**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**КУРСОВОЙ ПРОЕКТ**

**на тему «Проектирование мобильного приложения для багетной мастерской ООО «Сириус»»**

Специальность 09.02.07 «Информационные системы и программирование»

Дисциплина «Технология разработки программного обеспечения»

|  |  |
| --- | --- |
| Преподаватель:  Коцюба Игорь Юрьевич  «30» декабря 2021г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы № Y2431  Головачев Данила Андреевич |

Санкт-Петербург

2021/2022

СОДЕРЖАНИЕ

введение 3

1 анализ поставленной задачи 4

1.1 Формулировка поставленной задачи 4

1.2 Описание предметной области 4

1.3 Обзор и сравнение аналогов 5

1.4 Функциональные требования 7

2 проектирование пиложения 8

2.1 Описание входных и выходных данных 8

2.2 Моделирование системы 9

2.3 Выбор архитектуры системы 10

2.4 Обоснование выбора технологий и программных средств 11

2.5 Выбор методов тестирования 12

заключение 15

список источников 16

ПРИЛОЖЕНИЕ А МОДЕЛИ РАЗРАБОТКИ 17

введение

Целью проекта является проектирование мобильного приложения для багетной мастерской ООО «Сириус».

Целевой аудиторией являются работники багетной мастерской ООО «Сириус».

Для проектирования системы необходимо выполнить следующие задачи:

* анализ предметной области;
* формирование функциональных требований к системе;
* определение входных и выходных данных;
* выбор и обоснование средств и технологий реализации;
* определение архитектуры системы;
* построение функциональной модели разработки;
* детальное проектирование.

Разделы, представленные в пояснительной записке:

1. Введение.
2. Раздел 1 Анализ поставленной задачи.
3. Раздел 2 Проектирование системы.
4. Заключение.
5. Список источников.
6. Приложение А Модель разработки.
7. анализ поставленной задачи
   1. Формулировка поставленной задачи

Целью проекта является проектирование мобильного приложения для оптимизации работы багетной мастерской ООО «Сириус», которое позволит автоматизировать процесс раскроя рейки для производства рам и подрамников, уменьшить количество избытков материала, обеспечить удобные коммуникации с клиентами.

Целевой аудиторией являются работники багетной мастерской ООО «Сириус».

* 1. Описание предметной области

Багетная мастерская занимается обеспечением художников всем необходимым для работы. Часть принадлежностей закупаются уже в готовом виде, однако есть и то, что, делают в багетной мастерской.

Для того, чтобы создать холст (полотно, натянутое на деревянную раму, на котором будет писать художник), нужно:

* получить подрамник:
  1. получить карту раскроя для рейки;
  2. нарезать рейку для подрамника;
  3. собрать подрамник.
* получить холст:

1. загрунтовать холст;
2. нарезать.

* Натянуть холст на подрамник.

В приведенном алгоритме все этапы кроме 1.А являются этапами механического воздействия и оптимизируются только при помощи оборудования.

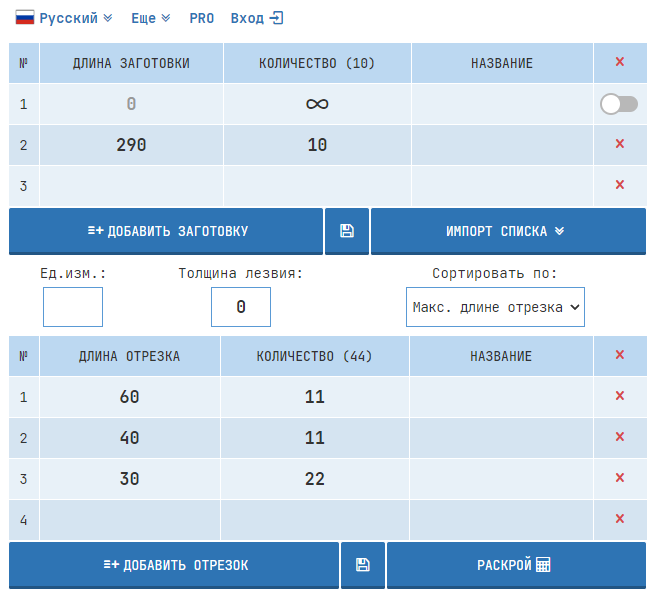
На данный момент этап 1.А проводится вручную и отнимает очень много времени и трудозатрат у работников предприятия. Заказчику требуется оптимизировать этот процесс при помощи мобильного приложения.

* 1. Обзор и сравнение аналогов

Для выявления актуальности разработки и необходимых функций проведено сравнение аналогичных решений.

Основной функцией приложения является метод линейного раскроя. В результате поиска аналогичных решений были выбраны следующие аналоги.

Smartcut.pro (URL: https://smartcut.pro/ru/ (дата обращения: 10 сентября 2021г.)). (Интерфейс для расчета в Smartcut.pro изображен на рисунке 1) – веб-ресурс, который позволяет составить оптимальную схему раскроя линейных изделий (арматура, брус, кабель, веревка и т.п.).



*Рисунок 1 – Интерфейс для расчета в Smartcut.pro*

Smartcut.pro (URL: https://smartcut.pro/ru/ (дата обращения: 10 сентября 2021г.)). (Интерфейс для расчета в Smartcut.pro изображен на рисунке 1) – про-версия – веб-ресурс, который позволяет составить оптимальную схему раскроя линейных изделий, при учете дополнительных параметров.

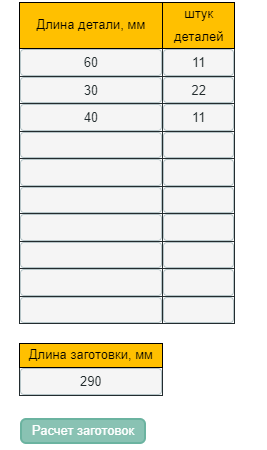
По сравнению с базовой версией появляются следующие отличия:

* поиск лучшего раскроя для ограниченного количества заготовок;
* задать размер лезвия.

slpl.ru (URL: http://slpl.ru/node/544 (дата обращения: 10 сентября 2021г.)). (Интерфейс для расчета в slpl.ru изображен на рисунке 1) – веб-ресурс для расчета минимума обрезков. Данный ресурс является минимальной версией метода раскроя.

Ресурс не имеет возможности задавать такие параметры метод, как:

* количество заготовок;
* размер лезвия;
* заготовки разного размера.

******

*Рисунок 2 - Интерфейс для расчета slpl.ru*

Таблица 1 – Сравнение аналогичных решений

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  аналогов → | Smartcut.pro | Smartcut.pro + | slpl.ru |
| Функция ↓ |
| Задать размер лезвия | Нет | Да | Нет |
| Визуализация раскроя | Да | Да | Нет |
| Поиск лучшего раскроя для ограниченного количества заготовок | Нет | Да | Нет |
| Платно | Нет | Да | Нет |
| Учет погрешности ручной работы | Нет | Нет | Нет |

Таблица 1 показывает, что ни в одном из аналогов не реализованы все требуемые функции. Учет погрешности ручной работы не учитывается нигде.

* 1. Функциональные требования

На основе анализа предметной области были выявлены следующие функциональные возможности в разрабатываемом приложении:

* метод линейного раскроя для определенного количества заказанной продукции;
* метод линейного раскроя для ограниченного количества заготовок;
* подсчет стоимости заказа;
* графический вывод метода раскроя.

1. проектирование пиложения
   1. Описание входных и выходных данных

Входными данными системы являются данные, представленные в таблице 3.

Таблица 3 – Входные и выходные данные.

|  |  |  |
| --- | --- | --- |
| Функция | Входные данные | Выходные данные |
| 1 | 2 | 3 |
| Авторизация | * логин; * пароль. | * Маркер успешности авторизации. |
| Раскрой | * тип раскроя; * количество заготовок * размер заготовки; * количество требуемых деталей; * размер деталей. | * + наилучшая карта раскроя текстом;   + наилучшая карта раскроя графически; * процент избыточности наилучшей карты. |
| Добавление товара | * + тип товара;   + название товара;   + артикул товара;   + коэффициент товара (если нужен); * описание товара. | * маркер успешности добавления. |
| Вывод списка товаров | * таблица товаров. | * + тип товара;   + название товара;   + артикул товара;   + коэффициент товара;   + описание товара. |
| Расчет цены | * название товара; * количество товара; * количество позиций; * скидка. | * + список позиций;   + общая цена сборки. |

* 1. Моделирование системы

Для проектирования системы были использованы:

* нотация IDEF0;
* нотация UML.

IDEF0 - нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции.

В конце 1980-х годов получили большое распространение объектно-ориентированные языки программирования. Тенденции их активного использования определили задачи разработки языка моделирования, дающего возможность реализовать объектно-ориентированный подход и построить наилучшую модель системы с указанием ее значимых свойств. Этим языком стал UML. В настоящее время UML как нотация моделирования ИС поддерживает ряд объектно-ориентированных CASE-продуктов.

Диаграмма вариантов использования (диаграмма прецедентов, use case diagram) – диаграмма, предназначенная для построения на концептуальном уровне моделей поведения пользователей в системе.

Диаграмма компонентов (component diagram) отображает иерархию подсистем, структурных компонентов и зависимостей между ними. Физическими компонентами выступают базы данных, исполняемые файлы, системы, библиотеки, интерфейсы ИС и т.д. В случае использования диаграммы компонентов для отображения внутренней структуры компонентов, интерфейсы составного компонента делегируются в определенные интерфейсы внутренних компонентов.

Основными целями построения диаграмм компонентов являются:

* определение архитектуры проектируемой системы;
* построение концептуальной и физической моделей баз данных;
* представление структуры исходного и специфики исполняемого кода системы;
* многократное использование определенных фрагментов программного кода.

Диаграмма деятельности (activity diagram) – диаграмма, использующаяся при моделировании бизнес-процессов, на которой представлено разложение на составные части некоторой деятельности, а именно: скоординированного выполнения отдельных действий и вложенных видов деятельности, которые соединяются между собой потоками от выходов одного узла к входам другого, с указанием их исполнителей.

* 1. Выбор архитектуры системы

Архитектура системы определяет основные элементы сети, характеризует ее общую логическую организацию, техническое обеспечение, программное обеспечение, описывает методы кодирования. Архитектура также определяет принципы функционирования и интерфейс пользователя. Наиболее популярными видами архитектур являются три вида:

* архитектура терминал – главный компьютер;
* одноранговая архитектура;
* архитектура клиент-сервер.

Архитектура терминал – это концепция информационной сети, в которой вся обработка данных осуществляется одним или группой главных компьютеров.

Одноранговая архитектура – это концепция информационной сети, в которой ее ресурсы рассредоточены по всем системам. Данная архитектура характеризуется тем, что в ней все системы равноправны. К одноранговым сетям относятся малые сети, где любая рабочая станция может выполнять одновременно функции файлового сервера и рабочей станции. В одноранговых ЛВС дисковое пространство и файлы на любом компьютере могут быть общими. Чтобы ресурс стал общим, его необходимо отдать в общее пользование, используя службы удаленного доступа сетевых одноранговых операционных систем. В зависимости от того, как будет установлена защита данных, другие пользователи смогут пользоваться файлами сразу же после их создания.

Архитектура клиент-сервер – это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов. Рассматриваемая архитектура определяет два типа компонентов: серверы и клиенты.

Сервер – это объект, предоставляющий сервис другим объектам сети по их запросам.

Клиенты – это рабочие станции, которые используют ресурсы сервера и предоставляют удобные интерфейсы пользователя.

В моем проекте используется архитектура терминал.

* 1. Обоснование выбора технологий и программных средств

При проектировании системы использовались следующие программные средства:

− Visual Paradigm;

− Draw.io.

Visual Paradigm относится к профессиональным инструментам работы со стандартом UML. При помощи встроенного функционала данный пакет способен поддерживать весь рабочий цикл программы: анализ, ориентированный на объекты, дизайн, ориентированный на объекты, конструкция, тестирование и разработка.

Draw.io - инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, отношений сущностей, программных блоков и другого. Обладает богатым набором функций для визуализации большинства задач пользователя.

Для реализации системы были выбраны следующие программные средства:

Java – это объектно-ориентированный язык программирования, является основным языком разработки для операционной системы Android. Программы на Java транслируются в байт-код, который затем выполняется виртуальной машиной Java (JVM).

Android Studio — нтегрированная среда разработки для работы с платформой Android, официальная среда Java-разработки, признанная компанией Google.

SQLite - это встроенная библиотека, которая реализует автономный, без серверный, нулевой конфигурации, транзакционный механизм СУБД SQL. Имеет маленький размер, поэтому часто используется при создании мобильных приложений.

XML — расширяемый язык разметки, который используется для хранения и передачи информации в удобном для человека и компьютера виде.

* 1. Выбор методов тестирования

Методы тестирования классифицируются:

* + 1. По знанию внутренностей системы:
* черный ящик - метод, основанный на незнании внутреннего устройства объекта тестирования;
* белый ящик – метод основывается на знании об устройстве и логике работы тестируемого объекта;
* серый ящик – метод, в котором используются паттерн поведения пользователей, зная внутреннее устройство системы.
  + 1. По объекту тестирования:
* функциональное тестирование;
* тестирование интерфейса пользователя;
* тестирование локализации;
* тестирование скорости и надежности;
* тестирование безопасности;
* тестирование опыта пользователя;
* тестирование совместимости.
  + 1. По субъекту тестирования:
* Альфа-тестировщик - субъект тестирования, работающий в компании разработчике, который профессионально или непрофессионально проводит тестирование;
* бета-тестировщик - Субъект тестирования, который не является сотрудником компании и которому предоставляется возможность пользоваться новой системой до того, как она станет доступна всем остальным.
  + 1. По времени проведения тестирования:
* до передачи пользователю
* альфа-тестирование;
* после передачи пользователю.
  + 1. По критерию “позитивности” сценариев:
* позитивное тестирование;
* негативное тестирование.
  + 1. По степени изолированности тестируемых компонентов:
* компонентное тестирование;
* интеграционное тестирование;
* системное тестирование.
  + 1. По степени автоматизации тестирования:
* ручное тестирование;
* автоматизирование тестирование;
* смешанное тестирование.
  + 1. По степени подготовки к тестированию:
* тестирование по документации;
* эд хок-тестирование.

Тестирование системы должно проводиться на следующих уровнях:

Модульное тестирование на этапе разработки проводится непосредственно самим разработчиком, методом Белого ящика. Набор соответствующих тестов также определяется разработчиком.

заключение

В результате выполнения проекта было спроектировано мобильное приложения для багетной мастерской ООО "Сириус".

В ходе выполнения проекта были выполнены следующие задачи:

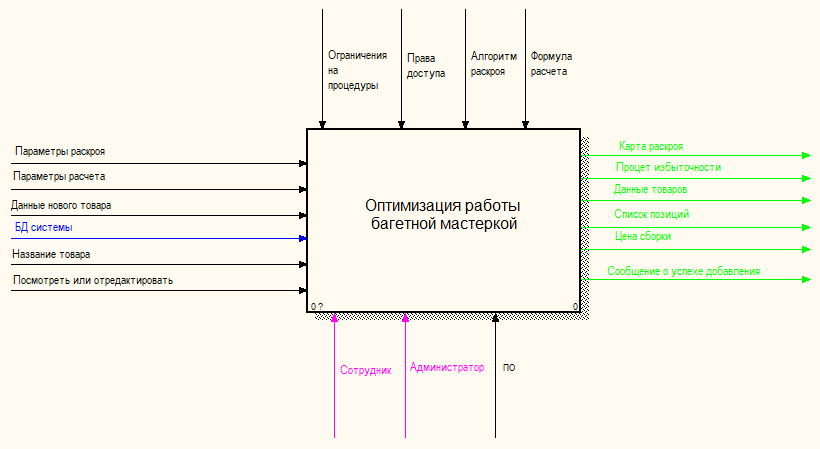
* анализ предметной области;
* формирование функциональных требований к системе;
* определение входных и выходных данных;
* выбор и обоснование средств и технологий реализации;
* определение архитектуры системы;
* построение функциональной модели разработки;
* детальное проектирование.

Таким образом, все задач, поставленные на проект, выполнены.

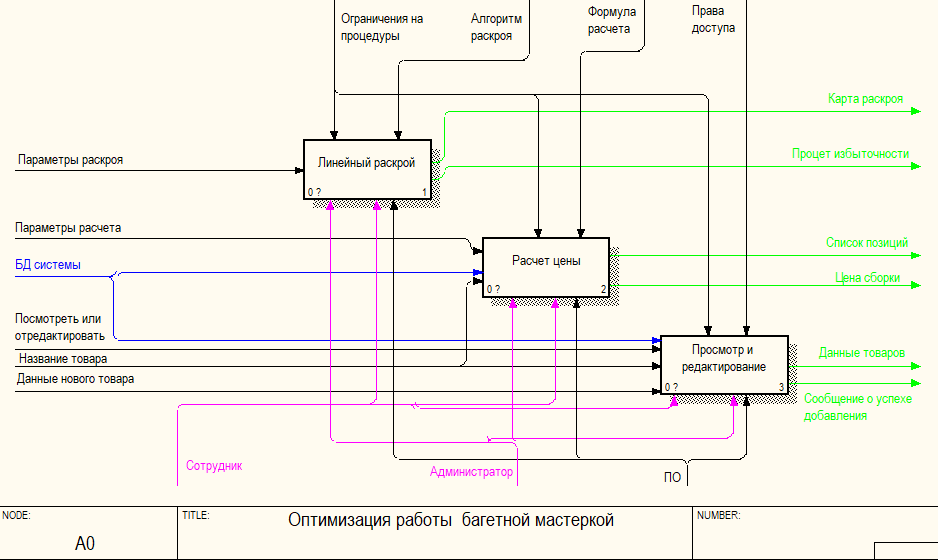
список источников

1. Что такое BPMN-схема и как ее создать? [Электронный ресурс] // Lucidchart URL: https://www.lucidchart.com/pages/ru/bpmn (дата обращения 15.12.2021).
2. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем. Учебное пособие. – СПб: Университет ИТМО, 2015. – 206 с.
3. Visual Paradigm [Электронный ресурс] // Официальный сайт компании Visual Paradigm: 2020. URL: https://www.visual-paradigm.com/ (дата обращения: 15.12.2021)
4. MySQL [Электронный ресурс] // Официальный сайт MySQL. URL: https://www.mysql.com/ (дата обращения: 15.12.2021).
5. Springer Link [Электронный ресурс] // Официальный сайт. URL: https://link.springer.com/book/10.1007%2Fb135457#editorsandaffiliations (дата обращения: 15.11.2021).
6. IntelliJ IDEA: функциональная и эргономичная IDE для разработки на Java [Электронный ресурс] // JetBrains. URL: https://www.jetbrains.com/ru-ru/idea/ (дата обращения: 12.12.2021)
7. Developer Guides | Android Developers [Электронный ресурс] // Android Developers. URL: https://developer.android.com/guide (дата обращения: 10.12.2021).
8. UML [Электронный ресурс] // Руководство Microsoft. URL: https://www.microsoft.com/ruru/microsoft365/businessinsightsideas/resources/guide-to-uml-diagramming-and-databasemodeling (дата обращения: 10.12.2021)
9. Java [Электронный ресурс] URL: https://web-creator.ru/articles/java (дата обращения: 10.12.2021)

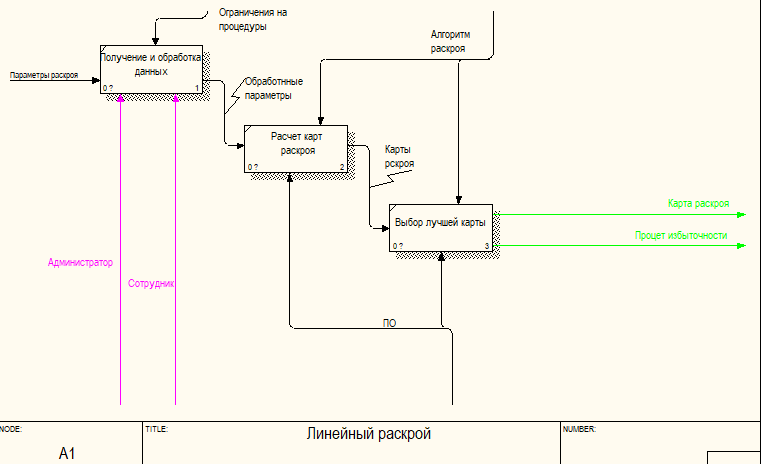
ПРИЛОЖЕНИЕ А МОДЕЛИ РАЗРАБОТКи



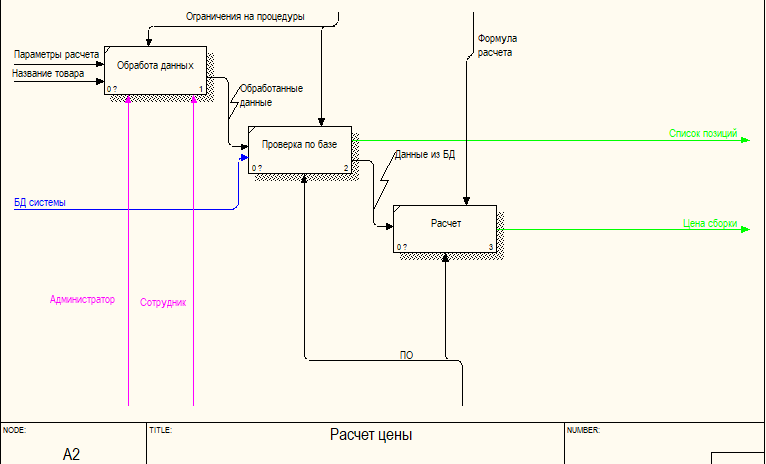
*Рисунок А.1 – Функциональная модель: главный процесс*



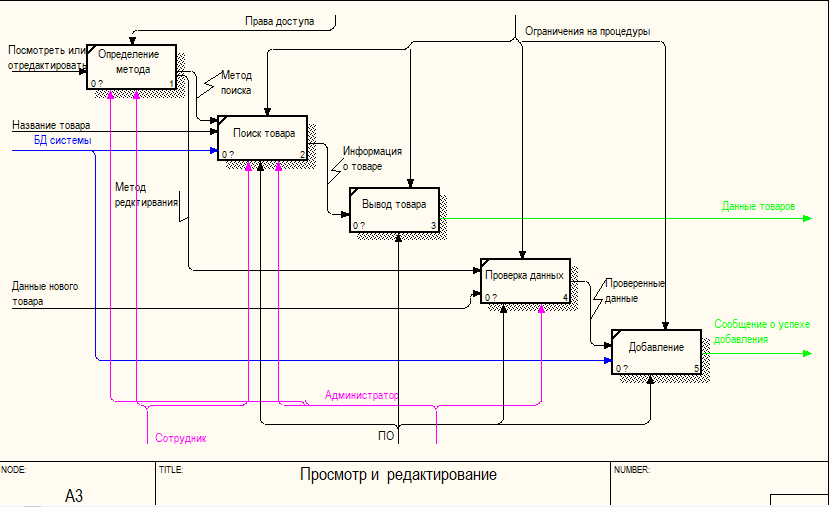
*Рисунок А.2 – Функциональная модель: детализация главного процесса*



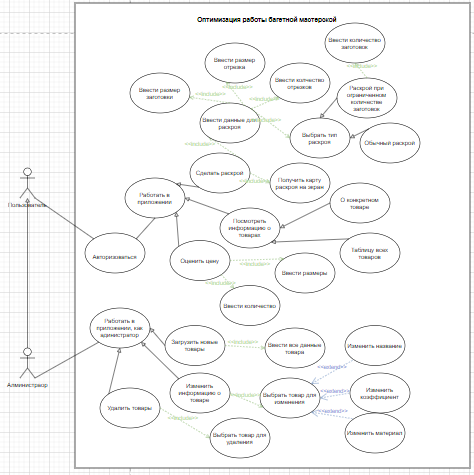
*Рисунок А.3 – Функциональная модель: детализация подсистемы линейного раскроя*



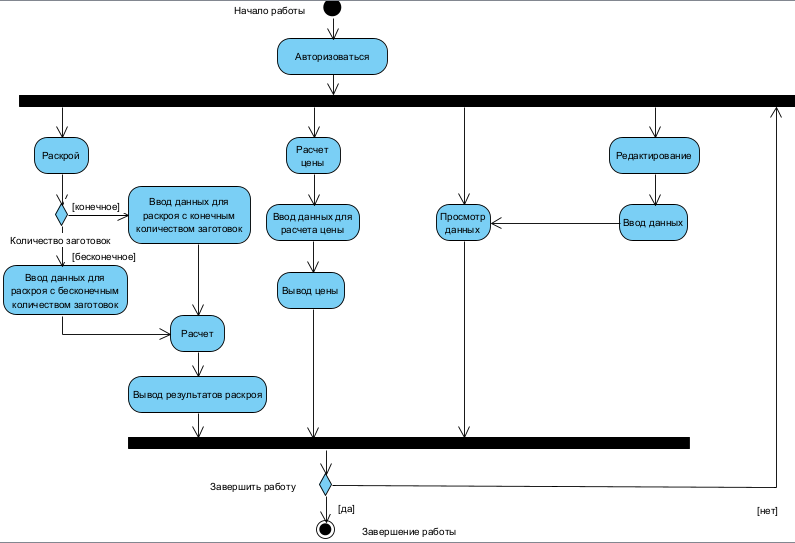
*Рисунок А.4 – Функциональная модель: детализация подсистемы расчета цены*



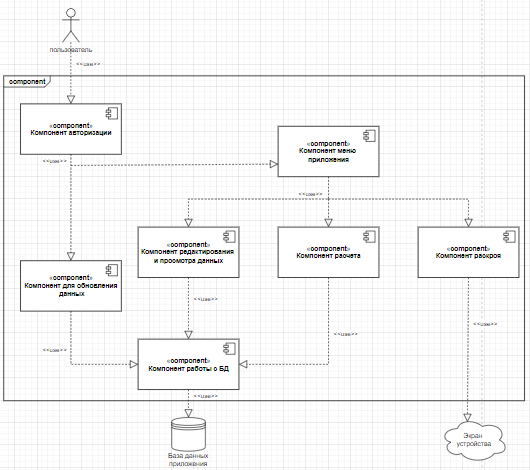
*Рисунок А.5 – Функциональная модель: детализация подсистемы просмотра и редактирования*



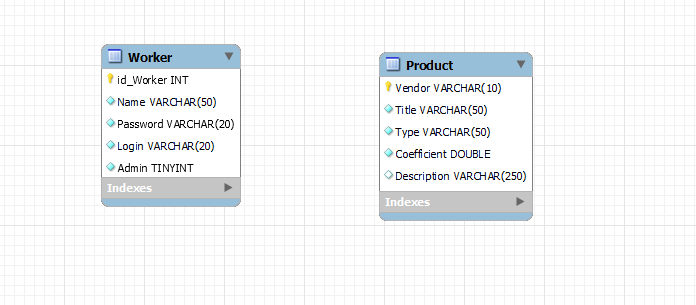
*Рисунок А.6 – Поведение пользователей системы*



*Рисунок А.7 – Логическая модель системы*



*Рисунок А.8 – Физическая модель системы*



*Рисунок А.9 – Структура данных в БД*