций (минимум оплаты труда), руб./мес.

C_{3n}^{min} - 1100 руб./мес.

K_8 - тарифный коэффициент 8-го разряда ЕТС.

$K_8 = 2,34$.

$C_{3n} - 1100 * 2,34 = 2574$ (руб./мес.)

K_d - коэффициент доплат.

При разработке программного продукта, как правило, учитываются доплаты за вредные условия труда и за сверхурочное время работы. Для плановых целей данный коэффициент выбирается в диапазоне $K_d = 1,2 \dots 1,4$.

Принимаем $K_d = 1,3$.

K_{np} - коэффициент, учитывающий премиальные выплаты за качественное и своевременное выполнение планового задания.

Данный коэффициент принимается по среднему уровню премиальных выплат в промышленности России за аналогичный вид работ: $K_{np} = 1,25$.

$K_{сн}$ - коэффициент, учитывающий отчисления на социальные нужды $K_{сн} = 1,26$

K_p - районный коэффициент. $K_p = 1,5$.

$$C_{тз} = 3,9 * 2574 * 1,4 * 1,25 * 1,26 * 1,5 = 30831,05 \text{ (руб.)}$$

Стоимость израсходованных материалов

Стоимость израсходованных на создание программного продукта материалов (материальные затраты) рассчитывается по формуле:

$$C_m = \sum_{i=1}^n C_i Q_i, \text{ руб.} \quad (3.4)$$

где

Π_i - цена i -го вида материалов,

Q_i - количество израсходованных материалов i -го вида.

Расчет стоимости израсходованных материалов представим в виде таблицы (табл. 11).

Таблица 11 - Смета материальных расходов на создание приложения

Наименование материала	Единица измерения	Фактический расход материалов в натуральных единицах измерения	Цена приобретения материалов, руб./натуральную единицу измерения	Стоимость материалов, руб. (гр.3 * гр.4)
Жесткий диск	шт.	1	4650	4650
Компакт-диск DVD-RW	шт.	2	180	360
Бумага листовая	пачка 500 л. формата А4	1	220	220
Тонер	шт.	1	250	250
Итого:				5480

Затраты на электроэнергию

Стоимость затраченной электроэнергии рассчитывается по формуле:

$$C_3 = \left(\sum_{j=1}^m \Pi_{1\text{Квт}} * W_j * T_{\text{obj}} \right) + \Pi_{1\text{Квт}} * W_{\text{осв}} * t_{\text{осв}}, \text{ руб.} \quad (3.5)$$

где

$\Pi_{1\text{Квт}}$ – цена за 1 Квт. электроэнергии для юридических лиц на момент разработки ВКР.

$\Pi_{1\text{Квт}}$ – 1,5 руб.

W_j - энергопотребляющая мощность задействованного оборудования вида j . Выбирается на основе технического паспорта оборудования.

T_{obj} - время использования оборудования вида j , час. Определяется исходя из фактического использования оборудования каждого вида в период

разработки ВКР.

Для определения расходов на электроэнергию по задействованному оборудованию составим расчетную таблицу (табл. 12).

Таблица 12 - Потребление электроэнергии работающим оборудованием

Наименование оборудования	Энергопотребляемая мощность оборудования, Квт.	Время использования оборудования,	Фактически потребленная электроэнергия, Квт-час. (гр.2*гр.3)
Компьютер (системный блок)	1,7	311	528,7
Монитор	0,7	311	217,7
Принтер	1,2	16	19,2
Сканер	0,8	4	3,2
Итого:			768,8

$W_{\text{осв}}$ - энергопотребляемая мощность осветительных приборов в расчете на нормативную площадь рабочего места программиста, Квт.

$W_{\text{осв}}$ - 0,15 кВт

$t_{\text{осв}}$ - время работы осветительных приборов, час. Определяется по формуле:

$$t_{\text{осв}} = t_{\text{пп.час}} * K_{\text{попр}}, \text{ час.} \quad (3.6)$$

$t_{\text{пп.час}}$ - продолжительность разработки программного продукта, час. Определяется по формуле: $t_{\text{пп.час}} = t_{\text{nn}} * 176$.

$$t_{\text{пп.час}} = 3,9 * 176 = 686,4 \text{ (час)}$$

$K_{\text{попр}}$ - поправочный коэффициент, учитывающий разную продолжительность светового дня в различное время года.

Принимаем $K_{nonp} = 0,75$.

$t_{oce} = 686,4 * 0,75 = 514,8$ (час).

$$C_9 = 1,5 * 684,4 + 1,5 * 0,150 * 514,8 = 1026,6 + 115,83 = 1142,43 \text{ (руб.)}.$$

Амортизационные затраты

Амортизационные затраты рассчитываются по формуле:

$$C_a = \left(\sum_{j=1}^m N_{a.obj} C_{obj} T_{obj} / 1600 \right) + N_{a.зд} C_{зд} t_{пп.час} / 1600, \text{руб. (3.7)}$$

где:

$N_{a.obj}$ – норма амортизации задействованного оборудования вида j , в долях;

$$N_{a.obj} = 0,33;$$

C_{obj} - первоначальная стоимость задействованного оборудования вида (ЭВМ и оргтехника (сканер, принтер, внешний модем и т. п.)).

1600 - плановый годовой фонд рабочего времени, час.

Для определения амортизации задействованного оборудования составим аналитическую таблицу (см. табл. 13).

Таблица 13 - Амортизация задействованного оборудования

Наименование оборудования	Первоначальная стоимость оборудования ($C_{обв}$), руб.	Время использования оборудования, $T_{об}$, час.	Амортизация оборудования, руб. гр.2*0,33*гр.3/1600
Компьютер (системный блок)	44400	311	2848
Монитор	9700	311	608,2
Принтер	3300	16	10,9
Сканер	1200	4	0,9
Итого			3468,1

$H_{a,зд}$ - норма амортизации здания (помещения), в котором разрабатывался программный продукт. $H_{a,зд} = 0,05$.

$C_{зд}$ - стоимость здания (помещения), в котором разрабатывался программный продукт, руб. Определяется по формуле:

$$C_{зд} = S_n * C_{кв.м}, \text{руб.} \quad (3.8)$$

где

S_n - нормативная площадь рабочего места программиста; кв. м.

Нормативная площадь рабочего места учитывает эргономические характеристики работы программиста с набором оргтехники, включающим компьютер (системный блок, монитор, клавиатура), сканер, принтер, ксерокс, а также наличие места для работы вне технического окружения. Принимается $S_n = 14$ кв.м.

$C_{кв.м}$ - стоимость одного квадратного метра общей площади здания (помещения), в котором разрабатывался программный продукт, руб.

$$C_{кв.м} = \frac{C_{п.с,зд}}{S}, \text{руб./кв.м.} \quad (3.9)$$

где

$C_{п.с,зд}$ - первоначальная (сметная) стоимость здания, руб.

$C_{п.с,зд} = 168000000 \text{ руб.};$

S - общая (эксплуатационная) площадь здания, кв. м.

$S = 100000$ кв.м.

$$C_{кв.м} = \frac{168000000}{100000} = 1680 \text{ (руб. кв. м.)}.$$

$$C_{зд} = 14 * 1680 = 23520 \text{ (руб.)}.$$

$$C_a = 3468,1 + (0,05 * 23520 * 686,4 / 1600) = 3972,6 \text{ (руб.)}$$

Прочие расходы

Прочие расходы включают в себя почтовые расходы, оплату телефонных переговоров, расходы на консультации, транспортные расходы и т.п.

$$C_{пр} = (C_{тз} + C_m + C_э + C_a) * 0,05, \text{ руб.} \quad (3.10)$$

$$C_{пр} = (33505 + 4365 + 1125 + 4863) * 0,05 = 2192,9 \text{ (руб.)}$$

Коммерческие расходы

Коммерческие расходы при создании эксклюзивного программного продукта включают в себя расходы на презентацию программного продукта ($C_{през}$), оформление авторских прав ($C_{авт}$) и сертификацию ($C_{серт}$):

$$C_{кр} = C_{през} + C_{авт} + C_{серт}, \text{ руб.} \quad (3.11)$$

Расходы на презентацию рассчитываются по следующей формуле:

$$C_{през} = (C_{тз} + C_m + C_э + C_a + C_{пр}) K_{през}, \text{ руб.} \quad (3.12)$$

где

$K_{през}$ - коэффициент сравнительных затрат времени на создание и презентацию программного продукта.

$$K_{през} = \frac{t_{през} * K_{эмп}}{t_{пп.час}}, \quad (3.13)$$

где

$t_{през}$ - планируемая продолжительность презентации программного продукта, час.

Для расчетов принимаем $t_{през} = 0,5$ часа.

$K_{\text{эмп}}$ - эмпирический коэффициент, учитывающий трудозатраты на подготовку и организацию презентации, а также трудозатраты принимающей презентацию комиссии. Принимаем $K_{\text{эмп}} = 100$.

$$K_{\text{през}} = \frac{0,5 * 100}{686,4} = 0,07$$

$$C_{\text{през}} = (30831,05 + 5480 + 1142,43 + 3972,6 + 2192,9) * 0,07 = 43618,98 * 0,07 = 3053,3286$$

Принимаем $C_{\text{авт}} = 450$ руб., $C_{\text{серт}} = 1850$ руб.

$$C_{\text{кр}} = 3053,33 + 450 + 1850 = 5353,33 \text{ (руб.)}$$

Расчет информационной ренты

Расчет информационной ренты выполняется по следующей формуле:

$$H_{\text{инн}} = C_{\text{ан.пп}} - (C_{\text{тз}} + C_{\text{м}} + C_{\text{э}} + C_{\text{а}} + C_{\text{пр}} + C_{\text{кр}}) K_{\text{выг}}, \text{руб.} \quad (5.14)$$

где

$C_{\text{ан.пп}}$ - рыночная цена на аналогичный программный продукт, руб.

$K_{\text{выг}}$ - коэффициент выгоды приобретателя; принимаем $K_{\text{выг}} = 0,9$.

Средняя цена аналогичных программных продуктов – 78000 р.

$$H_{\text{инн}} = (78000 - (30831,05 + 5480 + 1142,43 + 3972,6 + 2192,9)) * 0,9 = 34868,63 \text{ (руб.)}$$

Итого, стоимость программного продукта:

$$C_{\text{пп}} = 30831,05 + 5480 + 1142,43 + 3972,6 + 2192,9 + 5353,33 + 34868,63 = 83840,94 \text{ (руб.)}$$

Расчет дополнительных затрат

Дополнительные затраты, связанные с внедрением программного

продукта, включают следующие единовременные затраты:

- затраты на приобретение и установку дополнительного оборудования, с помощью которого будут реализовываться функции программного продукта ($C_{об.доп}$);
- затраты на реконструкцию существующих рабочих мест, необходимую для эксплуатации программного продукта ($C_{рек}$);
- затраты на обучение и переобучение персонала, который будет работать с внедряемым программным продуктом ($C_{обуч}$).

$$Z_{доп} = C_{об.доп} + C_{рек} + C_{обуч}, \text{руб.} \quad (3.15)$$

Рассмотрим способы расчета каждой составляющей дополнительных затрат.

Внедрение на предприятии программного продукта не требует приобретения дополнительного оборудования. Затраты на приобретение дополнительного оборудования примем за 0.

Внедрение программного продукта не предполагает реконструкцию существующих на предприятии рабочих мест. Затраты на реконструкцию рабочих мест примем за 0.

Внедрение на предприятии программного продукта не требует дополнительного обучения персонала.

Сведем все расчетные показатели в таблицу (см. табл. 14).

Таблица 14 - Требуемый объем инвестиций в программный продукт

Наименование расходов	Сумма, руб.
1. Стоимость программного продукта, всего	83840,94
в том числе:	
- трудоzатраты;	30831,05

- расходы на материалы;	5480
- расходы на электроэнергию;	1142,43
- амортизация;	3972,6
- прочие расходы на создание программного продукта;	2192,9
- коммерческие расходы;	5353,33
- информационная рента.	34868,63
2. Дополнительные затраты, всего	0
в том числе:	
- затраты на приобретение оборудования;	0
- затраты на реконструкцию рабочих мест;	0
- затраты на обучение персонала.	0
Итого	83840,94

ЗАКЛЮЧЕНИЕ

В результате выполнения работы была достигнута цель – разработан модуль учета материальных ценностей в условиях предприятия.

Целью исследования является разработка веб-приложения для автоматизации учета материалов на примере предприятия ООО «МДМ».

В процессе достижения цели были решены следующие задачи:

- проведен анализ предметной области, изучены особенности учета материальных ценностей и систем, которые автоматизируют процесс учета;
- определены инструменты проектирования и разработки модуля учета материальных ценностей;
- разработана база данных и модуль учета материальных ценностей;
- протестирована работоспособность информационной системы;

В ходе анализа предметной области изучены особенности учета материальных ценностей и деятельности предприятия, разработаны бизнес-модели «AS-IS» которые показывают особенности процесса учета материальных ценностей, рассмотрены программные продукты, которые позволяют автоматизировать процесс учета материалов.

При рассмотрении аналогов проектируемого программного средства рассмотрен их функционал и проведено сравнительное исследование, показавшее, что основную задачу учета материалов рассмотренные программы решают. Но не все удовлетворяют требованиям организации. Таким как невысокая цена, простота обучения и работы с системой, учет специфики функционирования организации. В связи с этим принято решение в проектировании и разработки современной информационной системы, которая бы позволила автоматизировать проведение учета материалов. Также были определены инструменты и технологии проектирования и разработки веб-

ориентированной информационной системы для учета материальных ценностей.

Для реализации информационной системы выбран язык программирования PHP, который предназначен специально для создания веб-приложений. В качестве СУБД для проекта выбрана MySQL, которая является достаточно надежной и преимущественно хранит данные веб-приложений, которые созданы с помощью языка программирования PHP. Созданная информационная система представляет собой веб-приложение, реализованное посредством языка программирования PHP функционал которого, включает возможность осуществления учета материальных ценностей. Приложение позволит ООО «МДМ» автоматизировать процессы учета материальных ценностей, и таким образом повысить эффективность и надежность процессов учета, от чего напрямую зависит эффективность деятельности предприятия.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Анализ финансово-хозяйственной деятельности: учебник / Л. Н. Чечевицына, К. В. Чечевицын. – Ростов-на-Дону: Феникс, 2013. – 368 с.
2. Баронов, В. В. Автоматизация управления предприятием: Учебник / В. В. Баронов, Г.Н. Калянов, Ю.И. Попов, А.И. Рыбников. - М.: ИНФРА-М, 2012. – 239с.
3. Басовский Л. Е. Управление качеством: Учебник / Л. Е. Басовский, В. Б. Протасьев. – М.: ИНФРА-М, 2008.-212с.
4. Большаков, С. Н. Маркетинговые коммуникации и информационные ресурсы предприятия / С. Н. Большаков, В. А. Ачкасова. - М.: Издательство Санкт-Петербургского университета, 2014. – 132 с.
5. Бухгалтерский учет: Учебник для студентов вузов/ Ю.А. Бабаев, И.П. Комиссарова, М.С. Крашенинникова; Под ред. Ю.А. Бабаева. – М.:ЮНИТИ-ДАНА,2003. – 476 с.
6. Вересников Ю. К. О надежности систем обработки информации / Ю. К. Вересников //Актуальные проблемы современной науки. - 2011. - № 2 (58). - С. 193-195.
7. Верещагин С. А. Основные средства. Новый порядок учета / С. А. Верещагин. - М.: Эксмо, 2014. – 256 с.
8. Власенко А. А. Разработка адаптивной системы дистанционного обучения в сфере информационных технологий / А. А. Власенко // Автореферат диссертации на соискание ученой степени кандидата технических наук. Воронеж. - 2014. – 14 с.
9. Гвоздева В. А. Введение в специальность программиста / В. А. Гвоздева. - 2-е изд., испр. и доп.-М.: ИД «ФОРУМ»; ИНФРА-М, 2010. - 208 с.

10. Гейц И. В. Учет основных средств по-новому ПБУ 6/01 и в соответствии с главой 25 НК РФ / И. В. Гейц. - М.: Дело и сервис; Издание 2-е, перераб. и доп., 2016. – 176 с.
11. Голицына О. Л. Базы данных: учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. - М.: ФОРУМ: ИНФРА-М, 2011. – 400 с.
12. Горбаченко В. И. Проектирование информационных систем с CAERwin Modeling Suite 7.3: учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154 с.
13. Городилов А. А. Математическая модель динамических структур данных автоматизированной информационной системы / А. А. Городилов // Вестник Сибирского государственного аэрокосмического университета имени академика М. Ф. Решетнева. – 2009. – С. 90 – 95.
14. ГОСТ Р ИСО 19011-2003 «Руководящие указания по аудиту систем менеджмента качества и/или систем экологического менеджмента». – М.: Изд-во стандартов, 2003. - 19 с.
15. ГОСТ Р ИСО 9001-2001 «Система менеджмента качества. Требования» – М.: Изд-во стандартов, 2003. - 21 с.
16. ГОСТ Р ИСО 9004-2001 «Система менеджмента качества. Рекомендации по улучшению деятельности» – М.: Изд-во стандартов, 2001. – 35 с.
17. Государственные элементные сметные нормы на монтаж оборудования. ГЭСНм-2001. Часть 40. Дополнительное перемещение оборудования и материальных ресурсов сверх предусмотренного в государственных элементных сметных нормах на монтаж оборудования. - М.: ФГУ ФЦС, 2014. - 650 с.
18. Евдокимов Д. К. Нормирование материальных ресурсов / Д. К. Евдокимов, Г. М. Покараев. - М.: Экономика, 2013. - 200 с.

19. Евдокимов Д. К. Нормирование материальных ресурсов. Словарь-справочник / Д. К. Евдокимов, Г. М. Покараев. - М.: Экономика, 2012. - 199 с.
20. Зудилова, Т. В. Web-программирование HTML / Т. В. Зудилова, М. Л. Буркова - СПб: НИУ ИТМО, 2012.– 70 с.
21. Изобретая информационные системы будущего. Теория и практика. — Сургут, 2017. — 192 с.
22. Ильина Л. О. Рынок труда и управление человеческими ресурсами / Л. О. Ильина. - М.: Феникс, 2012. – 416 с.
23. Косенко В. В. Популярные языки программирования с позиций системного программирования / В. В. Косенко // Информационные технологии и вычислительные системы. – 2013. - № 1. – С. 54-59
24. Коцюба И. Ю. Основы проектирования информационных систем. Учебное пособие / И. Ю. Коцюба, А. В. Чунаев, А. Н. Шиков. – СПб: Университет ИТМО, 2015. – 206 с.
25. Крутякова, Т. Рациональные схемы бухгалтерского и налогового учета основных средств и нематериальных активов / Т. Крутякова. - М.: АКДИ Экономика и жизнь", 2010. – 416 с.
26. Крятова Л. А. Бухгалтерский учет основных средств и нематериальных активов / Л.А. Крятова. - М.: Издательско-книготорговый центр Маркетинг", 2011. – 220 с.
27. Кудряев В. А. Защита информационных ресурсов в негосударственной сфере / В.А. Кудряев, Е. А. Степанов. - М.: Государственный Университет Управления, 2010. – 63 с.
28. Кузнецов, М. В. РНР на примерах / М. В. Кузнецов, И. В. Симдянов. – СПб.: BHV, 2012. – 400 с.
29. Либерти Д. Программирование на C#. Создание .NET приложений. Программирование на C# / Д. Либерти. – М.: Бином 2010. – 684 с.

30. Ляндау Ю. В. Концепции моделирования бизнес-процессов / Ю. В. Ляндау, М. А. Пономарев // Вестник ИжГТУ. 2013. № 2(58) – С. 83 – 87.
31. Майстрах, К. В. Организация здравоохранения / К.В. Майстрах. - М.: Государственное издательство медицинской литературы, 2014. - 270 с.
32. Маклаков С. В. Моделирование бизнес-процессов с AIFusion Process Modeler. - Диалог-МИФИ, 2004. - 240 с.
33. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство [Текст] / Б. Маклафлин. – СПб.: Питер, 2014. – 544 с.
34. Мирошниченко, Г. А. Реляционные базы данных. Практические приемы оптимальных решений [Текст] / Г. А. Мирошниченко. - "БХВ-Петербург", 2005. – 400 с.
35. Михайлов П. В. База данных: цель создания, требования / П. В. Михайлов // Межвузовский сб. научных статей. Вып. 3 / Под ред. А. А. Кабанова. – СПб.: СПб ун-т МВД России, 2008. – С. 6.
36. Репин В. В. Бизнес-процессы. Моделирование, внедрение, управление / В. В. Репин. - М.: Манн, Иванов и Фербер, 2013. - 512 с.
37. Рогозов Ю. И., Свиридов А. С., Кучеров С.А. Архитектура информационных систем: учебное пособие. – Ростов-на-Дону: Изд-во ЮФУ, 2014. – 117 с.
38. Росс В. С. Создание сайтов: HTML, CSS, PHP, MySQL [Текст]. Учебное пособие, ч. 2 / В. С. Росс — МГДД(Ю)Т, М. - 2011 - 68 с.
39. Рудаков А. В. Технология разработки программных продуктов: учеб. пособие для студ. / А. В. Рудаков. - 5-е изд., стер. - М.: Издательский центр «Академия», 2010.-208с.
40. Самохвалова Ю.Н. Бухгалтерский учет: Практикум: Учеб. Пособие. – М.: ФОРУМ: ИНФРА-М, 2003. – 224 с.

41. Терехова В.А. Международные и национальные стандарты учета и финансовой отчетности: Учеб. Пособие для студентов вузов/ В.А. Терехова. – 2-е изд., перераб. И доп. – СПб.: Питер, 2003. – 267 с. – (Учеб пособие).

42. Швецкая В.М. Бухгалтерский учет: Учебник / В.М. Швецкая, Н. А. Головкин. – М.: «Дашков и Ко», 2012. – С. 120.